

# DESTINATION CISSP

A CONCISE GUIDE



SECOND  
Edition

Rob Witcher | John Berti | Lou Hablas | Nick Mitropoulos

# DESTINATION **CISSP**

A CONCISE GUIDE

Rob Witcher | John Berti | Lou Hablas | Nick Mitropoulos

Copyright © 2024, Destination Certification, Inc.

All rights reserved. No part of this book may be used or reproduced by any means—graphic, electronic, or mechanical, including any information storage retrieval system—without the express written permission from the author, except in the case of brief quotations for use in articles and reviews wherein appropriate attribution of the source is made.

Library of Congress Control Number: 202292270

ISBN: 979-8-9874077-2-1 (Paperback) ISBN: 979-8-9874077-3-8 (Ebook) Because of the dynamic nature of the Internet, web addresses or links contained in this book may have been changed since publication and may no longer be valid. The content of this book and all expressed opinions are those of the authors.

This work does not constitute the authors engaging in the rendering of professional advice or services.

Neither the authors nor any associated with the publication process shall be held liable or responsible for any loss or damage allegedly arising from any suggestion or information contained in this work.

Layout designer: Kelly Badeau eBook designer:

[Booknook.biz](http://Booknook.biz)

Cover illustrator: Aleksei Diuzhov Cover designer:

Vinod Kumar Palli Images: [unsplash.com](https://unsplash.com), [pexels.com](https://pexels.com),  
[pixabay.com](https://pixabay.com)

Icons: [thenounproject.com](https://thenounproject.com)

# Acknowledgments

To you dear reader: I hope this book helps you achieve your goals!

To gyönyörű with whom I have the privilege of sharing my life, and who created our two incredible and wonderful little menaces. To my mom for always being there for me. And to John (that John) for introducing and guiding me to the extremely fulfilling profession of teaching.

—Rob Witcher

To my mom and dad for instilling the proper values. To my amazing brother, whom I admire and drives me to be more like him. To my incredible daughters who make me proud every single day. And to my beautiful wife, who makes my life so incredibly amazing. And finally, to Hal Tipton, who inspired my passion and dedication and made me realize how gratifying this would be.

—John Berti

To each member of this book dream team: Bravo, we did it!

To Rob and John: It is an honor and privilege to work with you!

To my parents: Thank you for your selfless love, care, and guidance. I miss you.

To Beth, Madison, and Patrick: Thank you for your love and support — always and especially during the past several years, as this CISSP journey has unfolded and continues to bloom.

—Lou Hablas

To Rob, John and Lou, who have invited me to be a part of this journey. It is an honour to have co-authored this book with them and be part of a truly gifted team giving back to the security community.

To my son, coming into our lives in a few weeks. The world is your oyster little one. Enjoy every minute.

—Nick Mitropoulos



## OVERVIEW OF CONTENTS

[Why This Book](#)

[About the Exam](#)

[Mindset](#)

[About the Authors](#)

[Notes on the Book](#)

[Introduction](#)

[Domain 1—Security and Risk Management](#)

[Domain 2—Asset Security](#)

[Domain 3—Security Architecture and Engineering](#)

[Domain 4—Communication and Network Security](#)

**Domain 5—Identity and Access Management (IAM)**

**Domain 6—Security Assessment and Testing**

**Domain 7—Security Operations**

**Domain 8—Software Development Security**

**References and Further Reading**

**Acronyms**

**Index**

**Proven Exam Strategies**

**Destination Certification CISSP MasterClass**



## CONTENTS

### Why This Book

Who is the CISSP meant for?

Value of the CISSP certification

How to best use this book

What are “Core Concepts” and “Expect to be tested on?”

### About the Exam

April 2024 exam change summary

### Mindset

### About the Authors

Rob Witcher

John Berti

Lou Hablas

Nick Mitropoulos

### **Revision Editor**

Josh Lake

### **Technical Reviewer**

Taz Wake

### **Notes on the Book**

What's up with the mixed case in the titles?

Hey! I found a mistake in the book!

## **INTRODUCTION**

### **DOMAIN 1: Security and Risk Management**

1.1 Understand, adhere to, and promote professional ethics

1.1.1 ISC2 Code of Professional Ethics

1.1.2 Organizational Code of Ethics

1.2 Understand and apply security concepts

**1.2.1 Confidentiality, Integrity, Availability, Authenticity, and Nonrepudiation**

**1.3 Evaluate, apply, and sustain security governance principles**

**1.3.1 Alignment of the Security Function to Business Strategy, Goals, Mission, and Objectives**

**1.3.2 Organizational Processes**

**1.3.3 Organizational Roles and Responsibilities**

**1.3.4 Security Control Frameworks**

**1.3.5 Due Care versus Due Diligence**

**1.4 Understand legal, regulatory, and compliance issues that pertain to information in a holistic security context**

**1.4.1 Cybercrimes and Data Breaches**

**1.4.2 Licensing and Intellectual Property Requirements**

**1.4.3 Import/Export Controls**

**1.4.4 Transborder Data Flow**

**1.4.5 Issues Related to Privacy**

## **1.4.6 Contractual, Legal, and Industry Standards and Regulatory Requirements**

**1.5 Understand requirements for investigation types (i.e. administrative, criminal, civil, regulatory, industry standards)**

**1.6 Develop, document, and implement security policies, procedures, standards, baselines, and guidelines**

**1.7 Identify, Analyze, assess, prioritize, and implement Business Continuity (BC) requirements**

**1.8 Contribute to and enforce personnel security policies and procedures**

**1.8.1 Candidate Screening and Hiring**

**1.8.2 Employment Agreements and Policy Driven Requirements**

**1.9 Understand and apply risk management concepts**

**1.9.1 Risk Management**

**1.9.2 Asset Valuation**

**1.9.3 Risk Analysis**

**1.9.4 Annualized Loss Expectancy (ALE) Calculation**

**1.9.5 Risk Response/Treatment**

**1.9.6 Applicable Types of Controls**

**1.9.7 Categories of Controls**

**1.9.8 Functional and Assurance**

**1.9.9 Selecting Controls**

**1.9.10 Risk Management Frameworks**

**1.10 Understand and apply threat modeling concepts and methodologies**

**1.11 Apply supply chain risk management (SCRM) concepts**

**1.11.1 Risks Associated with the Acquisition of Products and Services from Suppliers and Providers**

**1.11.2 Risk Mitigations**

**1.12 Establish and maintain a security awareness, education, and training program**

**1.12.1 Methods and Techniques to Increase Awareness, Training, and Education**

**1.12.2 Periodic Content Reviews to Include Emerging Technologies and Trends**

### **1.12.3 Program Effectiveness Evaluation**

#### **MINDMAP REVIEW VIDEOS**

#### **REVIEW QUESTIONS**

### **DOMAIN 2: Asset Security**

**2.1 Identify and classify information and assets**

**2.1.1 Asset Classification**

**2.1.2 Classification Process**

**2.1.3 Classification versus Categorization**

**2.1.4 Labeling and Marking**

**2.2 Establish information and asset handling requirements**

**2.2.1 Media Handling**

**2.3 Provision information and assets securely**

**2.3.1 Data Classification Roles and Responsibilities**

**2.3.2 Data Classification Policy**

**2.4 Manage data life cycle**

**2.4.1 Information Life Cycle**

**2.4.2 Data Destruction**

**2.5 Ensure appropriate asset retention**

**2.5.1 Data Archiving**

**2.6 Determine data security controls and compliance requirements**

**2.6.1 Protecting Data at Rest**

**2.6.2 Protecting Data in Transit**

**2.6.3 Protecting Data in Use**

**2.6.4 Information Obfuscation Methods**

**2.6.5 Digital Rights Management (DRM)**

**2.6.6 Data Loss Prevention (DLP)**

**MINDMAP REVIEW VIDEOS**

**REVIEW QUESTIONS**

**DOMAIN 3: Security Architecture and Engineering**

**3.1 Research, implement, and manage engineering processes using secure design principles**

**3.1.1 Security's Involvement in Design and Build**

### **3.1.2 Determining Appropriate Security Controls**

## **3.2 Understand the fundamental concepts of security models**

### **3.2.1 Security Models**

### **3.2.2 Enterprise Security Architecture**

### **3.2.3 Layer-based Models**

### **3.2.4 Rule-based Models**

### **3.2.5 Certification and Accreditation**

### **3.2.6 Evaluation Criteria (ITSEC and TCSEC)**

### **3.2.7 Common Criteria**

## **3.3 Select controls based upon systems security requirements**

### **3.3.1 Security Control Frameworks**

## **3.4 Understand security capabilities of information systems**

### **3.4.1 RMC, Security Kernel, and TCB**

### **3.4.2 Processors (CPUs)**

### **3.4.3 Process Isolation**

**3.4.4 Types of Storage**

**3.4.5 System Kernel**

**3.4.6 Privilege Levels**

**3.4.7 Middleware**

**3.4.8 Abstraction and Virtualization**

**3.4.9 Layering/Defense-in-Depth**

**3.4.10 Trusted Platform Modules (TPM)**

**3.5 Assess and mitigate the vulnerabilities of security architectures, designs, and solution elements**

**3.5.1 Vulnerabilities in Systems**

**3.5.2 Hardening**

**3.5.3 Risk in Mobile Systems**

**3.5.4 OWASP Mobile Top 10**

**3.5.5 Distributed Systems**

**3.5.6 Inference and Aggregation**

**3.5.7 Industrial Control Systems (ICS)**

**3.5.8 Internet of Things (IoT)**

**3.5.9 Cloud Service and Deployment Models**

**3.5.10 Compute in the Cloud**

**3.5.11 Cloud Forensics**

**3.5.12 Cloud Computing Roles**

**3.5.13 Cloud Identities**

**3.5.14 Cloud Migration**

**3.5.15 Edge Computing**

**3.5.16 XSS and CSRF**

**3.5.17 SQL Injection**

**3.5.18 Input Validation**

**3.6 Select and determine cryptographic solutions**

**3.6.1 Introduction to Cryptography**

**3.6.2 Cryptographic Terminology**

**3.6.3 Substitution and Transposition**

**3.6.4 Steganography and Null Ciphers**

**3.6.5 Symmetric Cryptography**

**3.6.6 Asymmetric Cryptography**

**3.6.7 Hybrid Key Exchange**

**3.6.8 Message Integrity Controls**

**3.6.9 Digital Signatures**

**3.6.10 Digital Certificates**

**3.6.11 Public Key Infrastructure (PKI)**

**3.6.12 Key Management**

**3.6.13 S/MIME**

**3.7 Understand methods of cryptanalytic attacks**

**3.7.1 Cryptanalysis**

**3.7.2 Cryptanalytic Attacks Overview**

**3.7.3 Cryptographic Attacks**

**3.8 Apply security principles to site and facility design**

**3.8.1 Intro to Physical Security**

**3.8.2 Layered Defense Model**

**3.9 Design site and facility security controls**

**3.9.1 Security Survey**

**3.9.2 Perimeter**

**3.9.3 Closed-circuit TV (CCTV)**

**3.9.4 Passive Infrared Devices**

**3.9.5 Lighting**

**3.9.6 Doors and Mantraps**

**3.9.7 Locks**

**3.9.8 Card Access/Biometrics**

**3.9.9 Windows**

**3.9.10 Walls**

**3.9.11 Automated Teller Machine (ATM) Skimming**

**3.9.12 Power**

**3.9.13 Heating Ventilation and Air Conditioning (HVAC)**

**3.9.14 Fire**

**3.10 Manage the Information System Lifecycle**

**MINDMAP REVIEW VIDEOS**

**PRACTICE QUESTIONS**

**Domain 4 Communication & Network Security**

## **4.1 Implement secure design principles in network architectures**

**4.1.1 Open System Interconnection (OSI) Model**

**4.1.2 Layer 1: Physical**

**4.1.3 Layer 2: Data Link**

**4.1.4 Authentication Protocols**

**4.1.5 Layer 3: Network**

**4.1.6 Logical Addressing**

**4.1.7 Layer 4: Transport**

**4.1.8 Layer 5: Session**

**4.1.9 Layer 6: Presentation**

**4.1.10 Layer 7: Application**

**4.1.11 Network Administrator**

**4.1.12 Convergence and Voice Over IP (VOIP)**

**4.1.13 Network Security Attacks**

**4.1.14 Wireless**

**4.1.15 VLAN and SDN**

**4.1.16 Wide Area Networks (WAN)**

## **4.2 Secure network components**

### **4.2.1 Network Architecture**

### **4.2.2 Firewall Technologies**

### **4.2.3 Firewall Architectures**

### **4.2.4 IDS and IPS**

### **4.2.5 Sandbox**

### **4.2.6 Honeypots and Honeynets**

### **4.2.7 Endpoint Security (e.g., host-based)**

## **4.3 Implement secure communication channels according to design**

### **4.3.1 Tunneling and VPNs**

### **4.3.2 IPsec**

### **4.3.3 SSL/TLS**

### **4.3.4 Remote Authentication**

## **MINDMAP REVIEW VIDEOS**

## **PRACTICE QUESTIONS**

## **DOMAIN 5: Identity & Access Management (IAM)**

## 5.1 Control physical and logical access to assets

### 5.1.1 Access Control

### 5.1.2 Administration Approaches

## 5.2 Design identification and authentication strategy

### 5.2.1 Access Control Services

### 5.2.2 Identification

### 5.2.3 Authentication by Knowledge

### 5.2.4 Authentication by Ownership

### 5.2.5 Authentication by Characteristics

### 5.2.6 Factors of Authentication

### 5.2.7 Credential Management Systems

### 5.2.8 Single Sign-on (SSO)

### 5.2.9 CAPTCHA

### 5.2.10 Session Management

### 5.2.11 Registration and Proofing of Identity

### 5.2.12 Authenticator Assurance Levels (AAL)

### 5.2.13 Federated Identity Management (FIM)

**5.2.14 Federated Access Standards**

**5.2.15 Accountability = Principle of Access Control**

**5.2.16 Just-in-time (JIT) Access**

**5.3 Federated identity with a third-party service**

**5.3.1 Identity as a Service (IDaaS)**

**5.4 Implement and manage authorization mechanisms**

**5.4.1 Discretionary Access Control (DAC)**

**5.4.2 Mandatory Access Control (MAC)**

**5.4.3 Non-discretionary Access Control**

**5.4.4 Access Policy Enforcement**

**5.5 Manage the identity and access provisioning life cycle**

**5.5.1 Vendor Access**

**5.5.2 Identity Life Cycle**

**5.5.3 User Access Review**

**5.5.4 Privilege Escalation**

**5.5.3 Service Account Management**

**5.6 Implement Authentication Systems**

## **5.6.1 Authentication Systems**

### **MINDMAP REVIEW VIDEOS**

### **PRACTICE QUESTIONS**

## **DOMAIN 6: Security Assessment and Testing**

**6.1 Design and validate assessment, test, and audit strategies**

**6.1.1 Validation and Verification**

**6.1.2 Effort to Invest in Testing**

**6.2 Conduct security control testing**

**6.2.0 Testing Overview**

**6.2.1 Testing Techniques**

**6.2.2 Vulnerability Assessment and Penetration Testing**

**6.2.3 Vulnerability Management**

**6.2.4 Vulnerability Scanning**

**6.2.5 Log Review and Analysis**

**6.2.6 Limiting Log Sizes**

**6.2.7 Operational Testing—Synthetic Transactions and RUM**

**6.2.8 Regression Testing**

**6.2.9 Compliance Checks**

**6.3 Collect security process data (e.g., technical and administrative)**

**6.3.1 Key Risk and Performance Indicators**

**6.4 Analyze test output and generate report**

**6.4.1 Test Output**

**6.5 Conduct or facilitate security audits**

**6.5.1 Audit Process**

**6.5.2 System Organization Controls (SOC) Reports**

**6.5.3 Audit Roles and Responsibilities**

**MINDMAP REVIEW VIDEOS**

**PRACTICe QUESTIONS**

**DOMAIN 7: Security Operations**

**7.1 Understand and comply with investigations**

7.1.1 Securing the Scene

7.1.2 Evidence Collection and Handling

7.1.3 Locard's Exchange Principle

7.1.4 Digital/Computer Forensics

7.1.5 Chain of Custody

7.1.6 Five Rules of Evidence

7.1.7 Types of Investigations

7.2 Conduct logging and monitoring activities

7.2.1 Security Information and Event Management (SIEM)

7.2.2 Continuous Monitoring and Tuning

7.2.3 Security Orchestration, Automation, and Response (SOAR)

7.3 Perform configuration management (CM)

7.3.1 Asset Inventory

7.3.2 Configuration Management

7.4 Apply foundational security operations concepts

7.4.1 Foundational Security Operations Concepts

**7.5 Apply resource protection techniques**

**7.5.1 Protecting Media**

**7.6 Conduct incident management**

**7.6.1 Incident Response Process**

**7.7 Operate and maintain detective and preventive measures**

**7.7.1 Malware**

**7.7.2 Anti-malware**

**7.8 Implement and support patch and vulnerability management**

**7.8.1 Patch Management**

**7.9 Understand and participate in change management processes**

**7.9.1 Change Management**

**7.10 Implement recovery strategies**

**7.10.1 Failure Modes**

**7.10.2 Backup Storage Strategies**

**7.10.3 Spare Parts**

**7.10.4 Redundant Array of Independent Disks (RAID)**

**7.10.6 Recovery Site Strategies**

**7.11 Implement disaster recovery (DR) processes**

**7.11.1 BCM, BCP, and DRP**

**7.11.2 RPO, RTO, WRT, and MTD**

**7.11.3 Business Impact Analysis (BIA)**

**7.11.4 Disaster Response Process**

**7.11.5 Restoration Order**

**7.12 Test disaster recovery plans (DRP)**

**7.12.1 BCP and DRP Testing**

**7.13 Participate in business continuity (BC) planning and exercises**

**7.13.1 Goals of Business Continuity Management (BCM)**

**7.14 Implement and manage physical security**

**7.15 Address personnel safety and security concerns**

**MINDMAP REVIEW VIDEOS**

**PRACTICe QUESTIONS**

## **DOMAIN 8: Software Development Security**

**8.1 Understand and integrate security in the software development life cycle (SDLC)**

**8.1.1 Security's Involvement in Development**

**8.1.2 SDLC and SLC**

**8.1.3 Development Methodologies**

**8.1.4 Maturity Models**

**8.1.5 DevOps**

**8.1.6 Canary Testing and Deployments**

**8.2 Identify and apply security controls in software development ecosystems**

**8.2.1 Software Development Overview**

**8.2.2 Code Obfuscation**

**8.2.3 DBMS, Concurrency, and Lock Controls**

**8.2.4 Metadata**

**8.2.5 Development Ecosystems**

**8.3 Assess the effectiveness of software security**

**8.3.1 Software Security Assessment Methods**

## **8.4 Assess security impact of acquired software**

### **8.4.1 Acquiring Software**

## **8.5 Define and apply secure coding guidelines and standards**

### **8.5.1 Secure Coding Guidelines**

### **8.5.2 Buffer Overflow**

### **8.5.3 Application Programming Interfaces (APIs)**

### **8.5.4 Secure Coding Practices**

### **8.5.5 Software Development Vulnerabilities**

## **MINDMAP REVIEW VIDEOS**

## **PRACTICE QUESTIONS**

## **REFERENCES AND FURTHER READING**

## **ACRONYMS**

## **INDEX**

## **PROVEN EXAM STRATEGIES**

**The CISSP Exam — What to Expect**

**How to Read and Understand the Question**

**How to Select the BEST Answer**

**The CISSP Mindset**

**Final Preparations and Exam Day**

## WHY THIS BOOK

Welcome to *Destination CISSP: A Concise Guide*. We're glad you've invited us to join you on your journey toward earning the highly coveted Certified Information Systems Security Professional (CISSP) designation. More than joining you, we hope you'll allow us to guide you to success with our cumulative years of practical experience as security professionals and educators.

The goal of this guide is simple: to help you pass the CISSP exam and to provide you with a foundation of security knowledge that will equip you to be a better security professional and benefit you throughout your career.

As a CISSP, you will be viewed as a competent and knowledgeable security professional, a decision-maker, and a leader. The CISSP exam requires candidates to assimilate a vast array of information spread across eight (8) domains that make up the Common Body of Knowledge (CBK) of security. This book was created with one unique goal in mind: to be a concise guide that still has enough information to help readers understand the concepts behind each domain in a simple and digestible manner.

As many people say, the CISSP exam is “a mile wide and an inch deep,” and most candidates bring experience and

strength in only a handful of the domains. This guide will help you supplement knowledge you lack and enhance your current understanding of concepts to a deeper degree, which is required for success on this very difficult exam.

## **Who is the CISSP meant for?**

The CISSP certification is excellent for managers, executives and security practitioners who wish to demonstrate their knowledge and competence across a range of security domains. Common roles include:

- High-Level Management
- Chief Information Security Officer (CSO/CISO)
- Chief Information Officer (CIO)
- Director of Security/Risk & Compliance
- IT Director/Manager
- Security Systems Engineer
- Security Analyst

## **Value of the CISSP certification**

The CISSP certification is a gold standard certification that is globally recognized and respected. Earning the CISSP

certification tells employers and peers that you are an extremely knowledgeable and competent security professional; additionally, it indicates that you are disciplined and committed to professional security development.

## **How to best use this book**

Everybody learns differently, but for the sake of your exam preparation, our guide may serve as one of your study resources. Based upon our years of teaching experience, we've aligned the guide with the 2024 CISSP Exam Outline and included what we believe to be the most relevant and important material that will help you prepare efficiently and confidently. We recommend that you read the material as presented in order to most effectively synthesize the information and concepts pertinent to the exam. Additionally, through our own experience, we've found that reading with a highlighter and notebook in hand can help you identify and retain important information.

## **What are “Core Concepts” and “Expect to be tested on?”**

The “Core Concepts” and “Expect to be tested on” callouts sprinkled throughout the Guide reflect what we believe to be the most important information a student should focus

on. By specifically calling out concepts and information, our goal is to help students stay focused and on track.

#### CORE CONCEPTS

- Summary of key concepts in each section

- Refers to specific topics that you might see questions about on the CISSP exam

For purposes of your studies, you should pay particular attention to these items.

## **ABOUT THE EXAM**

The broad spectrum of topics included in the CISSP Common Body of Knowledge (CBK) ensure its relevance across numerous disciplines in the field of cybersecurity. Successful candidates are expected to be competent in the following eight domains: 1. Security and Risk Management 2. Asset Security 3. Security Architecture and Engineering 4. Communication and Network Security 5. Identity and Access Management (IAM) 6. Security Assessment and Testing 7. Security Operations 8. Software Development Security If you want to review the full CISSP exam outline, you can find it at <https://www.isc2.org/CISSP-Exam-Outline>.

The CISSP exam tests much more than simply rote memorization. It tests the deep comprehension and application of knowledge. The exam tests understanding and application of concepts, but more importantly, it tests competence; this is why you'll commonly see multiple answers that seem correct for an exam question, but only one of those answers is in fact the correct (best) answer. Another way of putting this is that the exam answers may offer several true statements, but only one will be the most correct answer based on what the question is asking.

Oftentimes, a question will seek the answer that represents the best, least, or most appropriate answer, among the

options offered, and usually two or more possible answers will seem to fit the criteria. However, only one answer fits the criteria in the best possible way. Sometimes, no answer may seem correct. The material needs to be understood at a deep level to answer the most challenging questions.

There are three different types of questions that you may encounter on the CISSP exam:

- **Multiple-choice:** where you will be given a question and four answers, and you must select the BEST answer. The vast majority of questions on the exam will be multiple choice.

- **Scenario-based:** where a scenario will be presented, essentially a few paragraphs of text, and there will be a multiple choice-question pertaining to the scenario.

- **Drag-and-drop:** where two lists—one on the left and one on the right—need to be matched. Terms and definitions are most often used in these types of questions.

The exam typically requires at least two to four months of preparation. This timing will heavily depend on your existing skills, knowledge, background, experience, and the training courses you take. Once you achieve your CISSP certification, Continuing Professional Education (CPE) requirements

ensure ongoing professional and knowledge development in all the areas covered by the exam.

## April 2024 Exam Change Summary

The CISSP exam is constantly being reviewed, and ISC2 typically refreshes the exam outline about every three years to ensure alignment with the ever-evolving security landscape. In the words of ISC2:

*ISC2 has an obligation to its membership to maintain the relevancy of its credentials. These enhancements are the result of a rigorous, methodical process that ISC2 follows to routinely update its credential exams. This process ensures that the examinations and subsequent continuing professional education requirements encompass the topic areas relevant to the roles and responsibilities of today's practicing cybersecurity professionals with the knowledge, skills and abilities to lead an organization's information security program.*

The 2024 refresh resulted in very minor overall changes to the exam.

However, there have been slight changes to the weights:

Domain	2021 Weight	2024 Weight	Change
1 Security & Risk Management	15%	16%	+1%

# MINDSET

The most important concept to highlight for passing the CISSP exam is adopting a **managerial mindset** when answering questions versus reverting to a deeply technical “fix-it-now” approach.

We highly recommend that you watch our free video on how to **“Think like a CEO”**: [dcgo.ca/thinkCEO](http://dcgo.ca/thinkCEO)

**Thinking Like a CEO** is how we summarize this extremely important mindset for every student in our classes. You need to approach your studies and especially the exam with the right mindset. You must be focused on helping the business achieve its goals and objectives. You must be focused on helping the organization create more value. Essentially, as a security professional, you are an enabler. You must focus on ensuring that the goals and objectives of the security function are aligned with the goals and objectives of the organization. You need to **think like a CEO**.

One of the authors of this guide, John Berti, has been involved with ISC2 for over twenty years; in fact, he was one of the authors of the first Official CISSP guidebook, the *Official ISC2 Guide to the CISSP Exam* (ISC2 Press), in the mid 2000s. According to John, the questions that make it into the exam pool have gone through multiple rounds of

vetting and consideration to make them perfect questions that thoroughly test competence. **Finding the best answer to most questions is a lot easier if you have the foundation of knowledge as well as the perspective that comes from thinking like a CEO.**

Testing science is applied to the creation of real CISSP exam questions, which makes them excellent to test competence. Ultimately, this enhances the value of the CISSP certification as well as the difficulty of the exam. It's not a "study, memorize, and pass" exam; it's an exam requiring competence, experience, and a methodical approach.

## **ABOUT THE AUTHORS**

### **Rob Witcher**

Rob is one of the driving forces behind the success of the Destination Certification CISSP program. He is a technical wizard, directing the creation of the integrated intelligent learning system.

Rob has over twenty years of intense security, privacy, and cloud assurance experience, including:

- Guiding multiple companies in responding to and recovering from (global headline level) security and privacy breaches
- Leading PCI readiness engagements, SOC2 audits, cloud assessments, and security maturity reviews
- Managing the development of multiyear security strategies and enterprise-wide privacy operating models
- Acting as the CIO of a global mining company

Rob has delivered hundreds of CCSP, CISSP, and ISACA classes globally.

Rob is a dedicated security professional and creative instructor who is deeply invested in the success of our students. He brings an entertaining delivery style that is grounded in years of experience and a deep understanding of what is required for success on the CISSP exam.

### **John Berti**

John is the other driving force behind the success of the Destination Certification CISSP program. With over thirty years in the field, a wealth of global experience, and an exceptional ability to make complex topics simple, John brings the CISSP concepts to life through out of the box teaching approaches that lead to our industry-high exam success rates.

John is one of Canada's leading Information Security professionals with outstanding credentials:

- Over thirty years of Cyber Risk and Security experience in the industry.

- Over twenty years of practical involvement in, experience with, and advising to ISC2.
- Coauthored the best-selling CISSP exam preparation guide *Official ISC2 Guide to the CISSP Exam*.
- Relevant involvement in helping ISC2 develop materials for the official CISSP curriculum, the CCSP curriculum, and sample CISSP exam questions.

John has facilitated hundreds of classes worldwide and quite literally wrote the book on CISSP exam preparation.

## **Lou Hablas**

Lou has almost 30 years of working in the technology industry, with roles that have included:

- Managing the

Identity and Access Management function at an Olympic venue during the 1996 Olympic Games ■ Network Administrator for the retail securities division of a major Southeastern bank ■ Consultant with a leading Microsoft-centric management consulting and technology services firm ■ IT Director for a global non-profit Lou enjoys helping others succeed and is passionate about using written and verbal communication to simplify and convey concepts. He especially enjoys celebrating students after they pass the challenging CISSP exam!

## **Nick Mitropoulos**

Nick has two decades of experience in Security Operation Centres, threat intelligence, data loss prevention, and incident handling. He is a world-renowned SANS, CompTIA and ISC2 instructor, travelling the world and providing security trainings and insight at various public conferences and privately held events. He holds a BSc with distinction in Information Technology and Telecommunications as well as an MSc with distinction in Advanced Security and Digital Forensics and numerous accolades and certifications in the industry, like the following: ■ More than forty-five security certifications, including:

- ISC2 – CISSP, CSSP, and SSCP.
- ISACA – CISM.

- GIAC – GSEC, GCLD, GBFA, GWAPT, GPEN, GCIH and GISF.
- Member of the SANS global CISO network, GIAC advisory board, senior IEEE member, BCS, ISACA, Cisco Champion, and EC-Council global CISO advisory board.
- Author of McGraw-Hill's *SSCP Systems Security Certified Practitioner Practice Exams* and *GCIH GIAC Certified Incident Handler All-in-One Exam Guide* and co-author of *Destination CISSP: A Concise Guide*.
- Winner of the CEH Hall of Fame 2021 and United Nations Hall of Fame awards.

If you want to reach out to Nick for security advice or to engage for training or public speaking, please don't hesitate to do so by using [info@scarlet-dragonfly.com](mailto:info@scarlet-dragonfly.com). He's also on X ([@MitropoulosNick](https://twitter.com/MitropoulosNick)) and LinkedIn (<https://www.linkedin.com/in/nickmitropoulos>).

**REVISION EDITOR****Josh Lake**

Josh is a cybersecurity writer, researcher, and editor with nearly a decade of experience. He has written widely about a range of privacy and information security issues, with a particular interest in cryptography and IAM. He is passionate about making complex topics easier to understand and helping students further their careers.

## **TECHNICAL REVIEWER**

### **Taz Wake**

Taz has worked in a variety of security roles since 1993. Since then, his work has taken him across the globe in a variety of roles for government agencies and private sector organizations. Moving into the private sector, Taz founded Halkyn Consulting as a boutique security and risk management consultancy delivering technical security advice to businesses worldwide. Since forming the consultancy, Taz has developed CISRTs for multinationals, provided expert Digital Forensics and Incident Response services to a range of companies, and regularly provided specialist training to forensic science labs ■ Holds multiple physical and cybersecurity certifications including CPP, CISSP, CISM, CRISC, CEH, GXPN, GCFA, GCFE, GCIH, GCIA, and more ■ SANS instructor for FOR508 Advanced Incident Response, Digital Forensics, and Threat Hunting course ■ SANS course author for FOR608 Enterprise Incident Response ■ SANS “Lethal Forensicator” and multiple challenge coin winner; 2x winner of the Core Netwars tournament ■ Regularly active on Hack the Box, Try Hack Me, Immersive Labs, and other CTF platforms.

# NOTES ON THE BOOK

## **What's up with the mixed case in the titles?**

As you navigate this guidebook, you might find yourself wondering about the use of mixed case in the titles and headings. For level 1 and 2 titles/headings, we used the exact wording and case as shown in the CISSP Exam Outline published by ISC2. Titles at levels 3 and 4 reflect our preference and tend to be more uppercase.

## **Hey! I found a mistake in the book!**

We're a small team that worked incredibly hard to create a CISSP guidebook that we hope will be instrumental in helping you pass the CISSP exam. We have devoted a huge amount of effort into making this book and it's gone through multiple rounds of reviews. However, we are only human, and as much as it irks us, we're pretty sure the odd mistake has evaded us. If you find a mistake, we'd greatly appreciate it if you could let us know so that we can fix it:  
[cisspguide@destcert.com](mailto:cisspguide@destcert.com)

Thanks, and all the very best in your studies!

Rob, John, Lou, & Nick



## INTRODUCTION

The role of security has evolved significantly over the years. Simply focusing on protecting data on a server is no longer enough. Threat actors now target an incredibly broad spectrum of assets across an organization, including a variety of devices such as mobile phones, tablets, industrial controllers, and even smart fridges and sensors. The attack vectors have also evolved, and there's a large increase of phishing emails and other social engineering attacks that try to bypass defenses and take advantage of the weakest element in the security chain: people.

Given the evolution of the security field, one of the fundamental questions for any security professional to consider is: **What is the role of the security function in every organization?**

A solid understanding of the answer to this question will not only make you a better security professional but will also

make it much easier to pass the demanding and difficult CISSP exam.

Answers to the question “What is the role of the security function in every organization?” will vary, depending on who is answering. Often, answers will include items such as:

- Reduce risk
- Protect information, IT assets, the company, and its reputation
- Preserve confidentiality and integrity
- Manage availability
- Ensure compliance

All the items listed above equate to one phrase that corporate governance focuses on: **Organizational VALUE**.

Security cannot focus solely on protecting data or information, as these are just some of the things that represent value to any organization.

**Security must enable and support the organization in achieving its goals and objectives.** Gone are the days where security existed only to minimize risk or tick a box. While it is still necessary to conduct risk analysis and implement controls to address risks, this needs to be done with a top-down approach and direct input from upper management to ensure the security controls that are implemented help the business achieve its goals and objectives.

Security also protects people, hardware, software, intellectual property, concepts, products, services, and

corporate reputation—anything of value. It allows an organization to achieve compliance with laws, regulations, and industry standards, and it protects against various risks.

How can security address all these things if it is reporting to Information Technology (IT)?

The CEO (Chief Executive Officer) is accountable for managing the organization in such a way that ultimately allows it to increase its value, through adhering to a set of rules, practices, and processes; this is governance.

In many organizations, the security function is led by the CISO (Chief Information Security Officer). Information is just one example of the important assets of any organization that security needs to protect. Another frequently used title for those leading a security function is CSO (Chief Security Officer). Often enough, the CSO then reports to the chief information officer (CIO), which can hinder the goal of security. **Security nowadays needs to be empowered to protect ALL the assets of the organization and to do that, it needs to report to those who are accountable for the company. That is, either the CEO or the corporate Board of Directors.**

The key takeaway is that to be a better security professional and to pass the CISSP exam, you must first understand

security from a management point of view rather than simply a technical one.

**As a security professional, you must always focus on helping the organization achieve its goals and objectives. You must be an enabler to the business.**



DOMAIN 1

# Security and Risk Management

## DOMAIN 1

### **SECURITY AND RISK MANAGEMENT**

The first CISSP domain focuses on the fundamentals of security and how to assess and manage risk. You will learn the concepts of the CIA triad, gain insight into core organizational roles and how they relate to security, and understand the important difference between accountability and responsibility, in addition to corporate laws (policies) and key processes like risk analysis. This domain also focuses heavily on the key factors of governance and compliance, and how security helps by being aligned and contributing to each.

#### **1.1 Understand, adhere to, and promote professional ethics**

##### **1.1.1 ISC2 Code of Professional Ethics**

As a CISSP candidate, you are responsible for understanding and complying with the ISC2 Code of Professional Ethics, which applies to CISSP holders around the globe. In fact, the CISSP exam will most likely ask at least one question on this topic. The Code of Ethics Preamble and Canons are noted below. **It is important that the Preamble and the Code of Professional Ethics Canons be understood fully in the**

**context of corporate and industry application, and the Canons should be memorized and adhered to in the order presented.**

### **ISC2 Code of Ethics Preamble**

- The safety and welfare of society and the common good, duty to our principals, and to each other, requires that we adhere, and be seen to adhere, to the highest ethical standards of behavior.
- Therefore, strict adherence to this Code is a condition of certification.

Agreement with and strict adherence to this code is a condition of gaining and maintaining the CISSP certification. The ISC2 Code of Ethics consists of the Canons outlined in [Table 1-1](#).

#### **Wording and order of the ISC2 Code of Ethics Canons**

<b>ISC2 Code of Ethics Canons</b>	
<b>1</b>	Protect society, the common good, necessary public trust and confidence, and the infrastructure.
<b>2</b>	Act honorably, honestly, justly, responsibly, and legally.

3	Provide diligent and competent service to principals.
4	Advance and protect the profession.

Table 1-1: ISC2 Code of Ethics Canons

### How to apply Ethics Canons in various scenarios and contexts

In both the Preamble and the Canons, the topics are in order of importance, and again, **all these items should be memorized as presented. Remember, if a scenario is presented in which there's a conflict in the Canons, they need to be applied in order.**

#### 1.1.2 Organizational Code of Ethics

##### CORE CONCEPTS

- Ethics are based on doing nothing that is harmful to anyone else
- For organizations to have consistent ethics, they must be codified in corporate laws (policies)
- ISC2 Code of Ethics

Ethics are a foundational element to a successful security program and should be adhered to throughout the organization. The success of any security program requires

the proper ethical support from every level of the organization and therefore needs to be driven by management and instilled through proper support, direction, and enforcement through high-level management. Proper ethical behavior is based upon one belief: abide by the rules and do nothing that is harmful to anyone else. However, this belief comes in the form of a challenge: Though almost every professional follows some form of ethics, they tend to vary widely due to upbringing, culture, education, life experiences, religious beliefs, and so on. Thus, most people will pursue a course of action—a course of ethical behavior—based upon what they believe is ethically correct. So, although ethical behavior can help promote a good and secure working environment, there are likely a wide variety of ethical lenses forming the work landscape, especially in a large organization. How, then, can ethical behavior be pursued in a consistent manner to ensure that all employees employ the same set of ethics?

Within an organization, the best way to prescribe, promote, and instill consistent ethical behavior is through the use of corporate rules or laws, more appropriately referred to as policies. Policies that promote sound and consistent ethical behavior help make an organization a better place to work and more valuable to shareholders and to the communities where they operate. Policies must be legal, and adherence to and promotion of them must start with senior

management and be consistently communicated to every employee.

## 1.2 Understand and apply security concepts

### CORE CONCEPTS

- Security must support the business in achieving its goals and objectives
- Security must increase the value of the organization

As outlined in the introduction, the role of security has evolved to become more fully integrated with business processes. For example, for many years, the IT or Information Security function didn't consider physical security to be part of their purview. However, there are a lot of physical assets that an organization owns that don't strictly relate to data—like people—that need protection. Security focuses on anything that represents value, better referred to as assets, and implements controls that ultimately increase the value of those assets. Security should not focus only on information, or data, as this is just one example of assets that represent value to organizations and therefore need to be protected based on that value.

In summary, the focus of the security function is to:

1. Allow and enable the organization to achieve its goals and objectives
2. Increase the organization's value

Security, therefore, is in a support role. Through proper security governance, those who are accountable for increasing the value of the organization can be supported and enabled to achieve their goals.

### **1.2.1 Confidentiality, Integrity, Availability, Authenticity, and Nonrepudiation**

#### **CORE CONCEPTS**

- **Confidentiality:** Protects and prevents unauthorized disclosure
- **Integrity:** Protects and adds value to assets by making them more accurate, more timely, more current, more meaningful
- **Availability:** Ensures organizational assets are available when required by stakeholders
- **Authenticity:** Proves the source and origin of important valuable assets
- **Nonrepudiation:** Provides assurance that someone cannot deny having done something

#### **Definitions of Confidentiality, Integrity, Availability, Authenticity, and Nonrepudiation**

[Figure 1-1](#) depicts a classic security model known as the CIA triad. The CIA triad is a foundational model that helps organizations design, structure, and implement the security function.



**Figure 1-1: CIA Triad**

The elements of the CIA triad are outlined in [Table 1-2](#):

<b>Confidentiality</b>	Protects assets using important principles such as need-to-
------------------------	---

	know and least privilege; prevents unauthorized disclosure
<b>Integrity</b>	Protects and adds value to assets by making them more accurate, more timely, more current, more meaningful; prevents unauthorized or accidental changes to assets such as information
<b>Availability</b>	Protects critical assets based on value to ensure organizational assets are available when required by stakeholders

Table 1-2: **CIA Triad**

These are the core pillars of security, and, even though referred to as the goals of information security, this is a narrow view of what security needs to focus on today. The goals of the three pillars—Confidentiality, Integrity, and Availability—need to be applied to information and everything else (assets) that represents value to the organization. In other words, security and the core pillars should be referred to as the “goals of asset security” and not just “the goals of information security.”

The traditional pillars of security have been increased to include authenticity and nonrepudiation, outlined in [Table 1-3](#). Together with the CIA triad, we often refer to these as the **five pillars of information security**.

<b>Authenticity</b>	Proves assets are legitimate and bona fide, and verifies that they are trusted and verified. Proves the source and
---------------------	--

	origin of important valuable assets. Also referred to as "proof of origin."
<b>Nonrepudiation</b>	Provides assurance that someone cannot dispute the validity of something; the inability to refute accountability or responsibility. Also, inability to deny having done something.

**Table 1-3: Authenticity and Nonrepudiation**



## **1.3 Evaluate, apply, and sustain security governance principles**

### **1.3.1 Alignment of the Security Function to Business Strategy, Goals, Mission, and Objectives**

#### **CORE CONCEPTS**

- The goal of governance is to enhance organizational value
- Corporate governance is based upon the goals and objectives of the organization
- Security must be managed top down instead of bottom up
- Scoping and tailoring are used to align security objectives with organizational goals and objectives
- Security governance must be aligned with corporate governance

The word *governance* can be defined as the act of governing or overseeing the process of directing something. In other words, governance means to govern properly to allow the organization to achieve its goals and objectives focused on increasing the value of the organization. Those activities can be referred to as corporate governance activities, and examples may include creating new processes, new products, new services, striking new relationships with third parties, improving margins and cash flow, creating new systems and procedures, reducing risk, meeting compliance requirements, and so on. All of these

are just a few examples of corporate governance activities that organizations will implement/create to increase value.

Security governance will therefore include all those activities, initiatives, and programs that the security function will drive, initiate, and support, which should always be aligned, focused, and contributing toward those corporate governance activities mentioned above that will ultimately increase the value of the organization. The important point to remember is that this alignment can only be assured in a top-down structure. Those who are accountable for corporate governance activities need to be the ones who drive what security needs to do, to ensure alignment and proper contribution from security to add value and ultimately achieve the goals and objectives of the organization.

Security should be a proactive enabler rather than a reactive function, but this requires senior management to have strong convictions about the need for security. If there is a lack of support, and lack of conviction on management's part, what do you do? You could educate and convince them of the value of security via an internal champion or perhaps even via the hiring of external consultants.

Security needs to enable the organization's goals and objectives, not just enforce information processes or fix

technical issues. Security governance must align with corporate governance, and security's goals and objectives should be driven by the organization's goals and objectives.

To best understand what is meant by the terms *corporate governance* and *security governance*, it is important to first understand what is meant by the term *governance*. At the heart of the word *governance* is *govern*, which means "to lead;" but, for what purpose does governance exist? When officials are elected to government, whether at the local, state, national, or federal level, what are they being elected to do? Ultimately, they're being elected to enhance or increase the value of whatever jurisdiction they will govern by providing better services and better meeting the needs of their constituents. Extending this definition, organizations need people to govern too, with the goal of increasing the value of the organization. Just like every country has officials who are elected to provide for the people and the country itself—in other words, to provide governance—organizations need a similar structure. Who are the members of the governing body of any organization? Typically, this would be the Board of Directors, the CEO, and senior management; their goal is to increase the value—the prosperity, the sustainability, and the viability—of the organization.

Instead of enacting things like local and municipal laws, organizations enact corporate laws called policies that allow the organization and its stakeholders to thrive. The Board of Directors should establish the organizational goals and objectives and set the tone for governance, but it can't necessarily oversee the continuous monitoring and proper implementation of the elements related to these principles. This is why the Board usually appoints an individual to be accountable for corporate governance. This individual—the CEO—therefore becomes directly accountable for corporate governance, or all the activities and initiatives that the organization undertakes to achieve its goals and objectives.

Extending the top-down perspective, it follows that security in any organization is only as good as its leadership; in other words, for security to be effective and for employees to be committed to the need for good security, the Board, CEO, and senior management must adopt, promote, and consistently communicate a security culture.

### **Aligning security governance with corporate governance**

Security governance can be best aligned with corporate governance when it draws on the knowledge and experience of senior and upper management, HR, Legal, IT,

and key functional areas of the organization. Specifically, based upon the expertise from functional areas like Legal, for example, security can know which laws and regulations need to be followed by the organization. The best way any organization can establish and maintain sound organizational governance that aligns with security is through the establishment of an Organization Governance committee, charged with establishing and promoting the top-down governance structure and tone that is critical to an organization achieving its goals and objectives. This committee should meet regularly and include security goals and objectives in its organization. Put simply, the goals and objectives of the security function must be directly aligned with the goals and objectives of the organization.

## Scoping and tailoring

Scoping and tailoring are important processes to ensure controls are properly aligned with organizational goals and objectives. **Scoping** looks at potential control elements and determines **which ones are in scope**—for example, security control elements that could adhere to applicable laws and regulations—and **which ones are out of scope**. In other words, based on the previous example, those security elements that best align with and support the goals and

objectives of an organization from a legal perspective would be considered in scope.

**Tailoring** looks specifically at applicable—in scope—security control elements and further **refines or enhances** them so they're most effective and aligned with the goals and objectives of an organization. This is done from the perspective of each functional area. They should be cost-effective in relation to what they're protecting, and they should ultimately help add value to the organization. When done well, security governance is completely aligned with corporate governance, and the goals and objectives of an organization can be fulfilled in a manner that is cost-effective and adds value.

If the Board of Directors and senior management don't support the security function, security simply becomes a reactive nuisance versus a proactive enabler.

Starting from the top, if the Board and senior management are convicted—*absolutely convinced*—of the need for robust security that is aligned with the strategy, goals, mission, and objectives of the organization, the security function will be viewed as a great asset and an organizational enabler.

As was mentioned earlier with regards to governance, the CEO is ultimately accountable for guiding the organization

and helping it achieve its goals and objectives in order to add value. However, as the roles and responsibilities listed above allude to, accountability can exist elsewhere within an organization. For example, the CFO often is accountable for the accuracy of financial reports, and the Data Controller is accountable for privacy.

### 1.3.2 Organizational Processes

Security needs to be an integral part of all organizational processes. One example that requires special consideration is during **acquisitions of other companies**. Organizations face increased risk during acquisitions and mergers, because they have limited visibility and control over the other organization being acquired. Similarly, if an organization goes through the process of **divestiture** and sells off some of its assets, the sales process needs to ensure that the organization's security, compliance, and other obligations are not compromised. **Governance committees** that include a focus on security can play a vital role in protecting organizations during these periods.

## Accountability versus Responsibility

### CORE CONCEPTS

- Only one person or group or entity must be accountable
- Multiple people can be responsible

■ Accountability can never be delegated

■ Responsibility can be delegated

At this point, it is important to understand the difference between two very important terms that are sometimes mistakenly used interchangeably: **accountable and responsible**. These two words do not share the same meaning. The word accountable was used earlier very deliberately. How does being accountable differ from being responsible? If someone is accountable for something, that accountability can never be delegated to anyone else. That person will always remain accountable. Responsibility, on the other hand, can be delegated, but the delegator will remain accountable. This explains why security is everyone's responsibility, yet the accountability for security remains with those who are focused on corporate governance—the Board, the CEO and other C-level members, and the owners of assets. If something that negatively impacts value happens in an organization, the CEO is ultimately accountable.

From a functional point of view, delegating responsibilities to the right person or team makes perfect sense and is usually the most effective and efficient means by which an organization achieves its goals and objectives. It's important to know and understand the difference between

accountability and responsibility. [Table 1-4](#) highlights the major differences between them:

Accountability	Responsibility
Where the buck stops	The doer
Ultimate ownership and liability	In charge of a task or process
Only one person or group can be accountable	Multiple people can be responsible
Sets rules and policies	Develops plans and implements controls

**Table 1-4: Accountability and Responsibility**

### Accountability vs. responsibility

Even if certain functions of the organization are managed by a responsible third party, like a payroll or Cloud Service Provider (CSP), **accountability still resides with the owner of the assets being managed**. To expand on this thought, because it's more and more applicable these days due to the prevalence of cloud-based computing, the owner of any and all data stored in the cloud is accountable for that data. A CSP will often have a contractual-based responsibility for protecting the data, but **the owner of the data is always**

**accountable for the data and therefore liable if there is a data breach.**

### **Who is ultimately accountable for security? Upper management, the CEO, or the Board of Directors?**

Ultimately, the individuals accountable for every single asset in the organization are the Board and the CEO. Senior management are also accountable for the assets that they manage. This is important, because it's obviously not practical for the CEO to be accountable for every single asset in a large company. On the CISSP exam, if a question asks who is ultimately accountable for the finance system, the best answer would be the VP of Finance—but if they aren't listed, the next best answer is the person above them in seniority. Accountability can never be delegated.

What accountability does the security function hold? The security function is accountable for security governance activities that have been driven or initiated by those who are accountable: the Board, the CEO, other C-suite executives, and upper management.

#### **1.3.3 Organizational Roles and Responsibilities**

##### **CORE CONCEPTS**

- The role of security is to be an enabler

- **The owner/controller is the person that created, bought, or is most familiar with an asset**
- **The processor is the person, function, or group responsible for data and who do things on behalf of the controller**

**Table 1-5** outlines some of the key functions typically found in an organization and their accountabilities and responsibilities from a security perspective.



<b>Owners / Controllers/ Functional Leaders / Senior Management</b>	<b>Accountable</b> for: <ul style="list-style-type: none"> <li>■ Ensuring that appropriate security controls, consistent with the organization's security policy, are implemented to protect the organization's assets ■</li> <li>■ Determining appropriate sensitivity or classification levels</li> <li>■ Determining access privileges</li> </ul>
---	--

<b>Information Systems Security Professionals / IT Security Officer</b>	<p><b>Responsible</b> for:</p> <ul style="list-style-type: none"> <li>■ Design, implementation, management, and review of the organization's security policies, standards, baselines, procedures, and guidelines</li> </ul>
<b>Information Technology (IT) Officer</b>	<p><b>Responsible</b> for:</p> <ul style="list-style-type: none"> <li>■ Developing and implementing technology solutions</li> <li>■ Working closely with IS and IT Security Professionals and Officers to evaluate security strategies</li> <li>■ Working closely with Business Continuity Management (BCM) team to ensure continuity of operations should disruption occur</li> </ul>
<b>IT Function</b>	<p><b>Responsible</b> for:</p> <ul style="list-style-type: none"> <li>■ Implementing and adhering to security policies</li> </ul>
<b>Operator / Administrator</b>	<p><b>Responsible</b> for:</p> <ul style="list-style-type: none"> <li>■ Managing, troubleshooting, and applying hardware and software patches to systems as necessary</li> <li>■ Managing user permissions, per the owner's specifications</li> <li>■ Administering and managing specific applications and services</li> </ul>
<b>Network Administrator</b>	<p><b>Responsible</b> for:</p> <ul style="list-style-type: none"> <li>■ Maintaining computer networks and resolving issues with them</li> <li>■ Installing and configuring networking equipment and systems and resolving problems</li> </ul>
<b>Information Systems Auditors</b>	<p><b>Responsible</b> for:</p> <ul style="list-style-type: none"> <li>■ Providing management with independent assurance that the security objectives are appropriate</li> <li>■ Determining whether the security policy, standards, baselines, procedures, and guidelines are appropriate and effective to comply with the organization's security objectives</li> <li>■ Determining whether the objectives have been met</li> </ul>

Users	<b>Responsible</b> for: ■ Adherence to security policies ■ Preserving the availability, integrity, and confidentiality of assets when accessing and using them
-------	--

Table 1-5: Roles and Accountabilities/Responsibilities

In [Domain 2 \(section 2.3—Provision Information and Assets Securely\)](#), additional roles and responsibilities will be covered specific to information security, including: Data Owner/Controller, Data Processor, Data Custodian, Data Steward, and Data Subject.

One of the roles described in Domain 2—**Custodians**—is often confused with **Owners** (mentioned in [Table 2-3](#)). Where does the word *custodian* originate? The word *custodian* comes from the word *custody*, and it follows that custodians are people or functions that have custody of an asset that does not belong to them; custodians are caretakers or users. The asset belongs to an owner, but the custodian is entrusted with it and is responsible for protecting the asset while it's in their custody. In this case, *protect* means to ensure that the asset's value is not negatively impacted. For example, related to database access, a custodian is responsible for ensuring that the database is available to the users or applications that need access to it; or, regarding confidential assets, a custodian is responsible for ensuring that information is not divulged that might negatively impact the asset's value.

However, what about a situation where, for example, a custodian is responsible for protecting an asset—data, for instance—and the asset becomes corrupted and unusable? Who is responsible? Who is accountable? Referring back to [Table 1-4](#), the responsibility for the corruption rests with the custodian. However, accountability for the corruption rests with the **asset owner**. In other words, referring to points made earlier, it's critical that owners manage their accountability well and ensure that custodians are equipped to manage their responsibility well. Custodians can only take care of this responsibility if security helps. For a custodian to protect the assets in their custody, the right tools, architecture, security controls, knowledge, and skills must exist and be in place. The asset owner is accountable for ensuring this, and this is achieved through the support that the security function should provide.

Who provides all these tools? The security function. The security function performs two critical tasks: 1) makes it easy for custodians and users to perform their job while accessing assets and 2) security enables and equips owners to protect assets in the most efficient, cost-effective way possible.

**Who is accountable for what? Who is responsible for what?**

## **Who is specifically responsible for security? Everyone.**

Everyone has some degree of responsibility for security in their role; for example, the janitor of a locked building must make sure they're not taking confidential papers off someone's desk and that they're disposing of confidential recycling properly. However, asset owners are accountable for telling people what their responsibilities are. Asset owners are in the best position to know the value of the assets they control, and they can best determine how much security is needed to protect those assets. They also need to communicate what should be protected, who should protect it, and how to do so. Security professionals provide advice, but it's not up to them to secure anything. Security is ultimately everyone's responsibility.

Security frameworks, which will be discussed in more detail later, provide guidelines on how to align the security function with corporate governance. Frameworks like NIST, ISO, COBIT, ITIL, and more will be described more fully. For now, it's important to know that security frameworks provide comprehensive guidance on how to structure security properly.

Before moving to the matter of compliance with laws and regulations, let's examine another key component of security management embodied in two phrases: due care and due diligence.

### **1.3.4 Security Control Frameworks**

Most of these topics are discussed in section [\*\*3.3.1 Security Control Frameworks\*\*](#). SABSA is briefly discussed in [\*\*3.2.2 Enterprise security architecture\*\*](#).

### **1.3.5 Due Care versus Due Diligence**

#### **CORE CONCEPTS**

- **Due care is the responsible protection of assets**
- **Due diligence is the ability to prove due care**

[Table 1-6](#) details the basic principles surrounding due care and due diligence.

<b>Due Care</b>	<b>Due Diligence</b>
<p><b>Accountable protection of assets based on and aligned with the goals and objectives of the organization</b></p> <p>This definition aligns what security should be doing with what the organization should be doing. It aligns accountable protection of assets based on the goals and objectives of the organization. This is what due care means from a security perspective.</p>	<p><b>Ability to prove due care to stakeholders—upper management, regulators, customers, shareholders, etc.</b></p> <p>Due diligence is what is done to prove due care on a regular basis to organization stakeholders.</p>

**Table 1-6: Due Care vs. Due Diligence**

**Definitions of due care and due diligence as they relate to security**

Consider penetration testing as a representative example. Due care would be the owner of a system requesting that a penetration test be performed and then authorizing the remediation of the vulnerabilities identified by the penetration test. Due diligence would then be providing proof that the vulnerabilities were addressed in a cost-effective and efficient way to management and other relevant stakeholders (e.g., customers).

**1.4 Understand legal, regulatory, and compliance issues that pertain to information in a holistic security context**

**1.4.1 Cybercrimes and Data Breaches**

Due to the importance of information security, every organization should be asking fundamental questions like:

- How is/are our information/assets protected?

- What are the issues pertaining to information security for our organization in a global context?

- What does the current threat landscape look like?

It's important for organizations to understand the threat landscape, especially the current cybercrime trends. These insights can help organizations better deploy security and other defense-related resources in the most effective manner. Not every attack can be prevented, but effective security strategies can reduce attacks by making them:

- Not worthwhile
- Too time-consuming
- Too expensive

Bottom line: **Don't be the low-hanging fruit that can be easily picked!** As a security professional, it's important to implement effective security measures. Additionally, the security function needs to work with the compliance and legal functions to understand legal and regulatory issues in a global context because these could factor into how security is developed. Security professionals must understand global threats to their organization and respond in a manner that acknowledges them.

### 1.4.2 Licensing and Intellectual Property Requirements

#### CORE CONCEPTS

- The goal of intellectual property laws is to encourage the creation of intellectual goods (inventions, literary and artistic works, designs, symbols, and names) and to protect the same.

## What do trade secrets, patents, copyrights, and trademarks protect?

Intellectual property is any intangible product (invention, formula, algorithm, literary work, song, symbol, etc.) of the human intellect that the law protects from unauthorized use by others. Intellectual property laws (outlined in [Table 1-7](#)) help protect intellectual property assets with the goal of encouraging the creation of a wide variety of intellectual goods. Though intellectual property laws and regulations vary quite a bit from country to country, the basic premises remain the same, as noted in [Table 1-7](#).

	Protects	Disclosure Required	Term of Protection	Protects Against
Trade Secret	Business information	No	Potentially infinite	Misappropriation
Patent	Functional innovations Novel idea/inventions	Yes	Set period of time	Making, using, or selling an invention
Copyright	Expression of an idea embodied in a fixed medium (books, movies, songs, etc.)	Yes	Set period of time	Copying or substantially similar work
Trademark	Color, sound, symbol, etc. used	Yes	Potentially infinite	Creating confusion

	to distinguish one product/company from another			
--	--	--	--	--

Table 1-7: Intellectual Property Laws

### 1.4.3 Import/Export Controls

Import and export controls are country-based rules and laws implemented to manage which products, technologies, and information can move in and out of those countries, usually meant to protect national security, individual privacy, economic well-being, and so on.

### The Wassenaar Arrangement

Encryption is a powerful technological tool that can have immense value, but it can also pose a significant threat if it gets into the wrong hands. Cryptography is heavily used in the context of military and government agencies. In the United States, organizations like the National Security Agency (NSA) actively seek to deduce cryptographic keys to decrypt and understand secret communications of governments around the world in an effort to keep the country safe. As a result of the inherent value and potential threat that cryptography represents, global laws and regulations restrict the use of cryptography; and, in many cases, import/export restrictions to certain regions exist.

These laws and regulations often pertain to the sales of weapons, but they also pertain to the underlying technology—computers, network infrastructure, and more—that can be used to develop military systems.

The Wassenaar Arrangement was put in place to manage the risk that cryptography poses, while still facilitating trade. It allows certain countries to exchange and use cryptography systems of any strength, while also preventing the acquisition of these items by terrorists.

Participating members can exchange cryptography of any strength, but countries that are not a member are excluded from data exchange.

### **International Traffic in Arms Regulations (ITAR)**

This is a US regulation that was built to ensure control over any export of items such as missiles, rockets, bombs, or anything else existing in the United States Munitions List (USML) (<https://www.ecfr.gov/current/title-22/chapter-I/subchapter-M/part-121>) [note that this URL is case sensitive, so if you are typing it in, make sure to capitalize the “I” and “M”]. The responsible agency is the US Department of State, Directorate of Defense Trade Controls (DDTC).

## **Export Administration Regulations (EAR)**

EAR predominantly focuses on commercial-use related items like computers, lasers, marine items, and more. However, it can also include items that may have been designed for commercial use but actually have military applications. The responsible agency is the US Department of Commerce, Bureau of Industry and Security (BIS).

### **1.4.4 Transborder Data Flow**

#### **CORE CONCEPTS**

- **Transborder data flow laws restrict the transfer of data across country borders**
- **When sharing data across borders, applicable laws must be considered**
- **Enforcement of requirements in one country may not apply to other countries**

#### **Challenges associated with sharing data across international borders**

Many countries have enacted laws commonly referred to as transborder data flow regulations, data residency regulations, and data localization laws. These require that specific data remain within the country's physical borders.

These laws primarily relate to personal data. The idea is to protect a country/state/province/region's citizens' personal data. If an organization is collecting citizens' data, then they are accountable for the protection of that data. As privacy laws and the protection of personal data vary significantly around the world, this has prompted the creation of these transborder data flow laws. If a country/state/province/region has strong privacy laws, then they may wish to prevent personal data from being stored or processed in other countries/states/provinces/regions that may have weaker laws. Hence, transborder data flow laws prevent personal data from leaving the physical borders of a country/state/province/region.

Given these laws, organizations must consider the potential implications of the flow of data across physical borders. This can be very challenging to organizations to keep track of with the proliferation of service providers and global cloud services.

General Data Protection Regulation (GDPR), enacted in May 2018, is a great example of a data residency regulation that specifically requires that personal data of European Union citizens be stored and processed only within the physical borders of the European Union.

## 1.4.5 Issues Related to Privacy

### CORE CONCEPTS

- **Privacy is the state or condition of being free from being observed or disturbed by other people**
- **Personal data is information on its own or in combination that uniquely identifies an individual**

In the context of asset protection and security, it might seem odd to include this topic. In fact, the topic of privacy is very relevant, especially in today's globally connected world. Information that is collected from clients and visitors to websites could be considered very valuable—perhaps as much or more valuable than other organizational assets. If personal information is disclosed as the result of a breach or carelessness, it harms both the individual to whom that personal data refers to and the value of the information itself. Additionally, the organization could face significant fines or damage to corporate reputation. Depending on the nature of the business and industry, the organization may not recover or even survive. Therefore, regardless of the value, it's essential that personal data is well protected to comply with current privacy laws and to protect the value of the information and of the organization itself.

This can become complex for multinational organizations since there's a significant variation around the world in both

the definition of personal data and the laws that determine how to protect it. When dealing with personal data, organizations must tread carefully and work closely with their legal departments to identify all the applicable laws and regulations. After consulting with the legal department, it is the security function's responsibility to make sure that the correct controls are in place to achieve privacy. To have privacy, you need security.

## Definition of privacy

Let's consider the topic of privacy. First, what's the definition of privacy? **Privacy is the state or condition of being free from being observed or disturbed by other people.**

This is a fundamental concept in privacy laws around the world, based upon the premise that if an organization is allowed to collect personal information, that information might be used in an unauthorized manner or such that causes harm. This explains why privacy laws like Europe's GDPR are becoming more and more common around the globe, and they apply to both government and private business organizations. Generally, privacy laws around the world have, and continue to, become more stringent and more restrictive, requiring the perfect implementation of security controls to ensure compliance.

A very important question that comes to mind is, who and what is impacted if personal data (also referred to as Personally Identifiable Information, or PII) is disclosed? Certainly, the individual to whom the personal data refers is affected. Additionally, the value of the organization that allowed the disclosure is also affected. This could be in the form of significant fines, liability, loss of corporate reputation, or any combination thereof. The organization may not be able to sustain these operations, depending on the industry and sector in which they have activities. For example, imagine how difficult it would be for an incident response company to be able to offer their services after being the subject of a breach. As such, it's essential that personal data is well shielded to comply with current privacy laws and to protect the value of the information and the overall value of the organization itself.

## Personal Data

Depending on the location in the world, personal data may be referred to in different ways, and what constitutes personal data can vary significantly. [Figure 1-2](#) contains the various categories of sensitive data types, like PII, PHI, and IP.

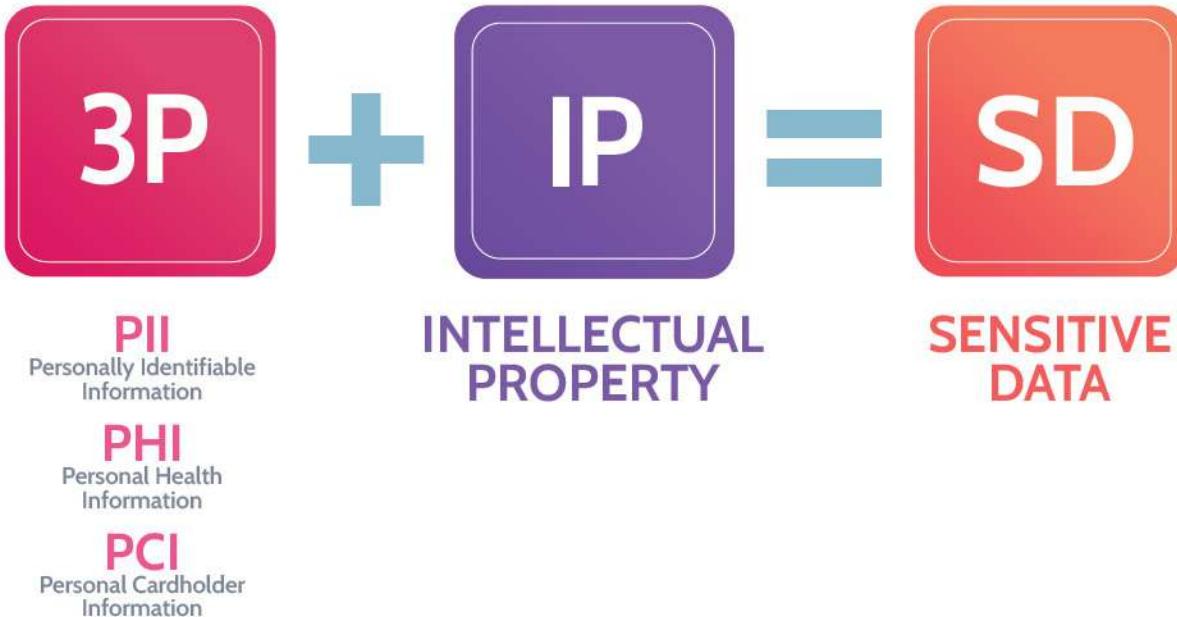


Figure 1-2: Personal Data Types

The simplest definition of personal data is data that can be used on its own or in combination to identify an individual. Personal data can be referred to as: ■ **PI**—Personal Information ■ **PII**—Personally Identifiable Information ■ **SPI**—Sensitive Personal Information ■ **PHI**—Protected Health Information How is personal data defined in a little more detail? As noted above, the definition of personal data varies quite significantly around the world. In the context of one privacy law or regulation in one part of the world, a telephone number might be considered personal data; in a different context, perhaps not. The same is true for IP addresses, email addresses, and many other types of information. For example, consider the difference between a business and a personal phone. A business phone would

need to be known to prospective clients, while a personal phone would not. The same is true for IP addresses, email addresses, and many other types of information. There is no perfect definition of personal data, because it varies significantly around the world, and this points to the notion of direct and indirect identifiers.

**Direct identifiers** include information that relates specifically to an individual, such as their name, address, biometric data, government ID, or other uniquely identifying number.

**Indirect identifiers** include information that on its own cannot uniquely identify an individual but can be combined with other information to identify specific individuals, including, for example, a combination of gender, birth date, geographic indicators, and other descriptors. Other examples of indirect identifiers include place of birth, race, religion, weight, activities, employment information, medical information, education information, and financial information.

In general, these definitions clearly describe each type of identifier. However, as a security professional, it's important to communicate with the legal team to be absolutely clear about what constitutes personal data and what jurisdictions and regulations apply. This approach allows everybody in

the organization to be on the same page, and for the proper security controls to be implemented. Some examples of direct, indirect, and online identifiers are outlined in [Table 1-8](#):

Direct	Indirect	Online
<ul style="list-style-type: none"><li>■ Name ■ Phone number ■ Government ID (e.g. SIN, SSN, driver's license) ■ Account numbers ■ Certificate/license numbers ■ Biometric data</li></ul>	<ul style="list-style-type: none"><li>■ Age ■ Gender ■ Ethnicity ■ City ■ State ■ Zip/postal code</li></ul>	<ul style="list-style-type: none"><li>■ Email address ■ IP Address ■ Cookies</li></ul>

**Table 1-8: Categories of Identifiers**

## Privacy Requirements

### CORE CONCEPTS

- Supervisory authorities are independent authorities in each EU state that investigate privacy complaints

- General Data Protection Regulation (GDPR) principles

## ■ Organization for Economic Cooperation and Development (OECD) principles

### ■ Role of supervisory authority

## Privacy Policy Requirements

Everyone deserves a reasonable expectation of privacy. When someone enters personal details at the doctor's reception area or when booking a hotel room and providing credit card details, they expect their data to be adequately protected.

Along the same lines, companies must adhere to agreements and controls that comply with applicable laws and regulations. [Table 1-9](#) contains a summary of the key roles within the privacy realm, while [Table 1-10](#) summarizes some key privacy regulations in different countries.

<b>Data Owners</b>	<p>Owners need to have clearly defined accountabilities including:</p> <ul style="list-style-type: none"><li>■ Defining classification ■ Approving access ■ Retention and destruction</li><li>■ Different types of owners:</li><ul style="list-style-type: none"><li>■ Data owners ■ Process owners ■ System owners</li></ul><p>Companies that collect personal data about customers are <b>accountable</b> for the protection of the data.</p></ul>
<b>Data Custodians</b>	<p>Need to have clearly defined <b>responsibilities</b></p> <p>Protect data based on input of the owners.</p>

	Custodians also need tools, training, resources, etc. And who provides all this? Typically the owners.
<b>Data Processors</b>	Need to have clearly defined <b>responsibilities</b> . Processes personal data on behalf of the controller / owner.
<b>Data Subjects</b>	Individual to whom personal data relates

Table 1-9: **Data Owners, Custodians, and Processors**

<b>GDPR</b>	<ul style="list-style-type: none"> <li>■ A single set of rules applies to all EU member states ■ Each state establishes an independent <b>Supervisory Authority (SA)</b> to hear and investigate complaints ■ <b>Data subjects shall have the right to lodge a complaint with a SA</b></li>   <li>■ Seven principles describe lawful processing of personal data: ■ Lawfulness, fairness, and transparency ■ Purpose limitation ■ Data minimization ■ Accuracy ■ Storage limitation ■ Integrity and confidentiality (security) ■ Accountability ■ <b>Privacy breaches must be reported within 72 hours</b></li> </ul>
<b>United States</b>	<ul style="list-style-type: none"> <li>■ Gramm–Leach–Bliley Act (GLBA) ■ Health Insurance Portability and Accountability Act (HIPAA) ■ Sarbanes–Oxley Act (SOX) ■ Children's Online Privacy Protection Act (COPPA) ■ California Consumer Privacy Act (CCPA) – Similar to the GDPR</li>   <li>■ California Privacy Rights Act of 2020</li> </ul>
<b>Canada</b>	<ul style="list-style-type: none"> <li>■ Personal Information Protection and Electronic Documents Act (PIPEDA)</li> </ul>

<b>China</b>	■ Personal Information Protection Law
<b>South Africa</b>	■ Protection of Personal Information Act
<b>Argentina</b>	■ Personal Data Protection Law Number 25,326 (PDPL)
<b>South Korea</b>	■ Personal Information Protection Act (PIPA)
<b>Australia</b>	■ Privacy Act ■ Australian Privacy Principles (APPs)

**Table 1-10: Privacy Regulations in Different Countries**

The list above illustrates just a few of the privacy laws around the world, and the requirements in these laws vary from country to country. You are not expected to be a privacy expert for purposes of the CISSP exam; but as a security professional you should understand that privacy cannot be achieved without security. Security must be involved in implementing the required security controls to achieve the required privacy requirements.

One privacy law that you should have a slightly deeper understanding of is GDPR. The reason is that GDPR is considered by many to be the bellwether for privacy laws in countries around the world. GDPR is one of the most comprehensive privacy laws in the world, and many countries have modeled, or are in the process of modeling their privacy laws on GDPR or plan to in the future. Some of

the basic information you should know about GDPR is listed in [Table 1-10: Privacy Regulations in Different Countries](#).

For multinational organizations, it can be quite complex and challenging to keep track of the varying privacy requirements around the world. In response to this problem, the Organization of Economic Cooperation and Development (OECD) has created guidelines that offer a simple set of principles that organizations can use to structure their privacy practices.

### **OECD Privacy Guidelines**

The **Organization for Economic Cooperation and Development (OECD)** is an international organization that is focused on international standards and policies, and finding solutions to social, economic, and environmental challenges. One such challenge that they have been driving for decades is privacy.

Working with its member states, OECD has developed **guidelines** that would help to harmonize national privacy legislation and, while upholding such human rights, would also prevent interruptions in international flows of data. OECD represents a consensus on basic principles that can be built into existing national legislation or serve as a basis for legislation—those countries that do not yet have adequate privacy legislation. These guidelines have

consistently been updated to reflect new requirements as technology has advanced.

Are the OECD guidelines mandatory for organizations to comply with? No, usually they're considered a prudent course of action. This is precisely how the OECD Guidelines should be viewed. They are intended as suggestions, as common “best practices” related to privacy and conducting business, regardless of the location around the globe. In other words, the OECD Guidelines can be useful to organizations, as they can provide guidance on how to achieve compliance to privacy requirements. Does this mean a perfectly implemented privacy program, based on the OECD Guidelines, is compliant everywhere? No, but following those guidelines is likely to meet most requirements in a given locale. However, it's not a replacement for reviewing the specific laws and regulations you need to follow. Security professionals can use the guidelines as a starting point for the fundamental security controls organizations should put in place. Once they've done so, it's still necessary to consult with legal experts about the specific laws and regulations they need to comply with, depending on the country in which they are operating. Subsequently, specifics related to that jurisdiction can be considered further for inclusion. OECD's privacy guidelines can be seen in [Table 1-11](#).

<b>Collection Limitation Principle</b>	Limit the collection of personal data to only what is needed to provide a service, obtain the personal data lawfully and, where appropriate, with the knowledge or consent of the data subject.
<b>Data Quality Principle</b>	Personal data should be relevant, accurate, and complete, and it should be kept up to date.
<b>Purpose Specification Principle</b>	The purposes for which personal data is collected should be clearly specified at the time of collection.
<b>Use Limitation Principle</b>	Personal data should only be used based on the purposes for which it was collected and with consent of the data subject or by authority of law; in other words, if an organization says it has collected personal data for a specific purpose, they should only use the personal data for that purpose.
<b>Security Safeguards Principle</b>	Personal data should be protected by reasonable security safeguards against loss, unauthorized access, destruction, use, modification, etc. Essentially, security controls must be put in place, because privacy is unattainable without security.
<b>Openness Principle</b>	The culture of the organization collecting personal data should be one of openness, transparency, and honesty about how personal data is being used and in what context.
<b>Individual Participation Principle</b>	When an individual—data subject—provides personal data to an organization, that individual should have the right to obtain their data from the data controller as well as have their data removed. In other words, the individual should have the chance to participate or choose whether to share their personal information or withhold it. The

	term <i>data subject</i> refers to the individual to whom the personal data pertains.
<b>Accountability Principle</b>	A data controller should be accountable for complying with the other principles. What this basically means is that an organization that collects personal data is now accountable for the protection of that information.

Table 1-11: OECD Privacy Guidelines

## Privacy Assessments

### CORE CONCEPTS

- **Privacy Impact Assessment (PIA) is a process undertaken on behalf of an organization to determine if personal data is being protected appropriately and to minimize risks to personal data where appropriate.**

- **What is a PIA/DPIA?**
- **How often does a PIA/DPIA need to be conducted?**

With the protection of privacy becoming more important with each passing day, requirements calling for privacy and **Data Protection Impact Assessments (DPIA)** have become equally important. In fact, Article 35 of the GDPR legislation includes a provision for Data Protection Impact Assessments (DPIA) and outlines when they're required and how they should be carried out. Additionally, ISO/IEC 29134:2017 describes a process on privacy impact

assessments (PIA) and a structure and content of a PIA report. The NIST Technology Innovation Program includes information about PIAs. The European Data Protection Board, other organizations, trade groups, and independent businesses and vendors have and will continue to provide guidance, tools, checklists, and templates.

## **What is a privacy impact assessment?**

A **Privacy Impact Assessment (PIA)** is a process undertaken on behalf of an organization to determine if personal data is being protected appropriately and to minimize risks to personal data as appropriate. Any system that processes personal data could be included in a PIA. Like many other risk management processes, a PIA is not a one-time assessment. Rather, it should be performed each time it's necessary, especially when risk represented by personal data processing operations has changed. Additionally, along with the assessment of risks, accompanying mitigation measures should be included.

## **Why are they important?**

A PIA is performed with a goal to:

1. Identify/evaluate risks relating to privacy breaches

2. Identify what controls should be applied to mitigate privacy risks
3. Offer organizational compliance to privacy legislations

## Privacy/Data Protection Impact Assessment Steps

There are no all-inclusive templates for conducting a PIA/DPIA, but the steps outlined in [Table 1-12](#) summarize the core elements of a PIA/DPIA.

### What are the steps required to conduct a PIA?

## Privacy Impact Assessment Steps

<b>1. Identify the need for a DPIA</b>	Use legislative guidelines, like GDPR, European guidelines, federal and state laws, industry regulations, etc. to determine if a DPIA is required. If any doubt exists, it's best to err on the side of caution and conduct an assessment.
<b>2. Describe the data processing</b>	This involves two steps. The first step answers questions such as: <ul style="list-style-type: none"><li>■ How is data being collected/used?</li><li>■ Where is data being gathered from?</li><li>■ How much data is being gathered, and how many data subjects are involved?</li><li>■ Is this data being stored with any third-party entities?</li></ul>

	<ul style="list-style-type: none"> <li>■ What are the purposes of processing?</li> <li>■ Are the interests of the data controller legitimate?</li> </ul> <p>The second step considers information gathered via questions like those mentioned earlier and further defines the purpose—the what, how, and why—of data processing activities as they relate to the goals of the project.</p>
<b>3. Assess necessity and proportionality</b>	<p>Data processing activities should always correlate with what is actually required for the goals and objectives of a project. The DPIA should confirm this is the case, and this can be done by answering questions such as:</p> <ul style="list-style-type: none"> <li>■ Does a legal basis for collected personal data exist?</li> <li>■ Do data subjects have the right to opt out or in with relation to their personal data?</li> <li>■ Does a precedent exist for the collecting and processing of data?</li> <li>■ How are the rights of data subjects protected?</li> </ul>
<b>4. Consult interested parties</b>	<p>As part of any DPIA, several key parties should be consulted, including the data protection officer, project stakeholders, and data subjects (or their legal representatives).</p>
<b>5. Identify and assess risks</b>	<p>This step is critically important and likely the most important component of the DPIA, as it involves thoroughly assessing risks to personal data. While some risks might be project dependent, key considerations with any project should include asking:</p> <ul style="list-style-type: none"> <li>■ Is data being stored in unsafe locations?</li> <li>■ Are appropriate access control lists being utilized?</li> <li>■ What data retention policies are currently in place?</li> </ul>

<b>6. Identify measures to mitigate the risks</b>	<p>Once risks associated with a project have been identified, it's imperative that corresponding steps to mitigate those risks be identified and implemented, based upon the cost-effectiveness of doing so.</p> <p>Additionally, this is where a defense-in-depth approach that involves the use of complete controls should be utilized to:</p> <ul style="list-style-type: none"> <li>■ protect personal data from unauthorized internal and external access,</li> <li>■ remove data that is no longer required, via relevant data retention policies and processes,</li> <li>■ maintain visibility over personal data,</li> <li>■ automate remediation action when and where possible for the sake of data removal, cleanup, and classification.</li> </ul>
<b>7. Sign off and record outcomes</b>	<p>After risks and associated mitigation steps have been identified, all details should be documented and signed off on by relevant parties that could include the data protection officer, senior management, process and project stakeholders, and data subjects.</p>
<b>8. Monitor and review</b>	<p>A PIA should be performed when necessary, especially when risk represented by personal data processing operations has changed. This fact points to the need for ongoing monitoring and review of operations, processes, and all facets of a business that involve handling of personal data.</p>

**Table 1-12: PIA/DPIA Core Elements**

Additionally, Article 35 of the GDPR offers the minimum features of a DPIA:

The assessment shall contain at least:

1. a systematic description of the envisaged processing operations and the purposes of the processing, including, where applicable, the legitimate interest pursued by the controller; 2. an assessment of the necessity and proportionality of the processing operations in relation to the purposes; 3. an assessment of the risks to the rights and freedoms of data subjects; and 4. the measures envisaged to address the risks, including safeguards, security measures, and mechanisms to ensure the protection of personal data and to demonstrate compliance with this regulation considering the rights and legitimate interests of data subjects and other persons concerned.



## 1.4.6 Contractual, Legal, and Industry Standards and Regulatory Requirements

### CORE CONCEPTS

- Controls should align with compliance requirements
- Legal, privacy, and audit/compliance functions are usually the best sources to determine compliance requirements

Establishing the right security controls isn't just about the internal needs of an organization. There is a plethora of contractual, legal, industry, and regulatory requirements that should inform how different assets are protected—also referred to as compliance requirements. [Table 1-13](#) shows how an organization can determine compliance needs and requirements by defining the most common compliance requirements an organization would need to consider.

<b>Laws</b>	<ul style="list-style-type: none"><li>■ Specific laws that an organization must comply with are based on the assets owned or managed, or the industry, jurisdiction, or country in which the organization operates ■ Examples of laws: Health Insurance Portability and Accountability Act (HIPAA), Gramm–Leach–Bliley Act (GLBA), Consumer Online Privacy Rights Act (COPRA), Family Educational Rights and Privacy Act (FERPA), General Data Protection Regulation (GDPR), Federal Information Security Modernization Act (FISMA), and Digital Millennium Copyright Act (DMCA)</li></ul>
<b>Regulations</b>	<ul style="list-style-type: none"><li>■ Specific regulations that an organization must comply with are based on the assets owned or managed, the industry, jurisdiction, or country in which the organization operates ■ Examples of regulations: International Traffic in Arms Regulations (ITAR), Export Administration Regulations (EAR), and Encryption Export Controls</li></ul>
<b>Industry Standards</b>	<ul style="list-style-type: none"><li>■ Specific industries often have associated standards—procedural and technical rules—that help guide the activities of organizations ■ Examples of industry standards: Critical Infrastructure Protection (NERC CIP),</li></ul>

	National Institute of Standards and Technology (NIST), and International Organization for Standardization (ISO)
<b>Import/Export Controls</b>	See <a href="#">Domain 1 (section 1.4.3)</a>
<b>Transborder Data Flow Regulations</b>	See <a href="#">Domain 1 (section 1.4.4)</a>
<b>Assets</b>	See <a href="#">Domain 2 (section 2.1)</a>
<b>Personal Data</b>	See <a href="#">Domain 2 (section 1.4.5)</a>
<b>Corporate Policies</b>	See <a href="#">Domain 1 (section 1.6)</a>

**Table 1-13: Compliance Requirements**

The legal, privacy, and audit/compliance functions must work together to ensure compliance, which requires the drive and initiative that security will ultimately design and implement as security controls. The compliance function will monitor compliance, the security function will advise on and enforce controls, and legal and privacy functions will determine organizational compliance needs. As a security professional, it's important to understand what the organization needs to be compliant with and what controls must be in place to adhere to these requirements. This implies that security must know what compliance needs

exist, and the best resource to identify and understand these compliance needs is usually the legal function.

Once management understands compliance needs, they can work with security to implement controls. A big part of implementing the right controls is having the right roles and responsibilities defined, to determine who is accountable and who is responsible. Certain people within an organization are going to be accountable for the protection of personal information; many others are going to be responsible for it. Owners need to have clearly defined accountabilities related to compliance, including:

- Defining classification
- Approving access
- Retention and destruction

**1.5 Understand requirements for investigation types (i.e., administrative, criminal, civil, regulatory, industry standards)**

We discuss the various investigation types in [Domain 7](#) ([section 7.1.7](#)).

## **1.6 Develop, document, and implement security policies, procedures, standards, baselines, and guidelines**

### **CORE CONCEPTS**

- Policies = Corporate Laws
- Policies document and communicate management's goals and objectives

- **Overarching security policy must come from upper management (tone from the top)**
- **Procedures = Step-by-Step instructions**
- **Standards = Specific information related to solutions**
- **Baselines = Defined minimal implementation levels**
- **Guidelines = Recommendations or suggestions**

Earlier, it was discussed in detail how security must be aligned with organizational goals and objectives. A top-down approach that incorporates a governance committee to help design policies is required. The committee, reporting to the Board of Directors and CEO, should develop an overarching security policy that is aligned with organizational goals and objectives that covers the entire organization and clearly articulates the goals and objectives of the security function. Policies, as we've already briefly touched upon earlier, are corporate laws that reflect the goals and objectives of the organization. They dictate and communicate management's intentions. The overarching security policy is critical, as it sets the tone and helps create the culture necessary for effective organizational security to exist. This policy must be communicated from the CEO, or even the Board of Directors, to be most effective and impactful. It also needs to be consistently communicated and demonstrated by those at the top of the organization.

- Who should write policies and who owns policies?
- How often should policies be reviewed?
- How are policies implemented through standards, procedures, baselines, & guidelines?

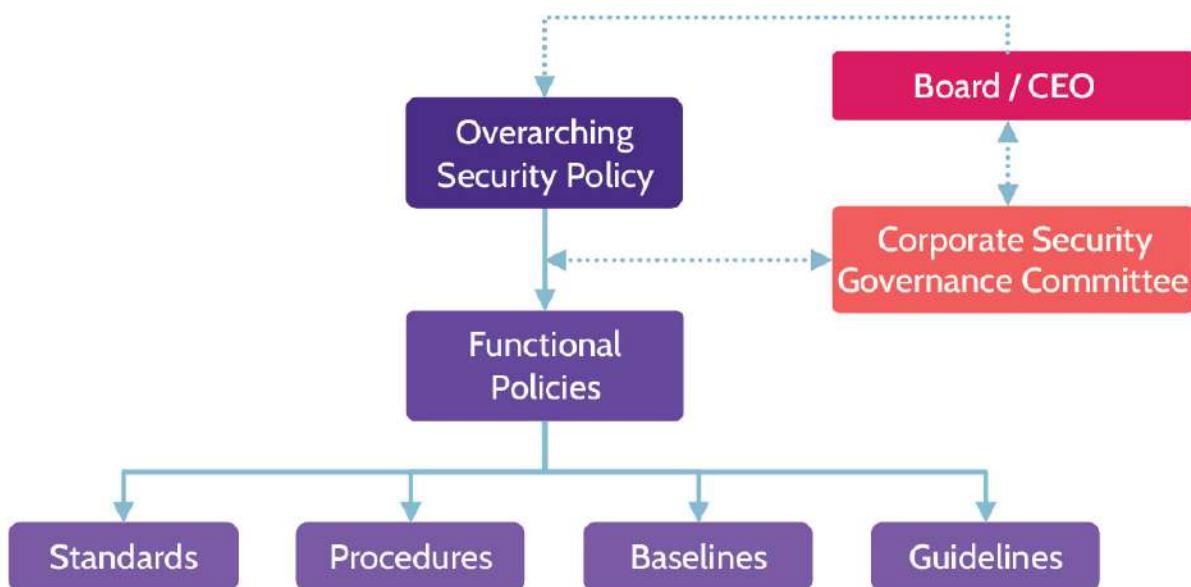
The overarching policy should be very simple. It needs to be communicated by the CEO and will clearly spell out how the CEO and organization is accountable for protecting all assets that represent value to the organization—that the CEO and upper management are ACCOUNTABLE, but also that EVERYONE in the organization is RESPONSIBLE for security and protecting the value of assets. The CEO must clearly communicate this and remind the entire organization on a regular basis. If this is done, it creates the proper culture and tone that security needs to be an enabler to the organization. It also ensures that everyone understands the importance of security and that it is everyone's responsibility.

If this is done properly, the security function is seen as an enabler and helper, as opposed to the traditional view of security as an obstacle—where the business goes to be told they can't do something.

Specific functional security policies will flow from the overarching policy. Functional security policies include

standards, procedures, baselines, and guidelines that outline how to enact them. While policies don't need to be reviewed every year, standards, procedures, baselines, and guidelines may need to be updated frequently. Any combination of these elements will typically be put in place to support functional policies; together, the compendium of functional policies will be defined, supported, and informed by many standards, procedures, baselines, and guidelines.

Therefore, let's take a close look at a model for creating and maintaining security policies in an organization as depicted in [Figure 1-3](#).



**Figure 1-3: Security Document Hierarchy**

An organization might have a policy, created and owned by the Security Governance Committee, mandating the use of

anti-malware software. Functional policies would then need to be developed that dictate exactly how to enact that policy. Those functional policies might include a standard to specify the version of anti-malware software to use, a procedure to outline the steps to install it, and a guideline to suggest ideal goals for anti-malware efforts, such as heuristics in anti-malware software where possible. Each type of supporting document works together to ensure the policy is met.

- **Differences between policies procedures, baselines, and guidelines**
- **Identifying when something provided relates to a procedure, policy, baseline, or guideline**

The success of this model depends on each person performing their role well and supporting functional policies that make sense to the company. If the Board or CEO are unwilling to lead, failure from the top could follow. In our example, this might mean that although an organization would benefit from an anti-malware policy, the Board or CEO may not be working with security to create one. Similarly, if the supporting elements of functional security policies are not considered properly, implementation of security could fail. Thus, it's important

that all facets of the model be carefully considered when developing and implementing it.

Many companies have very different definitions of policies, standards, procedures, baselines, and guidelines—but there are industry standards for what each of these documents are meant to be used for. Knowing the precise definition of these standards—outlined in [Table 1-14](#)—will help you use them properly.

<b>Policies</b>	<ul style="list-style-type: none"><li>■ Documents that communicate management's <b>goals and objectives</b></li><li>■ Provide <b>authority</b> to security activity</li><li>■ Define the elements, functions, and scope of security team</li><li>■ Must be approved and communicated by management</li><li>■ Corporate laws</li></ul>
<b>Standards</b>	<p><b>Specific hardware and software solutions, mechanisms, and products</b></p> <p>Examples</p> <ul style="list-style-type: none"><li>■ Specific anti-virus software, e.g., McAfee</li><li>■ Specific access control system, e.g., Forescout</li><li>■ Specific firewall system, e.g., Cisco ASA</li><li>■ Published guideline (e.g., ISO 27001) adopted by an organization as a standard.</li></ul>
<b>Procedures</b>	<p><b>Step-by-step descriptions on how to perform a task; mandatory actions</b></p> <p>Examples</p> <ul style="list-style-type: none"><li>■ User registration or new hire onboarding</li><li>■ Contracting for security purposes</li><li>■ Information system material destruction</li><li>■ Incident response</li></ul>

	<b>Defined minimal implementation methods/levels for security mechanisms and products</b>
<b>Baselines</b>	<p>Examples</p> <ul style="list-style-type: none"> <li>■ Configurations for intrusion detection systems ■ Configurations for access control systems</li> </ul>
<b>Guidelines</b>	<p><b>Recommended</b> or <b>suggested</b> actions Examples</p> <ul style="list-style-type: none"> <li>■ Government recommendations ■ Security configuration recommendations ■ Organizational guidelines ■ Product/system evaluation criteria (Note: Guidelines allow an organization to suggest something be done without making it a hard requirement and thus cause a negative audit finding)</li> </ul>

**Table 1-14: Policies, Standards, Procedures, Baselines, and Guidelines**

## **1.7 Identify, analyze, and prioritize business continuity (BC) requirements**

See [Domain 7 \(section 7.11\)](#) to read about these topics in depth. The information for 1.7.1 is covered in [7.11.3 Business Impact Analysis \(BIA\)](#), while 1.7.2 External dependencies is discussed in [7.11.1](#).

This section addresses the importance of a BIA (Business Impact Analysis) as part of the overall Business Continuity Management processes of an organization. It also highlights the importance of understanding external dependencies, but is better addressed in domain 7 when we discuss the

entire business continuity management process and the involvement of security as part of that process. For now, here is a brief overview of each topic: **BIA (Business Impact Analysis)**

A BIA analyzes the consequences of a disaster to an organization and allows the organization to understand priorities and gather the information needed to develop recovery strategies.

### External Dependencies

As part of business continuity management of an organization, it is important to understand the external factors that can impact the criticality of valuable functions and processes within the organization. As part of the BIA process, a company needs to map out the inter-dependencies between their critical functions, processes, assets, applications, systems, etc., as well as others that are outside of their control, such as suppliers, vendors, and other third parties.

## 1.8 Contribute to and enforce personnel security policies and procedures

### CORE CONCEPTS

- Hiring, onboarding, and terminating employees
- Employment controls and associated cost-effectiveness

- Dealing with potential violations identified in a security assessment
- Dealing with employee terminations and resignations
- Employee duress

## Personnel Security Policies

Companies need clearly documented and communicated personnel security policies that are implemented through procedures, to address the needs related to the use and protection of valuable assets of the organization. Some of the best practices for protecting the business and its important assets are listed below. These best practices all have to do with how people and the organization work together to support the business.

### 1.8.1 Candidate Screening and Hiring

New personnel represent a risk to security; every organization needs personnel security policies that address and mitigate this risk with the right security controls.

Examples of personnel security policies and controls include background checks, access badges, ID cards, what you're allowed to bring in and out of the building, acceptable use policies, code of conduct, employee handbook, and so on.

## **1.8.2 Employment Agreements and Policy Driven Requirements**

As part of bringing a new employee into an organization—also referred to as onboarding—company security policies, acceptable use policies, and similar agreements should be reviewed and agreed upon prior to giving a new employee their badge and any system credentials.

Over the course of an employee's time at the company, controls like “separation of duties” and “job rotation” can be used to prevent fraud or violation of organizational policy. In addition to separation of duties and job rotation, two other controls often used are “least privilege” and “need to know.” These two access controls are often referred to together, and they help ensure that employees are given only the access they need to perform their job and no more.

Offboarding controls are used when an employee leaves an organization, whether through termination or resignation. Prior to an employee leaving, or in conjunction with it, user system access should be disabled, and the fact that the employee's employment is being terminated should be conveyed to all relevant parties within the organization. Usually, voluntary termination isn't too much of a security risk. However, involuntary termination is a big risk from a security perspective. If a terminated employee becomes

hostile, they might be tempted to lash out by stealing or tampering with data. For this reason, involuntary termination usually needs to be handled quite differently than voluntary termination. For example, in some situations, a member from physical security might even be physically present in the HR office while the person is being terminated to escort them out of the building.

## **Employee Duress**

An employee acting under duress may be forced to perform an action or set of actions that they wouldn't do under normal circumstances. For example, consider the scenario of a bank manager threatened by an attacker and told to open the bank's vault while held at gunpoint. In that scenario, the bank manager's life is at risk, so, acting under duress, they may give the attacker what they want. One common practice to handle these stressful situations is to have keywords that denote that an employee is acting under duress. Have you ever watched *The Bourne Identity*? You will notice that at some point one of the field agents calls the CIA headquarters and gives the all clear after inspecting a potentially compromised location. However, the agent in charge does a challenge-response check and expects a specific answer to be provided if the field agent is acting under duress. Training is key in these scenarios so

everyone will act calmly and denote they are potentially acting under duress.

Personnel security policies should also be extended to third parties. Third parties include contractors, companies, and anybody that may have access to company assets as part of the service provided to the organization. [Table 1-15](#) provides a list of important personnel security controls, while [Table 1-16](#) summarizes onboarding and offboarding processes.

## Personnel Security Controls

<b>Job rotation</b>	Job rotation is quite useful to protect against fraud and provide cross training. It entails rotating staff (especially individuals in key positions), so that an individual can't commit fraud and cover it up. For example, if someone is a loans officer at a major bank and is responsible for approving loans, they can easily defraud the bank by constantly approving loans for known individuals who pass them money as a reward. However, if that individual is rotated to another role, this won't be possible. In addition, this helps the organization to build personnel redundancy. If another staff member learns how to perform the loan officer's job, this can greatly help if that individual decides to leave the company.
<b>Mandatory vacation</b>	Mandatory vacation is a control also used by organizations to detect fraud. Employees are required to go on vacation for a set period of time, during which time another employee can step into the role and determine if any malicious or nefarious activity has taken place or is actively taking place.

<b>Separation of duties</b>	Separation of duties is used to prevent fraud, by requiring more than one employee to perform critical tasks. A good example of this can be found in Accounts Payable/Vendor Management department. For new vendors to be set up to receive payments, at least two people are typically involved: one person to enter the vendor or payment information and another to confirm the vendor or approve the check. This way, a check can't be submitted, reviewed, and processed by a single person, giving them an opportunity to commit fraud.
<b>Need-to-know and Least privilege</b>	Least privilege ensures that only the minimum permissions needed to complete the work are granted to any employee. Need to know ensures that access to sensitive assets is restricted only to those who require the information to complete the work.

**Table 1-15: Personnel Security Controls**

<b>Onboarding</b>	<b>Termination/Offboarding</b>
<ul style="list-style-type: none"> <li>■ Identity proofing (ability to identify individuals attempting to access a specific application or service)</li> <li>■ Signoff on policies and agreements</li> <li>■ Access provisioning based on least privilege and Need-to-Know</li> </ul>	<ul style="list-style-type: none"> <li>■ Timely removal of access</li> <li>■ Involuntary vs. voluntary</li> </ul>

**Table 1-16: Onboarding and Offboarding Processes**

## Enforce Personnel Security Controls

**CORE CONCEPTS**

- **Enforce organizational personnel policies and controls for contractors and vendors**
- **Contracts, agreements, and NDAs are tools that can be used to enforce personnel security controls**
- **Attestation and audit are tools that can monitor and show compliance with personnel security controls**

Enforcing personnel security controls commences with the hiring process, extends through the employment period, and ends only after the employee has left the organization. Security controls like job rotation, separation of duties, and the others mentioned earlier are important, but policies are the primary means by which these controls are enforced. They often include:

- Company security policies that align with and support organizational goals and objectives
- Acceptable use policies that outline the “do” and “don’t” behavior expected by the organization
- Additionally, personnel-focused policies are often further supported by things like:

  - Nondisclosure agreements (NDA)
  - Noncompete agreements (NCA)
  - Ethical guideline and requirement questionnaires and agreements
  - Vendor, consultant, and contractor agreements and controls

**Nondisclosure Agreements (NDAs)** are contracts through which the parties agree not to disclose information covered by the agreement. Organizations may require employees to agree to and sign an NDA before the employee is allowed to access sensitive information.

Personnel security policies should also be extended to third parties in the form of contracts and service level agreements (SLAs).

As employee actions and behavior are subject to and enforced by organization policies, third-party vendors should be equally subject to and held responsible for their actions and behavior. Contracts and SLAs, NDAs, attestation, and audit are tools that an organization can use to ensure compliance to organizational personnel policies.

Enforcement of organizational personnel policies and controls for vendors and other third parties are achieved through:

- Contracts/agreements
- NDAs
- Attestation/audit

## **1.9 Understand and apply risk management concepts**

### **1.9.1 Risk Management**

#### **CORE CONCEPTS**

- **Risk management is the identification, assessment, and prioritization of risks and the economical application of resources to minimize, monitor, and control the probability and/or impact of these risks**
- **Risk management steps: value, risk, and treatment**

Every organization (big or small) faces a similar challenge: limited resources are available to protect numerous assets. In those cases, what controls should be used, and what are the most effective ones? How can assets be adequately

protected when there are not enough resources present? Risk management aims to answer such questions.

Risk management is the identification, assessment, and prioritization of risks and the cost-efficient application of resources to minimize the probability and/or impact of these risks.

- **Risk management and relationship with risk analysis and threat analysis**
- **Risk management steps**

The value of an asset must be understood in order to identify and implement the most cost-effective security controls. If controls are inefficient and not cost-effective, the value of the organization is being eroded. For example, imagine applying a \$100,000 security control to a risk that has been calculated to only cost the organization \$1,000 per year. That isn't cost-efficient at all.

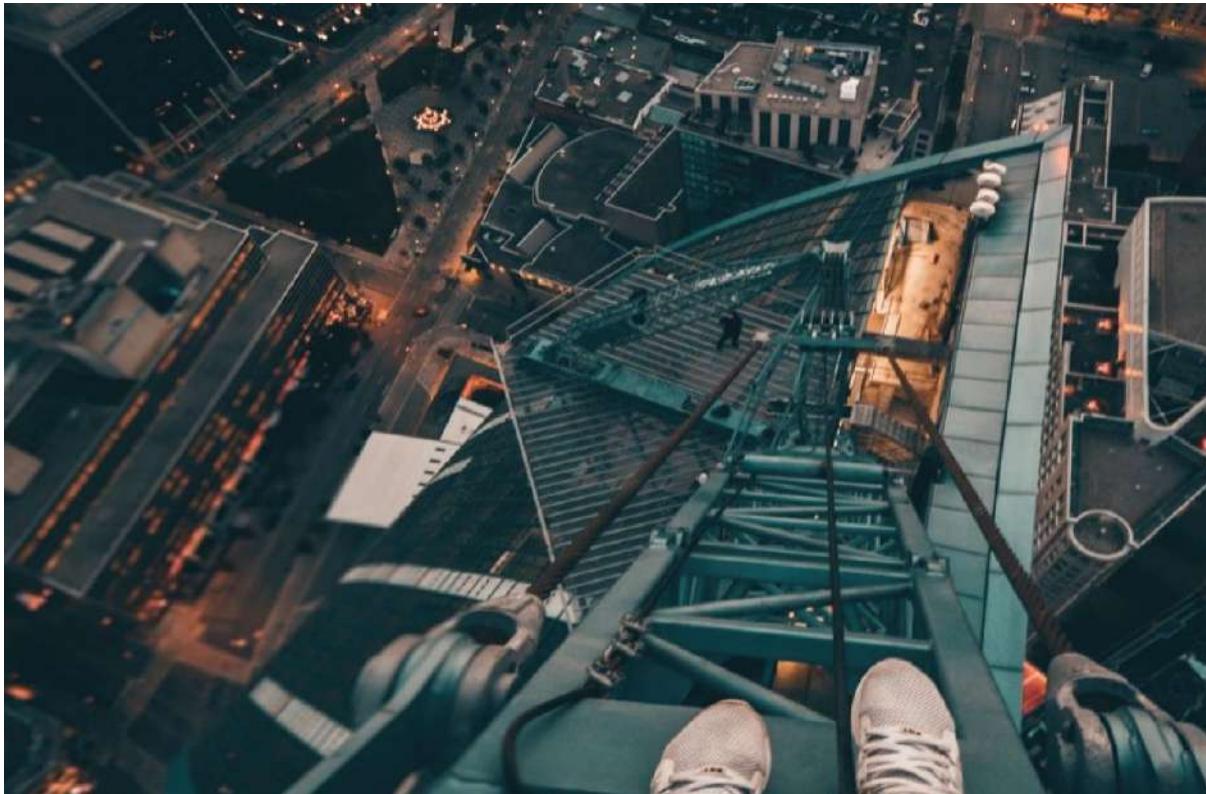


Table 1-17 provides an overview of the risk management process, and we'll delve into each step in more detail in subsequent sections.

<b>1. Value</b>	The first step is identifying the assets of the organization and ranking those assets from most to least valuable. This process is referred to as asset valuation, and the ranking of assets can be achieved via two methods or, most commonly, a combination of both: ■ Quantitative value analysis ■ Qualitative value analysis
<b>2. Risk Analysis</b>	Determine the risks associated with each asset via the risk analysis process. Risks are identified by determining the specific threats (threat analysis) that could harm the asset, the vulnerabilities (vulnerability analysis) of the asset, what the impact would be if a threat manifests or a vulnerability is

	<p>exploited, and the expected frequency of the risk occurring. Simple definitions of the four key components that must be identified as part of risk analysis follow:</p> <ul style="list-style-type: none"> <li>■ <b>Threat:</b> Any potential danger to an asset (could be environment, physical, people, technology).</li> <li>■ <b>Vulnerability:</b> Any weakness that exists that could be exploited by an attacker.</li> <li>■ <b>Impact:</b> The extent to which an asset would be negatively affected.</li> <li>■ <b>Probability/liability:</b> The chance that a risk might materialize due to a given threat or vulnerability being present.</li> </ul> <p>Based upon the findings from the risk analysis step, the next step is to rank the assets in order of the ones presenting the most risk to those with the least risk, using quantitative or qualitative analysis.</p>
<b>3. Treatment</b>	<p>Once identified, risks must be dealt with (treated), and there are four risk treatment methods:</p> <ul style="list-style-type: none"> <li>■ <b>Avoid</b> → Don't do whatever the risky thing is (e.g., implementing a certain system, moving to the cloud, jumping off a bridge, etc.)</li> <li>■ <b>Transfer</b> → Purchase an insurance policy (e.g., cyber insurance)</li> <li>■ <b>Mitigate</b> → Implement controls to reduce the risk</li> <li>■ <b>Accept</b> → The owner of an asset accepts a certain level of risk</li> </ul> <p>See <a href="#">section 1.9.5</a> for additional details on risk treatment options.</p>

**Table 1-17: Risk Management Process**

By following the steps outlined above, an organization can come to a full understanding of what comprises its most important and valuable assets as well as the risks associated with each of those assets. Once this understanding has

been reached, treatment of risks can be properly evaluated. Since an organization's technology, threat landscape, vulnerabilities, and the impact of risks occurring, are constantly changing, risk management is an ongoing and repetitive process.

## 1.9.2 Asset Valuation

### CORE CONCEPTS

- Before risks can be identified and managed, valuable assets of the organization must first be identified

### Asset valuation

During the first step of risk management, all efforts are focused on identifying the tangible and intangible assets that are of greatest value to the organization. These assets can vary widely: they can be buildings and equipment, critical business processes, the reputation of the company, and many others. Two different forms of analysis can be used to rank the assets of the organization from most to least valuable: qualitative and quantitative. Their main characteristics are depicted in [Table 1-18](#).

Qualitative Analysis	Quantitative Analysis
<ul style="list-style-type: none"><li>■ Does not attempt to assign</li></ul>	<ul style="list-style-type: none"><li>■ Assign objective monetary values</li></ul>

monetary value	
<ul style="list-style-type: none"> <li>■ Relative ranking system, based on professional judgment</li> <li>■ Uses words like "Low," "Medium," "High," "1-5," "Probability," or "Likelihood" to express value</li> </ul>	<ul style="list-style-type: none"> <li>■ Fully quantitative process when all elements are quantified</li> </ul>
<ul style="list-style-type: none"> <li>■ Qualitative analysis is relatively simple and efficient</li> </ul>	<ul style="list-style-type: none"> <li>■ Purely quantitative is difficult to achieve and time consuming</li> </ul>

**Table 1-18: Qualitative and Quantitative Analysis Characteristics**



### 1.9.3 Risk Analysis

**CORE CONCEPTS**

- Process of identifying threats and vulnerabilities related to an asset
- Identify risks and understand probability/impact of risk occurring

- Risk analysis steps
- Calculating residual risk

After the asset valuation process, related threats and vulnerabilities must be identified for each asset. Proper risk analysis takes time, effort, and resources. Without the support of senior management and asset owners, risk analysis is not going to be effective. Why? Owners best understand the value of an asset to the organization. Therefore, owners must be deeply involved in the risk analysis process.

## Threats and Vulnerabilities

There are three main components to a risk being present:

■ **Asset:** anything of value to the organization ■ **Threat:** any potential danger; anything that causes damage to an asset, like hackers, earthquakes, ransomware, social engineering, denial-of-service attacks, disgruntled employees, and many others.

■ **Vulnerability:** a weakness that exists; anything that allows a threat to take advantage of it to inflict

damage to the organization. Examples include open ports with vulnerable services, lack of network segregation, lack of patching, and OS updating.

**Table 1-19** contains some examples of threats and vulnerabilities that relate to them.

Risk Type	Threat	Vulnerability
<b>Natural/Environmental</b>	Flood	Building located on a floodplain
<b>Human</b>	Hacker	Employees that haven't been sufficiently trained and are susceptible to social engineering
<b>Operational/Process</b>	Process that's highly susceptible to fraud, e.g., issuing checks	No segregation of duties implemented to prevent fraud
<b>Technical</b>	Malware	Unpatched software
<b>Physical</b>	Power outage	No backup power system

**Table 1-19: Examples of Threats and Vulnerabilities**

**Figure 1-4** depicts how risk exposure occurs where there is an asset that is vulnerable, and a threat exists that can exploit the vulnerability.

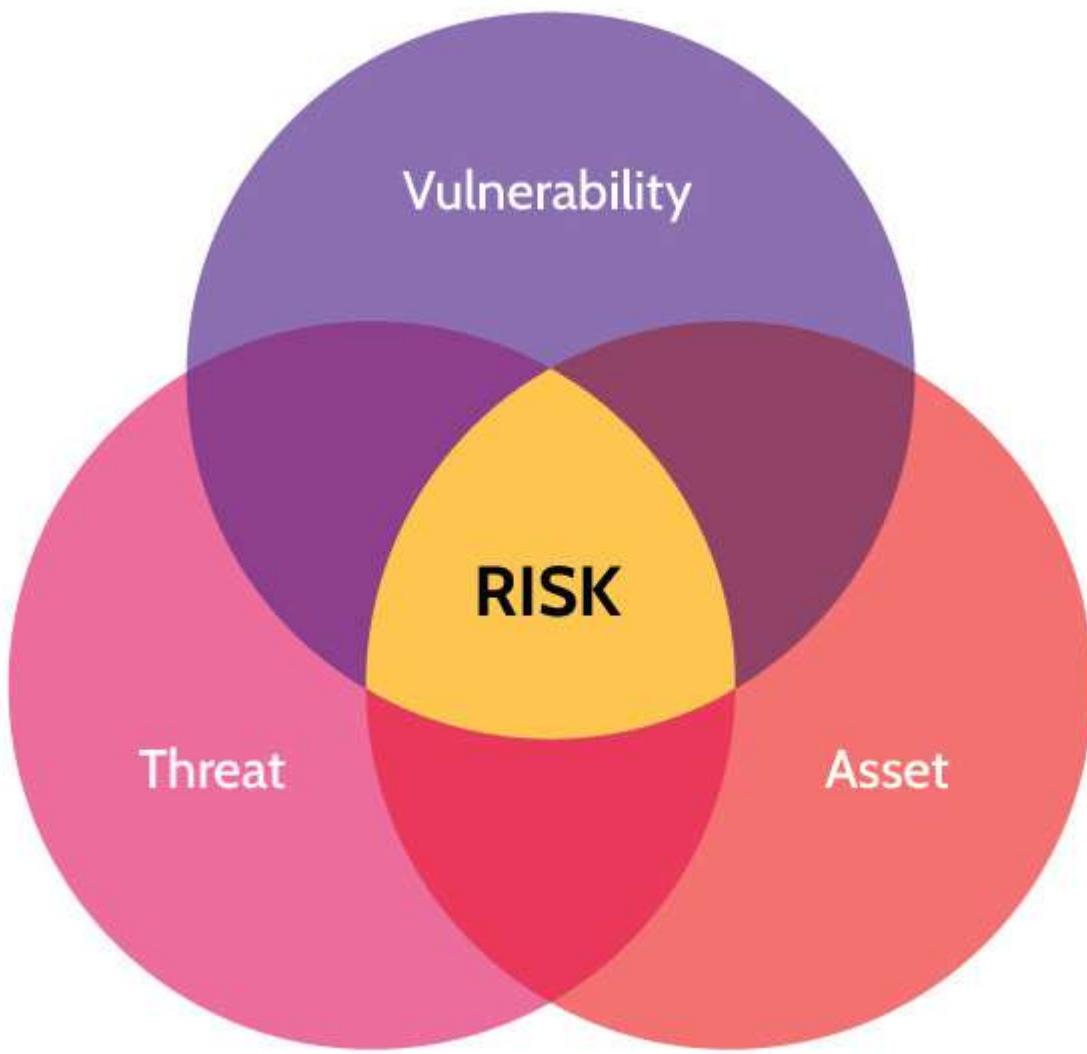
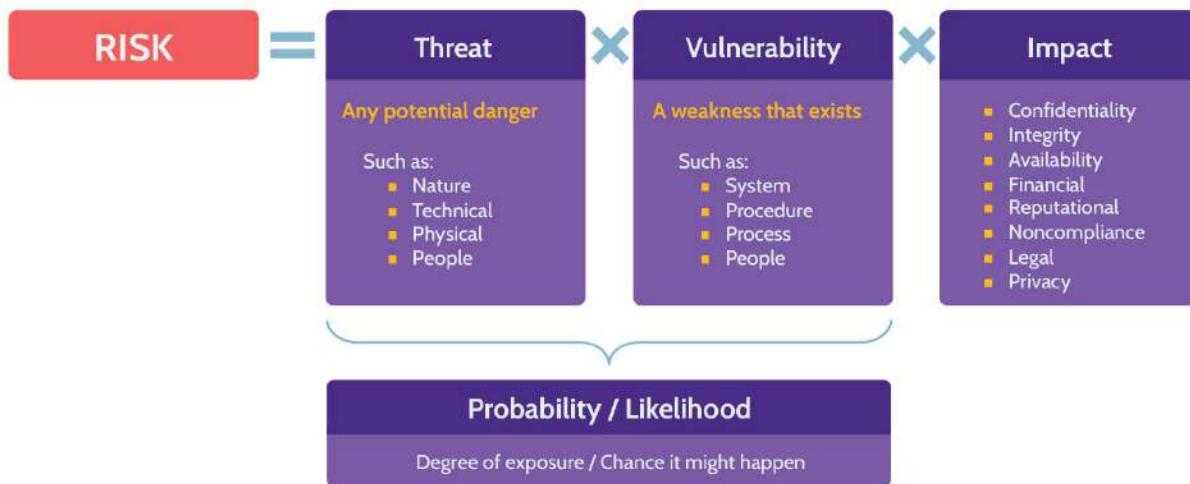


Figure 1-4: **Risk Exposure**

To fully understand each risk for a given asset, two additional pieces beyond threats and vulnerabilities must be considered: impact and probability. The impact is whatever negative consequences there may be to the organization if a risk occurs. Finally, the probability/liability is how often a given risk is expected to occur.

[Figure 1-5](#) summarizes how these components fit together and are used to identify the risks for a given asset.



**Figure 1-5: Relationship between Risk, Threat, Vulnerability, and Impact**

## Risk Management Terms

[Table 1-20](#) contains a list of core terms used in risk management and how they fit together.

<b>Threat Agent</b>	Entity that has the potential to cause damage to an asset (e.g., external attackers, internal attackers, disgruntled employees)
<b>Threat</b>	Any potential danger
<b>Attack</b>	Any harmful action that exploits a vulnerability
<b>Vulnerability</b>	A weakness in an asset that could be exploited by a threat

<b>Risk</b>	Significant exposure to a threat or vulnerability (a weakness that exists in an architecture, process, function, technology, or asset)
<b>Asset</b>	Anything that is valued by the organization
<b>Exposure/Impact</b>	Negative consequences to an asset if the risk is realized (e.g., loss of life, reputational damage, downtime, etc.)
<b>Countermeasures and Safeguards</b>	Controls implemented to reduce threat agents, threats, and vulnerabilities and reduce the negative impact of a risk being realized
<b>Residual Risk</b>	The risk that remains after countermeasures and safeguards (controls) are implemented

**Table 1-20: Risk Management Core Terms**

Figure 1-6 shows how all the terms mentioned in Table 1-20 interconnect.

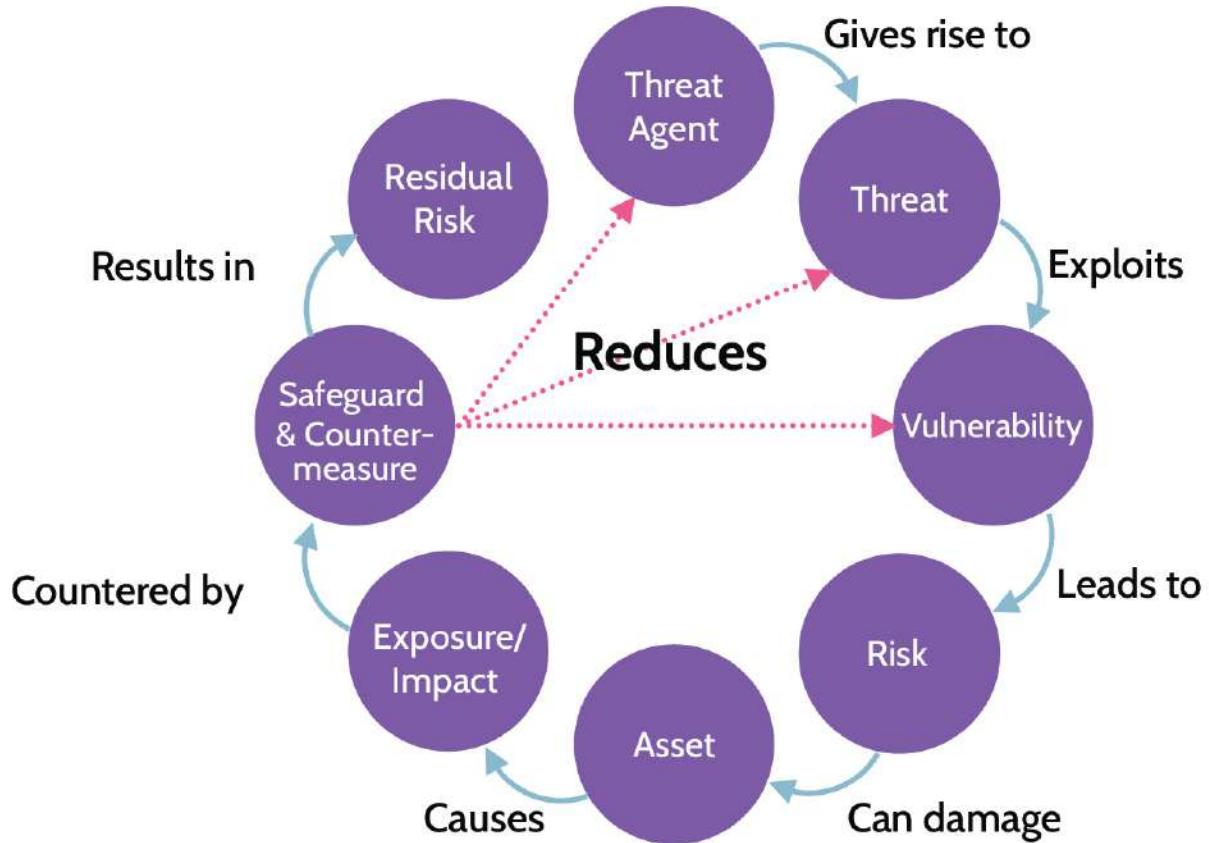


Figure 1-6: Connections Between Core Risk Management Terms

#### 1.9.4 Annualized Loss Expectancy (ALE) Calculation

##### CORE CONCEPTS

$$\text{■ ALE} = \text{SLE} (\text{AV} \times \text{EF}) \times \text{ARO}$$

- Definitions of ALE, SLE, AV, EF, ARO
- Simple calculations using a formula to calculate SLE and ALE

In the section on asset valuation, two ranking methods were mentioned: quantitative and qualitative analysis.

Quantitative analysis as part of ranking risks requires calculating how much a risk is expected to cost the organization annually—the Annualized Loss Expectancy (ALE). The ALE can be calculated using this formula: **ALE = SLE (AV x EF) x ARO**

Definitions of the five components of this formula are as follows, using a CCTV system as an example throughout: ■

**Asset Value (AV):** The cost of the asset in a monetary value, e.g., a CCTV system that costs \$2,000.

■ **Exposure Factor (EF):** Measured as a percentage and expresses how much of the asset's value stands to be lost in case of a risk materializing, e.g., if the voltage spikes excessively during certain periods of the year, a CCTV might lose three cameras to damage, thus costing \$200. This represents 10 percent of the total cost (which is \$2,000) and thus makes the EF be 10 percent. The EF will always be a percentage between 0 to 100 percent.

■ **Single Loss Expectancy (SLE):** Denotes how much it will cost if the risk occurs once. To calculate the SLE, simply multiply the AV by the EF:  $SLE = AV * EF$ ,

which in this example becomes  $\$2,000 * 10\% = \$200$ .

- **Annualized Rate of Occurrence (ARO):** Denotes how many times each year the risk is expected to occur. For example, if the voltage spikes excessively three times a year, the ARO is 3.
- **Annualized Loss Expectancy (ALE):** Expresses the annual cost of the risk materializing. To calculate it, use the following formula:  $ALE = SLE * ARO$ , which in this example becomes  $\$200 * 3 = \$600$ .

The ALE is a very useful figure, as it shows exactly how much money a given risk is expected to cost the organization per year, and can therefore provide guidance on what controls are cost-justified and should be put in place.

It is not a good business practice to implement controls that cost more than the risk they are meant to mitigate. If the cost of a control is more than the cost of the risk, a good business decision would be to accept the risk. Owners of an asset are best positioned to make this risk acceptance decision.

While the results of the ALE calculation are extremely useful, and quantitative analysis is highly preferred over

qualitative analysis, it's extremely difficult to perform this calculation. Most of the numbers you need are quite difficult to assess accurately. [Figure 1-7](#) graphically represents the formulas for the calculations mentioned above.



Figure 1-7: **SLE and ALE Calculations**

### 1.9.5 Risk Response/Treatment

#### CORE CONCEPTS

- Risk can be managed via four approaches
- Risk can never be entirely eliminated

After the risk analysis process, security should implement the most cost-effective treatments. The right approach depends on the value of the asset and type of risk identified

in the previous steps. [Figure 1-8](#) shows the four ways that risk can be managed, using a rather ridiculous diving board as an example.

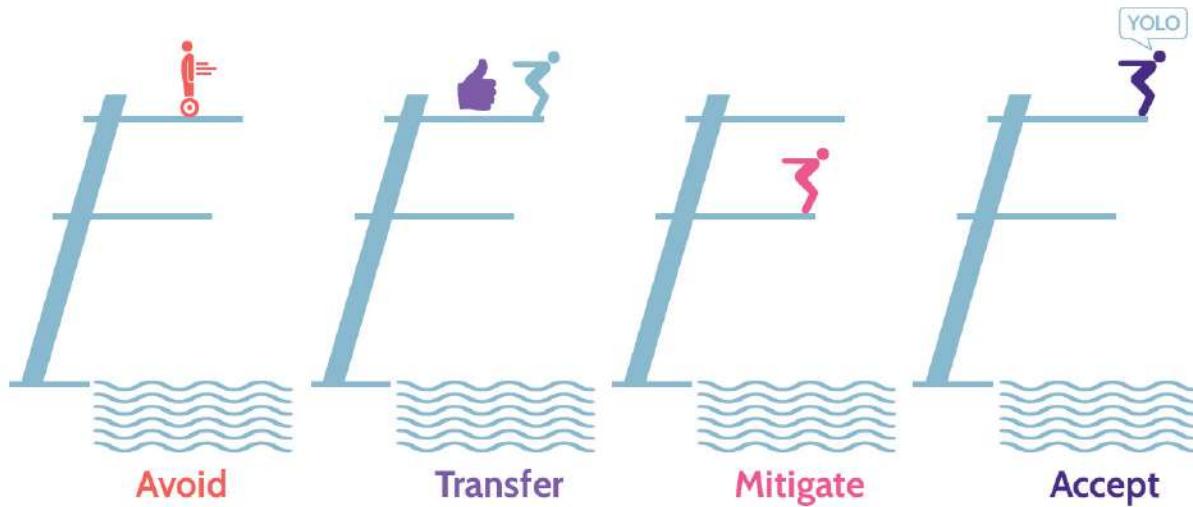


Figure 1-8: Risk Treatments

To **avoid** risk means to choose to stop doing whatever exposes the asset to risk. When risk is avoided, significant opportunities might also be lost—opportunity cost. In addition, it may also lead to other risks. For example, avoiding flying may lead to driving across different areas, which may have a higher risk than flying. Risk avoidance is not usually the first choice an organization makes when dealing with risk. The opportunity cost aspect is important. Organizations must always be taking a degree of risk to continue to expand, innovate, and remain relevant. If a risk is avoided, then all the potential upside of the risk is also

avoided. Therefore, risk avoidance should be used very selectively.

Using our diving board example, how do we avoid the risk? **Don't jump.** But of course, the opportunity cost is you miss out on the fun of jumping.

To **transfer** risk means to share some of the risk with another party, usually an insurance company. In this case, the insurer, because of what's called a premium payment, commits to paying the organization if a risk becomes reality. However, even when risk is transferred, ultimate accountability remains with the organization. Responsibility for managing the risk can be transferred, but accountability for the consequences of failing to manage it may never be transferred.

Using our diving board example, how do we transfer the risk? Get someone else to jump, or at least ensure your life insurance policy is up to date before jumping. Companies can take out cybersecurity insurance, which is a specialized insurance product designed to help organizations protect against financial losses resulting from cyber-related incidents.

To **mitigate** risk means to implement controls that reduce the risk to an acceptable level. Risk can never be eliminated or reduced to zero. However, it can be reduced enough that

residual risk (the risk that remains after controls have been put in place) can be accepted or transferred. Risk mitigation is where organizations typically focus most of their efforts, and types of risk mitigation controls are described in more detail below.

Using our diving board example, how do we reduce/mitigate the risk? Jump from the lower diving board.

To **accept** risk simply means taking no action or no further action where risk to a particular asset is concerned. This commonly happens when the cost of the control exceeds the value of the asset—the best business decision is to accept the risk. Another example of where an asset owner must accept risk is the residual risk that remains after mitigating controls have been implemented. In any case where risk is accepted, the person to make this decision should **always** be the asset owner or senior management—those who are **accountable**.

And finally using our diving board example, how do we accept the risk? **Just jump.** Right from the top. Who knows, you might make it!

Note that sometimes various companies choose to ignore a risk. Risk ignorance is not a viable approach to take, nor does it adhere to due care and due diligence. For example, a

security analyst mentions to the Chief Security Officer that multiple servers have no AV installed, thus putting them at risk of being affected by malware. If the Chief Security Officer chooses to ignore that risk that was just highlighted, the consequences can be dire for the business and can lead to financial fines and reputational damage.

### 1.9.6 Applicable Types of Controls

#### CORE CONCEPTS

- A complete control is a combination of preventive, detective, and corrective controls
- In defense-in-depth (layered security), a complete control should be implemented at each layer

#### Definitions and examples of the types of controls

Seven major types of controls can be put in place as shown in [Table 1-21](#). Understanding these different types of controls is crucial to carrying out defense-in-depth, which is an approach to security that involves multiple layers of controls. This is also sometimes referred to as layered security.

##### Directive

Directive controls direct, confine, or control the actions of subjects to force or encourage compliance with security policies. An example is a fire exit sign.

<b>Deterrent</b>	Deterrent controls discourage violation of security policies. An example is a sign warning that a piece of land is private property and trespassers will be shot. Nothing prevents someone from walking past the sign, but it's a good deterrent.
<b>Preventive</b>	Preventive controls can prevent undesired actions or events. For example, a fence that prevents someone from walking onto a private property. Or not having flammable materials around and therefore preventing a fire from starting.
<b>Detective</b>	Detective controls are designed to identify if a risk has occurred. Importantly, detective controls operate after an event has already occurred. An example is a smoke alarm detecting smoke.
<b>Corrective</b>	Corrective controls are used to minimize the negative impact of a risk occurring—minimize the damage. They are used to alleviate the impact of an event that has resulted in a loss and to respond to incidents in a manner that will minimize risk. An example is a fire suppression system activating.
<b>Recovery</b>	Recovery controls are designed to recover a system or process and return to normal operation following an incident. An example is a data backup policy allowing restoration of data on an affected server after an incident has taken place.
<b>Compensating</b>	Compensating controls are typically deployed in conjunction with other controls to aid in enforcement and support of the other controls. However, compensating controls can also be used in place of another control to provide the needed security. An example is deploying a Host Intrusion Prevention System (HIPS) on a critical server,

in addition to having a Network Intrusion Protection System (NIPS) operating on that server's subnet. This way, if any offending traffic manages to slip by the NIPS tool, the HIPS on the server may still be able to prevent malware from damaging it.

**Table 1-21: Types of Security Controls**

Remember that detective, recovery, and corrective controls are enforced after an incident is present. However, deterrent, directive, preventive, and compensating controls are applicable before an incident takes place. It is always better to stop something bad from happening than it is to deal with it after it has happened.

A concept that is pervasively used in security is a **complete control**. A complete control is a combination of preventive, detective, and corrective controls at a minimum. The idea being that whenever controls are implemented, ideally the possible preventive control is implemented to prevent risks from occurring, but there is no perfect preventive control, thus detective and corrective controls should also be implemented. This concept of a complete control should be used whenever controls are implemented. At a minimum, ensure that preventive, detective, and corrective controls are implemented at each layer of defense.

## 1.9.7 Categories of Controls

### CORE CONCEPTS

- **Safeguards = proactive — (before the fact)**
- **Countermeasures = reactive — (after the fact)**
- **Categories of controls: administrative, physical, and logical/technical**

A way to categorize the security controls we just reviewed is as safeguards or as countermeasures.

### Safeguards vs. countermeasures

**Safeguards** are proactive controls; they are put in place before risk has occurred to deter or prevent it from manifesting. Safeguards would be directive, deterrent, preventive, and compensating controls.

**Countermeasures** are reactive controls. They are put in place after risk has occurred and aim to allow us to detect and respond to it accordingly. Countermeasures would be detective, corrective, and recovery controls.

### Definitions & examples of administrative, technical/logical, and physical controls

Controls can be further classified in three main categories:

- **Administrative:** Policies, procedures, baselines, and guidelines are all classified as administrative controls. Items like background checks, acceptable use, network policy, onboarding/offboarding policies, and similar things fall in this category.
- **Logical/Technical:** Firewalls, IPS/IDS, AV, antimalware, proxies, and similar tools fall under the logical/technical security controls category.
- **Physical:** Doors, fences, gates, bollards, mantraps, guards, and CCTV fall under the physical security controls category.

Logical/technical controls are typically used synonymously, but there is an important distinction between them. Let's take a physical firewall as an example. To vastly oversimplify, a physical firewall is made up of two major components: the hardware (power supply, CPU, RAM, NICs, etc.) and the software installed and operating on the hardware. The software is the logical component, and the hardware is the technical.

[Table 1-22](#) illustrates various control types and categories that may be implemented in an organization. Note that it isn't exhaustive but provides a good comparison of typically used controls.

	<b>Administrative</b>	<b>Logical / Technical</b>	<b>Physical</b>
<b>Directive</b>	<ul style="list-style-type: none"> <li>■ Policy</li> <li>■ Procedure</li> </ul>	<ul style="list-style-type: none"> <li>■ Configuration standards</li> </ul>	<ul style="list-style-type: none"> <li>■ “Authorized Personnel Only” signs</li> <li>■ Traffic lights</li> </ul>
<b>Deterrent</b>	<ul style="list-style-type: none"> <li>■ Guideline</li> </ul>	<ul style="list-style-type: none"> <li>■ Warning banner</li> </ul>	<ul style="list-style-type: none"> <li>■ “Beware of Dog” signs</li> </ul>
<b>Preventive</b>	<ul style="list-style-type: none"> <li>■ User registration procedure</li> </ul>	<ul style="list-style-type: none"> <li>■ Login mechanism (security kernel)</li> <li>■ Operating system restrictions</li> </ul>	<ul style="list-style-type: none"> <li>■ Fence</li> <li>■ Radio Frequency (RF) ID badges</li> </ul>
<b>Detective</b>	<ul style="list-style-type: none"> <li>■ Review violation reports</li> </ul>	<ul style="list-style-type: none"> <li>■ SIEM system</li> </ul>	<ul style="list-style-type: none"> <li>■ CCTV</li> </ul>
<b>Corrective</b>	<ul style="list-style-type: none"> <li>■ Termination</li> </ul>	<ul style="list-style-type: none"> <li>■ Unplug, isolate, and terminate connection</li> </ul>	<ul style="list-style-type: none"> <li>■ Fire suppression system</li> </ul>
<b>Recovery</b>	<ul style="list-style-type: none"> <li>■ DR plan</li> </ul>	<ul style="list-style-type: none"> <li>■ Backups</li> </ul>	<ul style="list-style-type: none"> <li>■ Rebuild</li> </ul>
<b>Compensating</b>	<ul style="list-style-type: none"> <li>■ Supervision</li> <li>■ Job rotation</li> <li>■ Logging</li> </ul>	<ul style="list-style-type: none"> <li>■ CCTV</li> <li>■ Keystroke logging</li> </ul>	<ul style="list-style-type: none"> <li>■ Layered defense</li> </ul>

Table 1-22: Control Types and Categories



### 1.9.8 Functional and Assurance

#### CORE CONCEPTS

- **Functional = control must do what it is designed to do**
- **Assurance = control can be evaluated to confirm working correctly — (it provides proof and confidence)**

A good security control should always include two aspects: functional aspect and assurance aspect. [Figure 1-9](#) and [Table 1-23](#) depict and define the functional and assurance aspects.

# Controls



Figure 1-9: Functional and Assurance

Functional	Assurance
Control performs the function it was designed to address/does what it is meant to do. For example, a firewall filtering traffic between different subnets.	Control can be proven to be functioning properly on an ongoing basis. Usually proven through testing, assessments, logging, and monitoring, etc.

Table 1-23: Functional and Assurance

Anytime a control is implemented, it should include these two aspects. The control should perform some function (e.g., control the flow of network traffic, only allow authorized

employees to enter a building, detect smoke from a fire), and there should be some means of verifying/obtaining assurance that the control is working effectively on an ongoing basis.

### 1.9.9 Selecting Controls

#### CORE CONCEPTS

- Selected controls must support organizational goals and objectives
- Selected controls must be cost-effective

#### How to determine if a control should be implemented

#### How much security is enough?

When selecting appropriate security controls, there's a tendency to select the most expensive and top-performing solutions in an effort to provide the maximum level of security to the environment, but this doesn't necessarily make these cost-effective. Security is usually a balancing act between achieving the maximum level of security with the least amount of cost, and at the same time allowing proper functionality.

It is important to remember that implementing any security control has a negative impact on the organization. Security controls make systems more difficult to use, slower, more

complicated, and so on. Security for the sake of security must be avoided.

Criteria that should be evaluated as part of deciding what controls to implement include:

- Alignment to organizational goals and objectives—does a control help an organization achieve its goals and objectives, or is the control an impediment?

- Cost-effectiveness—every control must be cost-justified.
  - Complete control—a combination of preventive, detective, and corrective controls at a minimum.
  - Functional and assurance effectiveness
- Measuring Control Effectiveness and Reporting**

Once a control, or set of controls, has been decided upon and implemented, it is important to understand how well they're working. One of the best ways to do this is using metrics. To identify the metrics that will matter, the metrics that will be useful to implement and monitor, the target audience must be identified. Further, discussion and research must be done to understand what the target audience need to know—what metrics will provide them with the information they need.

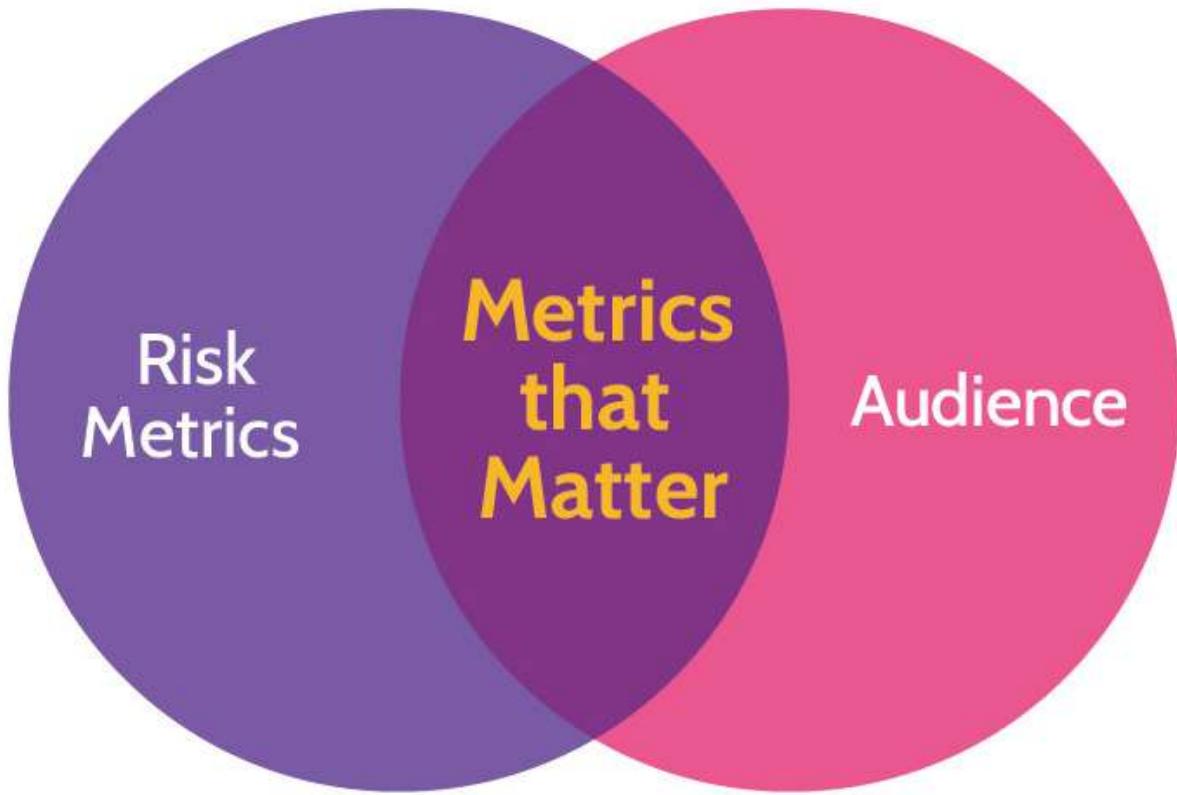


Figure 1-10: **Metrics that Matter**

Different metrics will be valuable to different audiences. For example, senior management will be more interested in “big picture” metrics, while the facilities operations team is more likely to be interested in more detailed metrics that apply directly to their everyday work. Metrics for control status can originate from multiple sources, such as internal monitoring, internal or external auditors, and third-party reports. In addition, the audience can vary and include management, regulators, internal teams, and customers. [Figure 1-10](#) depicts this concept of metrics that matter—

metrics that tell the intended audience what they need to know.

## **Continuous Improvement**

The landscape covered by the risk management process is ever-changing—new assets are added, old assets are retired, new threats and vulnerabilities are identified, the impact of a risks occurring changes, etc.—thus making risk management a continuous, arduous, and time-consuming process that needs to be continually updated. The Deming Cycle, sometimes also referred to as Plan Do Check Act (PDCA), shown in [Figure 1-11](#), outlines the cyclical nature of many processes in security, including risk management. The steps of the PDCA/Deming Cycle are defined in [Table 1-24](#).

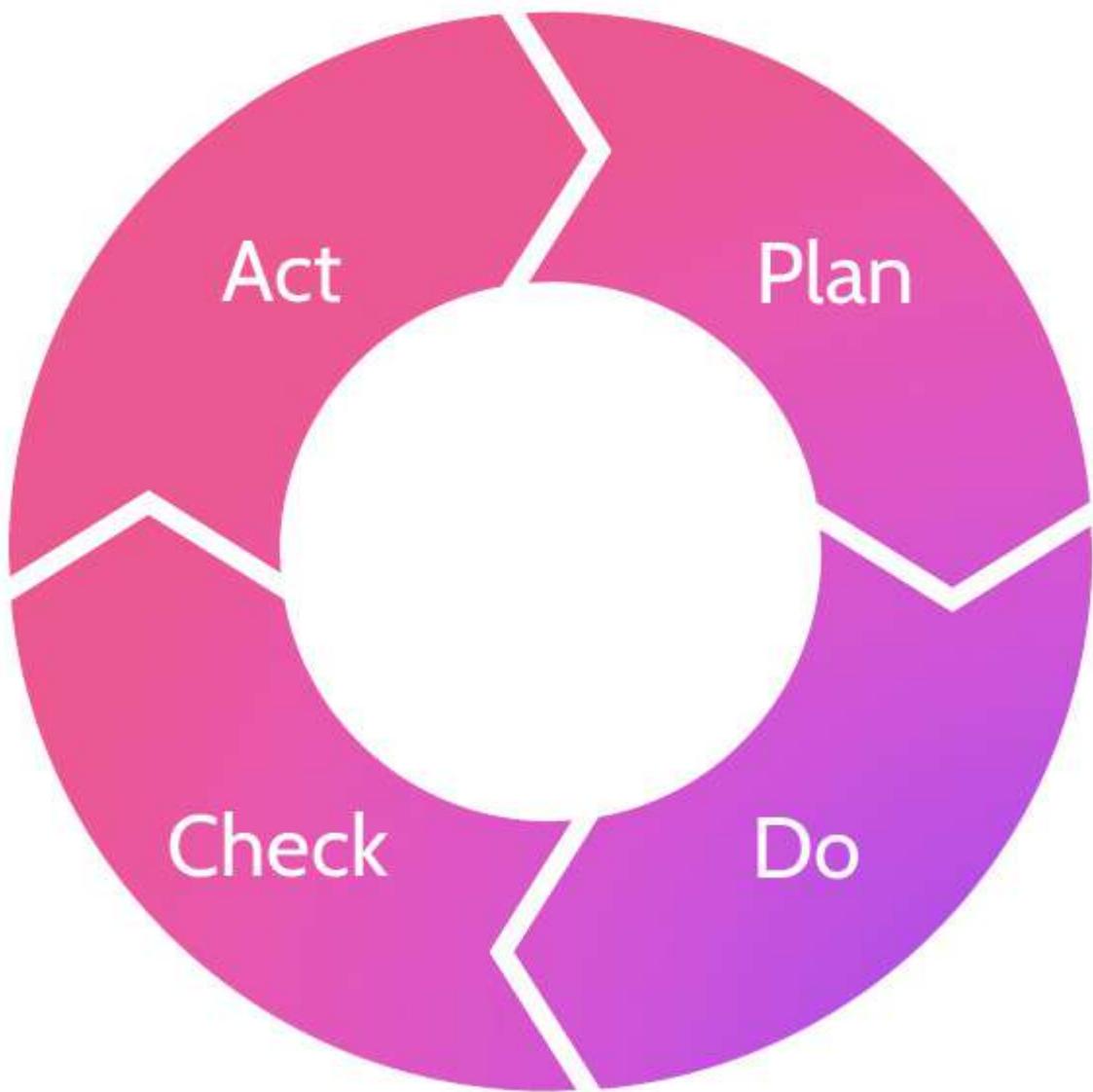


Figure 1-11: **Deming Cycle**

<b>Plan</b>	Determine which controls to implement based on the risks identified
<b>Do</b>	Implement the controls
<b>Check</b>	Monitoring and assurance; are the controls operating effectively?

Act	Based upon findings during the “Check” step, take additional actions as necessary (react), which leads back to planning.
-----	--

**Table 1-24: Deming Cycle Steps**

Risk management, like many processes in security, must be continually updated and improved. If a new asset is acquired, should a risk analysis be performed? What if a new, significant threat is identified? What if a new vulnerability is identified? What if a new potential impact has been identified? What if new regulations or laws apply? Any and all of these things should trigger an update to an organization’s risk matrix.

### **How often should a risk analysis be conducted?**

The perfect answer: as often as necessary. The frequency of risk analysis will depend on the nature of the business and associated risks, and should also be triggered by a change in the value of an asset.

#### **1.9.10 Risk Management Frameworks**

##### **CORE CONCEPTS**

- **Risk management frameworks provide comprehensive guidance for structuring and conducting risk management**

Imagine you're a newly hired risk manager, tasked with creating a risk management program: identifying all the assets, risks, threats, and vulnerabilities as well as leading the process of developing all the controls. It's a huge and complicated task, and you'd probably search for advice on best practices from someone who's done it before. Frameworks provide just that. They're collections of best practices that give you step-by-step guidance on how to perform certain activities, which controls to implement, and how to implement them. Frameworks allow you to take the collected wisdom of experts and apply it to your organization. The four risk management frameworks in [Table 1-25](#), are some examples of risk management frameworks used to address risks; NIST 800-37 is popular, so more detail on that framework is featured in [Table 1-26](#).

<b>NIST SP 800-37 (RMF)</b>	This guide describes the risk management framework (RMF) and provides guidelines for applying the RMF to information systems and organizations.
<b>ISO 31000</b>	ISO 31000 is a family of standards relating to risk management. The scope of ISO 31000 is to provide best practice structure and guidance to all organizations concerned with risk management.
<b>COSO</b>	COSO provides a definition to essential enterprise risk management components, reviews ERM principles and concepts, and provides direction and guidance for enterprise risk management.

## ISACA Risk IT Framework

ISACA's Risk IT Framework contains guidelines and practices for risk optimization, security, and business value. The latest version places greater emphasis on cybersecurity and aligns with the latest version of COBIT.

**Table 1-25: Common Risk Management Frameworks**

### NIST SP 800-37 Rev. 2

Though variations of the RMF exist, for purposes of the CISSP exam, NIST SP 800-37 Rev. 2 should be your focus. Understanding the RMF is critical, as it informs and underpins just about every facet of operational security governance within an organization. [Table 1-26](#) lists the seven steps of SP 800-37 Rev. 2.

### Steps in the NIST SP 800-37 Risk Management Framework (RMF)

1	<b>Prepare</b> to execute the RMF
2	<b>Categorize</b> Information Systems In this step, information systems are identified and categorized. It includes questions like "What do we have?"; "How does this system, its subsystems, and its boundaries fit into our organization's business processes?"; "How sensitive is it?"; "Who owns it and the data within it?" The purpose of this step is to determine any potential adverse impacts to the confidentiality, integrity, and availability of organizational operations and assets, thereby informing the organizational risk management process.

**3**

**Select Security Controls** After a risk assessment has been conducted, select, tailor, and document security controls necessary to protect the information systems. Security controls are management, operational, and technical safeguards or countermeasures embedded into information systems. They protect the confidentiality, integrity, and availability of those systems and the information contained therein; assurance provides evidence that the security controls within an information system are effective.

**4**

**Implement Security Controls** Activities at this step are based entirely on the controls selected in Step 3 and involve two key tasks: 1) implementing the selected controls in the security and privacy plans and 2) documenting the specific, baseline details of the control implementation. This latter task is critical and allows everybody to understand what controls exist and to understand the controls in the context of the larger operational framework of the organization.

**5**

**Assess Security Controls** Activities during this step help determine if the security controls are implemented correctly, operating as intended, and meeting the security and privacy requirements for the system and the organization. This step involves formulation of a comprehensive plan that must be reviewed and approved.

**6**

**Authorize Information System** This step requires senior management to decide whether it's acceptable to operate the system in question, given the potential risk, controls, and residual risk. In addition to determining if the risk exposure is acceptable, Senior Management should review the plan of action related to remaining weaknesses and deficiencies—the residual risk. Finally, this authorization or approval is usually given for a set period of time that is often tied to milestones in the Plan of Actions & Milestones (POA&M), which facilitates tracking and status of failed controls.

**7**

**Monitor Security Controls** Continuous monitoring of programs allows an organization to maintain the security of an information system over time, adapting to changing threats, vulnerabilities, technologies,

and mission/business processes. Milestones from the Authorize step are a key component of the Monitor step, which can also be considered the “continuous improvement” stage. During the Monitor step, questions like “Are the controls still effective?” and “Have new vulnerabilities developed?” are examined. Risk management can become near real-time using automated tools, although automated tools are not required. This helps with configuration drift and other potential security incidents associated with unexpected change on different core components and their configurations.

Table 1-26: NIST SP 800-37 Rev. 2 Steps

## 1.10 Understand and apply threat modeling concepts and methodologies

### Threat Modeling Methodologies

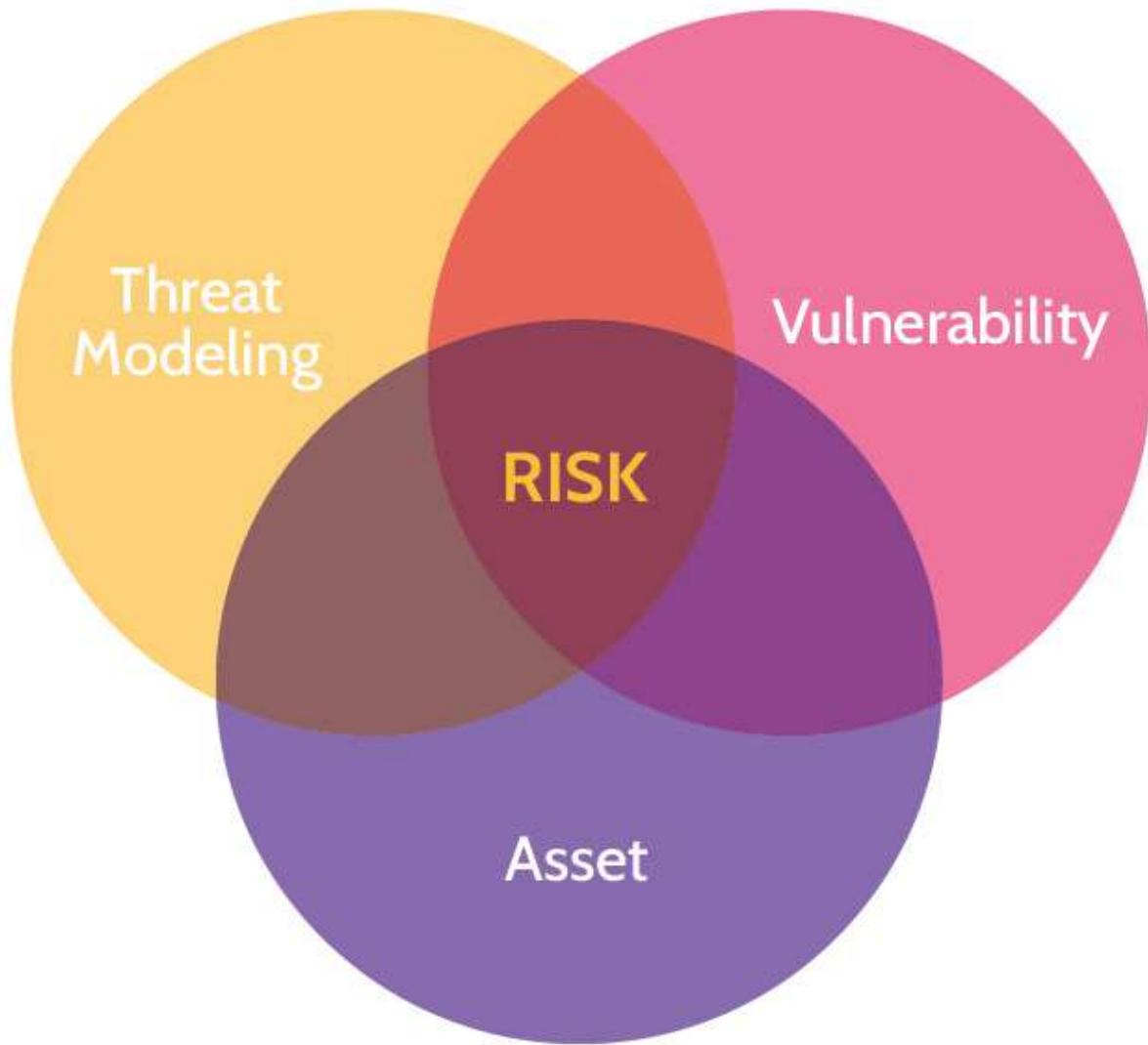
#### CORE CONCEPTS

- Threat modeling is used to systematically identify, enumerate, and prioritize threats related to an asset

#### Purpose of threat modeling

In order to perform proper risk management, it is important to identify the threats and vulnerabilities associated with each asset. Threat modeling methodologies aid in systematically identifying threats and their severity, which in turn makes risk management more accurate and effective.

[Figure 1-12](#) depicts how threat modeling fits in with overall risk analysis.



**Figure 1-12: Purpose of Threat Modeling**

Identifying all the threats to a complex asset, like a mobile phone, server, application, network, architecture, function, or process, can be a daunting task. So many possible threats exist, and it can be difficult to decide where to start and

how to proceed to ensure a systematic identification and prioritization of threats. This is where threat modeling methodologies can help. They enable: **the systematic identification, enumeration, and prioritization of threats related to an asset.**

Numerous threat modeling methodologies exist, and the primary goal of most is to provide a systematic and deliberate means of identifying and categorizing threats to a given asset. Three of the major threat modeling methodologies you need to know about for the exam are STRIDE, PASTA and DREAD.

## STRIDE

STRIDE was developed by Microsoft. Though it was initially developed as a means of assessing threats to applications and operating systems, it can be used in other contexts too. STRIDE is a threat-focused methodology that's less strategic and thorough than PASTA. STRIDE is an acronym as described in [Table 1-27](#).

### STRIDE vs. PASTA

Threat	Violation	Definition
S <b>Spoofing</b>	<b>Authentication</b>	An attacker pretends to be something or someone to gain

			unauthorized access
T	<b>Tampering</b>	<b>Integrity</b>	An attacker modifies data at rest (e.g., in a database) or in transit (e.g., over the network)
R	<b>Repudiation</b>	<b>Nonrepudiation</b>	An attacker performs an action on a system that is not attributable to them
I	<b>Information Disclosure</b>	<b>Confidentiality</b>	An attacker can read sensitive information
D	<b>Denial of Service</b>	<b>Availability</b>	An attacker prevents legitimate users from accessing an application/service
E	<b>Elevation of Privilege</b>	<b>Authorization</b>	An attacker gains elevated access rights (e.g., administrative/root access)

Table 1-27: STRIDE Model

## PASTA

Process for Attack Simulation and Threat Analysis (PASTA), contrary to STRIDE, is an attacker-focused, risk-centric methodology. It is much more detailed than STRIDE and performs threat analysis from a strategic perspective that includes input from governance, operations, architecture, and development. This is done from both business and technical viewpoints.

## Stages in PASTA

PASTA is a seven-stage threat modeling methodology, and each stage focuses on a specific set of goals and deliverables that must be achieved as seen in [Table 1-28](#):

1	<b>Define Objectives</b> —This considers the inherent application risk profile and addresses other business impact considerations early.
2	<b>Define Technical Scope</b> —The philosophy behind this stage is that you can't protect what you don't know. It's intended to decompose the technology stack that supports the application components that realize the business objectives identified from Stage 1.
3	<b>Application Decomposition</b> —This stage focuses on understanding the data flows among application components and services in the application threat model.
4	<b>Threat Analysis</b> —Reviews threat assertions from data within the environment as well as industry threat intelligence that is relevant to service, data, and deployment model.
5	<b>Vulnerability and Weakness Analysis</b> —Identifies the vulnerabilities and weaknesses within the application design and code and correlates to see if it supports the threat assertions from the prior stage.
6	<b>Attack Modeling</b> —This stage focuses on emulating attacks that could exploit identified weaknesses/vulnerabilities from the prior stage. It helps to also determine the threat viability via attack patterns.
7	<b>Risk and Impact Analysis</b> —This stage centers around remediating vulnerabilities or weaknesses in code or design that can facilitate

threats and underlying attack patterns. It may warrant some risk acceptance by broader application owners or development managers.

Table 1-28: **PASTA Stages**

## DREAD

DREAD is a threat model primarily used to measure and rank the severity of threats. DREAD is often used in combination with the STRIDE model, where STRIDE identifies the threats, and DREAD is then used to rank the severity of threats. Five key points are considered and a score for each is determined between 1 and 10 (1 being low-risk, and 10 being high-risk). The score from each of the five key points is then tallied up and divided by five to produce a final score out of ten. This final score is then used to understand the severity of a threat. DREAD is an acronym as described in [Table 1-29](#).

Key Point		Definition	Score
<b>D</b>	<b>Damage</b>	Total amount of damage the threat can cause?	<b>1-10</b>
<b>R</b>	<b>Reproducibility</b>	How easily can the threat be replicated?	<b>1-10</b>
<b>E</b>	<b>Exploitability</b>	How difficult is it to exploit the threat?	<b>1-10</b>

<b>A</b>	<b>Affected Users</b>	How many people, inside or outside the organization, will be affected by the threat?	<b>1-10</b>
<b>D</b>	<b>Discoverability</b>	How easily can the threat be discovered?	<b>1-10</b>

Table 1-29: DREAD Model

## Social Engineering

### CORE CONCEPTS

- Social engineering = manipulation of people's actions through intimidation and/or deception
- Social engineering is a prevalent means of attack against organizations
- Best way to combat social engineering is through awareness/education/training

**Social engineering can be defined as using deception or intimidation to get people to provide sensitive information that they shouldn't in order to facilitate fraudulent activities.**

In most organizations, the biggest security weakness exists between the keyboard and the back of the chair: **employees**. The way attackers persuade employees to do things they shouldn't do is through the use of social engineering techniques. Social engineering is a very

prevalent form of attack that exploits the inherent kindness and emotions of people. **It is so prevalent, because it's so effective.**

### Definitions of social engineering, phishing, vishing, and smishing

Common social engineering tactics include **intimidation** (involves inducing fear in order to manipulate someone into a specific course of action), **deception** (involves tricking someone in one manner or another), and **rappor** (building a gradual relationship with a victim in order to take advantage of it down the line).

An example of intimidation is blackmail, while lying is one of deception, and pretending to be from the IT team and wanting to help is an example of rapport.

**Table 1-30** shows some common forms of social engineering attacks.

<b>Phishing</b>	Phishing is where an attacker sends many emails with the hope that the target will open an email and click on a link or open a file that leads to a malicious action.
<b>Spear Phishing</b>	Spear phishing is a targeted form of phishing that typically focuses on certain individuals or groups of individuals. Through a bit of discovery, the attacker determines what might prompt the targeted individual(s) to click on a link in

	<p>an email, and the hook is then baited. A classic example is an attacker sending a malicious PDF posing as an invoice to the accounts payable team.</p>
<b>Whaling</b>	<p>Like spear phishing, this is also an email attack and targets the big fish—the whales—in an organization. Typically, people like the CEO, COO, and CFO are the targets of a whaling attack.</p>
<b>Smishing</b>	<p>Smishing is a form of phishing that targets mobile phone users. Typically, an attacker purporting to be from a legitimate company sends a fraudulent text/SMS message to a potential victim, with the hope that the target will click a link in the message. Smishing attacks can be simple, with the hope that a victim will click on a link and then reveal sensitive information, or they can be sophisticated and allow the attacker to control the victim's phone and thereby gain access to bank accounts, corporate resources, and other sensitive material.</p>
<b>Vishing</b>	<p>Vishing is another form of phishing, and the name refers to the way it is typically presented to a potential victim—via voice over IP (VoIP) phone systems (though attacks can take place over mobile phones, landlines, or voice mail).</p>
<b>Pretexting</b>	<p>Pretexting involves the attacker creating a scenario, almost like a script, that very ingeniously and subtly spurs the victim into action. Usually, the pretext will strike an emotional chord—whether it's your “bank” calling with news about suspicious activity related to your account, or a “friend” texting you with news about an unfortunate incident that's left them stranded someplace. Ultimately, a request is made for money, sensitive information, or both.</p>
<b>Baiting</b>	<p>Baiting is a form of social engineering that preys on people's curiosity via the use of physical tools, like USB drives. Usually, the attacker will drop some USB drives in a building parking</p>

	lot, a hallway, a convention hall, or other crowded area. Then, some employees, hotel guests, or convention attendees will find the device and plug it in to their computer to try and identify the owner and return it.
<b>Tailgating and Piggybacking</b>	Tailgating or piggybacking is the action of following a person who is authorized to enter a restricted area through a door and thus gaining unauthorized access. The difference is that in tailgating the attacker possesses a badge that is fake but looks real. In piggybacking, the attacker doesn't have any badge at all.

**Table 1-30: Social Engineering Attacks**

Mitigating most social engineering attacks can be done most effectively through awareness, training, and education. Strong security policies can also help in this regard. Additionally, there are practical steps that can and should be taken to prevent some of the attacks noted above. Some of the best steps include:

- Requesting proof of identity
- Requiring callback authorization for voice- or text-only requests for network alterations, sensitive information, etc.

- When sensitive information is being requested via email by a purported well-known entity, like a bank, contacting them “out-of-band.” In other words, not contacting the entity via any numbers in the email but rather contact them via the entity’s website or another confirmed source. Also, not reaching out

using random telephone numbers but only valid landline numbers that belong to the legitimate organization in question and can be easily found online.

## **1.11 Apply supply chain risk management (SCRM) concepts**

### **CORE CONCEPTS**

- Risk management methodologies should be applied to all vendors, suppliers, service providers

Risk management should also apply to your organization's suppliers and service providers. For example, if an organization is moving to the cloud, that should be factored as an inherent risk into their risk management process.

Even though a cloud service provider is responsible for storing data, owners are still accountable for that data. If the organization needs to be compliant with certain laws and regulations, the organization must ensure that their cloud service provider has the required controls in place to meet the organization's compliance requirements. Every organization has security dependencies with external entities—vendors, suppliers, customers, contractors—and risk management should apply to all of them, while including the following items:

- Governance review
- Site security review
- Formal security audit
- Penetration testing

Adherence to security baseline ■ Evaluation of hardware and software ■ Adherence to security policies ■ Development of an assessment plan ■ Identification of assessment requirements and which party will perform it ■ Preparation of assessment and reporting templates In short, owners need to define requirements for suppliers and communicate those requirements to all external suppliers, just as they should do for their processes. Vendors and suppliers perform a significant number of services for many organizations, and this fact should drive external risk analysis as much as internal risk analysis. An organization must be aware of and apply the same risk management process to its suppliers because accountability can't be outsourced. Supply chain risk analysis is as vital and important as any other type of risk analysis.

### **1.11.1 Risks associated with the acquisition of products and services from suppliers and providers**

**Table 1.31** discusses the main risks that should be considered when acquiring products and services.

<b>Product Tampering</b>	Unauthorized alteration or modification of a product after manufacturing, but before the product reaches the consumer. As an example, an attacker could intercept a keyboard delivery, solder a keylogger inside it, and then repackage it. The customer would have no idea that the keyboard had been tampered with. Product tampering can result in malfunctions, health hazards, the compromise of
--------------------------	---

	integrity, or the introduction of malicious code and vulnerabilities.
<b>Counterfeits</b>	Unauthorized replicas or imitations of products that are made to deceive consumers into thinking that they are purchasing the real thing. Counterfeits may result in reduced performance, hazards, regulatory violations, inappropriate security, or increased vulnerabilities.
<b>Implants</b>	Hardware or software components stealthily inserted into products to perform unauthorized activities, such as espionage or data theft. Implants can result in sensitive information being sent to unauthorized entities and unauthorized access to systems. They may grant attackers access for extended periods of time.

**Table 1.31: Product Tampering, Counterfeits, and Implants**

### 1.11.2 Risk Mitigations

[Table 1.32](#) discusses various supply chain risk mitigations.

<b>Third-party Assessment and Monitoring</b>	Evaluating and continuously monitoring the security practices and performance of third-party vendors or suppliers.
<b>Minimum Security Requirements</b>	Predefined baseline security standards that vendors must meet.
<b>Service-level Requirements</b>	Specifications set in contracts that dictate the expected performance, availability, and responsiveness of a service provided by a vendor.

<b>Silicon Root of Trust</b>	A secure cryptographic identity embedded in hardware, ensuring that the hardware starts in a trusted state and that the firmware loaded onto it is genuine. Having the cryptographic identity embedded in the hardware allows us to have a secure boot process.
<b>Physically Unclonable Function</b>	A hardware feature that uses the unique physical characteristics of semiconductor devices to generate cryptographic keys, ensuring that each device has a unique and unclonable identity. This mitigates the risk of counterfeits.
<b>Software Bill of Materials</b>	A comprehensive list of components, libraries and modules that are used to build a software product, often detailing versions, sources and dependencies. With this list, we can check what changes have been made in new versions. This can help us detect things like backdoors and other vulnerabilities.

Table 1.32: Supply Chain Risk Mitigations

## SLR, SLA, and Service Level Reports

### CORE CONCEPTS

- Security must be considered for all acquisitions
- Security must be part of procurement process
- Security requirements must be clearly communicated (e.g., SLAs) to suppliers/vendors/service providers
- Security metrics must demonstrate that security controls are operating effectively

## ■ Security's involvement in procurement

### ■ SLAs

### ■ Why security metrics are used

Acquisitions are usually made with the goal of adding value to an organization; but they often come with inherent risks, because a product or service from the outside world is being introduced to the organization. Even if the acquisition is of a well-known brand, product, or service, risks exist and must be evaluated as part of the acquisition, or procurement, process. This evaluation should take place as early as possible and include security considerations that minimize the risk that new acquisitions introduce to an organization. [Section 1.11](#) touched on the importance of conducting risk management for suppliers, and this section will provide more detail on exactly how to do it.

When an organization looks to acquire a new asset or service, any relevant security requirements must be identified and considered. Security needs to work with the owner to understand the business rationale for acquiring a new asset or service. If security does not understand how the asset will be used, who will access it, and what types of data the asset will store or be transmitted to a service provider, there is no way the right security controls can be

identified and evaluated as part of the procurement process.

After business requirements have been identified, they must be validated (confirmed), so the security requirements can be defined. The security requirements are documented in a service level requirements (SLR) document (explained in more detail below). The SLR document is used as part of the procurement process to help the organization evaluate different vendors and/or products against the documented security requirements. For example, a company employing a new credit card processing system might require that the system be PCI compliant; or if they're a health care provider, they might require that a service provider be HIPAA compliant.

Once an organization chooses a vendor or service provider, the requirements in the SLR should be included in the contract with the vendor, with stipulations about how the vendor will continue to meet the requirements on an ongoing basis. These stipulations usually take the form of a contract addendum known as the service level agreement (SLA), also explained below.

## **Service Level Requirements (SLR)**

With the acquisition of a service, additional organizational requirements must be considered, and this is done through

a document called an SLR. Specifically, an SLR outlines:

- Detailed service descriptions
- Detailed service level targets
- Mutual responsibilities

The SLR is a very important document during the procurement process, as it defines the security services and service level targets that each potential supplier can be evaluated against. When a winning supplier is selected in the procurement process, the SLR will then be used to inform the requirements that will be documented in the SLA.

## **Service Level Agreement (SLA)**

One important note: even though the agreement is between a service provider and a customer, the customer remains accountable for all customer data being processed by the provider. SLAs are addendums to the contract and are therefore enforceable. SLAs often include expectations and stipulations related to:

- Service Levels (performance levels)
- Governance—the customer and the service provider know who is responsible for what
- Security—expected security controls put in place by the service provider that speak to the topic of accountability and responsibility.

Accountability can never be outsourced; thus, the security controls needed to protect customer data must be very clearly defined by the customer and put in place to exact specifications by the service provider.

- Compliance with all laws and regulations that relate to the customer's industry or where the customer conducts business ■ Liability/Indemnification when any element of the SLA is not met or is below threshold standards To understand how a service provider is performing on behalf of a customer, and particularly to identify how well expectations defined in the SLA are being met, a service provider will provide service level reports on an ongoing basis.

## Service Level Reports

Service level reports are issued by a vendor or service provider to a client and provide insight and information about the service provider's ability to deliver services as defined by the SLA. The service level report compares anticipated and agreed upon service levels with actual service levels and documents the effectiveness of security controls, which allows the customer—the owner—to gain assurance that expectations are being met.

A service level report might contain any of the following components:

- Achievement of metrics defined in the SLA ■ Identification of issues ■ Reporting channels ■ Management ■ Third-party SOC reports, which

provide independent verification and assurance that the terms of the SLA are being met. There are cases, for example, when considering the acquisition of a cloud-based service, that the acquiring company will not be able to evaluate all facets of the service provider's offering or when a customer wants an outside company to evaluate whether the terms of an SLA are being met. In these situations, third-party assessment and monitoring tools and services can be utilized for the same purposes. Thus, even though a service provider may not allow an organization's auditors onsite to perform an audit, the potential customer can rely on the audit report (usually in the form of a SOC 2, Type 2 report) from a trusted third-party audit firm. This is known as third-party assurance, and it will be discussed in more detail in [section 6.5.2](#).

## **1.12 Establish and maintain a security awareness, education, and training program**

### **1.12.1 Methods and Techniques to Increase Awareness, Training, and Education**

#### **CORE CONCEPTS**

- **Everyone is responsible for security; however, they must know what to do**

- The goal of awareness is to change cultural sensitivity to a topic or issue
- Training provides specific skills
- Education provides understanding and decision-making capability

## Who is responsible for security?

**EVERYONE** in an organization is responsible for security. However, it's not nearly sufficient to simply say, "Everyone is responsible for security." Employees must understand and know how to execute their security responsibilities. This implies that organizations must provide awareness, training, and education so that everyone knows and understands their security responsibilities.

### Purpose of security awareness training

Awareness within an organization is fostered with the goal of creating cultural sensitivity to a given topic or issue. Awareness is usually done at an organization-wide level and is designed to get every employee on the same page, so they're all doing things related to security in a similar manner. Examples of awareness include internal phishing campaigns, lunch and learns, and awareness posters hung in visible places.

## Definition of awareness

Training provides specific skills needed to perform tasks related to security. It often focuses on the technical aspects of a role. Examples of training might include a firewall administrator learning how to write firewall rules or a security guard learning how to respond to different situations related to protecting a building and the assets within.

Finally, education helps people understand fundamental concepts and therefore develop decision-making skills and abilities. [Table 1-33](#) contains a comparison between these three.

Awareness	Training	Education
<ul style="list-style-type: none"><li>■ Raises cultural awareness and sensitivity within an organization ■ Organization-wide</li><li>■ Less time involved</li></ul>	<ul style="list-style-type: none"><li>■ More technical ■ Focuses on specific skills related to security-related task/role</li></ul>	<ul style="list-style-type: none"><li>■ Focuses on fundamental concepts ■ Develops decision-making skills</li></ul>

**Table 1-33: Awareness, Training, and Education Comparison**

## **Methods and Techniques to Provide Awareness and Training (e.g., social engineering, phishing, security champions, gamification)**

The key to engaging awareness, training, and education is to be creative and to use methods that effectively convey the message. Additionally, it's important to speak the audience's language; to talk in terms that will best resonate. In other words, the language used when speaking to members of upper management will be very different from the language used when speaking to members of the IT staff.

Common methods to accomplish this task include:

- Live in-person sessions.
- Live online sessions.
- Pre-recorded sessions.
- Requirements/rewards or gamification ■ Establish security champions, which are employees who promote security awareness and security issues among their colleagues. A good example is a developer who engages with the security team.
- Regular communications/campaigns.

## Prioritization of Topics

There is never enough time to train everyone on everything, so topics selected for awareness, training, and education should directly align with the organization's goals and objectives. A good source to aid in the identification of topics is the organization's risk register. Risk management identifies the most valuable assets and their associated risks that should help drive awareness, training, and education initiatives.

### **1.12.2 Periodic content reviews to include emerging technologies and trends**

Organizations and the surrounding threat landscape are constantly changing; therefore, awareness, training and education programs and materials should also evolve and be updated accordingly to be most effective. Technologies like blockchain (and the cryptocurrencies built on top of them) and AI have gained increasing prominence in recent years. However, changes may extend beyond the latest technology trends. Organizations should consider all changes to their threat environment when performing periodic content reviews. This helps to ensure that training, awareness and education materials are up-to-date and ready to prepare employees to tackle the latest security challenges and threats.

## Metrics to measure effectiveness of a security awareness and training program

### 1.12.3 Program Effectiveness Evaluation

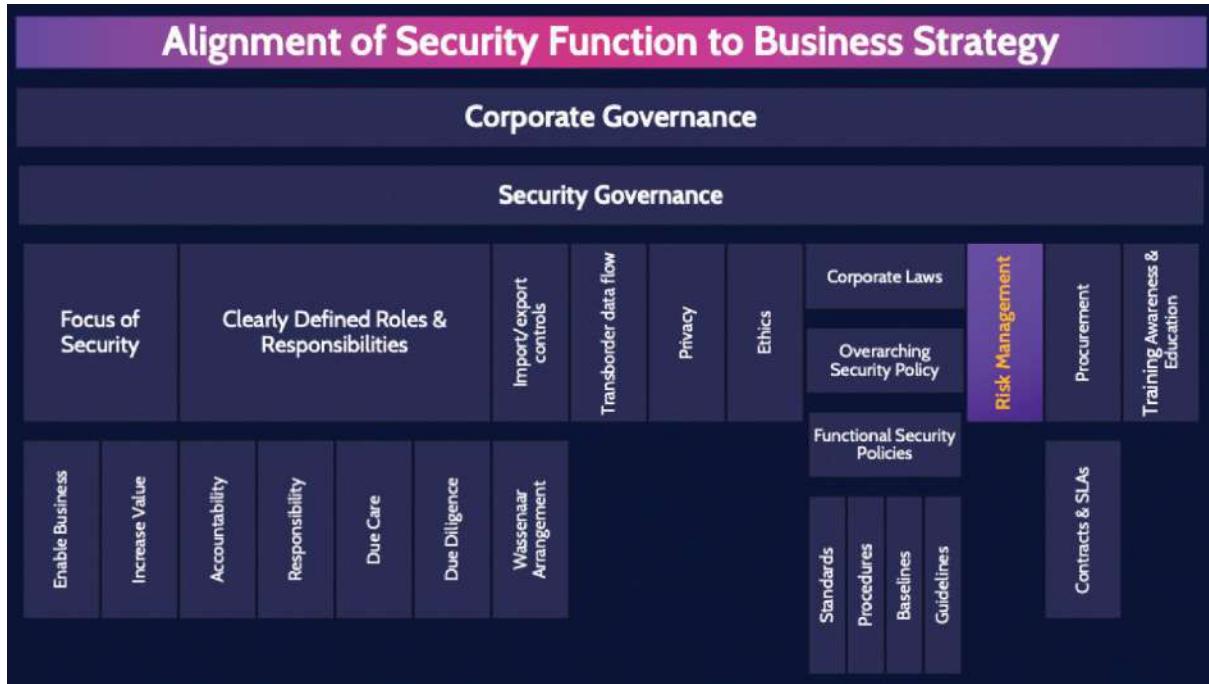
Speaking of effectiveness, program participants should be surveyed from time to time, and knowledge should be assessed via items like simulated phishing exercises or interactive multimedia presentations that include short quizzes.

Some key metrics that can be used to track effectiveness can be:

- Total number of people completing the awareness program
- Number of people providing feedback in comparison to total attendees
- Number of people reporting suspicious activities after training completion
- Tracking of how well staff members performed. For example, assuming a passing score of 75 percent:
  - Percentage passing with a score of 75 to 85 percent
  - Percentage passing with a score of 86 to 90 percent
  - Percentage passing with a score of 91 percent and above
- Total number of attempts the course was taken by each person

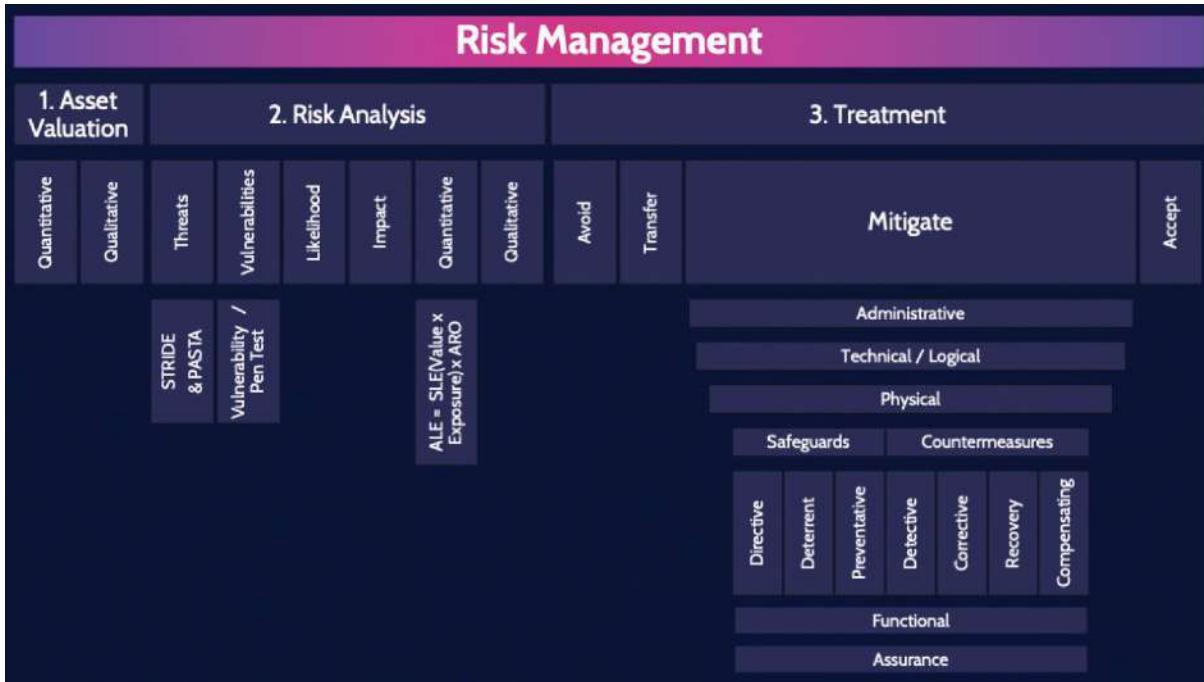


MINDMAP REVIEW **VIDEOS**



## Alignment of Security Function to Business Strategy

[dcgo.ca/CISSPmm1-1](http://dcgo.ca/CISSPmm1-1)



## Risk Management

[dcgo.ca/CISSPmm1-2](http://dcgo.ca/CISSPmm1-2)

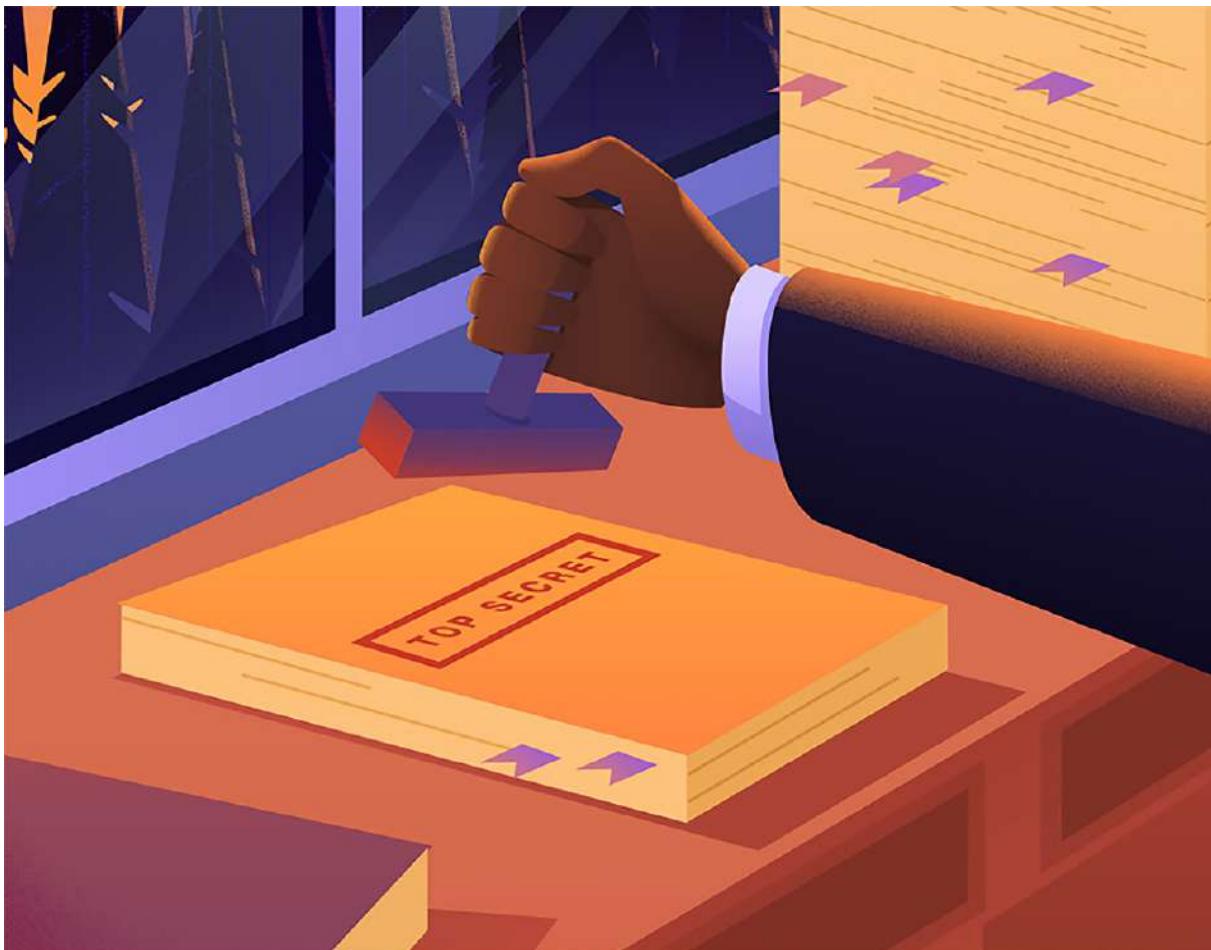
## Privacy

**State or condition of being free from being observed or disturbed by other people**

Privacy policy		Personal Data Information Lifecycle					OECD Guidelines					GDPR														
Standards	PII	Procedures	SPI	Baselines	PHI	Guidelines	PI	Direct Identifiers	Indirect Identifiers	Online Identifiers	Creation / Update	Store	Use	Share	Archive	Destroy	Collection Limitation	Data Quality	Purpose Specification	Use Limitation	Security Safeguards	Openness	Individual Participation	Accountability	Supervisory Authority (SA)	Cannot Achieve Privacy without Security

## Privacy

[dcgo.ca/CISSPmm1-3](http://dcgo.ca/CISSPmm1-3)



Domain 2

## Asset Security

## DOMAIN 2

# ASSET SECURITY

Asset security includes the concepts, structures, principles, and controls aimed at protecting assets—anything that represents value to the organization.

Security professionals need to be vigilant about asset protection. Even one minor vulnerability can leave a whole system exposed to a security breach, causing an organization to lose money and data, and possibly even compromise the entire company. Good security professionals cover their bases with a systematic approach to asset security: Know what you have, classify it, and protect it based on its classification level, which indicates its value to the organization. You can't protect something if you don't know it's there and the value it represents to the organization.

The concept is simple, but the execution is often incredibly difficult, especially for larger organizations with a lot of assets. Domain 2 gives an overview of the steps involved in asset security to address some of the issues that security professionals often encounter while implementing them.

## **2.1 Identify and classify information and assets**

### **2.1.1 Asset Classification**

#### **CORE CONCEPTS**

- **Asset classification policies, procedures, and processes help achieve proper protection of assets**

As we learned in Domain 1, protection of assets should always be based on the value of the asset. We also said that the owners of the asset are always in the best position to understand the true value that the asset represents to the organization.

One of the most common problems that organizations face is that they don't know what assets they have or how valuable those assets are. For example, a department manager might have signed up for a cloud service for use by their team, then forgot about it over time, or signed up for the service but never assessed the value of the data stored in it. This leaves the organization vulnerable, particularly if there's valuable data that isn't being protected adequately. Organizational policies, procedures, and processes should be put in place to address the requirement to protect valuable assets. An asset classification program and inventory system would be the starting point that organizations can use to address and

properly protect assets. In a large multinational organization, this can be a huge undertaking given the many types of assets within an organization. Additionally, the fact that assets can be created, purchased, rented, or taken over makes the task even more challenging.

In Domain 1, the concept of assets was introduced in addition to balancing the cost of their protection with their value to the organization. In other words, protecting assets should always be based on the value of the asset, and therefore, for security to be an enabler, protecting assets should always be cost-effective. As the value of an asset increases, so does the effort invested in protecting it. Less valuable assets might not warrant costly protection. The first step in asset classification addresses this issue with an asset classification system, a series of classes that represent the level of protection that each type of asset requires. For example, an asset that is classified as “top secret” or “proprietary” is going to have significantly more value than an asset classified as “confidential” or “public.” The security team must apply a valid and cost-justified baseline of controls for each level of classification; these controls will be used to create protection baselines for each classification level.

**Classification is driven by the value of the asset**

Proper asset classification ensures that assets receive an appropriate level of protection based on the value that they represent to the organization. Asset classification can be defined as ***assigning assets the level of protection they require, based on their value to the organization.***

Once assets are captured in an inventory system, the next step is to identify the owners of each asset. Once the asset classification system is in place and asset owners have been identified, security can then work with owners to assign assets an appropriate classification level that determines how the assets are protected. It's important to note that owners are ultimately accountable for ensuring their assets are classified and thus protected appropriately. In fact, it's not uncommon for owners to challenge ownership to avoid being accountable. When this happens, an organization's governance committee must set the tone from the top, stating that owners must own the asset, and the security function is there to support the implementation of suitable controls. In short, assets need to have an accountable owner to ensure proper controls are applied (more on this in [section 2.2](#)).

## Information Classification Benefits

The information classification process provides several benefits, including:

- Identification of **critical** information:

identifies information that the organization considers critical to business success.

- Identification of **sensitivity** to modification:  
classification helps identify data that must only be modified in specifically authorized ways.
- Commitment to **protect** valuable assets: creates awareness among users that the organization is committed to protecting assets from unauthorized access.
- Commitment to **confidentiality** where applicable:  
classification helps ensure that sensitive information remains confidential.

## 2.1.2 Classification Process

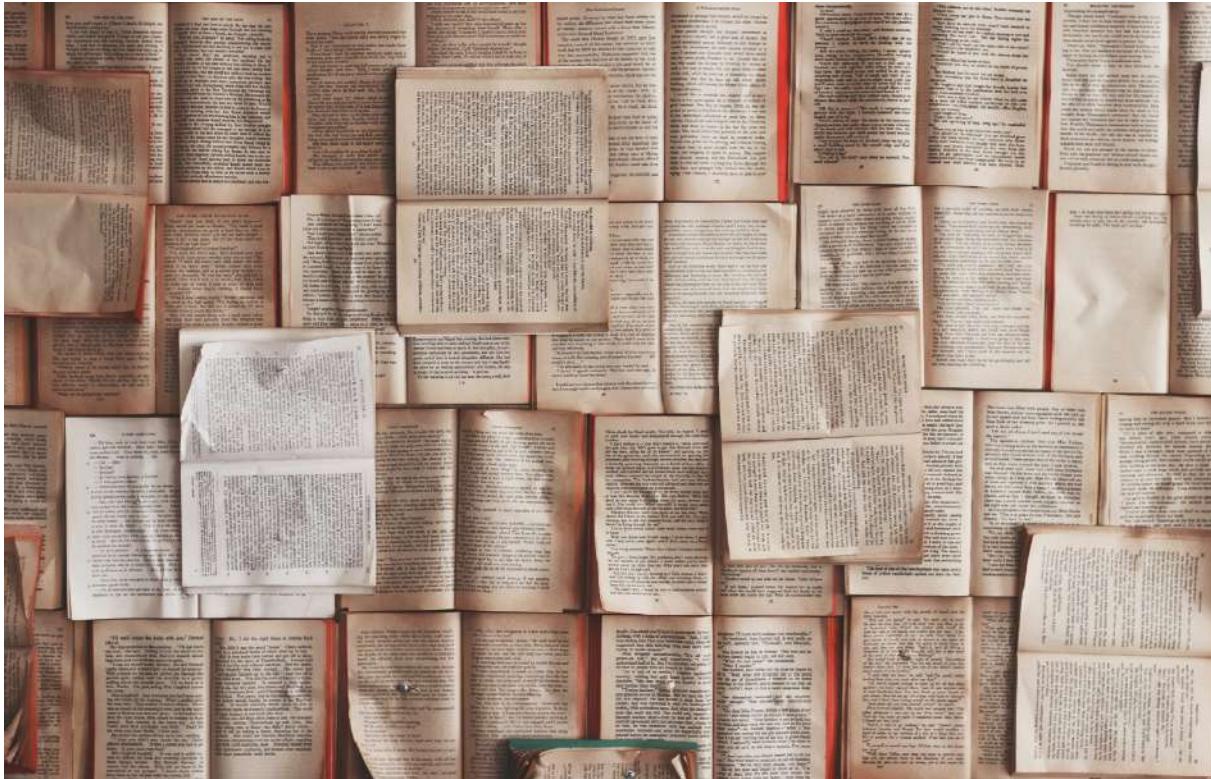
### CORE CONCEPTS

- Asset classification begins with a detailed asset inventory
- Asset owners determine the classification assigned to an asset
- Asset classification is an ongoing process

**Data classification** ensures that data receive an appropriate level of protection. It sounds simple, but it's a complex process. To be effective, it requires the right tools, procedures, education, and training. As a result, a lot of

organizations struggle with optimizing their data classification.

It's not sufficient to just talk about data classification. An organization needs more than just data classification, they need an entire "asset" classification system. An asset can be defined as something that represents value, either quantitative or qualitative, to an organization. When you think about it, data is only one example of an asset that represents value to an organization. Other assets that represent value need to be protected using classification systems, just like data. In fact, progressive companies, especially those that are heavily regulated, have expanded their data classifications systems to include all assets, not just data. They have realized that protecting all valuable assets can be best achieved by using asset classification systems that include all assets regardless of whether they are tangible or intangible.



If we remember the purpose of classification systems, which is to protect assets based on value, we also need to remember that value to assets can be represented through confidentiality (sensitivity), integrity (accuracy and meaningfulness), and availability (criticality). It stands to reason, then, that we need to classify assets based on those three characteristics. Value to assets can be represented in all three (sensitivity, accuracy, and criticality), which means that assets should be classified using three classifications, one for confidentiality, integrity, and availability.

Therefore, one of the other improvements that companies have made is to not only expand data classification systems to encompass all assets, but to also classify assets NOT only

based on confidentiality requirements, but also to include integrity and availability. Many asset classification frameworks and industry guidance today will encourage companies to classify valuable assets based on three classifications: one for confidentiality (sensitivity), one for integrity (accuracy), and one for availability (criticality).

For classification to be done properly, it needs to be driven by the owners. A risk with the classification process is that some owners tend to overprioritize their assets during the process. This is a common problem: every asset owner thinks their asset is among the most important. But when the time comes to calculate the costs for the protection of those high-importance assets, owners often take the opposite approach and under classify assets. The solution to this common problem is an asset classification committee or working group, comprised of qualified representatives from different areas of the organization. This committee or working group can provide a more objective classification process.

Another good way to address this problem is for organizations to ensure they have a consistent process to classify assets. This could be achieved through a consistent scoring system that is used by owners to understand the real value that assets represent to the organization. This scoring system and the way it has been used by specific

owners in the organization could be vetted and approved by an asset classification board or committee.

Asset classification is ongoing. Because asset classification helps identify the appropriate controls for a given asset, and the nature of assets changes over time, it means that classification levels can change over time too. An asset that is classified as “top secret” today might very well become less valuable to the organization tomorrow, which would warrant a different classification level. Accordingly, the classification of assets must be adjusted over time to ensure the assets are protected based on the value that they represent to the organization.

The classification of an asset also drives archiving and retention requirements. Laws, regulations, industry standards, privacy requirements, company policies, and related guidance might dictate that an asset be retained for a specific number of years, even though it's not being actively used by the organization. Similarly, the same guidance might indicate that the asset should be defensibly destroyed after a set period of time. Thus, the asset life cycle represents a continually changing paradigm that should be monitored and administered on an ongoing basis.

## Asset classification steps

A summary of the asset classification process is depicted in [Figure 2-1](#). The classification process begins with maintaining a continually updated asset inventory.

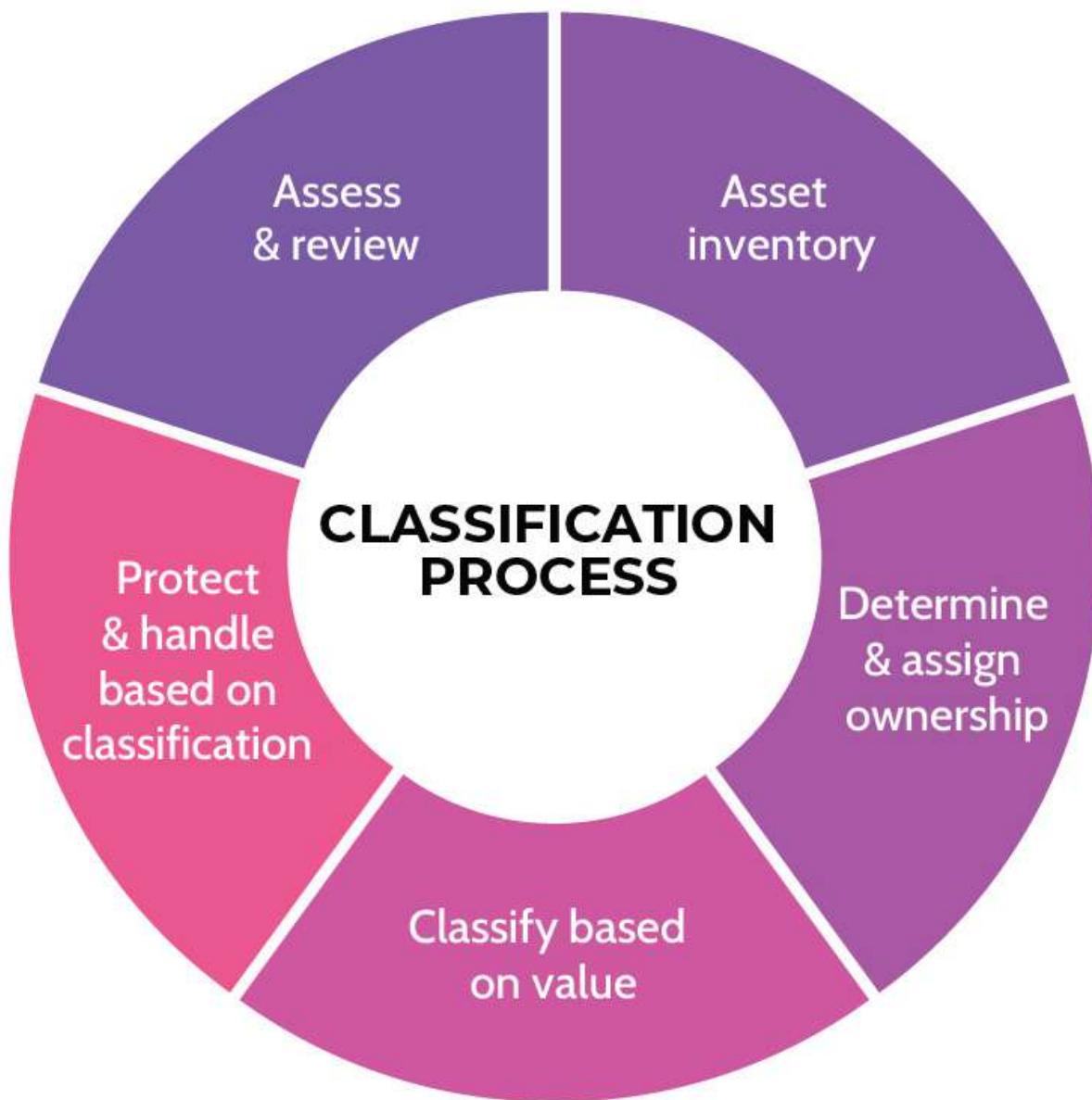


Figure 2-1: Asset Classification Process

An organization must know what assets it holds to protect them properly. Every asset must have an asset owner identified, as they are ultimately accountable for the protection of the value of each asset. Asset owners know the value of a given asset better than anybody else and can classify the asset based on this information. Once classified, assets can be properly protected and handled, based upon the assigned classification level, which indicates its value and how to protect it based on that value. Finally, as the value of assets changes—due to age, or to legal, regulatory, or compliance needs, or to any of a number of other reasons—asset classification should be assessed and reviewed periodically. Additionally, since organizations constantly add and remove assets, owners come and go, laws change, and so on, ongoing assessment and review of all steps in the classification process are required.

### 2.1.3 Classification versus Categorization

#### CORE CONCEPTS

- **Classification refers to a system of classes, ordered according to value**
- **Categorization refers to the act of sorting assets into defined classes**
- **Ideally, all assets should be categorized into a classification system to allow them to be protected based on value**

Now that we've described the concept of classification and how it allows the organization to protect assets based on

value, let's look at the difference between classification and categorization. *Classification by itself is simply a system of classes set up by an organization to differentiate asset values and therefore protection levels.* The act of assigning a classification level to an asset is called *categorization*. For example, if an owner says an asset should be assigned the classification “top secret,” this is the categorization of the asset. The major difference between classification and categorization is shown in [Table 2-1](#).

Classification	Categorization
System of <b>classes</b> ordered according to <b>value</b>	The <b>act of sorting</b> into defined classifications

**Table 2-1: Difference Between Classification and Categorization**

## Classification Examples

- Top secret, secret, confidential, sensitive but unclassified, and unclassified ■ Financially sensitive ■ Company restricted ■ Proprietary ■ Trade secret ■ Personally Identifiable Information (PII) Different organizations might use the same classification terminology and labels, but the corresponding value of each classification level in each organization could be completely different. Therefore, it's

imperative that the security function educates the owners as well as everybody else within the organization about the value of each classification level, so treatment of assets and understanding throughout the organization is consistent.

## **2.1.4 Labeling and Marking**

### **CORE CONCEPTS**

- **Labeling refers to the classification of the asset and is system-readable**
- **Marking refers to the handling instructions of the asset and is human-readable**
- **Should be consistently applied to all assets within an organization**
- **Labeling should be cost-effective**

Once assets are properly classified and categorized, they should be labeled and marked. Labeling and marking ensure that security operations stay consistent and that users handle and dispose of assets properly as they move through the asset life cycle. In NIST Special Publication 800-53A, labeling and marking are defined as follows:

*Organizations can define the types of attributes needed for selected information systems to support missions/business functions. The term security labeling refers to the association of security attributes with subjects and objects*

*represented by internal data structures within organizational information systems, to enable information system-based enforcement of information security policies. Security labels include, for example, access authorizations, data life cycle protection (i.e., encryption and data expiration), nationality, affiliation as the contractor, and classification of information in accordance with legal and compliance requirements. The term security marking refers to the association of security attributes with objects in a human-readable form, to enable organizational process-based enforcement of information security policies.*

### Main differences between labeling and marking

Though the terms sound similar, they in fact are very different. ***Labeling results in output that is system-readable*** and is dependent upon security attributes of subjects and objects as determined by security needs specific to the organization. Thus, labeling approaches will vary from one organization to the other. ***Marking extends the intent of labeling in a way that can clearly be understood and executed by humans***. Marking refers to specific asset data handling instructions, based on the asset label. Security marking is often used to direct handling of an asset according to external laws and organizational policies. For example, something labeled “top secret” might be marked with instructions not to remove it from the premises. A comparison between the two terms can be found in [Table 2-2](#).

<b>Labeling</b>	<b>Marking</b>
■ <b>System-readable</b>	■ <b>Human-readable</b>
■ Association of security attributes with <b>subjects and objects</b> represented by <b>internal data structures</b>	■ Association of security attributes with objects in a <b>human-readable form</b>
■ Enables <b>system-based</b> enforcement	■ Enables <b>process-based</b> enforcement

**Table 2-2: Comparison Between Labeling and Marking**

As noted, a characteristic of labeling is system-readability. For this reason, labeling often employs one of the following: ■ Metadata ■ Barcodes ■ QR codes ■ RFID tags ■ GPS tags

Each approach offers pros and cons, and the use of one or another should be predicated upon things like organizational needs, the value of the assets, and the associated approach the organization takes with respect to protecting them. For example, using global positioning system (GPS) tags may be challenging to implement. They are typically only cost-justified in situations where very valuable assets are being moved around and need to be tracked remotely. Radio frequency identification (RFID) tags are lower cost than GPS tags but still much more expensive than QR or bar codes. A cost-effective use case of RFID tags is in warehouses where inventory levels need to be tracked without having to individually handle and scan each item. An RFID reader can just be moved up and down the aisles. Bar codes have minimal cost and can be printed on packaging and are frequently used for scanning groceries in supermarkets. QR codes can hold more information than barcodes, can also be easily printed, and they can be scanned using a smartphone application.

## 2.2 Establish information and asset handling requirements

### 2.2.1 Media Handling

#### CORE CONCEPTS

- Handling requirements are based on the classification of the asset, not the type of media
- Owners determine who may access media, especially sensitive media

Information and handling requirements are another important element to consider when classifying assets. The more valuable the asset, the more controls are needed to restrict who can handle that asset, what they can do with it, and how they should do it. For example, many organizations use offsite storage for some assets but don't want highly classified assets to leave the premises. If those highly classified assets don't have proper handling requirements, they could be mishandled based on value, or they could be sent to offsite storage by mistake; there have been cases of valuable records that have gone missing under these exact circumstances. Asset handling requirements are clear procedures that mitigate risks like these by delineating the proper handling of assets. Handling assets properly, based on value, is an important requirement of any protection scheme.

As part of an asset classification policy, clear procedures for the proper handling of media should be delineated. Whether assets exist on hard drives, tapes, paper, or any other media, the requirements should clearly define and communicate procedures for handling the assets and media, based upon the classification system and storage requirements for each. Based upon policy, asset owners always remain accountable for the protection of their asset, and it is imperative that owners convey the responsibility of using assets to everyone. As such:

- Only designated individuals should have access to sensitive media
- Owners should define who is authorized to access that media

Additionally, handling requirements should ensure that the proper tools and technologies are available—for example, a shredder that would be used for safe document disposal—so users can follow appropriate procedures for the assets they use.

## Media Storage

Storage requirements for media are based on the classification of the data. For example, if you're storing top-secret data, then it would be mandated for that to be stored in an encrypted format with a very robust encryption algorithm like AES-256. Additionally, the media itself (e.g., tape, hard drive, etc.) should be stored in a physically secured location safe from unauthorized access, high humidity, and so on.

## Media Retention and Destruction

Retention and destruction are based on organizational data classification and data archiving policies. These can be heavily influenced by regulatory and contractual requirements. For example, PCI DSS requires audit log retention is set for a minimum of one year while it also requires audit logs ranging back ninety days to be available for immediate analysis. When it comes to disposal, PCI DSS requires all credit and payment card information to be destroyed as soon as it is no longer required for business or legal information.

## 2.3 Provision information and assets securely

### 2.3.1 Data Classification Roles and Responsibilities

#### CORE CONCEPTS

- Owners are accountable
- Assignment of ownership drives the data classification process
- Data classification roles and responsibilities

Identifying owners is an essential part of the classification process because they're the ones who are accountable for the assets being protected. Owners are the ones that create or procure the assets and work with them on a regular basis. If owners aren't assigned to assets, then no one is accountable for making sure the controls are in place to protect them. When this happens, security breaches tend to occur. Assets can only be properly classified and protected once owners are identified; every asset needs an assigned owner. Identifying and assigning owners is critical, and this is exactly why the concepts of *owners* and *ownership* exist.

While the CEO and upper management are the owners of an organization, they're not in the best position to protect each asset. They are, however, the most suitable people to promote the need for asset classification and empower the governance committee to set this mandate organization-wide. In turn, owners should understand the importance of following these mandates and the need to classify each asset they're accountable for.

**Owners are ultimately accountable for an asset and protecting its value**

In short, security must work with owners to determine the values of assets and how to protect them, but owners are ultimately accountable for the protection of their assets.

**The owner is:**

- The person who directly interacts with the asset the most. Due to this intimacy, they best understand the asset's value. For example, the HR director might be the owner of an HR

database.

- Even though IT might help manage the underlying systems related to the assets in question, they are only functioning as a custodian.

Owners need to have clearly defined **accountabilities**, including:

- Classifying and categorizing assets
- Managing access to assets
- Ensuring appropriate controls are in place based on asset classification

**Owners can delegate responsibility for an asset, but they always remain accountable for the protection of the asset. In other words, accountability cannot be delegated to anyone else. The owner can delegate the responsibility, but accountability remains with the owner.**

Different types of owners exist:

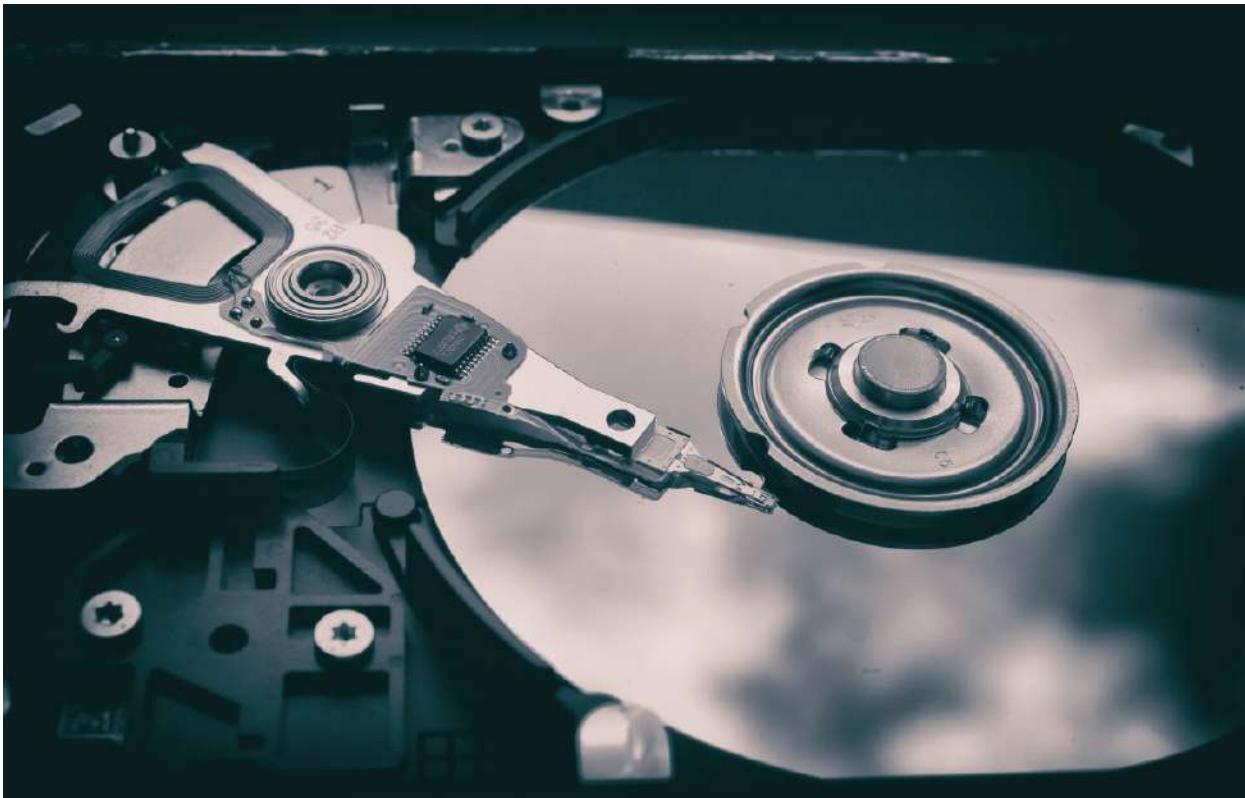
- Data owners
- Process owners
- System owners
- Product owners
- Service owners
- Hardware owners
- Applications owners
- Intellectual property owners
- Etc.

Regardless of the type of asset owned, all owners are accountable for protecting the value of that asset (including its classification and categorization) and approving access to it. The owner retains accountability throughout the asset life cycle, including its retention and end-of-life cycle, which is destruction.

The bottom line is that, regardless of the type of owner, each holds the same accountability, which is:  
***To understand the value of the assets to an organization and classify them properly and ensure appropriate protection as they progress through their life cycle.***

## Understand different roles and responsibilities

Table 2-3 summarizes the various roles and responsibilities relating to data protection within an organization.



<b>Data Owner/Controller</b>	<b>Accountable</b> for protection of data; holds legal rights and defines policies
<b>Data Processor</b>	<b>Responsible for processing</b> data on behalf of the owner/controller (a typical example of a processor is a cloud provider)
<b>Data Custodian</b>	<b>Technical responsibility</b> for data (e.g., data security, availability, capacity, continuity, backup and restore, etc.). Data custodians are responsible for custody of systems/databases—not necessarily belonging to them—for any period of time. Additionally, data custodians are responsible for things like network administration and operations, and for protecting assets in their custody.
<b>Data Steward</b>	<b>Business responsibility</b> for data (e.g., metadata definition, data quality, governance, compliance, etc.)
<b>Data Subject</b>	Individual to whom personal data pertains

Table 2-3: Roles and Responsibilities

### 2.3.2 Data Classification Policy

#### CORE CONCEPTS

- **Data classification policy is concerned with the management of information to ensure that sensitive and valuable information is protected and handled accordingly**
- **Data classification policy considers laws, regulations, privacy requirements, customer requirements, cost of creation, operational impact, liability, and reputation**

To be effective, asset classification needs to be done consistently, regularly, and thoroughly. When organizations don't follow a proper asset classification system, they will struggle to protect their assets and as a result, they can face fines, data breaches, and reputational impact. Having an asset classification policy formalizes the process so that everyone can follow the set of standards, procedures, baselines, and guidelines necessary to protect the assets as also depicted in [Figure 2-2](#).

Referring to Domain 1 and the perfect model of security, the asset classification policy should be governed by senior management. Because everyone in an organization will own or use these assets, this policy must apply to *everyone*. Like all policies, it should communicate *why* it exists, to *whom* it applies, its importance, who is accountable, who is responsible, who supports, and whatever else needs to be conveyed. The goals and objectives of the organization should drive the policy's structure, which should follow all applicable laws, regulations, and industry standards.

An asset classification policy should:

- Start with an asset policy, which drives the asset classification policy ■ Be accompanied by retention, destruction, and archiving policies ■ Clearly define accountability and responsibility ■ Define varying forms of asset media, such as digital, tape, paper, etc.
- Include all the factors that drive value, which should in turn determine how assets are protected
  - Outline asset liability and the consequences of regulatory oversight ■ Describe industry standards and how they impact organizational reputation ■ Involve security from a consulting and expertise perspective; owners should drive the process

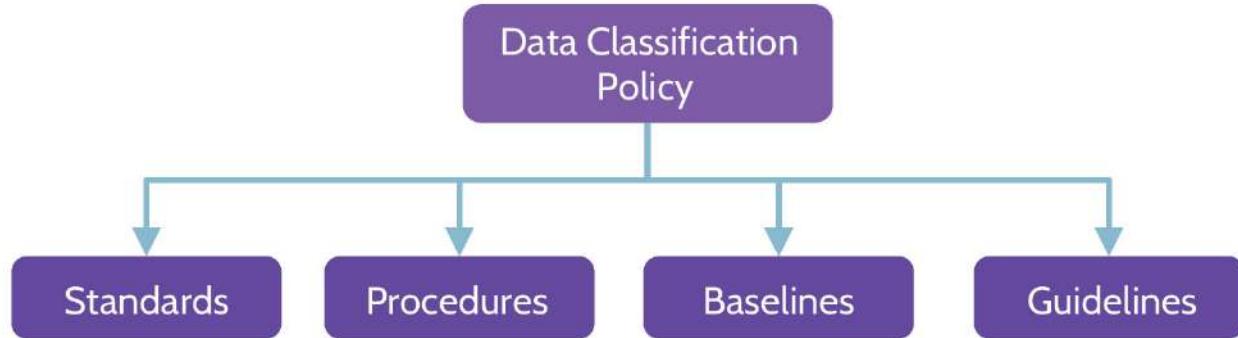


Figure 2-2: **Data Classification Policy**

At the end of the day, it is challenging to quantify the value of all assets, especially intangible assets such as data. This is why qualitative measures like the labels “Top Secret,” “Secret,” and “Public,” among others, are often used. Additionally, organizations will often create data classification boards that can provide an organization-wide perspective and offer asset classification guidance to asset owners.

Though not a complete list, the examples below help organizations determine the value of assets:

1. Laws and regulations

2. Privacy requirements
3. Creation cost
4. Operational impact
5. Liability
6. Reputation



## 2.4 Manage data life cycle

### 2.4.1 Information Life Cycle

#### CORE CONCEPTS

- **Information life cycle phases**
- **Data must be protected at each phase**

## Information Life Cycle

Data must be protected at each stage of its life cycle. The concept of the information life cycle is founded on the principle that proper controls should be in place at the time of creation and throughout the life cycle. Immediately upon *creation, collection, or update*, information should be assigned a classification (by the owner), which then drives all other activities, such as *storage, use, sharing, archiving, and final disposal or destruction, to protect information throughout its life cycle*. These stages are also highlighted in [Figure 2-3](#), while their definitions are contained in [Table 2-4](#). Also note that each data state might require different protective measures and handling practices. For example, data in use might require one set of procedures, while data in storage requires another. As information

moves through each stage of the life cycle, it may increase or decrease in classification or value, and it should be protected at its level.

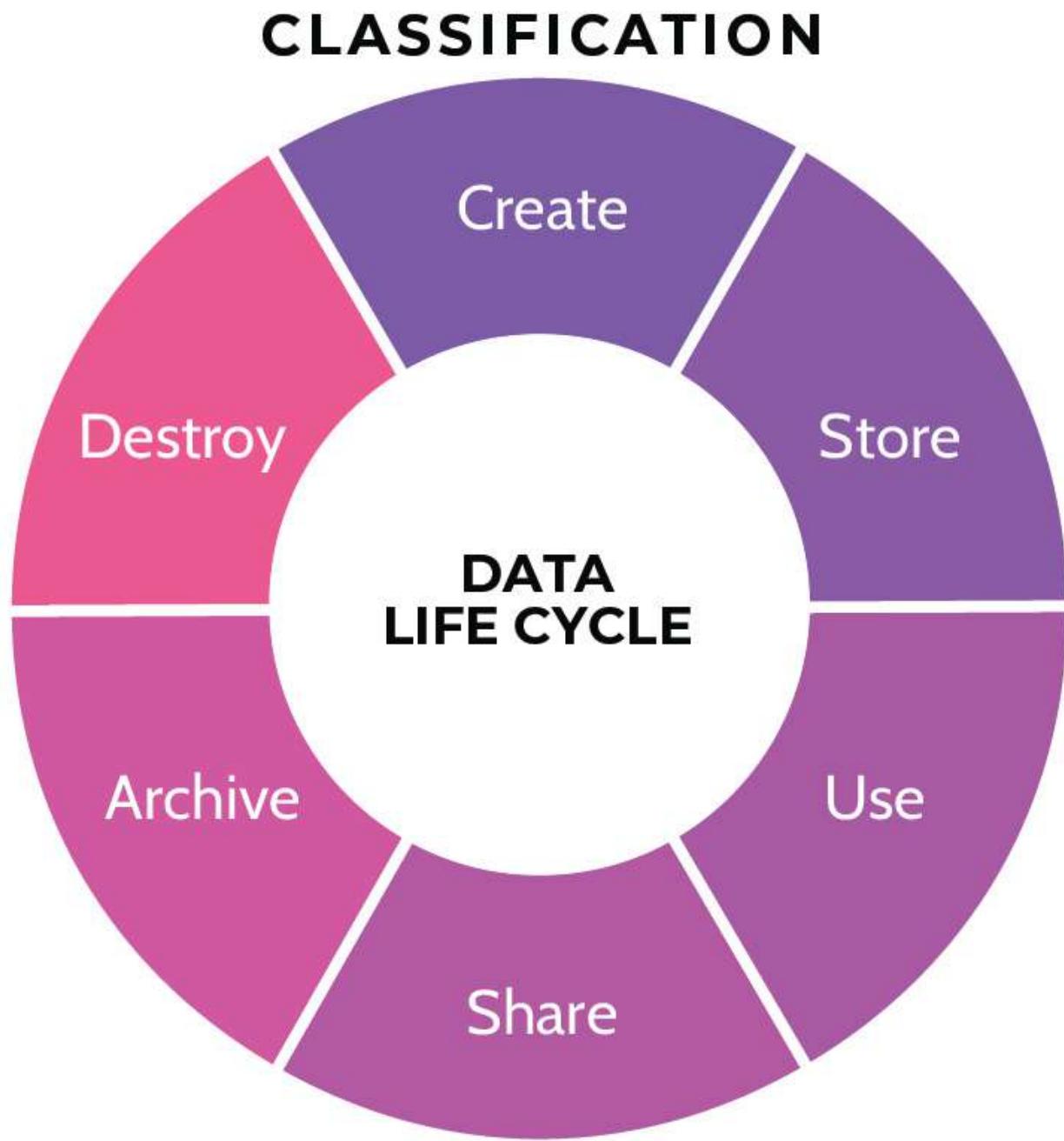


Figure 2-3: Data Life Cycle Stages

## Create

Generation of new digital content, or the alteration/updating/modifying of existing content

<b>Store</b>	Committing digital data to some sort of storage repository, which typically occurs nearly simultaneously with creation
<b>Use</b>	Data viewed, processed, or otherwise used in some sort of activity, not including modification
<b>Share</b>	Information made accessible to others, such as company users, customers, and partners
<b>Archive</b>	Data leaves active use and enters long-term storage
<b>Destroy</b>	Data is permanently destroyed using physical or digital means (e.g., crypto shredding)

Table 2-4: Definitions of Data Life Cycle Stages

## 2.4.2 Data Destruction

### CORE CONCEPTS

- **Data remanence** refers to residual representation of information even after attempts to securely delete or remove the data
- **Categories of sanitization—destruction, purging, clearing**
- **Secure removal of data in the cloud**

Depending upon how data is removed from media, remnants of that data typically exist. This means that a significant amount of that data may be recovered by a determined individual. Various methods exist for ensuring the removal of data and the focus these days is toward what is known as “defensible destruction.” **Defensible destruction** means being able to prove that there’s no possible way for anyone to recover data that has been securely destroyed. Data owners are responsible for ensuring the proper sanitization of the data assets they own.

### Categories of sanitization

Three primary data sanitization categories exist: destruction, purging and clearing as also seen in [Table 2-5](#). Every sanitization method fits into one of these categories as also seen in [Figure 2-4](#). **Note that destruction is the most effective.**

<b>Destroy</b>	<b>Physical destruction of media;</b> this is the <i>most effective</i> means of sanitization.
<b>Purge</b>	Logical/physical techniques used to sanitize; data <b>cannot</b> be reconstructed.
<b>Clear</b>	Logical techniques used to sanitize; data <b>may</b> be reconstructed. This is the least effective means of sanitization.

Table 2-5: Sanitization Categories

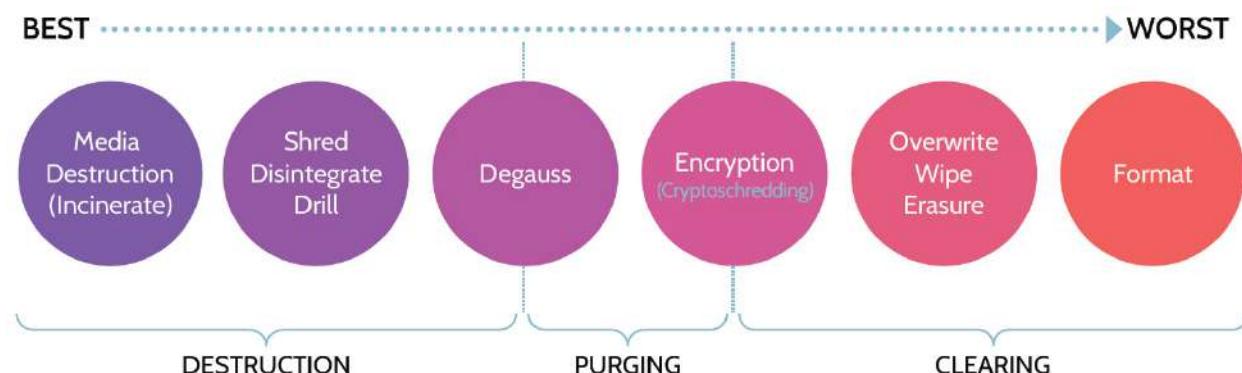


Figure 2-4: Sanitization Methods

NIST SP 800-88 revision 1 provides guidelines for media sanitization as outlined in [Table 2-6](#):

<b>Media Destruction (Incinerate)</b>	The most effective means by which data can be assuredly removed from media is through physical destruction, ideally in the form of incineration that renders a puddle of metal.
<b>Shred Disintegrate Drill</b>	If incineration is not an option, due to cost or availability, the next best sanitization methods include shredding, disintegrating, or drilling holes in the media. Though effective, these techniques are not foolproof because the right tools in the hands of a skilled and determined attacker could allow some data to be recovered. For example, though drilling a hole (or holes) in a hard drive may render the drive unusable, most of the data on the platters is still very much intact and accessible via different means.

<b>Degauss</b>	Degaussing is the process of applying a very strong magnetic field to magnetic media like hard drives or tapes. The strong magnetic field destroys the underlying data. However, degaussing may also render the media itself unusable. This explains why degaussing sits between destruction and purging in <a href="#">Figure 2-4</a> .
<b>Encryption (Crypto Shredding/ Crypto Erase)</b>	Crypto shredding, or cryptographic erasure, is a technique where data is encrypted using a very strong encryption algorithm, like AES-256, and after that's done, the encryption key is destroyed. By encrypting the data and then sanitizing every copy of the encryption key, the data has been effectively made unrecoverable. Crypto shredding fits between purging and clearing. As long as a copy of the key is never found or brute-forced, or a flaw in the underlying algorithm is never discovered, the data cannot be recovered and has been permanently purged. However, if a copy of the key is found or brute-forced, or the algorithm is compromised, then data may be recoverable and has thus only been cleared.
<b>Overwrite Wipe Erasure</b>	Overwriting, wiping, or erasure all refer to writing all zeroes or all ones or some combination of those to all sectors of a storage device, replacing the original data with this overwritten data. This process can be done multiple times, but even so, research has shown that no matter how many overwrite passes are done, some of the original data may still be recoverable. This explains why it is considered a clearing technique.
<b>Format</b>	The least effective method for destroying data is formatting the hard drive. Depending on the formatting method used it may be relatively easy to recover the data. For example, Windows “Quick Format” simply resets the folder and file address table on the drive; and most, if not all, data remains on the disk until it is overwritten by new information. This means it is possible, although potentially difficult, to recover most or all the data from a formatted drive, using software available from a number of vendors.

**Table 2-6: NIST SP 800-88 Sanitization Guidelines**  
**Object Reuse**

Though the name implies otherwise the term ‘object reuse’ refers to a security mechanism that uses *overwriting* to **TRY** to securely remove data from media, as illustrated in [Figure 2-5](#). Object reuse refers to the reassignment of media without the opportunity of data that previously resided on the media to be reconstructed (data remanence).

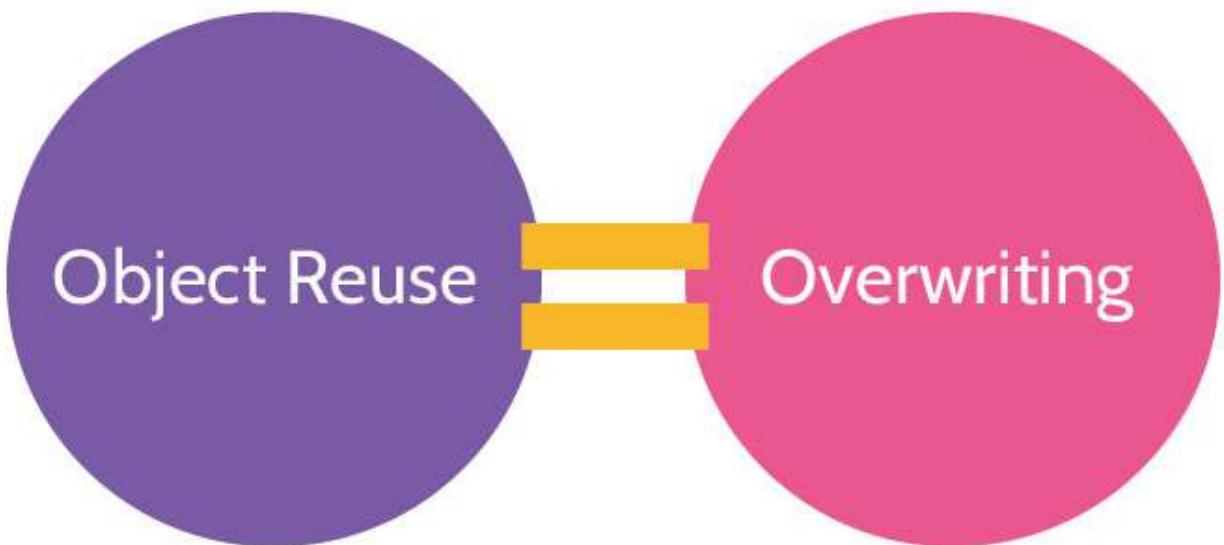


Figure 2-5: **Object Reuse = Overwriting**

The original definition of ‘object reuse’ comes from the Orange Book (TCSEC), and the requirement of certain levels of the Orange Book to ensure the secure reassignment of internal memory and system resources in a computer architecture. The most implemented way of achieving this was to ‘overwrite’ the system resources/memory space. Over the years, several organizations (NSA, DoD, etc.) have offered guidance on the number of overwriting passes required to securely prevent data remanence. This guidance has changed over the years as technology and data recovery techniques have evolved and improved. The important point is that object reuse is defined as: “The reassignment to some subject of a storage medium (e.g., page frame, disk sector, RAM, temporary files, etc.) that contained one or more objects. To be securely reassigned, no residual data can be available to the new subject through standard system mechanisms.” (<https://irp.fas.org/nsa/rainbow/tg018.htm>) The best way to achieve this was to use ‘overwriting’. ***Most experts today consider any number of overwriting passes as being ‘clearing’ and NOT ‘purging’.***

## Solid State Drive Data Destruction

Solid State Drive (SSD) technology presents potential problems where data remanence is concerned. Unlike traditional hard drives that use magnetic technology, SSDs are hard drives that use flash memory technology to represent ones and zeroes. The data on SSDs cannot be overwritten in the same way as a traditional magnetic hard drive. SSD vendors often provide tools or other means by which data can be securely removed because of this fact. From a security perspective, options available from a given SSD vendor should be the first choice when attempting to mitigate data remanence issues related to SSDs. Otherwise, the best way to mitigate data remanence on an SSD drive is to physically destroy it. To summarize:

- Some manufacturers provide sanitization or crypto erasure capabilities.

- The best option is always physical destruction of media.

## Encryption

Crypto shredding (also mentioned in [Table 2-6](#)), or crypto erasure, is the best method to use when attempting to remove data from a third-party environment, like that provided by a cloud provider. Crypto shredding encompasses encrypting the data and then securely destroying all copies of the key. After the key is securely destroyed, the data is effectively unrecoverable.

### Best method for dealing with data remanence in the Cloud

**Physical destruction of media is best but it may be too costly, unavailable or otherwise infeasible.**

## 2.5 Ensure appropriate asset retention

### 2.5.1 Data Archiving

#### CORE CONCEPTS

- **Data archiving is part of the asset life cycle**
- **Data archiving includes requirements for protecting archived data**
- **Data archiving should be driven by an appropriate retention policy**

As part of the asset life cycle, data in particular is often required to be kept for certain periods of time. Sometimes these “certain periods” can extend for many, many years. Laws, regulations, industry standards, and similar guidelines can all play a role in determining retention requirements and how data is archived. For example, health records often have stringent and long-lived retention requirements. Financial records also typically have lengthy retention requirements. A retention policy of 150 years (as may be the case in some organizations) poses a problem to security: What media should the organization store this data on? What format is the media in? A digital file that is readable today isn’t necessarily going to be readable 150 years from now. Archiving, especially for a long period of time, presents a challenge to security: What policy will ensure the data is both protected and recoverable over that span of time?

Depending upon the nature of an organization’s business, the creation of a subset of classification policies might be necessary. These subset policies would focus on archiving and retention, using standards, procedures, baselines, and guidelines. As with other policies and the supporting elements, legal, regulatory, and possibly industry-related requirements will drive the exact structure of each element. Essentially, data archiving policies answer two questions: How long do we need to keep this for, and how do we achieve that, technically speaking?

## Data Archiving

When archiving data, requirements for protecting it must be understood, including:

- Media type
- Security requirements
- Availability requirements
- Retention period
- Associated costs

## Considerations related to data archiving

### Data Archiving Policies

Typically, archiving and retention policies are part of the overall data classification policy. Some important items to keep in mind regarding them are:

- Archiving/retention policy are based on laws, regulations, industry standards, and business needs
- Classify records accordingly
- Educate employees and provide them with the right tools

### Elements of data archiving policies

#### Questions to Consider when Writing a Policy

- Who needs access to the data?
- Do access requirements change over time?
- How long does data need to be kept?
- What are the data disposal requirements?

## 2.6 Determine data security controls and compliance requirements

#### What is the best way to ensure data receives appropriate protection based on classification?

In Topic 2.1, we outlined the classification process, noting that specific security requirements and controls exist for each classification used within the classification system. For any given classification, it's important to know what controls are required to protect the asset appropriately. For example, a top secret asset has a different baseline of required security controls than a secret one, and so on. If baselines (minimum levels of security controls) exist for a given classification level, assets can be efficiently and effectively protected.

These baselines vary depending on value but also depend on the state the data is in. Specifically, data can be in one of three states at any given time: data at rest, data in transit, and data in use. The data security controls that protect data may be completely different depending on which of these states the

data is in. It is therefore important to understand what security controls are required for each state. For example, one of the ways to protect data in transit is using HTTPS for encryption of data between a client and a server, but that won't be relevant for data in use. While this topic doesn't provide in depth detail about how to protect data in each state (which is something discussed in detail in Domains 3 and 4), it introduces some of the appropriate security controls that can be used as mentioned in [Table 2-7](#).

Data at REST	Data in TRANSIT	Data in USE
<p>Inactive data that is stored (resting) on media: hard disks, tapes, databases, spreadsheets, etc.</p> <p><b>Protection:</b></p> <ul style="list-style-type: none"> <li>■ Encryption ■ Access Control</li> <li>■ Backup and Restoration</li> </ul>	<p>Data flowing across a network, such as the internet.</p> <p><b>Protection:</b></p> <ul style="list-style-type: none"> <li>■ Access Control</li> <li>■ Network Encryption</li> <li>■ End-to-end Link</li> <li>■ Onion</li> </ul>	<p>Data being used in computational activities.</p> <p><b>Protection:</b></p> <ul style="list-style-type: none"> <li>■ Homomorphic Encryption</li> <li>■ RBAC</li> <li>■ DRP</li> <li>■ DLP</li> </ul>

Table 2-7: Data Protection in Each State





## 2.6.1 Protecting Data at Rest

### CORE CONCEPTS

- Methods used to protect data at rest—encryption, access control, backup, and restoration

**Data at rest** refers to data that is stored somewhere. Examples of data at rest include files on a hard drive, a database, and similar states. Data at rest can be protected and secured through access control mechanisms, backups (and restores), and encryption.

**The best way to protect confidentiality of data being migrated to the cloud**

Additionally, as more and more organizations are migrating to the cloud, data should first be encrypted locally and then migrated, to best ensure the security and confidentiality of the information being migrated.

## 2.6.2 Protecting Data in Transit

### CORE CONCEPTS

- Methods used to protect data in transit—end-to-end encryption, link encryption, onion network

**Data in transit**, also sometimes referred to as **data in motion**, refers to data that is moving across networks, like an organization’s internal network or the internet. Like data at rest, methods used to protect data in motion include access controls, encryption, and perhaps redundancy. However, with regards specifically to encryption, two primary options exist to protect data in motion: end-to-end encryption and link encryption. A third option—an onion network—is also described.

## End-to-End Encryption

End-to-end encryption means the data portion of a packet is encrypted immediately upon transmission from the source node, and the data remains encrypted through every node—every switch, router, firewall—through which it passes while traveling to the destination node as also depicted in [Figure 2-6](#). Only upon reaching the destination is the packet decrypted. It’s a safe way for data to travel among many different nodes without becoming compromised. Though the data is never in plaintext while traversing nodes, routing information is visible—potentially allowing for inferences to be made about the nature of the data. So, the source and destination IP addresses, for example, are in plaintext and visible to anyone, and thus end-to-end encryption does not offer anonymity. This method is particularly useful in the context of virtual private networks (VPNs). In fact, a VPN is a perfect example of end-to-end encryption.

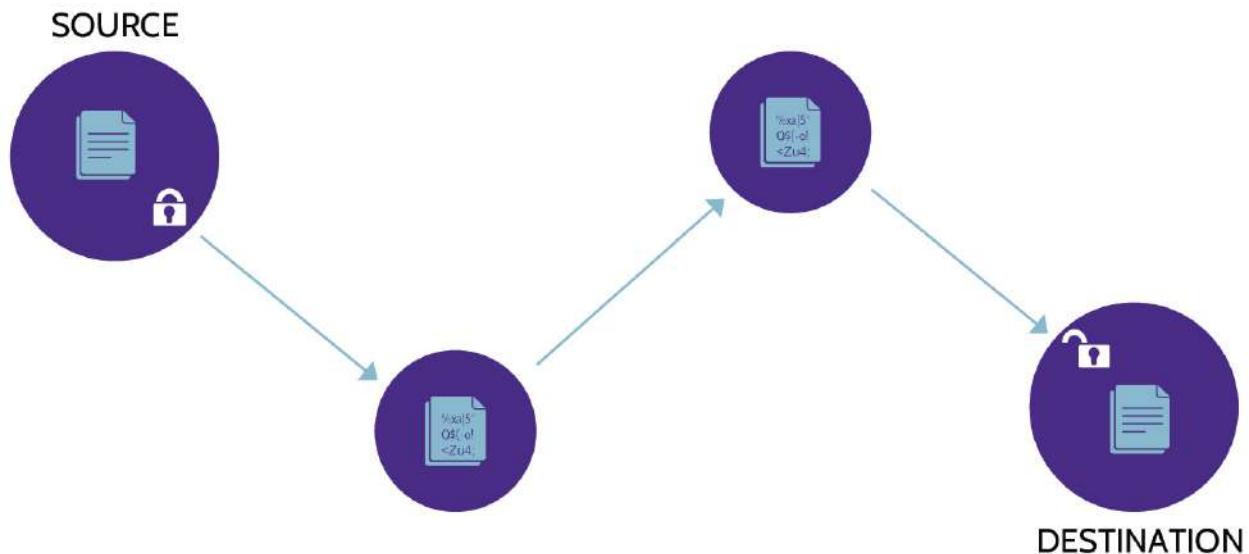
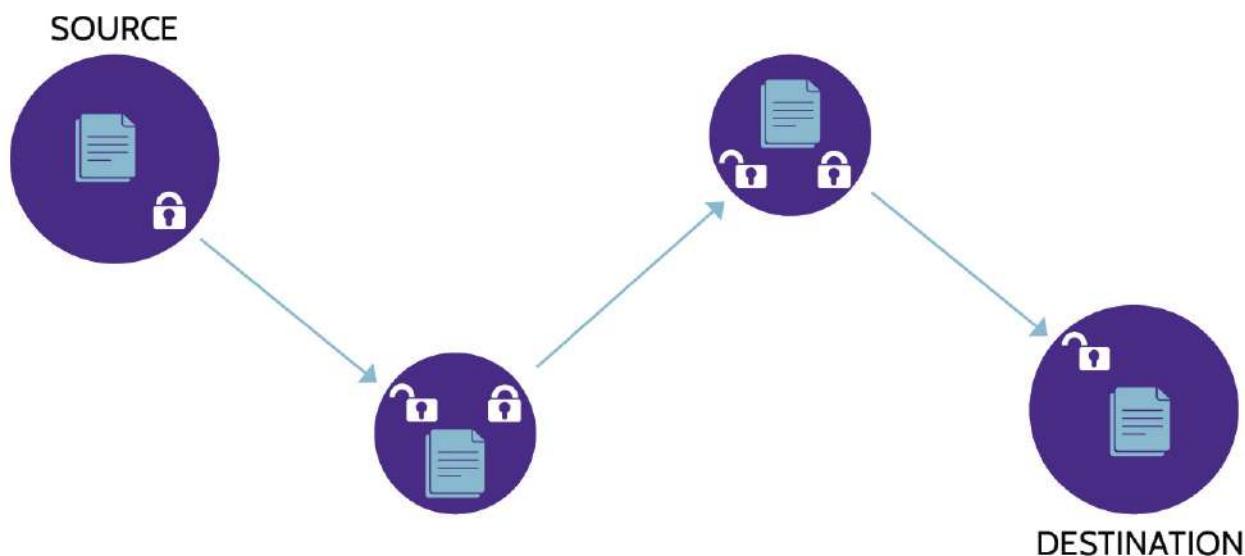


Figure 2-6: End-to-End Encryption

## Link Encryption

Link encryption means the packet header *and* data are encrypted *between* each node. Encrypting the packet header hides the routing information of packets traversing a network. However, unlike with end-to-end encryption, the header and data are decrypted at each node, so header information and plaintext content are also available at each node. As a result, every node becomes a potential attack or disclosure point.

To better understand how link encryption works, imagine several nodes across a network as depicted in [Figure 2-7](#). At the first node, the routing information determines where the data packet needs to go next, and the entire packet is then encrypted with a key that is shared with the next node and transmitted. At the destination, the entire packet (header information and data) is decrypted using the shared key, and the routing information for the next destination is determined. Again, the entire packet is encrypted using the shared key, and the packet is transmitted. This happens at every node, until the destination node is reached. Along the way, plaintext is available at every node, because the packet needs to be decrypted at every node, so the routing information can be determined prior to re-encryption and transmission. This does not best protect data, because every node is a potential attack point or disclosure point, as information is available in plaintext. The advantage is that routing information can be hidden but only from device to device.



[Figure 2-7: Link Encryption](#)

As also mentioned in NIST SP 800-12, link encryption is performed by service providers, such as a data communications provider. It encrypts all the data along a communications path (e.g., a satellite link, telephone circuit, T3 line).

In end-to-end encryption, data is encrypted when being passed through a network, but routing information remains visible.

## Onion Network

Compared to end-to-end and link encryption, an onion network describes a very effective method of protecting data in transit, as it essentially provides complete confidentiality and anonymity using multiple layers of encryption. Like the layers of an onion, multiple layers of encryption are wrapped around the data at the first node. As the encrypted data traverses each node, the outermost layer of encryption is removed, which reveals the address of the next node, as also seen in [Figure 2-8](#). This process takes place at every node, until the final node is reached, and the final layer of encryption is removed, revealing the plaintext data. As each node through which the data passes only knows the

address of the previous node and the next node, the source and destination addresses remain hidden throughout the transmission.

### An onion network provides anonymity and protection of data

By providing confidentiality of data as well as **anonymity**, an onion network makes it very difficult to determine the sender and receiver while data is in transit. With an onion network, the obvious significant advantage is that each node along the way only knows which node the packet came from and the next node. The source node and destination nodes are unknown, except to the nodes adjacent to each of them. Additionally, each node has no access to the encrypted data within the innermost layer. A perfect example of an onion network is The Onion Router—TOR. The big downside of course is performance, as it slows transmission speeds and requires higher-performance technology to be present to allow decryption to take place efficiently.

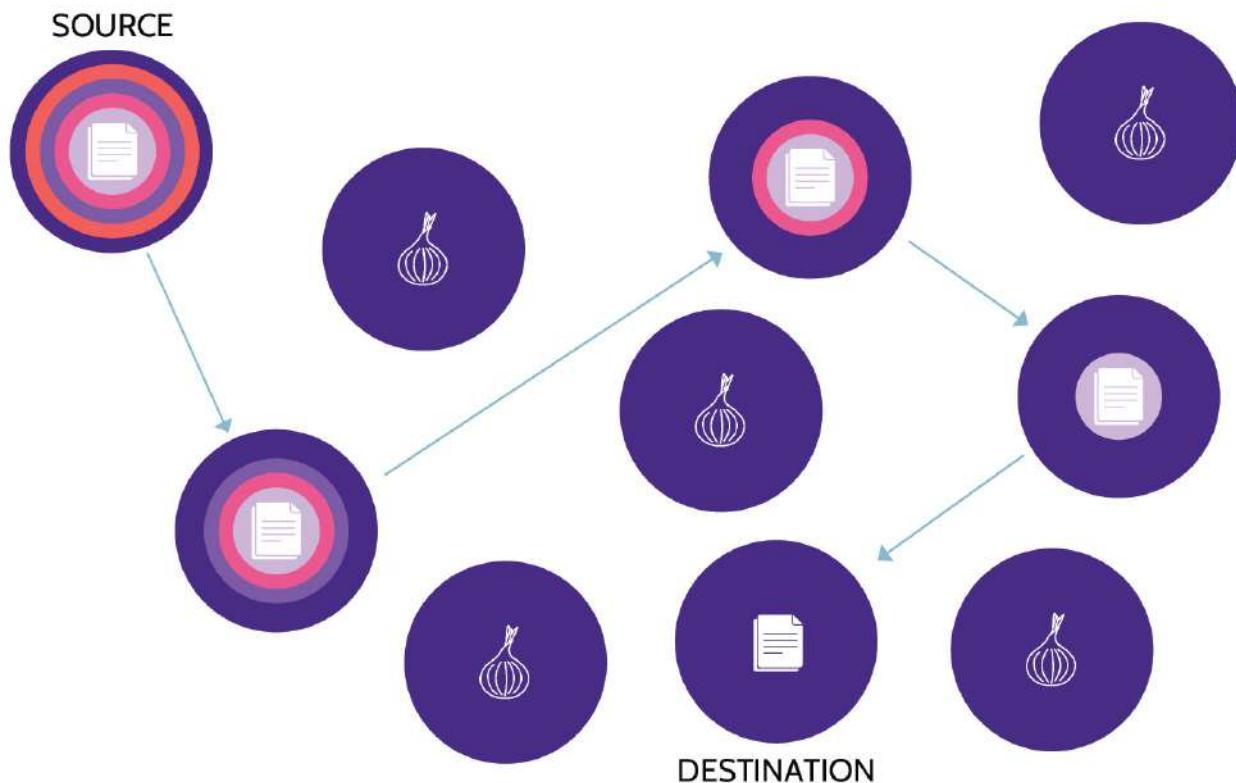


Figure 2-8: Onion Network Encryption

### 2.6.3 Protecting Data in Use

**Data in use** refers to data that is being used in some type of computational activity. One way to protect data in use is through homomorphic encryption, which allows calculations to be performed on data

while the data remains encrypted. Homomorphic encryption is groundbreaking as it gives an ability to process information while the data is encrypted and doesn't require access to a secret key.

Other ways to protect data in use are role-based access control (RBAC) and digital rights protection (DRP) or data loss prevention (DLP). With RBAC, access to specific data can be controlled to ensure only appropriate entities are able to access and process it based on specific roles and work groups. With DRP and DLP tools we can limit the specific actions someone can take when accessing information.

## 2.6.4 Information Obfuscation Methods

### CORE CONCEPTS

- **Obfuscation (or masking) makes something harder to understand**
- **Obfuscation methods**

To understand what is meant by the term **information obfuscation**, sometimes also referred to as **data masking**, it helps to understand what the word *obfuscation* means. Obfuscation is the action of making something obscure, unclear, or unintelligible; in other words, hiding it.

## Why obfuscation is used

Obfuscation serves to make something harder to understand, and information obfuscation serves to impair the ability of malicious actors from understanding data, code, and other information. At the same time, information obfuscation can be employed in such a way that the functionality of the system utilizing the technique still functions properly. Information obfuscation may also be used to hide proprietary information, to meet compliance requirements. For example, those outlined in the EU's GDPR legislation offers customers assurance that their information is private and protected. Developers use obfuscation very often in source code to obscure information like a serial number of a product or an application password. A summary of various obfuscation methods is provided in [Table 2-8](#).

<b>Concealing Data</b>	Concealing data, unlike pruning (noted below), completely removes access to sensitive data. Users do not have access, nor do they have visibility and the attribute field does not appear on computer screens and reports.
<b>Information Pruning/Pruning Data</b>	Information/Data pruning primarily takes place in nonproduction environments and involves the removal of sensitive data from attributes. The attribute will still be visible as a field on computer screens and reports, but it will not be populated with data.

<b>Fabricating Data</b>	<p>Especially when testing the functionality of a system or in cases where particularly sensitive data exists, fabrication of data is often used.</p> <ol style="list-style-type: none"> <li>1. Creating fake data to replace real data facilitates full functional testing.</li> <li>2. Creating fake data to replace sensitive data prevents unauthorized access and viewing of the actual data.</li> </ol>
<b>Trimming Data</b>	<p>Trimming data, unlike pruning, removes part of an attribute's value and is typically used for purposes of identification. Common examples of trimming involve Social Security numbers (SSN) and credit card numbers, where only the last four digits are visible and the remaining digits are masked.</p>
<b>Encrypting Data</b>	<p>Encryption creates ciphertext of a value and can be done at the attribute, table, or database levels. With access to the proper key, encrypted data can be decrypted, and the ciphertext can be changed to plaintext. Like with trimming, credit card numbers are often encrypted when stored (for example, in a database) or when being transmitted.</p>

Table 2-8: **Obfuscation Methods**

## 2.6.5 Digital Rights Management (DRM)

### CORE CONCEPTS

- **DRM protects intellectual property (IP) assets and the rights of asset owners**
- **DRM techniques**
- **Legal basis for protection in the United States through the Digital Millennium Copyright Act (DMCA)**

NIST SP 500-241 defines digital rights management (DRM) as “*a system of information technology (IT) components and services, along with corresponding law, policies and business models, which strive to distribute and control intellectual property and its rights. Product authenticity, user charges, terms-of-use and expiration of rights are typical concerns of DRM.*”

In short, DRM protects assets and the rights of the owners of those assets.

In the context of the NIST definition, intellectual property can include hardware as well as software, and DRM technologies typically focus on limiting the use, modification, and sharing of copyrighted or otherwise proprietary and protected works. Examples of intellectual property typically protected by DRM include:

- Movies and other audio and video works created by publishers
- Video games
-

Digital music ■ eBooks ■ Cable and satellite service providers In all the examples, DRM is employed to prevent intellectual property from being accessed, copied, or improperly used. It helps copyright holders maintain control over the content and it helps maintain income streams through licensing and rentals.

To achieve these ends, DRM techniques include:

- Licensing agreements that restrict access ■ Encryption ■ Embedding of digital tags that tie content to specific license holders, theoretically preventing sharing with others ■ Use of related technologies that restrict copying or viewing of certain content Additionally, in 1998, the Digital Millennium Copyright Act (DMCA) was signed into law in the United States and provides legal recourse for violation of DRM protections and the rights of intellectual property holders.

DRM technologies have primarily been used to protect mass-produced media such as songs and movies. The exact same techniques can be used to protect sensitive documents within an organization from unauthorized access and usage. This is referred to as **Information Rights Management (IRM)**, a subset of DRM.

## 2.6.6 Data Loss Prevention (DLP)

### CORE CONCEPTS

- **Data loss prevention focuses on the identification, monitoring, and protection of data.**
- **DLP data activities take place in three contexts: data in use, data in motion, data at rest.**
- **DLP takes place for multiple reasons.**

Relative to DRM, data loss prevention (DLP) is more all-encompassing. NIST defines DLP as:

*A system's ability to identify, monitor, and protect data in use (e.g., endpoint actions), data in motion (e.g., network actions) and data at rest (e.g., data storage) through deep packet content inspection, contextual security analysis of transaction (attributes of the originator, data object, medium, timing, recipient/destination, etc.), within a centralized management framework.*

As the definition spells out, DLP focuses on the three specific types of data that were mentioned earlier:

- Data in use ■ Data in motion ■ Data at rest In each context, DLP tools attempt to detect and prevent data breaches and potential data exfiltration. For example, in conjunction with acceptable use policies, DLP tools can allow endpoints and portable storage devices to be monitored to protect sensitive data from being used, transferred, or otherwise exposed outside of normal expected usage. Similarly, encryption can be used in a manner that prevents unauthorized access or viewing of things like PII or confidential data. Specific content-aware capabilities of DLP tools can help organizations better monitor data traversing internal and external endpoints in a network. Through content inspection,

logging, and development of organizational policies based upon the same, the movement of data that meet certain criteria can be blocked or redirected. Likewise, these capabilities can be used to scan and protect sensitive data stored on employee endpoints and network locations.

Protection of data is important for multiple reasons because it often encompasses more than organizational information, like trade secrets and other proprietary data. Customer, vendor, and employee data are equally important and should be afforded the same consideration and level of protection.

Protection of data has been a long-standing concern, but events over recent years including increasingly stringent regulations, laws, and industry requirements, and an increasingly litigious society, have made the protection of data more important than ever.



## MINDMAP REVIEW VIDEOS

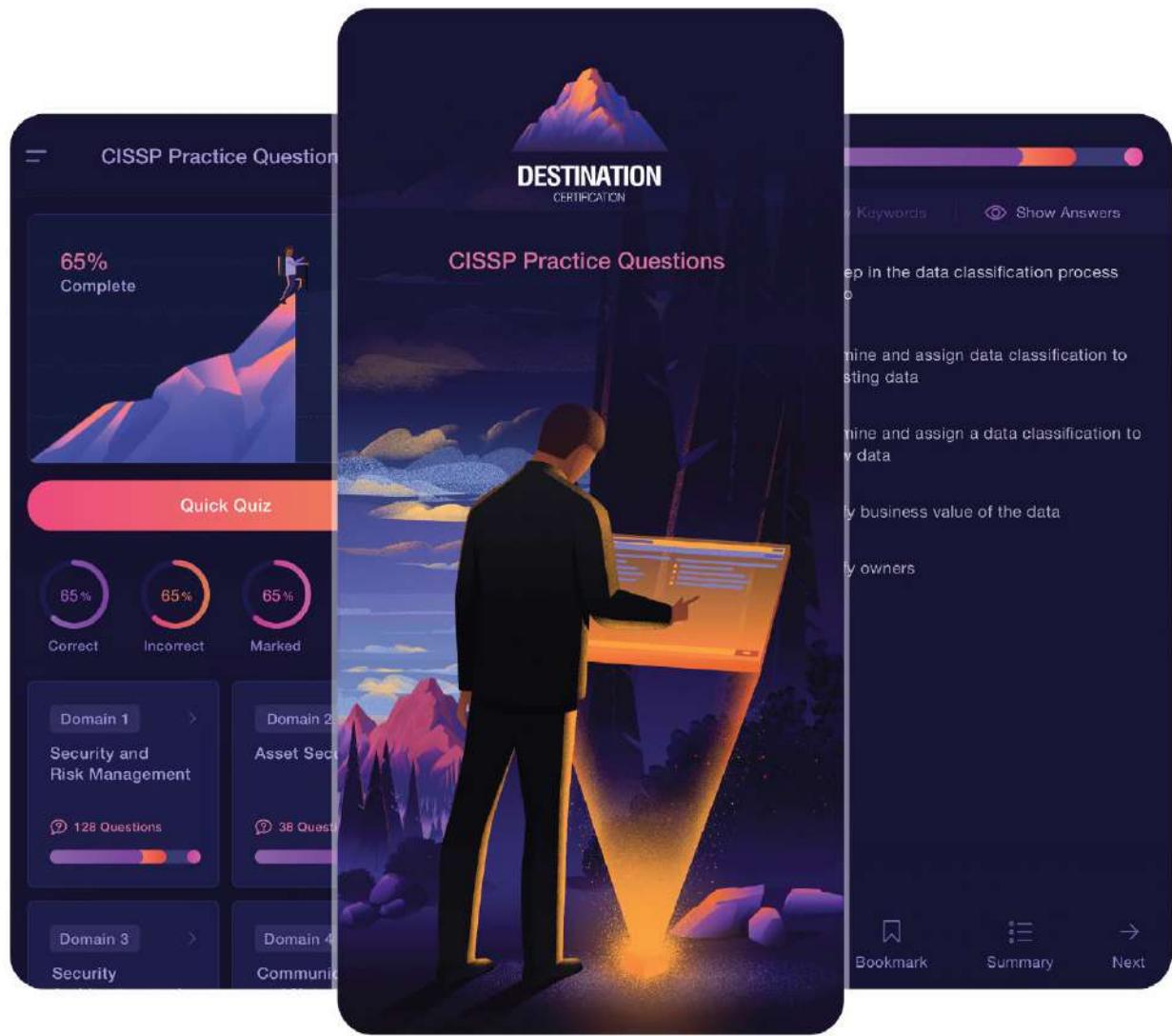


## Asset Classification

[dgo.ca/CISSPmm2-1](http://dgo.ca/CISSPmm2-1)

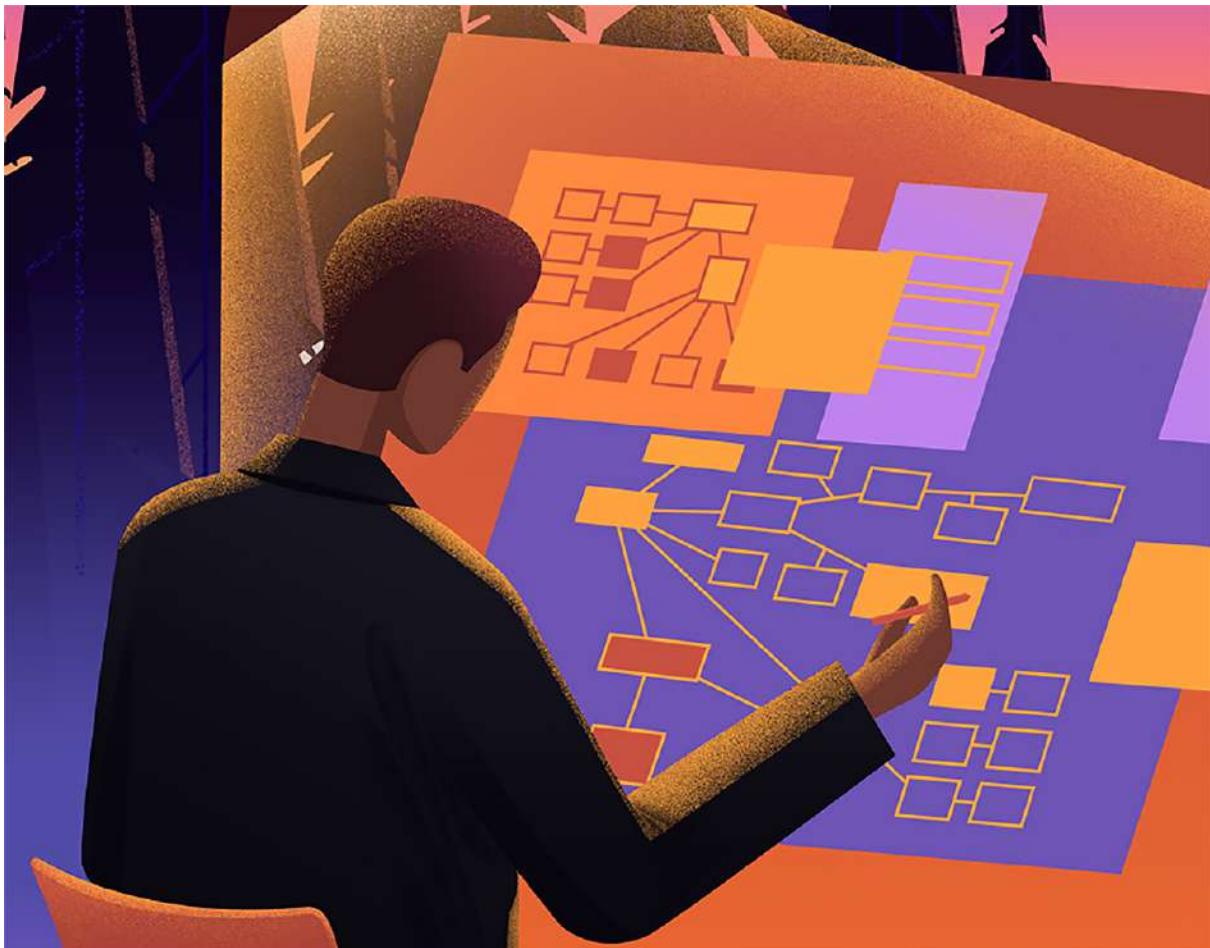


## CISSP PRACTICE QUESTION APP



**Download the Destination CISSP Practice Question app for Domain 2 practice questions**

[dcgo.ca/PracQues](http://dcgo.ca/PracQues)



Domain 3

# Security Architecture and Engineering

## DOMAIN 3

### **SECURITY ARCHITECTURE AND ENGINEERING**

The Security Architecture and Engineering CISSP domain contains the concepts, principles, structures, and standards used to design, implement, monitor, and secure various architectures such as systems, applications, operating systems, equipment, networks, and those controls used to enforce various appropriate levels of security.

This domain focuses on the key principles and concepts reflecting on the key responsibility of any security professional, which is to design, build, and implement security architectures based on corporate requirements that reflect the goals and objectives of the organization, ultimately to allow the organization to achieve its corporate governance initiatives in the most efficient and cost-effective way possible.

A good way to summarize the content and focus of this domain would be, “The Security Architecture and Engineering domain focuses on the different processes, standards, frameworks, and structures to design and implement secure architectures and how, in order to achieve that, the security function needs to be involved at

the start of the engineering life cycle and throughout each of the subsequent phases.”

### **3.1 Research, implement, and manage engineering processes using secure design principles**

#### **3.1.1 Security’s Involvement in Design and Build**

##### **CORE CONCEPTS**

- **Security must be involved in all phases of designing and building a product or system; it must be involved from beginning to end.**

Before diving deep into this domain, it's important to understand the meaning behind the title: “Security Architecture and Engineering.” What is meant by the word **architecture**? This can relate to a building, design, diagram, structure, or something similar. Really, architecture can mean anything. Items like office buildings, houses, firewalls, and computers can all be considered architectures. Why is that? Because they're all made up of different components. So the word *architecture* implies many components that work together to allow that architecture to be used for the purposes for which it was intended. Consider the example of a laptop. What are the components that make it up? At a high level, they include hardware (e.g., motherboard and processor), software (e.g., the operating system and applications), and firmware (allowing the hardware and

software to communicate and operate properly). The user is part of the architecture as well, whenever they're using the laptop.

Now, let's add the word *security*, to make *security architecture*. To secure the architecture, each individual component must also be secured and protected. Using the above example, to protect the prementioned laptop components, the hardware, software, and firmware must all be adequately protected. Security policies, knowledge, and experience must be applied to protect this architecture to the level of value relating to the individual components and to the overall architecture. This is what is meant by the term **security architecture**.

How about if we now add the word **enterprise** to the mix and thus make it **enterprise security architecture**. What does this mean? The word *enterprise* refers to the entire organization, and this now refers to a means by which the entire organization can be protected by breaking the enterprise into different components and protecting each component. What makes up any company or enterprise? Typical components include people, technology, processes, functions, information, hardware, and networks.

So, to protect the entire enterprise, each of these components must be protected based on its value.

However, this raises the question: To what level should each component be protected? Some may consist of additional components that also require protection. As always, the level and degree of protection should be based upon value and cost-effectiveness.

Now for the final question: What does the word **engineering** mean?

This commonly points to designing and building a solution by walking through a series of steps and phases to put the components together so they can work in harmony as an architecture. The building of any architecture requires a corresponding engineering life cycle. That typically starts with a concept, an idea that someone comes up with which then needs to be designed. Once designed, building and implementation can start. Testing also must take place, which happens during the design and build phases and must also be done prior to implementation. Once implemented, it is maintained and, at some point in the future, it is disposed of. These are the steps of the engineering life cycle, and they should be followed anytime something needs to be built.

One of the most important points to highlight is security's role in this process. Security should be involved from the very beginning and should be a consideration throughout

the life cycle. Unfortunately, most organizations do not follow this protocol, as security is often not considered at all, or it is an afterthought and pursued at the end. Security considered from the beginning leads to the best outcome, and it is the most cost-effective approach. It leads to what is known as security by design, which means that security should be embedded from the beginning and not just as an afterthought.

### 3.1.2 Determining Appropriate Security Controls

#### CORE CONCEPTS

- Regardless of the framework, model, or methodology used, the risk management process should be used to identify the most valuable assets and risks to those assets, and to determine appropriate and cost-effective security controls to implement this.

The design and creation of a secure system is not constrained by one framework, model, or methodology. In fact, many security concepts and principles can be applied to its design. A number of secure design principles are noted below, while some of them have been discussed elsewhere in the book (as also referenced).

#### Examples of Secure Design Principles:

- Threat modeling (discussed in [section 1.10](#)) ■ Least privilege (discussed in [section 1.8.2](#)) ■ Defense in

depth (discussed in [section 3.4.9](#)) ■ Secure defaults ■ Fail securely ■ Separation of duties (discussed in [section 1.8.2](#)) ■ Keep it simple and small ■ Zero trust or trust but verify ■ Privacy by Design ■ Shared responsibility ■ Secure access service edge **Secure Defaults**

Any default settings a system has should be secured to the extent possible, so no compromise is facilitated. For example, if the operating system allows an administrator account to exist with no password, that will make it easy to launch an attack.

## Fail Securely

If a system or its components fail, they should do so in a manner that doesn't expose the system to a potential attack. For example, if a safe has an electronic lock to protect its contents from theft, that lock should remain engaged in the event the electricity fails at the building.

## Keep It Simple and Small

Remove as much complexity from a situation as possible and focus on what matters most. In the context of designing and building a secure system, the same thinking applies and results in a number of benefits, including: ■ Smaller attack surface ■ Less errors and vulnerabilities ■ Less complex testing ■ Easier and more efficient troubleshooting

and problem resolution. If simplicity is ignored, it can lead to the development of complex mechanisms that may not be correctly understood, which makes configuration, implementation, maintenance, and use much more difficult.

In other words, the likelihood of a greater number of vulnerabilities increases as the complexity of the system design and code does.

### **Zero Trust and Trust but Verify**

With continued advancements in technology and especially with the growth of cloud computing, organizations are more intertwined and interconnected than ever before. As a result, organizations are also more vulnerable than ever before, and this has led to adoption of a security concept known as zero trust. Zero trust essentially means trust nothing, and it is based upon the premise that organizations should not automatically trust anything internal or external to enter their perimeter. Instead, prior to granting access to systems and individuals, those first need to be authenticated and authorized. Network segmentation can greatly be used in this approach, because it allows breaking the network into smaller parts and thus forces users and devices to authenticate each time they move from one network to another. With this perspective, organizations can take steps to adopt a zero trust posture.

such as those noted below:

- Micro-segmentation of networks
- Granular enforcement of perimeter ingress/egress points, based upon identity, user location, and other data to determine whether to trust the user, device, or application seeking access to enterprise resources.

Additionally, in a zero trust environment, inherent trust should not exist. Simply because a device is connected to a network does not mean the device should have access to anything on the network. Access to network resources should only come as a result of confidence gained through proper authentication, authorization, and accounting (AAA) of users, devices, and services.

Correspondingly, this confidence can only be gained by building trust into the user's identity (user authentication), associated user devices (device verification), and the services they access (service authorization and associated accounting, which is achieved by logging).

Ultimately, for this model to be effective, every connection to a service should be authenticated, and every device and connection should be authorized against a corresponding policy, regardless of the request's origin (internal or external).

To summarize, zero trust's "trust nothing" approach takes security very seriously and recommends the following features (with the zero trust principles being summarized in [Table 3-1](#)): ■ Strong user authentication ■ Authentication of services, which includes the following components: ■ User authentication ■ Device authentication ■ Service authorization ■ Logging and monitoring ■ Authorization using a corresponding policy

Principles for Zero Trust	
<b>1</b>	Know your <b>architecture</b> (users, devices, and services)
<b>2</b>	Know your user, service, and device <b>identities</b>
<b>3</b>	Know the <b>health</b> of your users, devices, and services
<b>4</b>	Use <b>policies to authorize</b> requests
<b>5</b>	<b>Authenticate everywhere</b>
<b>6</b>	<b>Focus your monitoring</b> on devices and services
<b>7</b>	<b>Don't trust any network</b> , including your own
<b>8</b>	Choose services <b>designed for zero trust</b>

**Table 3-1: Zero Trust Principles**

In essence, the term *trust but verify* is an oxymoron, because complete trust in something implies that no verification is needed. However, complete distrust is also not

viable. Locking down architectures, with an eye toward prevention and avoidance, is outdated. Rather, like the concept of zero trust, trust but verify really means being able to authenticate users and perform authorization based on their permissions to perform activities on the network so they can access the various resources. It also means that real-time monitoring is a requirement. In short, focus on employing complete controls that include better detection and response mechanisms. With the cloud and reliance on third-party services becoming more and more ubiquitous, adopting this “complete control” mindset and approach is especially relevant and valuable for an organization.

Additionally, due to the growth in reliance on third-party services, trust should be verified through assurance mechanisms like ■ Audits ■ Ongoing monitoring ■ SOC reporting ■ Contracts/agreements, like SLAs and SLRs

## **Privacy by Design**

Privacy by Design is premised on the belief that privacy should be incorporated into networked systems and technologies by default and designed into the architecture. As privacy is achieved through the proper implementation of security controls, the concept of Privacy by Design also means that security becomes an integral part of the design elements of an architecture. Security professionals often emphasize that “privacy by design” is simply another way to

reflect on “security by design,” since privacy requirements are enabled by security controls.

Privacy must be a priority and must become an integral part of organizational and project goals and objectives, design activities, and planning efforts. Privacy should be embedded into every standard, protocol, and process that touches people.

Privacy by Design incorporates seven foundational principles as also depicted in [Figure 3-1](#):

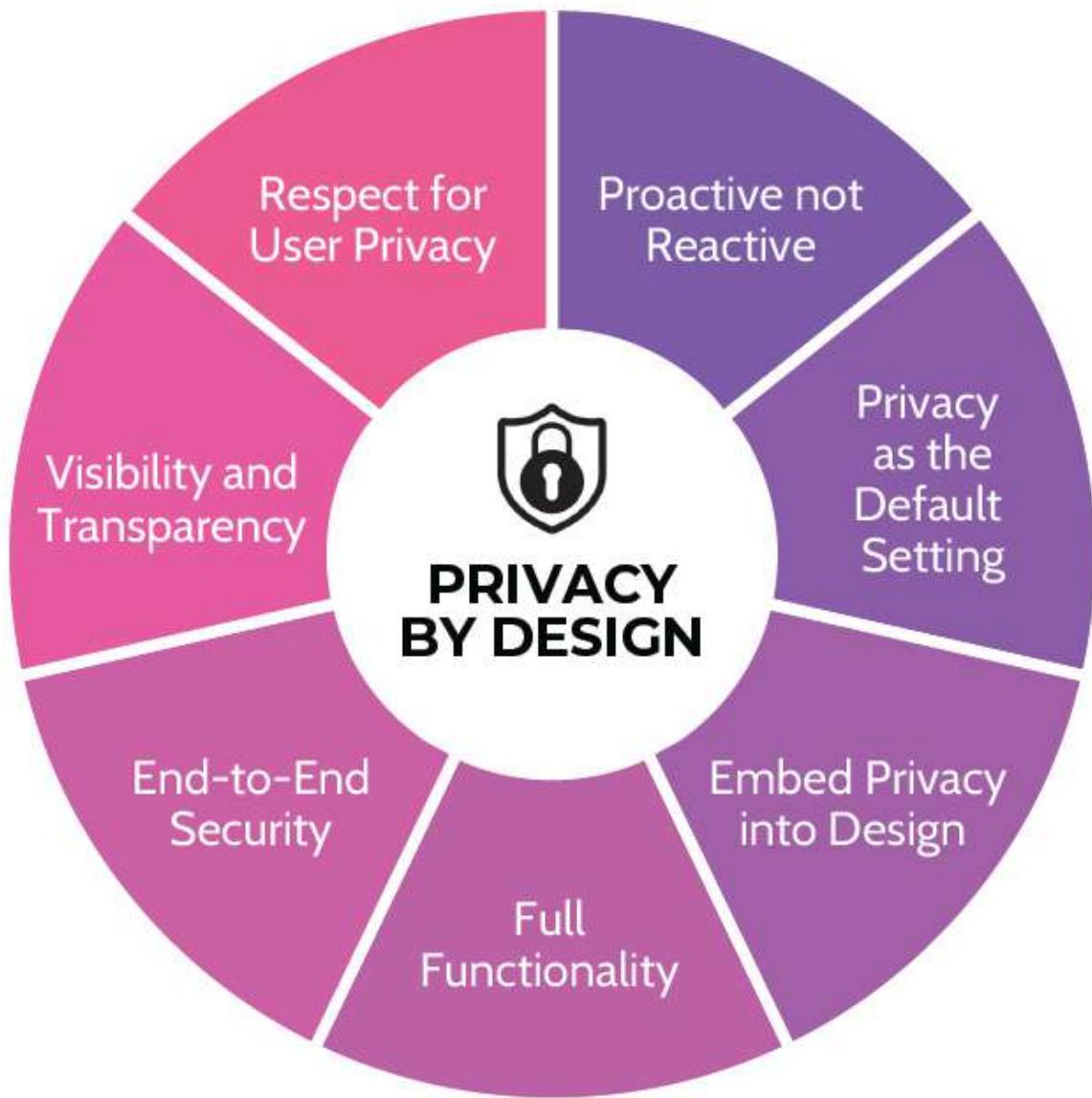


Figure 3-1: Privacy by Design

■ **Privacy as proactive and preventive, not reactive and remedial.**

Privacy by Design (PbD) should anticipate and prevent privacy shortcomings before they occur. PbD does not wait for risks to privacy to arise, nor does it attempt to resolve privacy breaches once they've taken place.

- **Privacy as the *default* setting.** Like many firewalls that include *implicit deny* as the default treatment of all traffic, PbD seeks to ensure the automatic—default—protection of personal data in IT systems and business practices. In other words, if an individual takes no action, their privacy is protected as the rule and not as the exception.
- **Privacy embedded into design.** PbD is considered from the inception of a system, application, or process and is included in the architecture, design, and development of the system, application, and process. This approach leads to a result that includes privacy as an essential part of the core functionality being delivered.
- **Full functionality within a given solution.** PbD seeks a “win-win” situation for all parties involved and attempts to accommodate all interests, goals, and objectives rather than demand trade-offs that reduce overall effectiveness.
- **End-to-End Security.** PbD, having been considered from the inception of a system, application, or process, should securely extend through the life cycle of the data involved. Strong security measures—from beginning to end, from cradle to grave—are essential to effective privacy.
- **Visibility and Transparency.** PbD calls for visibility and transparency of all components and processes related to the technology or business practice being used. Furthermore, PbD seeks to assure all relevant parties that whatever components and

processes are involved are being operated and used in a manner that is consistent with stated objectives and promises, subject to independent verification.

- **Respect for User Privacy.** PbD ultimately requires the individual to be treated with the utmost respect and care. Regardless of who is designing or working with a system, measures such as strong privacy by default, appropriate notice, and user-intuitive options that allow management of privacy should always be incorporated into the solution. User-centric considerations should always prevail.

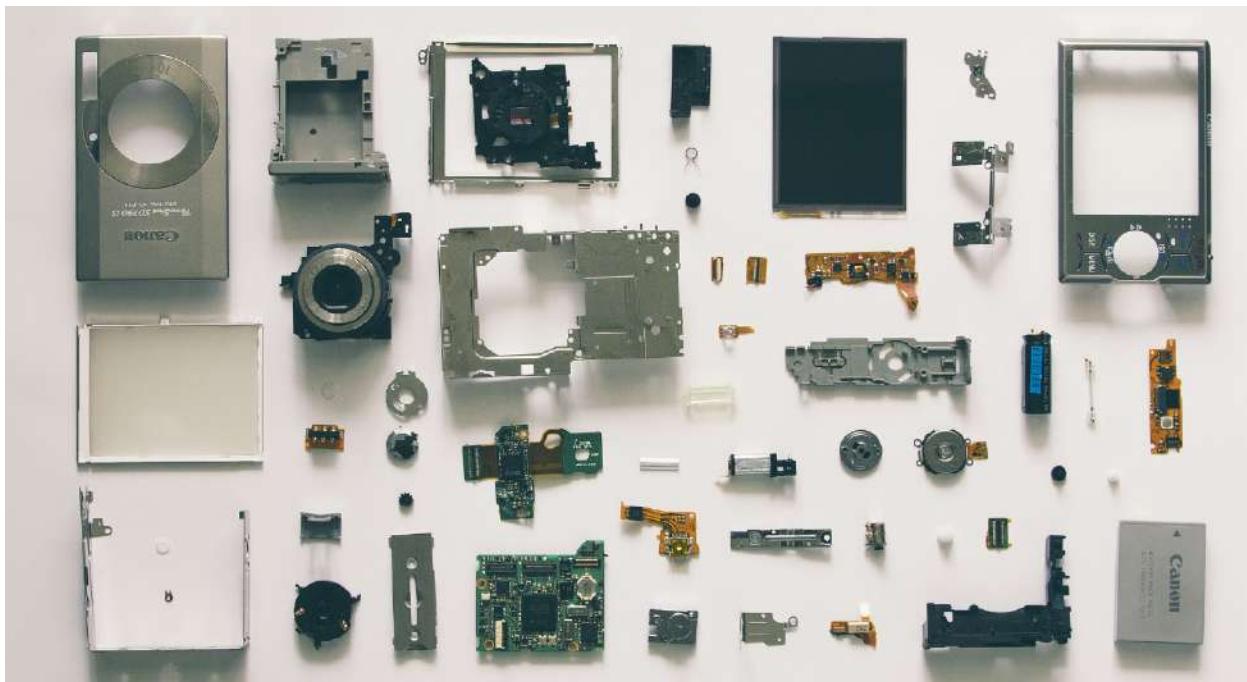
## Shared Responsibility

As noted already, the ubiquitous nature of the cloud and reliance on third-party services for support of day-to-day operations of organizations around the globe have shifted responsibilities from internal sources to a mix of internal and external sources. With most traditional on-premise IT infrastructure, accountability and responsibility for all facets of the same rests with the organization; with the cloud, depending upon the cloud deployment model being utilized, the same is not true—especially where responsibility is concerned.

Because of increased reliance on third-party services, a corresponding increase in clarity on shared security expectations should exist. The cloud customer and cloud service provider must clearly communicate expectations both ways as well as clearly define related responsibilities. Furthermore, responsibility and accountability in the eyes of the cloud customer must be clearly defined and understood, especially since accountability may never be

delegated or otherwise transferred, regardless of the customer–provider relationship.

To this end, consumers and providers must act on these responsibilities and define clear contracts and agreements, which can then be implemented through appropriate policies, procedures, and controls.



Another important concept you should be aware of is the cyber kill chain. Understanding the steps can help us to defend our environments. If we can break some of the links of the chain, we can prevent cyber attacks from succeeding. The steps involved in the cyber kill chain are outlined in [Table 3-2](#) and [Figure 3-2](#).

### The Cyber Kill Chain

<b>Reconnaissance</b>	Reconnaissance involves identifying a target and looking for information that will be useful to the attacker, such as details about the software, vulnerabilities, usernames, email addresses, phone numbers, etc.
<b>Weaponization</b>	This step involves building an exploit that aims to take advantage of any vulnerabilities identified in the previous step.
<b>Delivery</b>	Delivery involves the attacker launching their attack. Common delivery methods include sending malicious links or attachments to targets.
<b>Exploitation</b>	In this step, the attacker executes their malicious code on the target's systems.
<b>Installation</b>	The installation step comes immediately after exploitation, and the attacker now has software installed on the target's systems.
<b>Command and control</b>	Command and control allows the attacker to remotely control their malware running within the target's systems. The attacker can move laterally and install backdoors to help them continue accessing the target's systems.
<b>Actions</b>	The attacker carries out their initial objective, which could be things like exfiltrating data, or encrypting the target's files in a ransomware attack.

**Table 3-2: The Cyber Kill Chain**

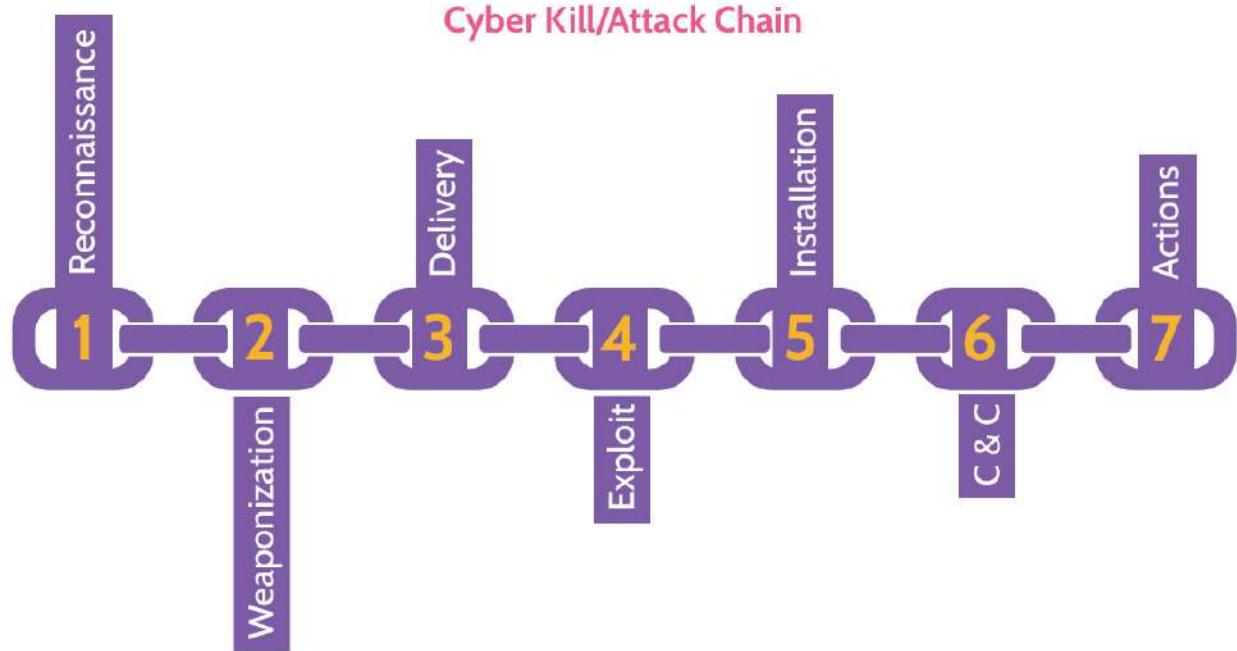
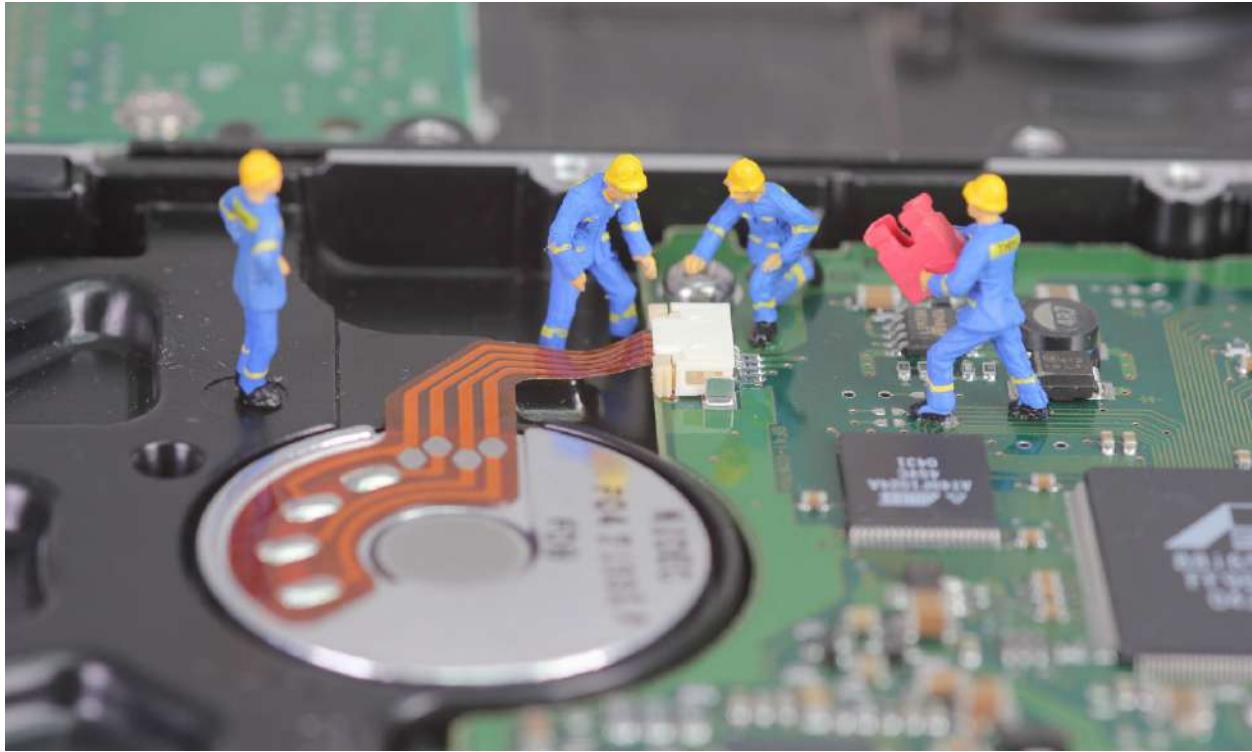


Figure 3-2: **The Cyber Kill Chain**

### Secure Access Service Edge

We discuss secure access service edge when we talk about edge computing in section [3.5.15](#).



## 3.2 Understand the fundamental concepts of security models (e.g., Biba, Star Model, Bell-LaPadula)

### 3.2.1 Security Models

#### CORE CONCEPTS

- A model is a representation of something real.
- A security model is a representation of what security should look like in an architecture.

#### What Is a Model?

A *security model* is a representation of what security should look like in an architecture being built. Security models have existed and have been used for

years. Some of these models include Bell–LaPadula, Biba, Clark–Wilson and Brewer–Nash (also referred to as the Chinese Wall model). These are simple models that provide the basis—the fundamental means—for building confidentiality or integrity into architectures that require these core principles. Like any model, security models represent what security needs to look like. Many people don't value the importance of security models because many of them date back to the early '70s, like Bell–LaPadula; but these still apply today. The conceptual, fundamental ways to address confidentiality that Bell–LaPadula highlights are the same today as they were in the '70s. The way that confidentiality is addressed today is also the same as it was years ago. Technology changes, but the fundamental rules on how confidentiality is addressed remain the same. It's valuable to understand these models and their underlying rules, as they govern the implementation of the model.

## Concept of Security

**Here's the bottom line:** To ensure the protection of any architecture, it must be broken down into individual components, and adequate security for each component needs to be put in place. Remember that a chain is only as strong as its weakest link. However, note that any system should be broken down and individual components secured to the degree that value dictates doing so.

### 3.2.2 Enterprise Security Architecture

#### CORE CONCEPTS

- An architecture is a group of components that work together.

■ Security architecture involves breaking down a system to its components and protecting each component based upon its value.

To implement an enterprise security architecture, frameworks exist to serve as guidelines. Three of the most popular enterprise security architectures are Zachman, Sherwood Applied Business Security Architecture (SABSA), and The Open Group Architecture Framework (TOGAF). Though each differs a bit in structure and terminology, they each basically do the same thing to protect any architecture. Zachman focuses on answering basic questions like how, where, who, when, and why by directing those to the various company teams (e.g., designers, owners, architects, strategists, engineers, operators) and acquiring their feedback. However, this is an older model dating back to the '70s and may not necessarily be the most suitable model today, as it merely focuses on classification and organization of enterprise security. SABSA is a newer framework (adopted in 1995) that focuses on security architecture risk and allows security to be embedded in IT functions. It is an open source framework that provides scalability and ease of implementation, facilitates compliance, and can help response prioritization. TOGAF focuses on efficient resource utilization and cost minimization while having a modular structure increasing its adoption, a content framework providing consistency, and a style that allows architectural flexibility.

## Security Models

Security models are pretty simple. Essentially, they're the rules that need to be implemented to achieve security. Many security models exist, but most of them are one of two types: lattice-based or rule-based. A good way to envision a lattice-based model is to think of a ladder, where a framework and

steps exist that look a bit like layers, going up and down. In other words, a lattice-based model is a layer-based model. It requires layers of security to address the requirements. Two lattice-based models exist: Bell–LaPadula and Biba. Bell–LaPadula addresses one primary component of the CIA triad: confidentiality. Biba addresses another component: integrity. All other models are rule-based, meaning specific rules dictate how security operates. People sometimes argue that lattice-based models also provide rules and ask why they’re not considered rule-based. Lattice-based models do include rules, but those rules are confined to layers within the model; hence, the term *lattice-based* is more applicable. [Table 3-3](#) provides a summary of the various lattice-based and rule-based security models.

Layer / Lattice-based Models	Rule-based Models
■ Bell–LaPadula ■ Biba	■ Information Flow ■ Clark–Wilson ■ Brewer–Nash (Chinese Wall) ■ Graham–Denning ■ Harrison–Ruzzo–Ullman

Table 3-3: Lattice-Based and Rule-Based Security Models

### 3.2.3 Layer-based Models

#### CORE CONCEPTS

- Layer-based security models are also considered lattice-based security models.
- Bell–LaPadula addresses only confidentiality.
- Biba addresses only integrity.

- Lipner implementation is not a model; it is an implementation that combines the best features of Bell-LaPadula and Biba.

As we mentioned earlier, lattice-based security models, like Bell-LaPadula and Biba, can also be thought of as layer-based security models. Based upon intersecting vertical and horizontal support elements, a lattice structure can be envisioned as having different layers. Extrapolating this concept further, each of the two security models can be viewed as having different layers of security.

## Bell-LaPadula

Bell-LaPadula is based on incorporating the necessary rules that need to be implemented to achieve confidentiality. Based upon this query, a lattice-based model was developed as also seen in [Figure 3-3](#).

**Bell-LaPadula addresses only confidentiality**

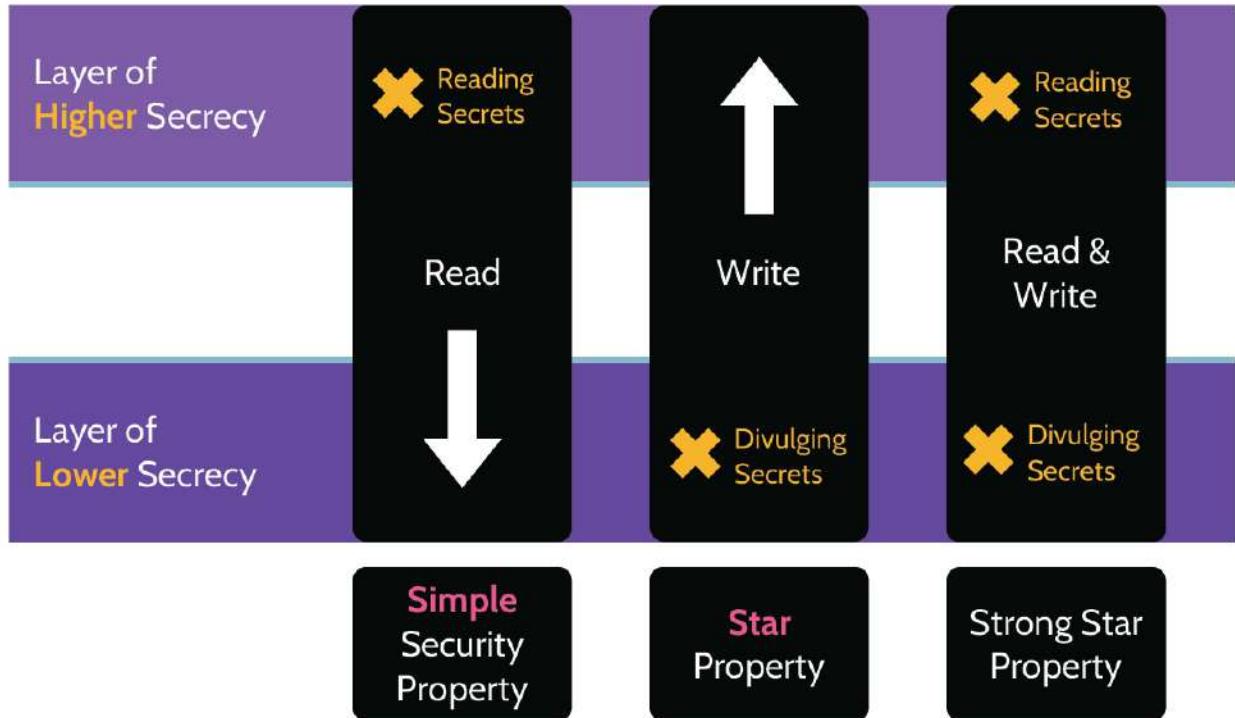


Figure 3-3: Bell–LaPadula Model

This model is based on three basic principles:

1. **Simple security property**, also known as “**no read up**” property, relates to reading and denotes that any subject at a particular security level may not read an object at a higher security level.
2. **The star (\*) property**, also known as “**no write down**” property, relates to writing and denotes that any subject at a particular security level may not write to an object at a lower security level.
3. **The strong star property** relates to both reading and writing. Having an ability to both read and write means a subject should be

able to read and write at their own layer, nothing higher and nothing lower.

## Biba

Biba focuses on ensuring data **integrity** as shown in [Figure 3-4](#). Integrity means accurate, relevant, or meaningful.

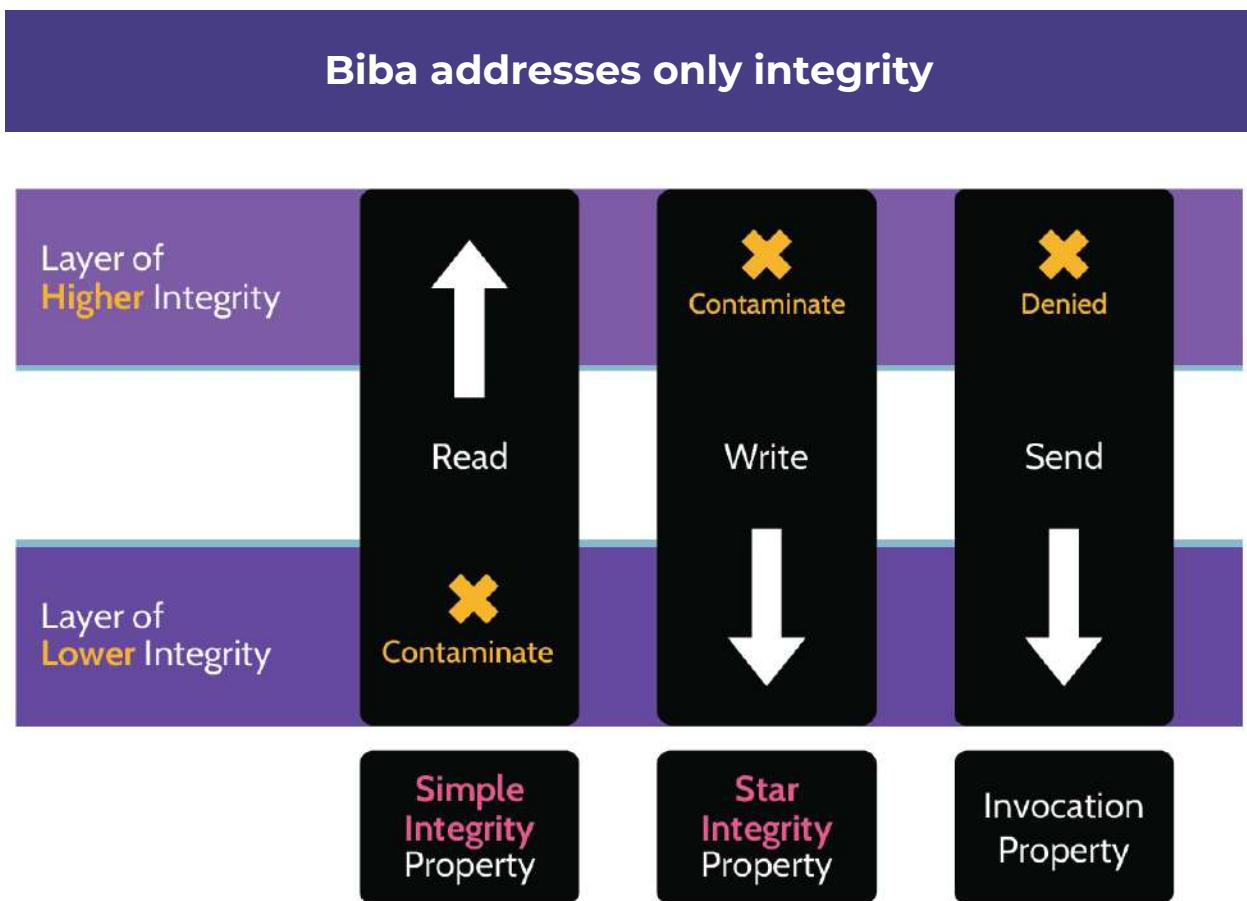


Figure 3-4: Biba Model

This model is also based on three basic principles:

1. **Simple integrity property**, also known as the “**no read down**” property, relates to reading and denotes that a subject at a particular level of integrity may not read an object at a lower integrity level.
2. **Star (\*) integrity property**, also known as “**no write up**,” relates to writing and denotes that a subject at a particular level of integrity may not write to an object at a higher integrity level.
3. **Invocation property** states that a subject can’t send information to someone that is rated at a higher layer of information than the current one the subject holds.

[Figure 3-5](#) provides a summary of the abovementioned models, with an emphasis on their two main principles.

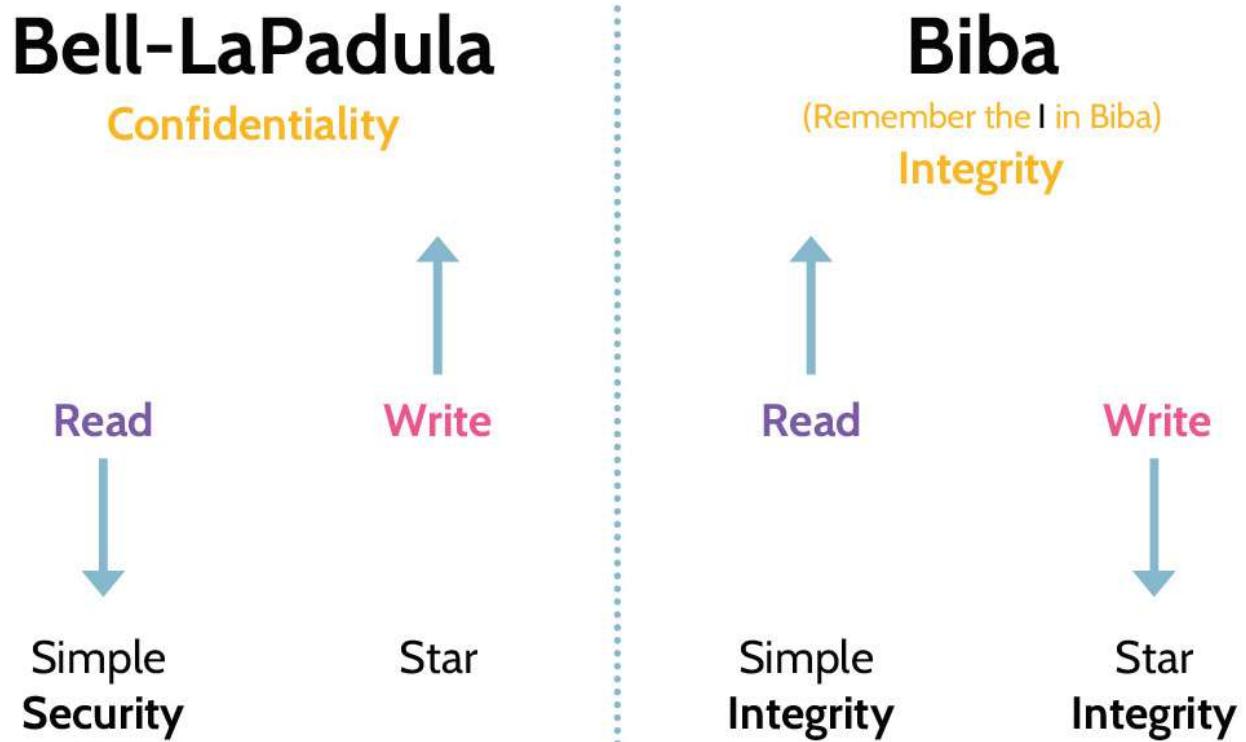


Figure 3-5: Summary of Bell–LaPadula and Biba

### Lipner Implementation

Now a question may come to mind: What happens if you want to have both confidentiality and integrity? The Lipner implementation, shown in [Figure 3-6](#), is the answer to how you can get the best out of both worlds.

It's simply an attempt to combine the best features of Bell–LaPadula and Biba regarding confidentiality and integrity. As such, *Lipner is not truly a model but rather an implementation of two models*. In theory, Lipner creates great security by combining two lattice-based models but based upon the way Bell–LaPadula and Biba work. Its principle is to separate objects into data and programs and apply sensitivity levels and job categories to subjects.

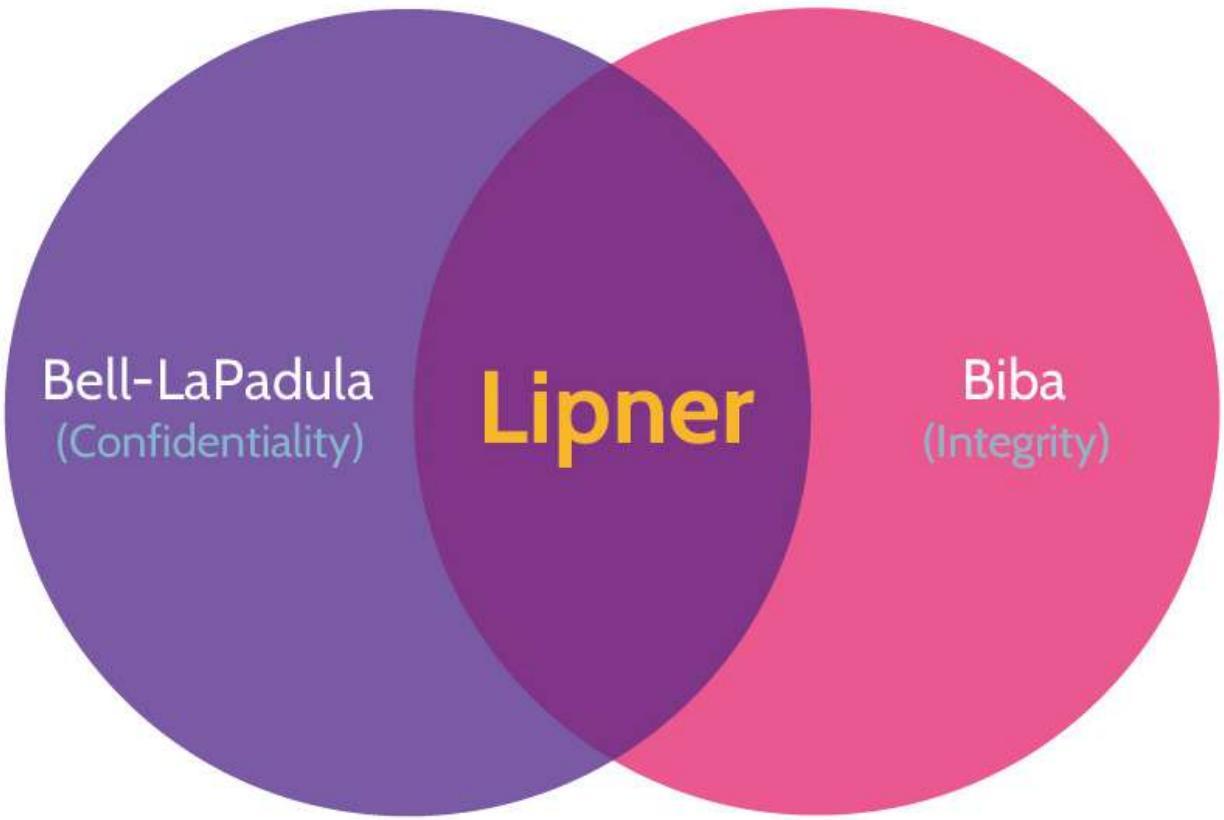


Figure 3-6: Lipner Implementation

### 3.2.4 Rule-based Models

#### CORE CONCEPTS

- Information flow models track the flow of information and can help uncover covert channels.
- Covert channels are *unintentional* communications paths; two types exist: storage and timing.
- Clark-Wilson is an integrity-focused rule-based model that includes three goals and three rules.
- Brewer-Nash—the “Chinese Wall” model—is designed to prevent conflicts of interest.

At the core of any rule-based model is a set of rules that mediate access between subject and objects. Depending on the model, the number and complexity of rules employed may vary widely, and the focus of the model can also vary. In addition to looking at some specific rule-based models further below, a basic understanding of information flow models and covert channels should first be covered.

## Information Flow

Information flow models track the flow of information. If the flow of information can be tracked, this implies it can be tracked throughout its life cycle; in other words, it can be tracked from the point of origin, whether collected or created, to its storage, use, dissemination, sharing with others, and eventually to its end of life (e.g., archival and destruction). Information flow can also help the identification of vulnerabilities and insecurities, like covert channels. Referring to lattice-based models, the information flow model serves as the basis for both Bell–LaPadula and Biba.

## Covert Channels

Covert channels are ***unintentional*** communication paths that may lead to the disclosure of confidential information. The key word is *unintentional*—they’re not meant to be there. However, because they’re there, the potential for confidential information to be disclosed exists.

A wonderful example of a covert channel occurred in the early ’90s and centered around the military operation known as Desert Storm. Some type of military action in the Gulf was supposedly imminent, but Pentagon officials

kept exact details and timing under wraps. However, borrowing from something—the “pizza index”—that some people might think of as myth and others fact, journalists and other interested parties kept track of the number of pizzas being delivered to the Pentagon each day. Typically, a small number of pizzas (e.g., three to four) was delivered but just prior to the start of the conflict, a much larger number (like thirty to forty pizzas) was delivered. Based upon the significant change in the number of pizzas being delivered, it was safe to assume or infer that a much larger number of people were working late at the Pentagon. Why? Because, among other things, they were likely helping with the final planning stages of military action. CNN and other news outlets were able to piece together the inadvertent disclosure of sensitive information and predict when military strikes in the Middle East would commence. This is a perfect example of a covert channel—unintentional and inadvertent disclosure of sensitive and confidential information.

**Covert channels are unintentional & may involve storage or timing; when they exist, confidentiality may be compromised**

Covert channels, also sometimes called *secret channels*, are *unintentional* communications paths, and two types of covert channels exist, as summarized in [Table 3-4](#). Storage refers to when storage capabilities can be exploited in such a way that confidential information is unintentionally disclosed or communicated. *Timing* refers to when the timing capabilities of a system can be exploited in a manner that allows confidential information to

be signaled. Looking back at the pizza example, what type of covert channel does it represent? You guessed it—timing—because that’s where the weakness is, in the timing mechanism. The number of pizzas ordered in a twenty-four-hour period signaled information that was considered confidential.

An example of a storage covert channel exists in most technology architectures. On a laptop, sensitive information could be placed in RAM, because a process needs to use it, but when that process finishes, the sensitive information remains present in memory. That could become available to other processes that are placed in memory and can read it. The sensitive information is not meant to be in RAM, but it is in fact there, and it could disclose confidential information. That is an example of a storage covert channel, which represents the majority of covert channels (roughly 99 percent).

Storage	Timing
Process writes sensitive data to RAM, and the data remains present after the process completes; now, other processes can potentially read the data.	An online web server responds to a user providing an existing username within three seconds, while if the username doesn’t exist it takes one second. That allows the attacker to perform username enumeration.

Table 3-4: Covert Channel Types

## Clark–Wilson

Clark–Wilson is an important, rules-based model that focuses only on **integrity**. Unlike, Biba, which only prevents unauthorized subjects from

making any changes, Clark–Wilson offers further protection and meets three **goals of integrity**: 1. Prevent unauthorized subjects from making any changes (this is the only of the three that Biba addresses) 2. Prevent authorized subjects from making bad changes

### 3. Maintain consistency of the system

#### The goals and rules of integrity found in Clark–Wilson

Biba only addresses #1 and therefore falls short of truly addressing security concerns related to the protection of all integrity, while Clark–Wilson addresses #1 and then further protects integrity through #2 and #3.

Clark–Wilson achieves each of the goals specifically through application of the three **rules of integrity** noted in [Table 3-5](#).

Well-Formed Transactions	Separation of Duties	Access Triple
Good, consistent, validated data. Only perform operations in a manner that won't compromise the integrity of objects.	One person shouldn't be allowed to perform all tasks related to a critical function.	Subject   Program   Object  A subject cannot directly access an object, i.e., in a database, access <i>must</i> go through a program that enforces access rules.

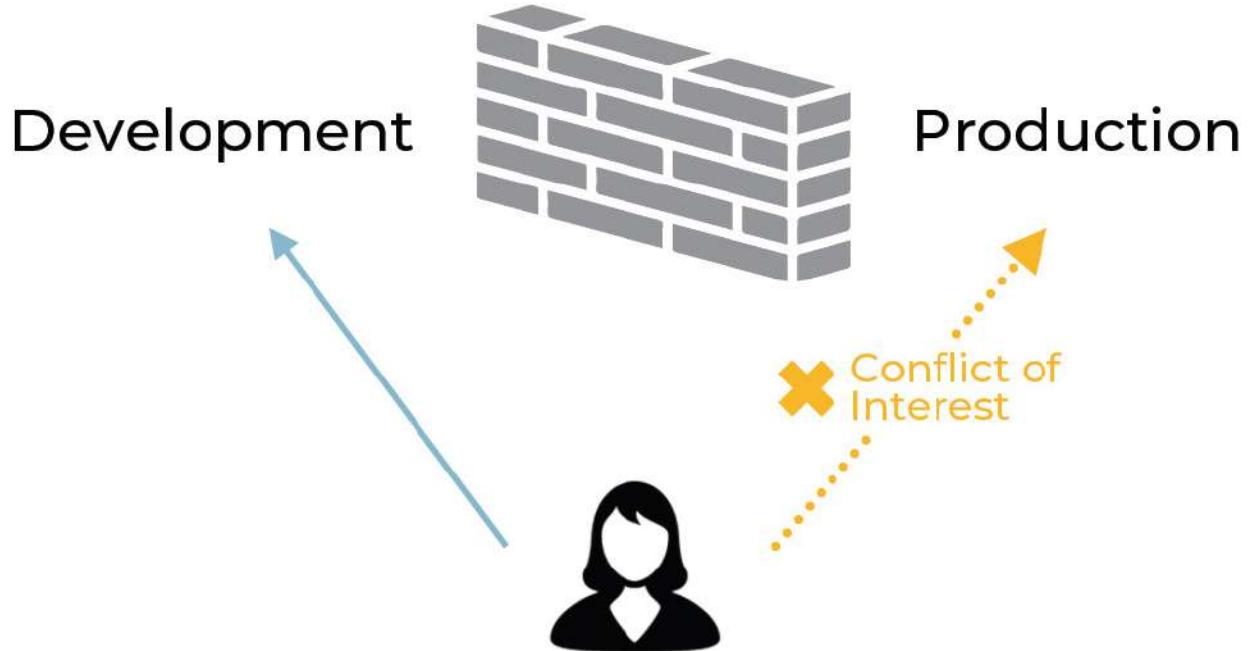
Table 3-5: Clark–Wilson Rules

## Brewer–Nash (The Chinese Wall) Model

Brewer–Nash is also known as “The Chinese Wall” and it is an information flow model that has one primary goal: *Preventing conflicts of interest.*

The model stipulates and ensures that information flows between subjects and objects are only allowed if the information does not provide a conflict of interest. An example of where Brewer–Nash might be implemented is between the Development and Production departments in an organization, as the two departments should not be able to influence each other or even allow access between each other. Another example is a big bank, between the Retail Investments and Mergers and Acquisitions (M & A) departments, as knowledge shared from the M&A department could significantly influence activities in the Investments department as also shown in [Figure 3-7](#).

Like Bell–LaPadula, Brewer–Nash primarily addresses issues related to *confidentiality.*



**Figure 3-7: Conflict of Interest between Two Company Departments**

### Graham–Denning Model

Graham–Denning is another lesser known, rule-based model that specifies rules allowing a subject to access an object.

### Harrison–Ruzzo–Ullman Model

Like Graham–Denning, Harrison–Ruzzo–Ullman is also a rule-based model that focuses on the integrity of access rights via a finite set of rules available to edit the access rights of a subject. It adds the ability to add generic rights to groups of individuals.

Remember that all security models are either classified as lattice-based or rule-based.

### **3.2.5 Certification and Accreditation**

#### **CORE CONCEPTS**

- **Certification is the comprehensive technical analysis of a solution to confirm it meets the desired needs.**
- **Accreditation is management's official sign-off of certification for a predetermined period of time.**

### **Evaluation Criteria**

This next section deals with what are known as “evaluation criteria systems.” The primary thing to remember is they’re measurement systems. When architectures—especially security architectures—are built, products are often purchased from vendors. Security today often relies on solutions and mechanisms provided by vendors. This fact introduces a potential problem: How do we know vendor solutions actually provide the level of security we think they provide? Any vendor is going to say they have the best products, the best solutions, the best architectures. For example, if a firewall needs to be purchased, every firewall vendor is going to say their firewall is the best one available and that it will meet our needs perfectly. How can statements like this be confirmed and verified? We would need an independent and objective measurement system that vendors can use for evaluation purposes and companies for purchasing purposes. Such a system could be used by any organization around the globe to make purchasing decisions and not need to rely on vendors themselves. With such a system, similar products from different vendors can easily be compared. If the product requirements are understood—for example, for the firewall noted above—evaluations from multiple vendor firewalls could be compared and the best one chosen. These

evaluations, these measurements, could be trusted, because they've been created using an independent, vendor-neutral, objective system. These measurement systems do, in fact, exist and are called "evaluation criteria systems." The most well-known evaluation criteria systems are called Trusted Computer System Evaluation Criteria (TCSEC)—also known as the Orange Book, the European equivalent of TCSEC called Information Technology Security Evaluation Criteria (ITSEC), and the latest one, an ISO standard, called the Common Criteria.



From the description above, it should be clear that vendors and consumers have an interest in using these measurement systems, and this explains why they've become very popular—especially the Common Criteria. Consumers want to shop for and buy products from vendors, and they want to know exactly what security levels the products provide and be able to trust those security capabilities. Likewise, vendors want their products measured and

rated, so they'll be more likely to be purchased by customers. For example, if every available firewall is Evaluation Assurance Level 4 (EAL4 indicates a very methodical design, test and review process has been implemented) capable, a firewall that receives a rating of EAL3 is most likely not going to sell very well, as some redesign and other changes will likely need to take place. Evaluation and rating systems like this benefit vendors by giving them confidence that consumers will be interested in the products if the products are rated well. When an independent certified company examines a product using objective and commonly accepted criteria versus a biased entity like the vendor or a related party using subjective and biased criteria, everybody can trust the results. Potential customers can examine the resulting documentation and make informed purchasing decisions, while vendors can examine the results and know exactly how their product compares to the same category of products from other vendors. This is what each of the evaluation criteria systems do, and the Common Criteria, in its third and most recent version, is by far the best one available today.

**Certification** and **Accreditation** are two related, yet different and very important concepts in security. Their definitions are outlined in [Table 3-6](#).

Certification	Accreditation
Comprehensive technical analysis of a solution or a product to ensure it meets the desired needs	Official management signoff of certification for a set period of time

Table 3-6: **Certification and Accreditation**

## **Understand the difference between certification and accreditation and what each provides**

Certification is the comprehensive technical analysis of a solution to make sure that it meets your needs. For example, for an organization that needs a firewall, how will the organization easily sift through the likely dozens of firewalls that exist and evaluate the one best suited to meet its needs? Before any decision or recommendation can be made, the requirements of the organization must be understood. Once the requirements are understood, a solution can more easily be identified. Among available firewalls, the one that best meets the requirements will be identified, and a comprehensive technical analysis of each potential solution is the basis for this identification. This is certification.

Accreditation is the official management decision to use a solution. Usually, this is done for a set period of time. One important note, accreditation is not performed by the security function; rather, it's performed by the asset owner or management. They will make the decision to accredit the solution for, let's say, the next eighteen months. Once this time period expires, the certification and accreditation process would be repeated. The same solution that's been in use might still be the best one, at which point management would re-accredit the solution for another set period of time; otherwise, a different solution would be selected.

This is the benefit of some of these evaluation criteria systems, especially of the Common Criteria. Once a vendor's product is evaluated, it becomes available to everybody. In addition, every capability and shortcoming of a

given product will be freely viewable. Most vendors commonly publish the results of a Common Criteria assessment on the products section of their website. The collection of products that have been evaluated using the Common Criteria continues to grow, which is great; it requires a massive amount of work, but the output of information is extremely helpful to consumers.

### **3.2.6 Evaluation Criteria (ITSEC and TCSEC)**

#### **CORE CONCEPTS**

- **TCSEC, also known as the Orange Book, is the first evaluation criteria system.**
- **ITSEC followed TCSEC and was developed by Europeans to include elements of the Orange Book and others.**
- **ITSEC applies to networked environments and measures functional and assurance elements separate from one another.**

## **Orange Book/Trusted Computer System Evaluation Criteria**

(TCSEC)

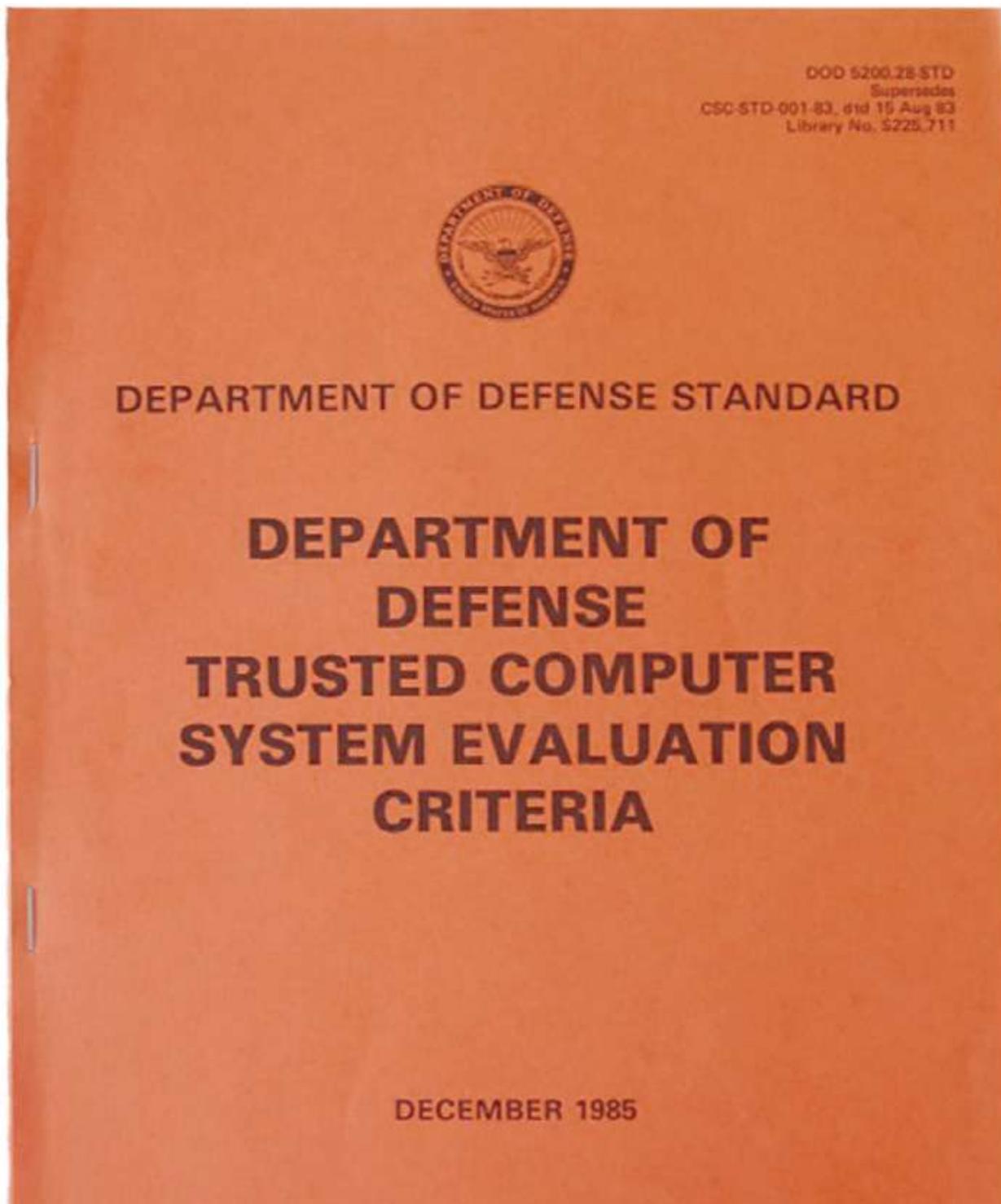


Figure 3-8: TCSEC Guide

The first evaluation criteria system created is often referred to as the “Orange Book,” depicted in [Figure 3-8](#), due to the fact the cover of the book is orange. It was written as part of a series of books known as the “rainbow series,” published by the US Department of Defense in the ’80s. Each book in the series deals with a topic related to security, and the cover of each is a different color, thus the nickname “rainbow series.” There’s one book called the Light Blue Book that deals with password guidelines, while the Red Book deals with network security, and the Orange Book focuses on measuring security products.

The classification levels—the criteria—used in the Orange Book are shown in [Table 3-7](#).

Functional Levels	
A1	Verified Design
B3	Security labels, verification of no covert channels, and must stay secure during start up
B2	Security labels and verification of no covert channels
B1	Security labels
C2	Strict login procedures
C1	Weak protection mechanisms
D1	Failed or was not tested

Table 3-7: Orange Book Evaluation Criteria

The Orange Book only measures confidentiality, which reflects the Department of Defense's mission in the early '80s, when confidentiality (especially in the context of the military) was so important. Even by today's standards and when many people say it's obsolete, if you're interested only in confidentiality, there's really no better system than TCSEC. Evaluation criteria goes from D1, which means no security, to A1, which means very robust and mathematically verified security as shown in [Table 3-7](#). Despite the possibility of an A1 rating, most products are typically rated at C2 or B1. Each rating level implies that a given product "inherits" all the characteristics of the previous level.

The Orange Book, however, only measures confidentiality. In addition, it only measures single-box type of architectures; it does not map well to networked environments. This is why a lot of organizations in Europe considered the model from a more current perspective and revamped it. They took what they considered to be a good idea and made it better in what is known as the Information Technology Security Evaluation Criteria (ITSEC).

## **Information Technology Security Evaluation Criteria (ITSEC)**

Unlike the Orange Book, ITSEC measures more than confidentiality, and it works well in a network environment. Also, when ITSEC was created, ways to measure function and assurance separate from each other were incorporated. When a product is considered through the lens of ITSEC, two ratings are given. One rating—the "F" levels—is a functional rating, like the ones used in the Orange Book. The other rating—the "E" levels—was introduced as part of ITSEC and refers to levels of assurance. E levels range from E0 to E6 (as also seen in [Table 3-8](#)), with E6 being the top and

representing the best level. Obviously, E6 would be a system that provides robust assurance while E0 not so much.

Assurance Levels	
<b>E6</b>	Formal end-to-end security tests + source code reviews
<b>E5</b>	Semi-formal system + unit tests and source code review
<b>E4</b>	Semi-formal system + unit tests
<b>E3</b>	Informal system + unit tests
<b>E2</b>	Informal system tests
<b>E1</b>	System in development
<b>E0</b>	Inadequate assurance

Table 3-8: ITSEC Assurance Levels (E levels)

Note that ITSEC improves on the Orange Book by providing:

- Functional measurements (same as the Orange Book)
  - Assurance measurements (E levels)
- ITSEC was replaced by Common Criteria in 2005.

### 3.2.7 Common Criteria

#### CORE CONCEPTS

- **Common Criteria evaluation criteria system is the best and most popular system.**

- Comprised of multiple components that work together and allow a globally recognized rating to be assigned to products.
- Common Criteria EAL rating levels

ISO 15408, better known as the Common Criteria, is the most used of the evaluation criteria systems and is also the most popular; most products are evaluated using it. As such, it's critical to understand Common Criteria components, Evaluation Assurance Levels (EAL), and the ramifications if changes to an EAL-rated system take place.

Like the other evaluation systems, the Common Criteria provides confidence in the industry for consumers and security functions as well as for vendors and others. The Common Criteria is the latest measurement system, and it's also an ISO standard (ISO 15408). It's called the Common Criteria because several countries joined together with a common goal: to create a common measurement system that could be trusted globally. For example, if a German-made product is rated, the rating can be trusted by US-based companies, because the rigorous rating process is independent and objective and globally applicable. To make this possible, globally dispersed, independent, Common Criteria–licensed organizations evaluate and rate products. A firewall vendor will regularly hire a Common Criteria–licensed company to evaluate and rate their firewall. The licensed organization would use a specific and rigorous process to measure and ultimately rate the firewall with an EAL level that objectively communicates what security it provides. The process also provides documentation that may prove useful to security professionals and other potential consumers of the firewall. [Figure 3-9](#) depicts the Common Criteria components and process flow.

## Common Criteria Process

The first component is the **Protection Profile (PP)**. The PP lists the security capabilities that a type or category of security products should possess. For example, there's a Protection Profile for firewalls; it lists the security capabilities that any firewall should contain—for example, two-factor authentication (2FA) capabilities, VPN capabilities, ability to encrypt to 128-bit encryption level, and secure logging, to name a few. Protection profiles exist for different categories of security products and serve to add uniformity to each category; there's one for firewalls, one for access controls, one for IDS systems, etc. This approach provides a consistent framework of capabilities that a specific category of product should possess.

### What the Common Criteria Protection Profile (PP) components represent

**Target of Evaluation (TOE)** is the next component. Using the earlier firewall example, if a vendor desires for their firewall to be rated according to the Common Criteria, the firewall would be considered the TOE. The TOE—the Target of Evaluation—is simply a vendor's product that's being rated and being assessed according to the Common Criteria.

The next component, the **Security Targets (ST)**, describe—from the vendor's perspective—each of the firewall's security capabilities that match up with capabilities outlined in the Protection Profile. When the firewall is measured, capabilities like VPN, encryption, two-factor authentication, secure logging, and so on are going to be compared against standards listed in

the protection profile and tested extensively. For example, the firewall may perform two-factor authentication very well, but it lacks strong VPN capabilities. Security targets are going to be scrutinized, and each will be scrutinized under the dual lens of **functional** and **assurance security** capabilities, the two prementioned pillars of a well-implemented security control.

The Security Targets is really where the value of the Common Criteria lies. As each capability is tested extensively, there is much documentation that is produced that highlights the security capabilities and deficiencies of each security target being tested. This is what provides potential consumers the information that is needed for them to evaluate vendor products.

The evaluation process is the part of the Common Criteria that creates meaningful documentation that becomes available to any interested parties. At the end, after the capabilities of the firewall have been evaluated, an overall EAL level will be assigned to the firewall. An EAL can range anywhere from 1–7, with 7 being the most thorough and exhaustive. The above process steps are summarized in [Table 3-9](#), while the different EAL levels are listed in [Table 3-10](#).



Figure 3-9: Common Criteria Process

<b>PP</b> Protection Profile	Specification of functional and assurance requirements for a specific type of security product
<b>TOE</b> Target of Evaluation	The specific product/system to be evaluated
<b>ST</b> Security Targets	Written statement by vendor explaining how functional and assurance specifications of the product meet the protection profile (PP) requirements
<b>Security Functional Requirements</b>	Security targets are evaluated from a functional perspective: what features exist and how well they work relative to the desired and expected security behavior.

<b>Security Assurance Requirements</b>	Security targets are evaluated from an assurance perspective: that the vendor's claimed security functionality and the CC evaluation process align.
<b>Evaluate</b>	Consider all of the components together
<b>Assign EAL (1-7)</b>	See the seven Common Criteria EAL levels below

**Table 3-9: Common Criteria Steps**

<b>EAL7</b>	Formally verified, designed, and tested
<b>EAL6</b>	Semi-formally verified, designed, and tested
<b>EAL5</b>	Semi-formally designed and tested
<b>EAL4</b>	Methodically designed, tested, and reviewed
<b>EAL3</b>	Methodically tested and checked
<b>EAL2</b>	Structurally tested
<b>EAL1</b>	Functionally tested

**Table 3-10: EAL Levels**

### EAL level definitions and order

Despite the table above illustrating multiple EAL levels, EAL7 is not necessarily the best for the sake of a vendor marketing and selling its product. In fact, most organizations will not purchase a product that is rated above EAL4. Operating systems are typically at EAL3 and firewalls at EAL4.

## **The potential negative implications if a product is rated too high**

If a product is at EAL7, it could become more vulnerable to compromise, due to being more complex and harder to maintain. Yes, the product might offer more security features and capabilities, but if they require extensive configuration, administrative skills, and maintenance, consumers will likely not use the features. This could ultimately leave an organization at greater risk. Vendors, therefore, must balance the trade-off between functionality and security. Too much of the latter always impacts product speed and administrative overhead, and it might lead to the creation of a very expensive product too. This explains why vendors often produce products that are configurable. Just because a firewall is EAL 4 capable, it doesn't mean it must be operated at EAL4; a customer could very well decide to configure and use it at an EAL3 level.

## **If a product is patched or receives software/firmware updates, the EAL level remains the same**

A final thing to note is that, after a product undergoes an evaluation and is assigned an EAL level, the EAL level for that product will remain the same throughout its life span, unless a major change in product functionality is introduced (in this case, the vendor may choose to have the product re-evaluated according to the Common Criteria). In other words, when a patch or software update to the product is made, the EAL level remains unchanged, unless the vendor decides to have it be re-evaluated. Evaluation through the

Common Criteria is a voluntary process that costs money but ultimately provides the vendor with trust and confidence in their products, and provides consumers with objective and independent trust mechanisms.

### **3.3 Select controls based upon systems security requirements**

#### **3.3.1 Security Control Frameworks**

##### **CORE CONCEPTS**

- Security control frameworks aid with the control selection process.
- Security control frameworks provide guidance, based upon best practices.
- Features from multiple frameworks can be used to meet the needs of an organization.

#### **What Do Security Control Frameworks Provide?**

This section focuses on the selection of controls to include in a system. Recall that this domain's focus is building systems and protection mechanisms to secure those systems, which requires breaking the systems down into components and protecting each component. Components should be protected based upon value, which drives the selection of controls. All this activity is predicated on risk management.

As controls are considered, especially mitigating controls, control frameworks can be utilized to aid with the control selection process. Control frameworks provide comprehensive guidance, based upon best practices. For

example, a security professional might be pondering how to protect a certain system component, like a storage device, CPU, or piece of memory, and turning to a control framework could offer insight into appropriate controls to consider. Additionally, as control frameworks provide guidance, the best and most applicable elements of multiple frameworks could potentially be utilized as part of the control selection process.

**Understand the major frameworks at a high level,  
especially ISO 27001/02, which is an internationally  
recognized framework**

From this perspective, the following section is a reminder of some of the major control frameworks available and how they can be applied. In addition to technical frameworks, many of the ones listed in [Table 3-11](#) focus on overall business processes.



<b>COBIT</b>	The Control Objectives for Information Technologies (COBIT) framework is particularly useful for <b>IT assurance</b> , such as conducting audits and gap assessments. It was created by Information Systems Audit and Control Assurance (ISACA), for information technology management and IT governance, and therefore it is particularly useful for IT assurance activities.
<b>ITIL</b>	Information Technology Infrastructure Library (ITIL) defines the <b>processes</b> in a well-run IT department, from the onboarding process to procurement, change management, configuration management, and access control, to name a few. ITIL defines the processes for IT service management that focuses on aligning IT services with business goals and objectives.
<b>NIST SP 800-53</b>	National Institute of Standards and Technology Special Publication (NIST SP) 800-53 is a set of best practices, standards, and recommendations that help an organization improve its cybersecurity controls.
<b>PCI DSS</b>	The Payment Card Industry Data Security Standard (PCI DSS) is a standard for organizations that <b>handle credit cards</b> like VISA, MasterCard, and American Express. The PCI Standard was created by the card brands, and it is administered by the Payment Card Industry Security Standards Council. The

## **ISO 27001**

standard was created to increase controls around cardholder data to reduce credit card fraud. The volume of transactions processed by a merchant helps determine the method used to validate compliance.

International Organization for Standardization/International Electrotechnical Commission (ISO/IEC) 27001:2022 specifies the requirements for establishing, implementing, maintaining, and continually improving an information security management system within the context of the organization. It also includes requirements for the assessment and treatment of information security risks tailored to the needs of the organization. The requirements set out in ISO/IEC 27001:2022 are generic and are intended to be applicable to all organizations, regardless of type, size, or nature.

**Organizations can be certified against ISO 27001.**

**Annex A** of the standard contains the following domains: 1. Information security policies

2. Organization of information security
3. Human resource security
4. Asset management
5. Access control
6. Cryptography
7. Physical and environmental security
8. Operations security

## **9. Communications security**

10. System acquisition, development, and maintenance
11. Supplier relationships

## **12. Information security incident management**

13. Information security aspects of business continuity management

## **14. Compliance**

<b>ISO 27002</b>	ISO/IEC 27002:2022 provides <b>guidelines</b> for organizational information security standards and information security management practices including the selection, implementation, and management of controls, taking into consideration the organization's information security risk environment(s). Essentially ISO 27002 provides guidance for implementing the controls in ISO 27001.
<b>COSO</b>	Committee of Sponsoring Organizations of the Treadway Commission (COSO) is a voluntary private sector initiative dedicated to improving organizational performance and governance through effective internal control, enterprise risk management, and fraud deterrence.
<b>HIPAA</b>	Health Insurance Portability and Accountability Act (HIPAA) relates to security controls in the <b>health-care industry</b> , and it focuses on the protection of protected health information (PHI) of individuals.
<b>FISMA</b>	Federal Information Security Management (FISMA) Act of 2002 requires US federal agencies to develop, document, and implement agency-wide security programs to provide information security for the operations and assets of the agency. FISMA further requires security programs for any other agencies, contractors, or service providers.
<b>FedRAMP</b>	Federal Risk and Authorization Management Program (FedRAMP) provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services. <b>Any cloud services that hold US federal government data must be FedRAMP authorized.</b>
<b>SOX</b>	Sarbanes-Oxley (SOX) Act is a direct result of the wild financial fraud at “Enron”. The US Congress decided better controls were needed to be in place to prevent similar incidents from happening again, and specifically enacted SOX to prevent financial fraud by public companies and thereby protect the financial interests of shareholders.

**Table 3-11: Security Control Frameworks**

## Rationalizing Frameworks

Figure 3-10 illustrates how all these different security frameworks relate to one another.

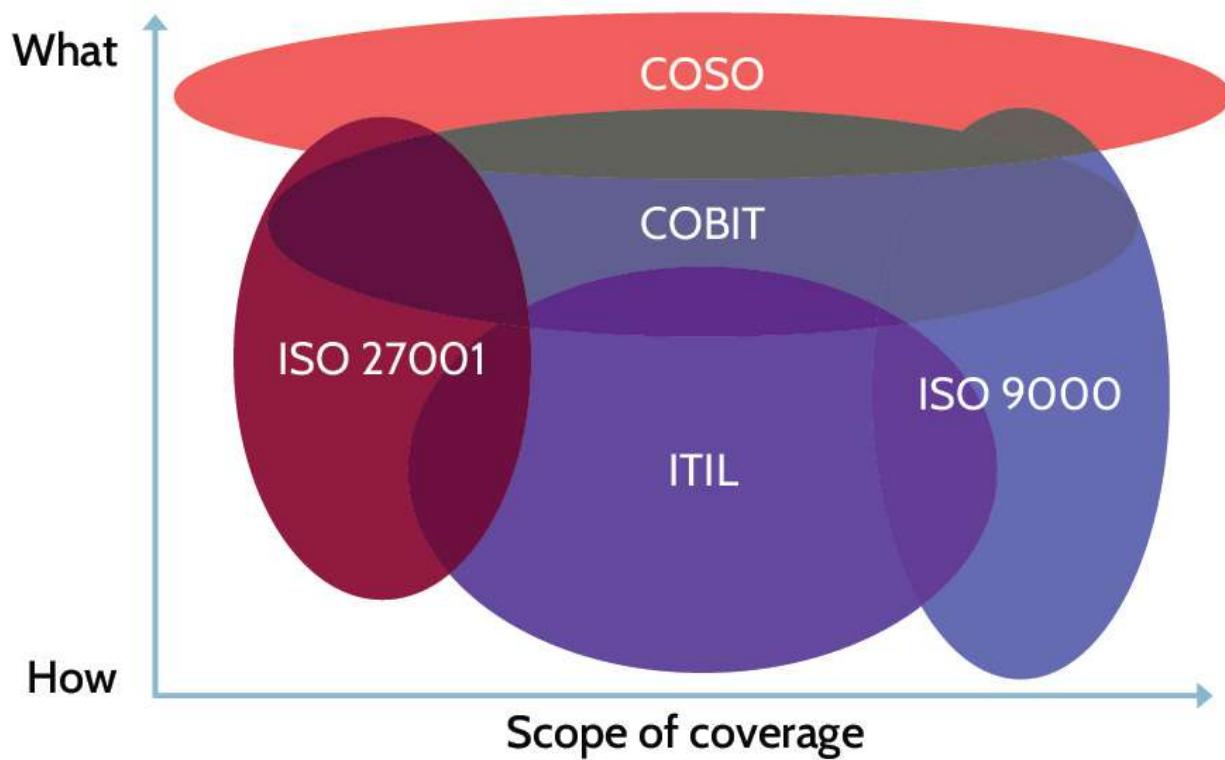
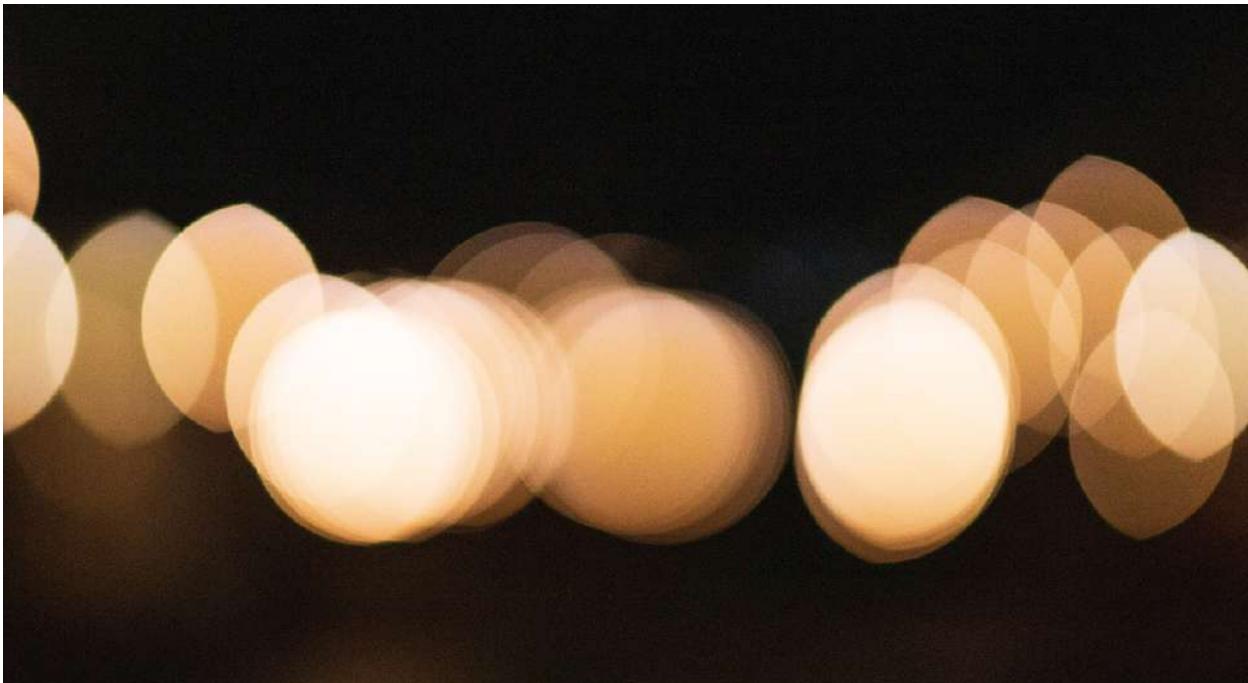


Figure 3-10: Framework Relations

Notice that they overlap, which means that frameworks can span contexts, and as also noted earlier, organizations will often choose to use features from multiple frameworks to meet their needs. Consider the example of a hospital, where patients are treated and payments are made. In this context, HIPAA and PCI DSS frameworks should undoubtedly be present, among other frameworks. Most organizations choose to use different ones for various contractual or legal reasons. Then they combine these and attempt to develop one overarching and simplified framework. It doesn't make sense to test a

very similar control multiple times. As such, controls are merged, rationalized, and then tested once. This is the approach most organizations take.



### **3.4 Understand security capabilities of information systems (e.g., memory protection, Trusted Platform Module (TPM), encryption/decryption)**

#### **3.4.1 RMC, Security Kernel, and TCB**

##### **CORE CONCEPTS**

- Security within information systems always pertains to subjects and objects.
- The Reference Monitor Concept (RMC) is a concept.
- Implementation of the RMC is known as a security kernel.

- A security kernel should consist of three properties, or characteristics: completeness, isolation, and verifiability.
- The term Trusted Computing Base (TCB) refers to all the protection mechanisms within an architecture; the TCB is the *totality* of protection mechanisms within an architecture.

## Subjects and Objects

Before diving into concepts like the RMC and security kernel, it's important to understand subjects and objects, as defined in [Table 3-12](#), as those concepts are heavily used throughout the following section.

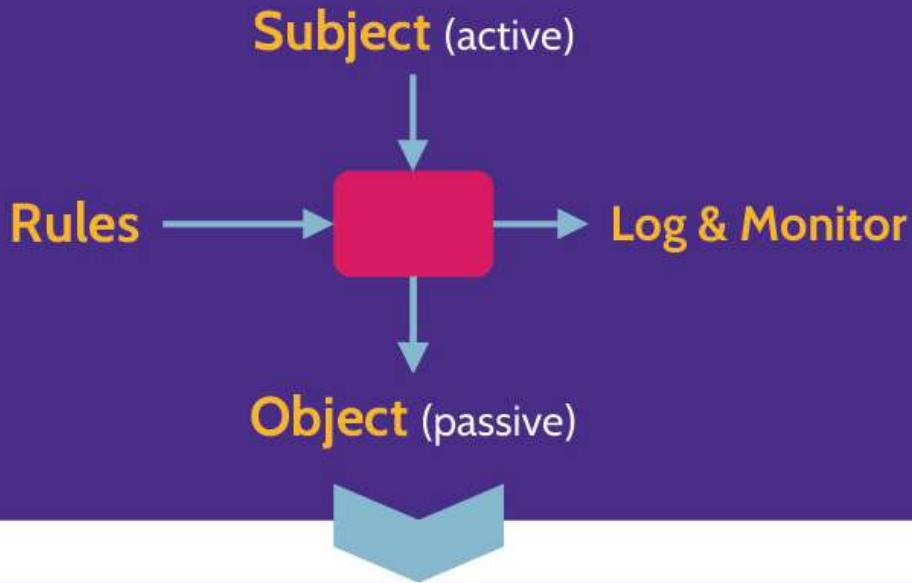
Subject	Object
<b>Active</b> entities A subject is a person, process, program, or anything similar that actively tries to access an object.	<b>Passive</b> entities An object is anything that is being passively accessed by a subject, like a file, server, process, or hardware component.

Table 3-12: Subjects and Objects

## Reference Monitor Concept (RMC)

### TRUSTED COMPUTING BASE (TCB)

#### Reference Monitor Concept (RMC)



Implementation of RMC = **Security Kernel**

**Completeness** – impossible to bypass

**Isolation** – tamperproof

**Verifiability** – verified correct

Figure 3-11: Reference Monitor Concept

The RMC is simply the concept of a subject accessing an object through some form of mediation that is based on a set of rules, with this access being logged and monitored. This is the “reference monitor concept,” depicted in

[Figure 3-11](#), which is prevalent throughout security and is a topic often seen on the exam. When a key is put into a door lock, the reference monitor concept is present; when logging on a computer system, the reference monitor concept is there—as a subject is accessing an object—based on a set of rules, and this activity is logged and monitored. This concept is constantly employed in security. There are a few key points to keep in mind about the reference monitor concept. RMC features include:

- Must mediate all access
- Be protected from modification
- Be verifiable as correct
- Always be invoked

It's equally important to remember that the reference monitor concept is just a concept. Unless the RMC is implemented, it's simply some good ideas. The *implementation of the reference monitor concept is known as a security kernel.*

## Security Kernel

Here's an important distinction: the security kernel controls access to any asset; the reference monitor concept defines a theory, a concept. *The security kernel is the implementation of the reference monitor concept.* It's important to understand this distinction. Any system that is actually controlling access must be an actual implementation. If it's implemented, it's a security kernel.

When implemented, a viable security kernel should contain three properties.

The first property is known as **completeness**. Completeness means it is impossible to bypass the mediation. If a subject is somehow able to bypass the mediation, there would be no point in having a security kernel. Completeness means it is impossible to bypass the security kernel; the subject must always go through the security kernel when accessing the object.

The next property is known as **isolation**. Isolation relates to the mediation rules and specifically ensures that the mediation rules are tamper-proof. Only authorized individuals should be able to change these rules.

The final property, **verifiability**, relates to the aspect of assurance. It means being able to verify that the security kernel is functioning correctly. How do we do that? This is done through logging and monitoring, including other forms of testing.

**Any time a security kernel is implemented, it should demonstrate the three characteristics or properties of the RMC: completeness, isolation, and verifiability**, which are also summarized in [Table 3-13](#).

Completeness	Isolation	Verifiability
Impossible to bypass mediation; impossible to bypass the security kernel	Mediation rules are tamperproof	Logging and monitoring and other forms of testing to ensure the security kernel is functioning correctly

Table 3-13: RMC Characteristics

## Trusted Computing Base (TCB)

Another term that encompasses all the security controls that are implemented to protect an architecture is the trusted computing base (TCB). This is the term used to refer to all the protection mechanisms within a system, within an architecture; the TCB is the *totality* of protection mechanisms within an architecture. Examples of components that would be within the TCB include

all the hardware, firmware, and software processes that make up the security system. For a large organization, things like policies and procedures, onboarding processes, change management, the entire network, security training and awareness programs, and similar things would be part of the TCB. Clearly, the TCB can be huge, and if an entire enterprise is being considered it can be massive.

It's worth highlighting that all components noted below are found in the TCB:

- Processors (CPUs) ■ Memory ■ Primary storage ■ Secondary storage ■ Virtual memory ■ Firmware ■ Operating systems ■ System kernel

### 3.4.2 Processors (CPUs)

#### CORE CONCEPTS

- A central processing unit (CPU) is the brain of a computer; it processes all of the instructions and ultimately solves problems.
- CPU processing involves an ongoing, four-step process: Fetch, Decode, Execute, and Store.
- A CPU operates in one of two states: the supervisor state or the problem state.

A CPU is the component within a computer that processes all the instructions. Essentially, a CPU is the brain of a computer. As shown in [Figure 3-12](#), a CPU constantly iterates through this four-step process: ► Fetching instructions and data ► Decoding instructions ► Executing instructions ► Storing results

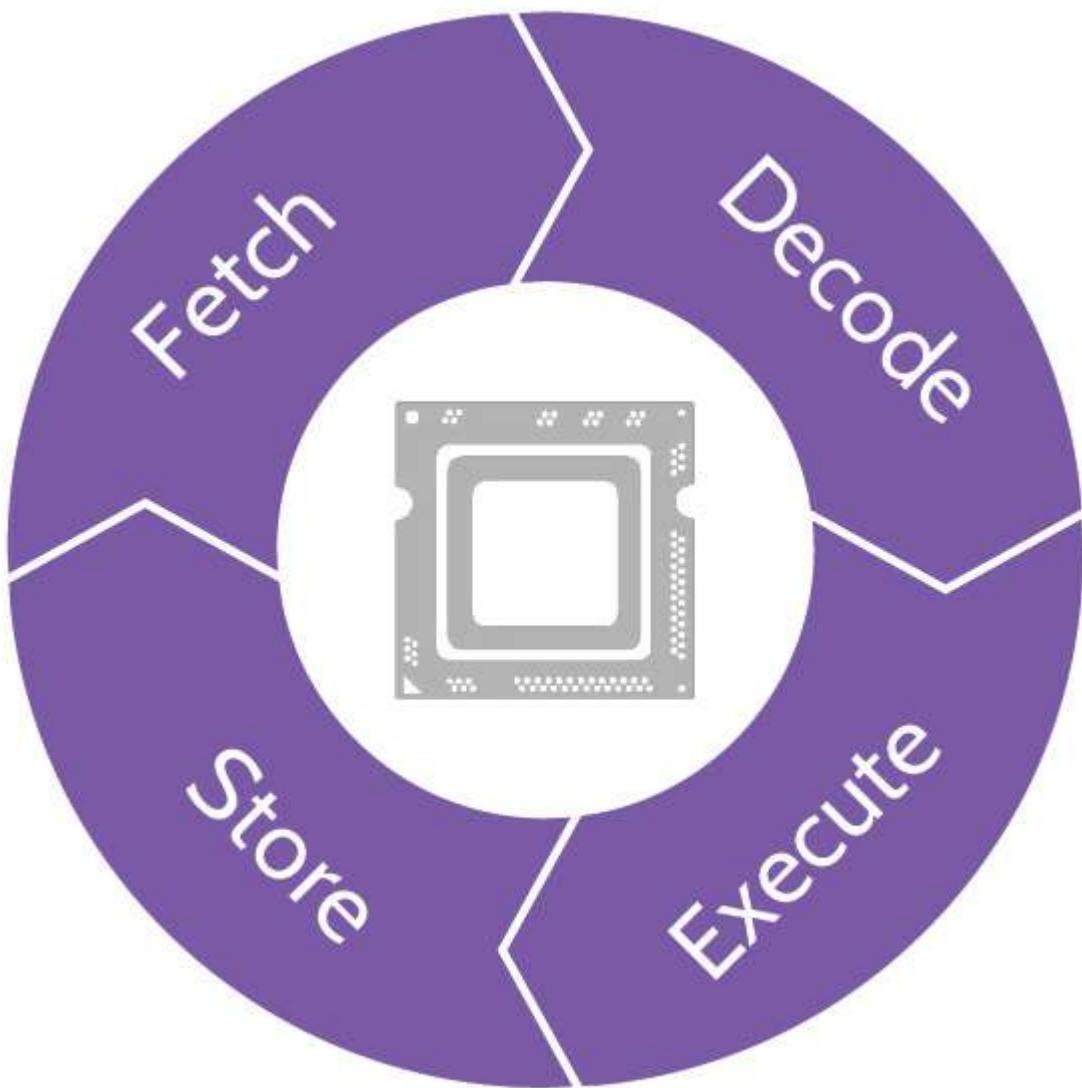


Figure 3-12: Four-Step CPU Processing

At their essence, CPUs solve problems.

## Processor States

From a security perspective, CPUs operate in one of two processor states: the supervisor or problem state. These states can also be thought of as privilege levels and are simply **operating modes for the processor that restrict the operations that can be performed by certain processes**. A summary of their characteristics is provided in [Table 3-14](#).

Supervisor State	Problem State
------------------	---------------

- |   |   |
|---|---|
| <ul style="list-style-type: none"> <li>■ Higher privilege level ■ Typically, where the system kernel runs, allowing full access to all of the instructions and capabilities of a CPU</li> </ul> | <ul style="list-style-type: none"> <li>■ Lower privilege level ■ Limited access to CPU instructions ■ The standard operating mode of a CPU</li> </ul> |
|   | <ul style="list-style-type: none"> <li>■ Known as “problem” state because fundamentally this is what a CPU does: solve problems</li> </ul>            |

Table 3-14: Processor States

### 3.4.3 Process Isolation

#### CORE CONCEPTS

- Prevents interactions that could result in negative consequences
- Two primary methods: memory segmentation and time-division multiplexing

If two applications are running on a computer, should one application be able to access the memory of the other application and manipulate it? Not really, unless the applications are specifically designed to allow interaction. If this was allowed, data could be corrupted, and inappropriate access could take place. A worst-case scenario could be one of the running processes accessing the operating system, which is another process. Processes need to be separated.

From a security perspective, process isolation is a critical element of computing, as it prevents objects from interacting with each other and their resources.

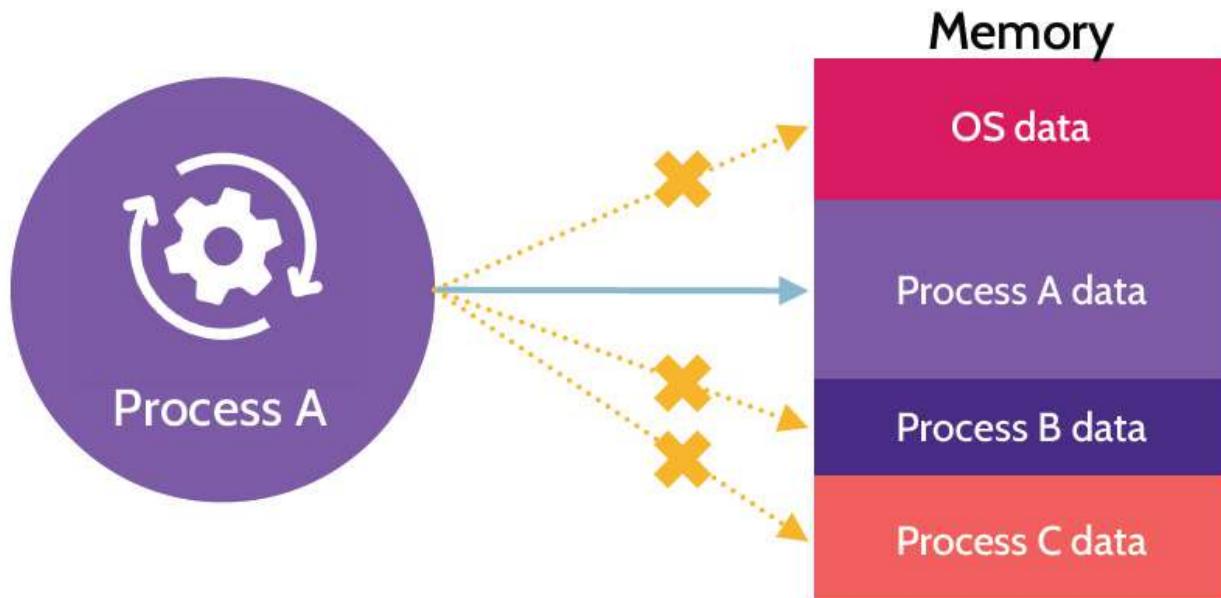
In other words, **the actions of one object should not affect the state of other objects.**

Process isolation is often accomplished using either of the two following methods:

- Memory segmentation ■ Time-division multiplexing **Time-division multiplexing** relates more to the CPU. With time-division multiplexing, process isolation is determined by the CPU. As before, when multiple applications are running, multiple accompanying processes are also running. In this case, the **CPU allocates very small slots of time to each process**. In reality, due to the extremely fast nature of even a slow CPU, being able to run multiple applications at the same time appears seamless. Under the hood, though, each application’s process is running in isolation, based upon the CPU’s processing time allocation.

**Memory segmentation** is all about separating memory segments from each other to protect the contents, including processes that may be running in those segments. In many cases it relates more to **Random-Access Memory (RAM)**—the high-speed volatile storage area found in computer systems. When applications are launched, a “loading” or “starting application” message sometimes accompanies that process. What’s actually taking place is program code is being moved from the hard drive into RAM, because hard drives tend to be much slower than RAM. With code in RAM, when the CPU

makes a request, subsequent processing can happen faster. However, if code from several different applications is loaded into RAM at the same time, one application shouldn't be able to access code from another application. **Memory segmentation ensures that the memory assigned to one application is only accessible by that application.** Based upon application needs, segments of memory are isolated and assigned to each application, and no other application should be able to access that particular segment, as depicted in [Figure 3-13](#).



[Figure 3-13: Memory Use](#)

The process isolation methods are summarized in [Table 3-15](#).

Memory Segmentation	Time-Division Multiplexing
Utilizes RAM and ensures that memory assigned to an application is not accessible by another application	CPU divides time into slices and allocates slots of time to different processes

[Table 3-15: Process Isolation Methods](#)

### 3.4.4 Types of Storage

#### CORE CONCEPTS

- Two types of storage: primary and secondary storage
- Primary storage is fast and volatile

## ■ Secondary storage is slow and non-volatile

The topic of storage, briefly mentioned earlier, requires a bit more elaboration. Storage is where data in a computer system can be found. At a high level, two main types of storage exist: primary and secondary storage. [Table 3-16](#) shows the characteristics of each type.

Primary Storage	Secondary Storage
<ul style="list-style-type: none"><li>■ Fast ■ Volatile—data is lost when device gets powered off ■ Small size ■ Examples of primary storage: ■ Cache ■ CPU registers ■ RAM</li></ul>	<ul style="list-style-type: none"><li>■ Slow ■ Non-volatile ■ Large size ■ Examples of secondary storage: ■ Magnetic hard drives ■ Optical media ■ Tapes ■ SSD</li></ul>

**Table 3-16: Primary (Volatile) and Secondary (Non-volatile) Storage**

Primary storage is also sometimes referred to as *volatile memory*. With volatile memory, anything stored there is temporary. In other words, if the power goes out, for example, anything stored in volatile memory goes away. With non-volatile memory, if the power goes out, data remains. Examples of volatile (or primary storage) include RAM and cache, to name a couple. If the power to a system gets cut, any data stored in volatile memory goes away. That's a big disadvantage. However, the major advantage is speed; primary memory is extremely fast. Examples of secondary storage (non-volatile memory) include hard drives, tapes, discs (CD/DVD), and similar items.

Another related concept refers to what happens because of RAM filling up when many applications are running at the same time. Data related to each program and running process are loaded into RAM, and if RAM fills up, the system will eventually crash. A way to mitigate this is using what's known as **paging**, or **virtual memory**, as depicted in [Figure 3-14](#). Even though multiple programs or processes might be running, the operating system keeps track of what is being accessed in RAM and how often. When data stored in RAM is not being used frequently, the operating system proactively moves it out of RAM and onto a portion of the hard drive dedicated for this purpose. This portion of the hard drive is often referred to as the “paging file” or virtual memory. When data stored in the paging file is needed, the operating system pulls it back into RAM. This process can lead to latency and slow performance, but it keeps a system from crashing; it's less of an issue today because RAM is inexpensive.

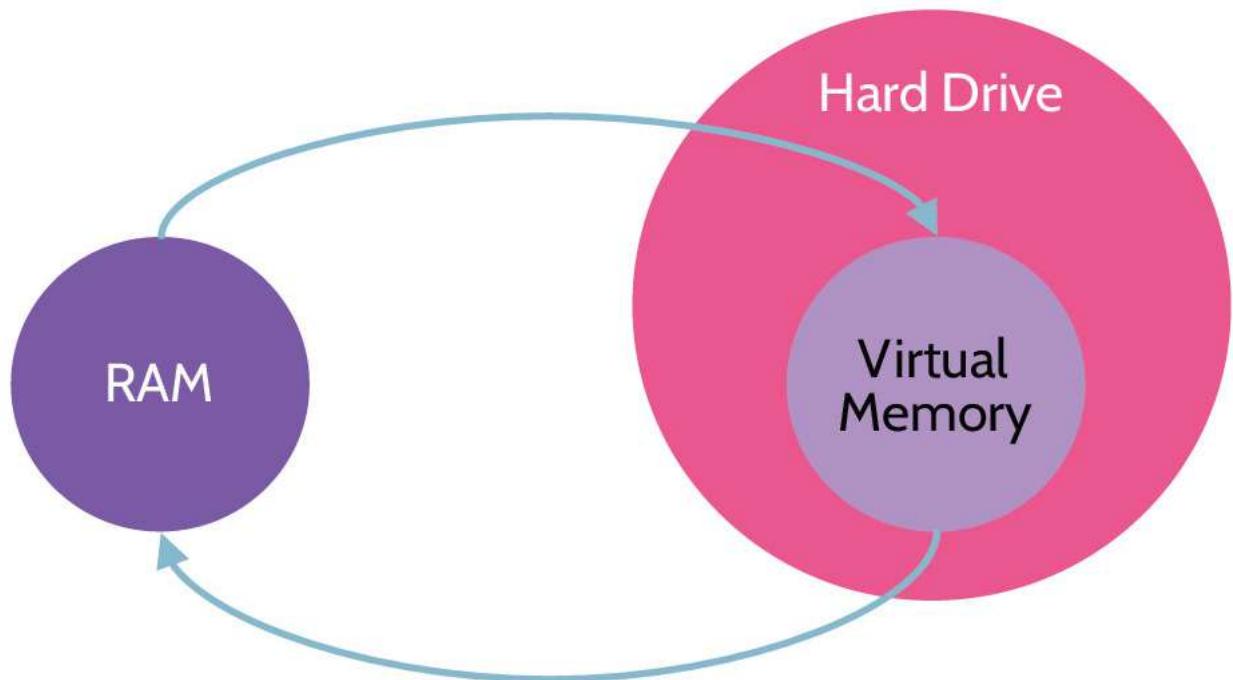


Figure 3-14: Virtual Memory

[Figure 3-15](#) depicts the major differences between primary (volatile) and secondary (non-volatile) storage.

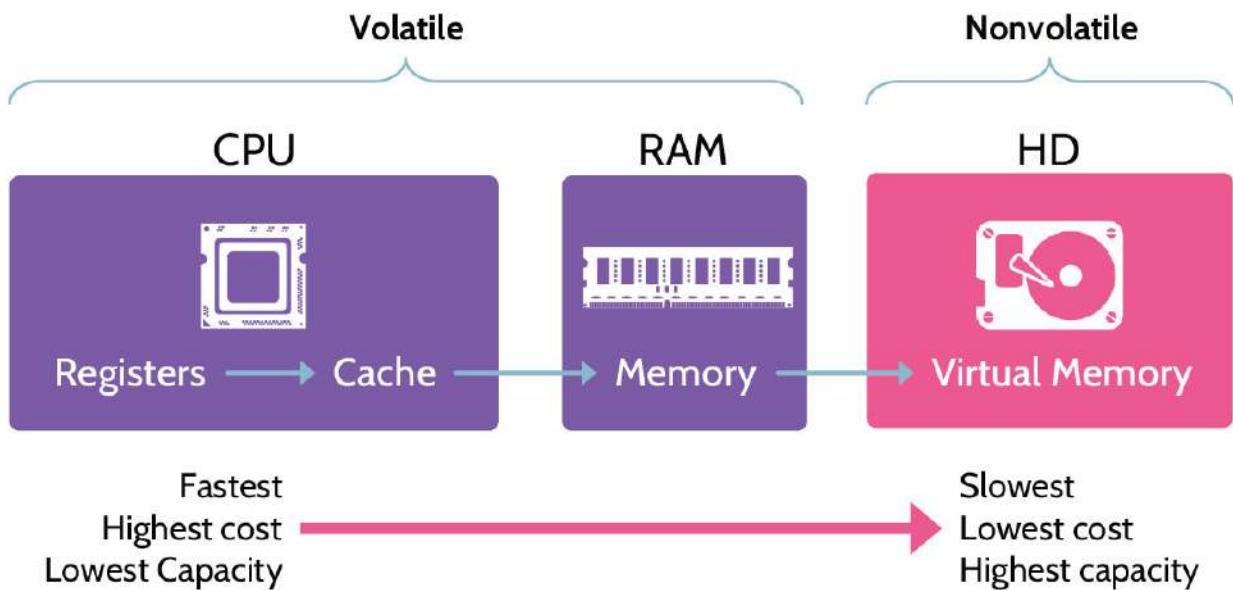


Figure 3-15: Volatile and Nonvolatile Storage



### 3.4.5 System Kernel

#### CORE CONCEPTS

- Core of the operating system that has complete control over everything in the system
- The system kernel and the security kernel are not the same thing.
- Relies on privilege levels for smooth and safe operation

The system kernel is the **core of the operating system** and **has complete control over everything in the system**. It has low-level control over all the fine details and operational components of the operating system. In essence, it has access to everything.

One important thing to note is that **the system kernel and the security kernel are not the same thing**. As noted, the **system kernel** drives the operating system. The **security kernel** is the implementation of the reference monitor concept.

From a security perspective, it's critical to protect the system kernel and ensure that it is operating correctly, and privilege levels aid in this regard.

### 3.4.6 Privilege Levels



#### CORE CONCEPTS

- **Privilege levels establish operational trust boundaries for software running on a computer.**
- **User mode results in lower trust and only allows access to a small subset of system capabilities.**
- **Privileged mode, also known as kernel mode, results in higher trust and allows access to more system capabilities.**
- **The ring protection model describes a form of CPU layering that is designed to protect critical elements of a computing system.**

Figure 3-16 depicts the relation between user and kernel modes.

Subjects of **higher trust** (e.g., the system kernel) can access more system capabilities and operate in **kernel mode**. Subjects with **lower trust** (e.g., most applications running on a computer) can only access a smaller portion of system capabilities and operate in **user mode**.

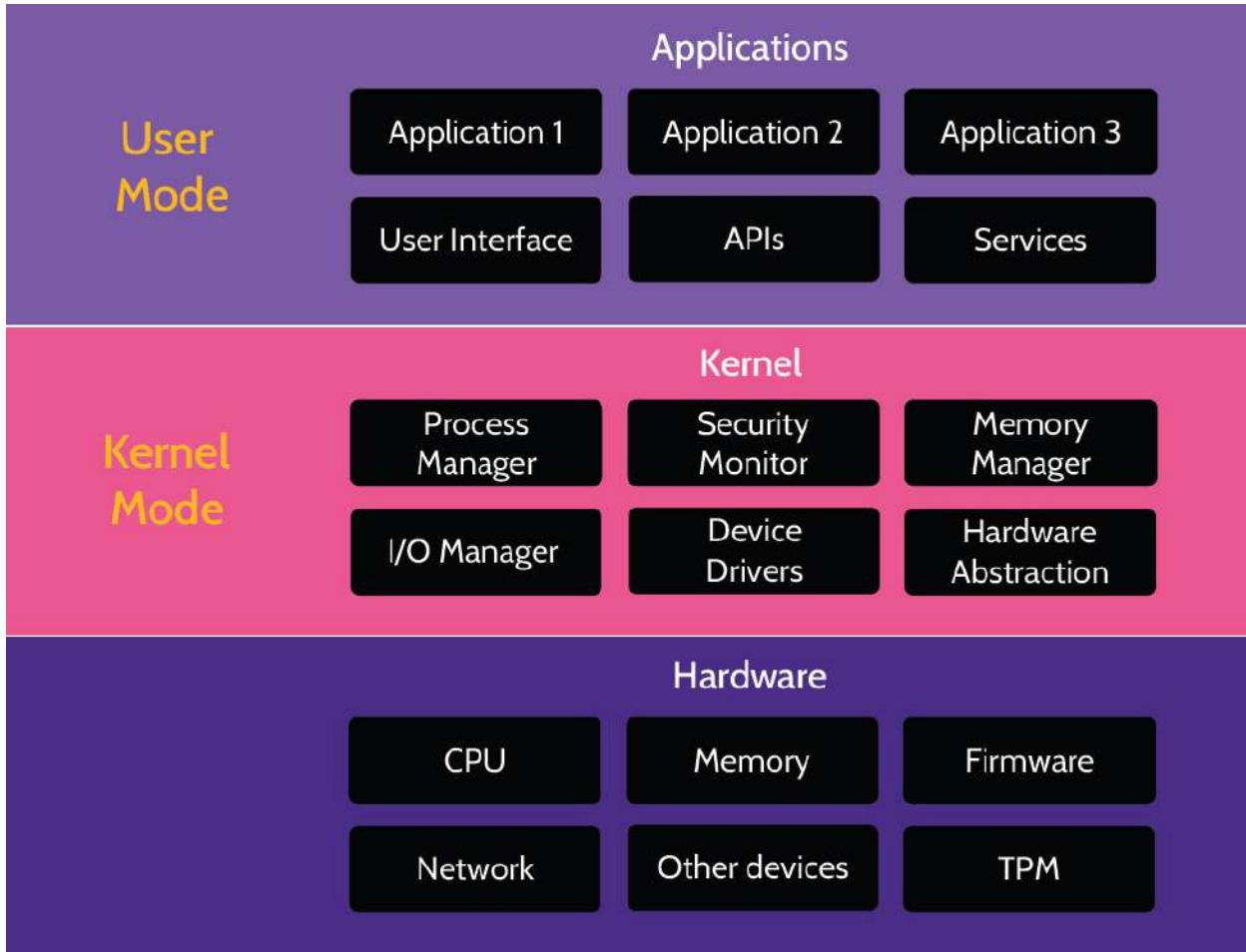


Figure 3-16: User and Kernel Modes

**Which ring is most critical from a security point of view and would likely support things like firmware and other critical systemrelated processes**

## Ring Protection Model

The ring protection model, depicted in [Figure 3-17](#), is a form of conceptual layering that segregates and protects operational domains from each other. Ring 0 is the most trusted and therefore the most secure ring. Firmware and other critical system-related processes run in Ring 0. Ring 3 (user programs and applications) on the other hand, is the least trusted and secure level, where the least access exists to protect the kernel from unwanted side effects like malware infecting the machine. The idea behind the model is that each ring communicates with the adjacent ring via system calls, and the outer rings can only communicate with the inner rings via the most trusted system calls.

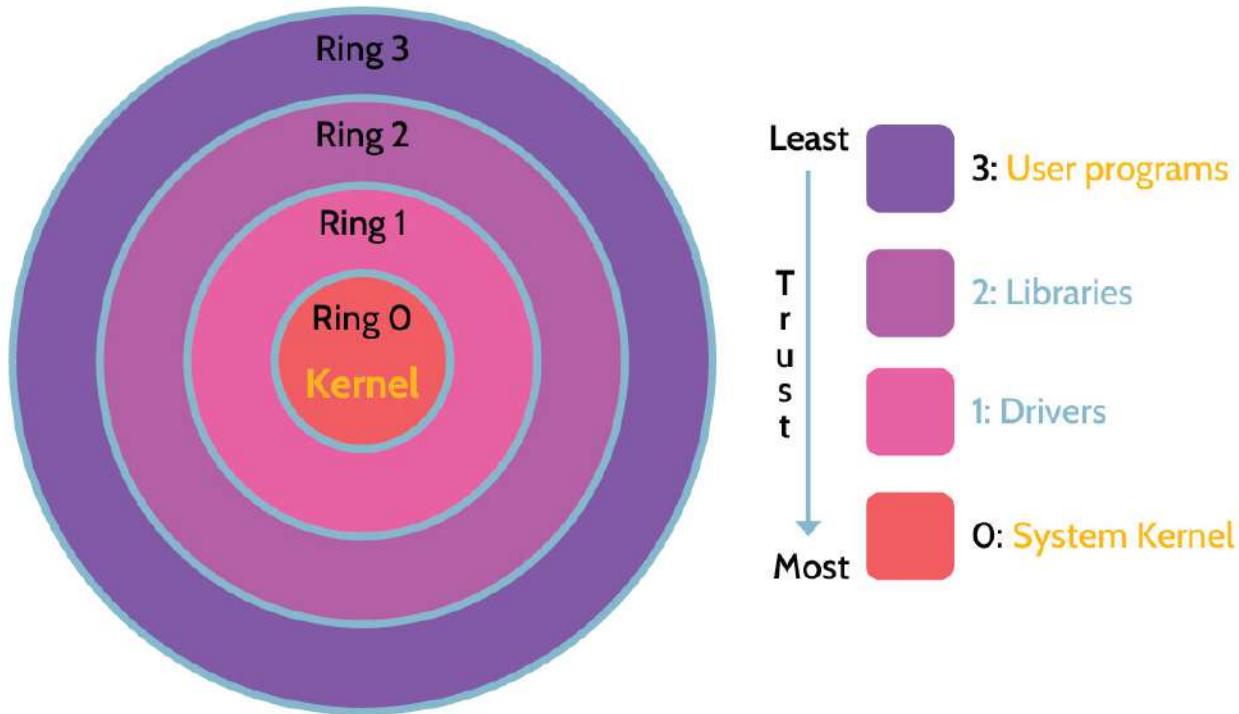


Figure 3-17: Protection Rings

## Firmware

Firmware is software that provides low-level control of hardware systems; it's the code that boots up hardware and brings it online. One of the challenges with firmware is that it is no longer hard-coded; therefore, it can be updated and modified, which makes it vulnerable to attacks. Changeable, updateable, or modifiable, firmware means that hardware itself is now vulnerable to attacks.

### 3.4.7 Middleware

#### CORE CONCEPTS

- **Middleware acts as an intermediary between two applications.**
- **Middleware is a layer of software that can speak the languages of two disparate applications and thereby facilitate communication between them.**

The idea of middleware is it's an intermediary; it's a layer of software that enables interoperability (glue) between otherwise incompatible applications. Think of mobile banking as an example. With mobile banking, a mobile device application exists that allows bank balances to be checked, funds transferred, and so on. However, what types of systems do most bank software run on? Most big banks, especially older ones, run on mainframe systems designed decades ago. When those systems were first

designed, there wasn't any concept of mobile and web-based banking. A mainframe system doesn't fundamentally understand APIs or things like web-based banking. If an older bank wants to develop a mobile application that can communicate with an underlying mainframe system, something needs to exist between the application and the mainframe to allow that. A translator—middleware—must be present. Middleware is an intermediary that allows disparate applications to communicate with each other. The mobile application speaks one language; the mainframe speaks another. Middleware speaks both languages and can thereby enable communication between two completely different systems that otherwise could not communicate with each other. In [Figure 3-18](#), the middle circle represents the software that acts as middleware, which allows a mobile application (Application A) and a mainframe computer (Application B) to communicate with each other.

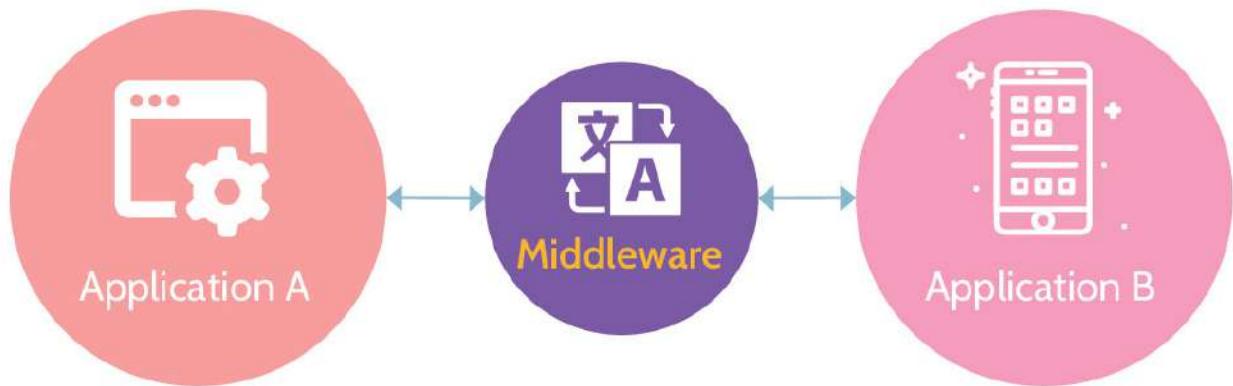


Figure 3-18: **Middleware Representation**

### 3.4.8 Abstraction and Virtualization

#### CORE CONCEPTS

- **Abstraction refers to the underlying complexity and details of a system being hidden.**
- **Examples of abstraction include driving a car and computing.**
- **Virtualization extends the computing example further.**

## Abstraction

Abstraction is a concept that is used extensively in computing. It is an idea that the underlying complexity and details of a system are hidden. Think of driving a car. To the driver, the act of driving involves simple steps that essentially boil down to inserting a key or pushing a button (newer models), putting the car in gear, and driving—using the steering wheel, accelerator, and brakes as necessary. However, while all of this is happening, a significant amount of abstraction is taking place. A driver does not need to worry about how the engine works, or the hydraulic or electrical system, or any other

components that ultimately make a car run. All this underlying complexity has been abstracted from the seeming simplicity of hopping in the car and driving away.

Abstraction is also used in programming. CPUs, at their core, understand 1s and 0s. From a human perspective, however, 1s and 0s are very hard to understand, and over the years numerous iterations of programming languages have evolved and abstracted the complexity of computing to human-readable form.

## Virtualization

Carrying the concept of abstraction further, **virtualization is the process of creating a virtual version of something to abstract away from the true underlying hardware or software**. Specifically, to facilitate virtualization, a hypervisor is employed. A hypervisor serves as a layer of abstraction between underlying physical hardware and virtual machines (VMs).

### 3.4.9 Layering/Defense-in-Depth

#### CORE CONCEPTS

- **Protection of an asset is best accomplished through the implementation of multiple control layers.**

Another important concept is the concept of layered defense or defense-in-depth. What this simply means is the protection of a valuable asset should never rely on just one control. If that control fails, the asset would be unprotected. Instead, multiple control layers should be implemented, and the control at each layer should be a complete control—a combination of preventive, detective, and corrective controls. Let's look more closely at this in the context of an example and while also considering the depiction of defense in depth provided in [Figure 3-19](#).

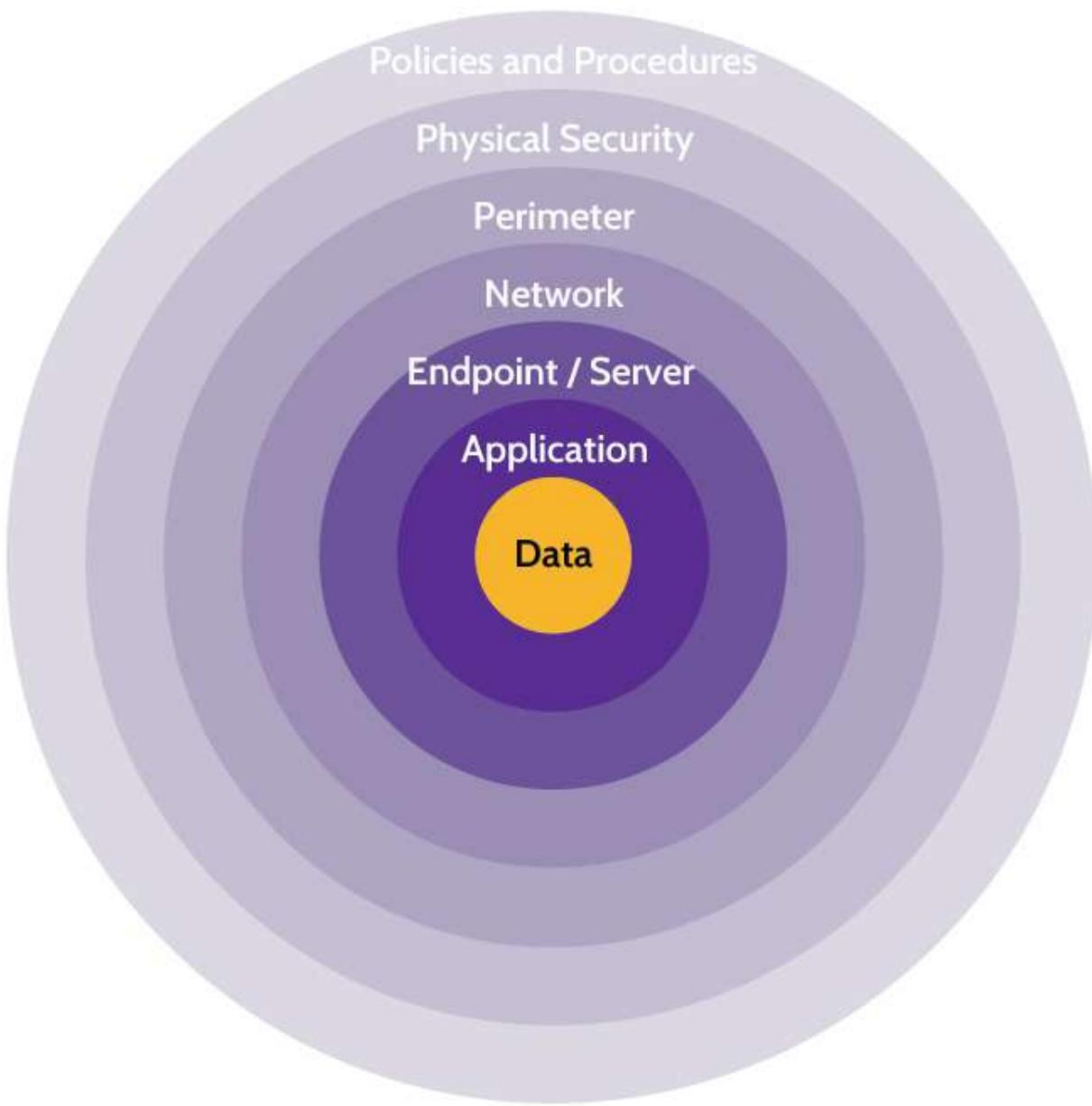


Figure 3-19: **Defense in Depth**

How many layers of defense does a company use to store and protect research and development information? Undoubtedly there's a fence around the building that could be serving as the first layer of defense. In addition, there might be CCTV cameras, and the fence could even be electric. The combination of these controls is preventive, detective, and corrective, thus constituting a complete control. After the fence, guards might be patrolling the area regularly. Next is the perimeter of the building. Similar to the fence, a combination of preventive, detective, and corrective controls exist. Walls, more cameras, and security guards collectively act as another layer of defense. Once inside the building, interior walls and locked doors are in place. Let's imagine that all these controls have somehow been bypassed, and someone reaches the computer system or the server room where highly sensitive data is stored. The system would need to be logged on to, and the related files would need to

be identified and then unencrypted. As should be clear, there is a combination of preventive, detective, and corrective controls at each layer. This is the concept of layered security, where multiple layers of controls exist. If there's a failure at one layer, controls at other layers can effectively protect whatever valuable asset, like sensitive research and development data.

### 3.4.10 Trusted Platform Modules (TPM)

#### CORE CONCEPTS

- A trusted platform module (TPM) is a piece of hardware that implements an ISO standard, resulting in the ability to establish trust involving security and privacy.
- A TPM is an independent component of a computing system and functions similarly to a black box.
- Binding and sealing are important elements that help a TPM maintain integrity.

A TPM is a chip that performs cryptographic operations like key generation and storage in addition to platform integrity. For example, when a machine boots the TPM can be used to identify if there has been any tampering of critical system components, in which case the system wouldn't boot. So, a TPM is a piece of hardware—usually installed on the motherboard—that incorporates the international standard denoted by ISO/IEC 11889 on computing devices, like desktop and laptop computers, and mobile devices, among others.

In many ways, a TPM is a black box, meaning that commands can be sent to the TPM, but information stored within the TPM cannot be extracted. TPMs do not rely on an operating system or components external to the device for processing instructions; they have their own internal circuits and firmware. Furthermore, every TPM chip is unique because a unique and secret endorsement key is burned into the chip during production. An endorsement key is a special purpose RSA key that remains hidden and can only be used for encryption, which allows for TPM authentication.

Computers that contain a TPM can create cryptographic keys and encrypt them—using the endorsement key—so only the TPM can be used for decryption. This process is known as **binding**, and it helps protect the key from being disclosed. Computers can also create a bound key that is also associated with certain computer configuration settings and parameters. This key can only be unbound when the configuration settings and parameters match the values at the time the key was created. This process is known as **sealing**, and it refers to associating the key to the TPM. Binding and sealing are particularly important as it relates to maintaining the integrity of the TPM host computing device.



- Binding – A cryptographic operation in which data is encrypted in such a way that it is tied (bound) to a specific TPM's hardware and software configuration. For example, encryption keys that are stored on a TPM can be bound to it, ensuring that keys are only accessible by that specific TPM and that the system's integrity has not been compromised.
- Sealing – A cryptographic operation that involves encrypting data. However, unlike binding, sealing is not tied to the TPM's state or configuration. Instead, sealing is used to only allow the data to be decrypted in certain conditions, such as in the presence of certain software or after user authentication. As an example, sealing can be used to ensure that certain data can only be decrypted by the TPM if a user logs into a system with the correct credentials.”

### **3.5 Assess and mitigate the vulnerabilities of security architectures, designs, and solution elements**

#### **3.5.1 Vulnerabilities in Systems**

**CORE CONCEPTS**

- A single point of failure is something that, if failure is realized, will result in negative operational impact.
- Redundancy can help alleviate the risk associated with a single point of failure, and it should be implemented where it is cost-justifiable.
- Bypass controls are a potential vulnerability, and their existence creates risks.
- The risks associated with bypass controls can be mitigated using segregation of duties, logging and monitoring, and physical security.
- Time-of-Check Time-of-Use (TOCTOU), also known as a race condition, represents a short window between two events, typically when something is used and when authorization for that use is checked.
- Frequent access or authorization checks can reduce the risk of race conditions.
- Emanations are unseen elements leaking out of systems that might reveal confidential and valuable information if captured and analyzed with the proper equipment.
- Shielding, white noise, and control zones can prevent emanations from being captured.

## Single Point of Failure

What does the term *single point of failure* mean? To answer this question, let's examine [Figure 3-20](#).

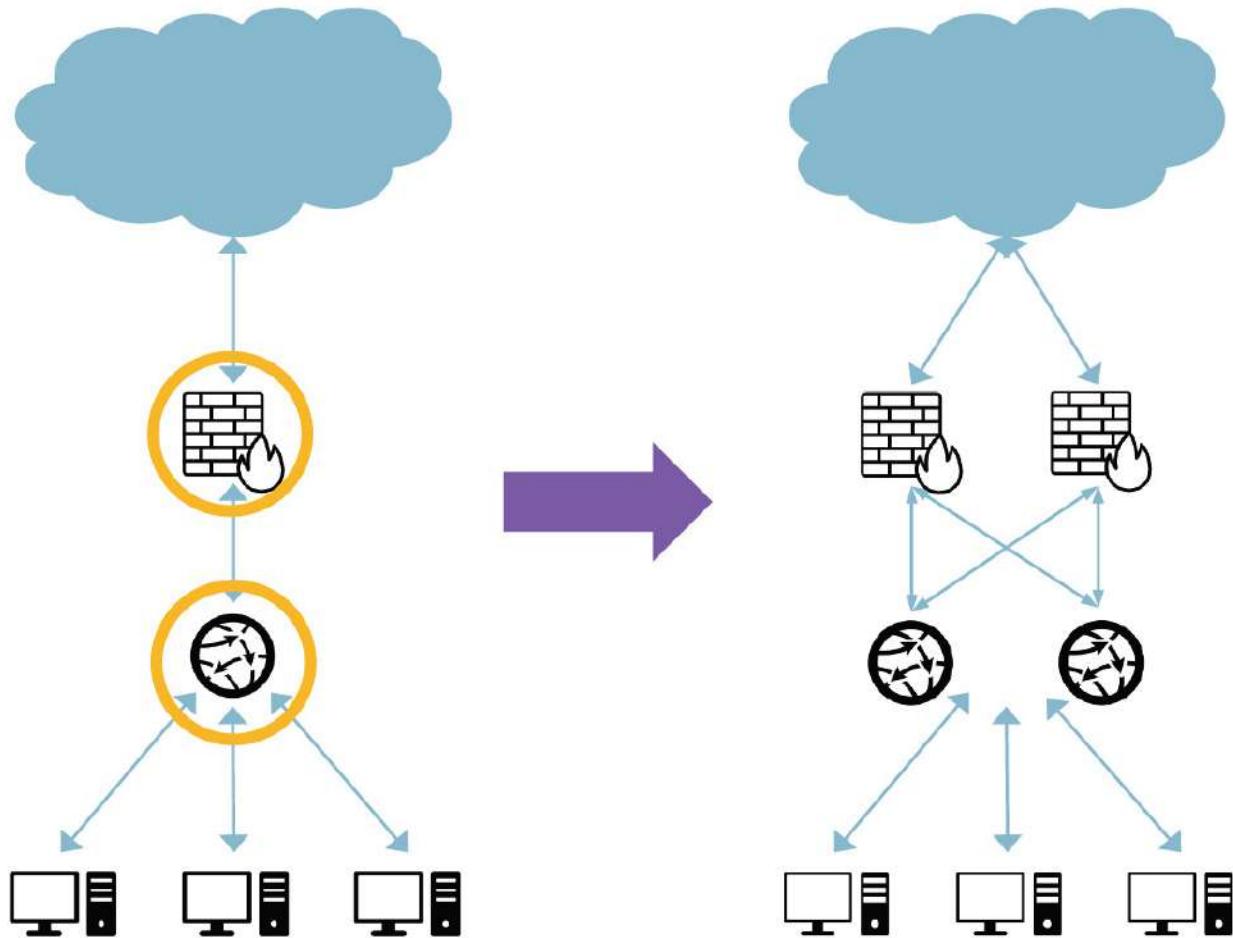


Figure 3-20: Single Point of Failure

The cloud represents the internet. Below it, the brick wall with the flame represents a firewall. Next, the ball with arrows pointing at every direction is a router. Finally, below the router are several computer systems. **What are the single points of failure in the diagram?** In this example, the firewall and the router are each considered a single point of failure; if *either* device fails, the connection to the internet is broken. In other words, a single point of failure means that when a single device or connection fails, it impacts the entire architecture.

## Reduce Risk of Single Point of Failure

Single points of failure can become very dangerous for any organization and need to be dealt with accordingly, usually by implementing *redundancy*. Looking at the previous example, two firewalls and two routers could be installed to create redundancy and mitigate the risk of single points of failure. Each pair can be configured in what is known as “high availability” so that if firewall 1 fails, traffic can be rerouted through firewall 2; if router 1 fails, traffic can be rerouted through router 2. However, one point that’s critical to keep in mind is that redundancy may not be feasible everywhere because it can be very costly. Firewalls and routers are expensive but often necessary. However, a solution like this should only be implemented where doing so is cost-justified.

## Bypass Controls

Bypass controls are a potential vulnerability or new source of risk, but *they are intentional*. Let's examine this concept through an example: You need to access the administrative settings of your home router, but for some reason you can't remember the password you set up last time you did this. Being able to perform a factory reset of that device would allow you to enter the configuration utility with default credentials and set up the device from scratch.

The reset process is a bypass control—it's intentional. In the above example, if the primary method of entering the root password fails, it might need to be reset. Bypass controls are intentional for situations just like the one described. They are intentionally built into systems, and they need to be there.

## Reduce Risk of Bypass Controls

The addition of bypass controls creates new risks. For example, if someone can gain physical access to a firewall, router, or similar system, they can reset the device. **Bypass controls are needed, and other compensating controls should always be implemented with them to mitigate or prevent their exploitation.** For example, physical security can be leveraged to protect the bypass control from being used. Only authorized people should have physical access to these devices. It's also worth noting that a bypass control isn't a covert channel because a covert channel is *unintentional* while a bypass control is intentional.

Ways to mitigate the risk associated with bypass controls include:

- Segregation of duties ■ Logging and monitoring ■ Physical security **TOCTOU or Race Condition**

What is TOCTOU? This stands for “Time-of-Check Time-of-Use” and essentially represents a short window of time between when something is used and when authorization or access for that use is checked. In other words, in that short time period, something unintended or malicious can transpire. This is also sometimes known as a *race condition*. A user or process attempts to “race in” and make changes to a system before another check to confirm that access is still appropriate. For example, assume there's 2 GB of RAM free and process 1 needs to use 1 GB. It checks and sees that is available. At the same time, process 2 needs 1.5 GB and checks and sees that it is available. A few milliseconds later, process 1 uses 1 GB, and immediately after that process 2 attempts to use 1.5 GB but is unable to, as that's not available anymore, so it unexpectedly crashes. That's an example of a race condition.

## Reduce Risk of Race Conditions

To mitigate the risk of race conditions, the frequency of access checks should increase. The more frequent the checks, the greater the frequency of re-authentication, thus reducing the overall risk. However, frequent prompts to re-authenticate could also frustrate users; systems would be very secure, but no work would be accomplished if a user needs to authenticate every fifteen minutes. So, a balance between security and functionality is important.

## What describes a race condition

### Emanations

Emanations manifest in the form of unseen things leaking out of systems, like radio or magnetic waves, light, sound, and so on. Examples of radio waves would be Bluetooth and Wi-Fi, to name a couple, while magnetic waves would include waves from hard drives and similar devices.

Emanations represent a valid security concern since any time a device is emanating, valuable data could be available that a properly equipped eavesdropper or system could collect. An example could be something as simple as shoulder surfing, where someone is looking over the shoulder of a user and reading what's on the screen, which is emanating light. Someone can look over the shoulder, see that light, and read what's on the screen. Other much more complicated and advanced ways of doing this exist too. Emanations can potentially be intercepted and unauthorized information gathered as a result. Various ways exist to protect from emanation as described in [Table 3-17](#).



WIFI



Light



Radio  
Waves



Micro  
Waves



Bluetooth



Sound

#### Shielding (TEMPEST)

Walls, Faraday cages, copper-lined envelopes, and other methods of preventing sensitive information from leaking out or being intercepted. TEMPEST is a specification that covers techniques for shielding equipment to prevent emanations from being detected

#### White Noise

Strong signal of random noise emanated where sensitive information is being processed

## Control Zones

Preventing access or proximity to locations where sensitive information is being processed

Table 3-17: Emanation Protection

**Shielding** is one of the best ways to protect against unauthorized capturing of emanations. Shielding could be as simple as putting up a wall. Walls with no windows could be installed around an office to protect from shoulder surfing or unauthorized access. More complicated shielding is needed when dealing with electromagnetic emanations, radio waves, and similar, and this is usually accomplished through use of a Faraday cage. Examples of Faraday cages could be the sophisticated installations at very secure facilities or something as simple as a copper-lined envelope that prevents emanations from things like RFID cards. These are forms of shielding, which mitigate emanations.

Another way to mitigate or prevent the pickup of emanations is using **white noise**. White noise is a strong signal of random noise emanated amid offices, where computers might be used to process sensitive information. Most emanations from computer systems are weak, and the stronger signal of random noise prevents the weaker computer signals from being intercepted.

Finally, using **control zones** to prevent access or proximity to a device is another way to prevent the picking up of emanations from being intercepted. As most emanations are very weak and only are available in a short distance, if someone can't get near the related devices, they can't intercept the signal. So, the last way to mitigate or prevent emanations is using control zones, which are basically just physical security. If different layers of physical security exist that prevent someone from getting near a device, the ability to pick up the emanations is eliminated.

## 3.5.2 Hardening

### CORE CONCEPTS

- Hardening is the process of looking at individual components of a system and then securing each component to reduce the overall vulnerability of the system.

## Vulnerabilities in Systems

Let's focus on different vulnerabilities and types of systems. This will be done in the context of mobile devices, desktops, laptops, servers, and so on. However, before moving forward, it's worth taking a step back and focusing on the fact that, at a basic level, all these devices share much in common. They all consist of hardware components (like a CPU and RAM) and all have operating systems and software applications. To protect them, they need to be broken down into components, and each component would need to be secured. Each component within a given device is secured based on value.

Organizational relevance is another term you need to be familiar with, and it indicates how valuable something is to the organization. This is a term you could possibly see on the exam, and you need to

remember that it implies value.

## Reduce Risk in Client and Server-Based Systems

Examples of hardening include doing things like disabling unnecessary services on a computer system or uninstalling software that shouldn't be there (like an SFTP server running on a user's endpoint). A service represents a small subset of code running on a system for a particular reason. Most computer systems consist of numerous running services, which directly relates to the potential attack surface they have. The more complicated the system and the more running services are present, the greater the attack surface becomes. Contrarily, the simpler and less complex the system, the smaller the attack surface will be, which makes it more difficult for an attacker to find a way into that system. Disabling unnecessary services, updating the operating system, and patching the machines are important parts of system hardening. Other ways to harden systems include:



- Installation of antivirus software ■ Installation of host-based IDS/IPS and firewall ■ Perform device configuration reviews ■ Implementation of full-disk encryption ■ Enforcement of strong passwords ■ Obtaining routine system backups ■ Implement sufficient logging and monitoring

To make hardening efficient, it's imperative that business requirements be understood. The most important question to ask is, "What is this system meant to do?" That will guide the hardening effort. If a system is supposed to act as a web server, then it shouldn't have fifty different ports open and services installed, as that heavily increases an attacker's chances of breaching it. As most systems contain a significant number of settings and configuration options that need to be considered, hardening checklists should be followed to ensure everything is set properly. Most vendors publish hardening guides for their systems. If a vendor checklist does not exist, Center for Internet Security (CIS) and similar organizations publish hardening guidelines, which are great starting points and can then be customized as needed. Each time a system is deployed, a hardening procedure should be followed, and after each hardening process the resulting configuration should be verified to confirm the system is working as expected. Ideally, the initial and ongoing verification process should be automated (especially for larger environments), but whether manual or automated, it's a critical aspect of the hardening process.

### 3.5.3 Risk in Mobile Systems

#### CORE CONCEPTS

- **Mobile device management (MDM) and mobile application management (MAM) solutions help organizations secure devices and the applications that run on them.**
- **Mobile device management solutions should particularly focus on securing remote access using a VPN and end-point security as well as securing applications on the device through application whitelisting.**

### Mobile Devices

Mobile devices are devices like iPhones, Android phones, iPads, and similar. They're small-form factor computing devices that are unbelievably powerful for their size and are typically carried in pockets and purses. The fact that they're small and powerful can allow them to store and access so much data and their mobility presents significant risk to most organizations.

### Reduce Risk in Mobile-based Systems

#### What is the primary difference between MDM and MAM?

Mobile devices are built in small sizes so their owners can carry them around easily, which is exactly why they are also often lost or stolen. This fact leads most organizations to install additional security controls on them. Mobile Device Management (MDM) and Mobile Application Management (MAM) software help organizations secure devices and the applications that run on them. For example, MDM software allows a security administrator to perform tasks like enforcing different security controls or

even wiping a device remotely, and mobile application management (MAM) software can secure applications that interact with corporate data. Note that oftentimes the two are included within a single application.

MDM and MAM can be combined with policy enforcement, application of device encryption, and related policies to adequately protect mobile devices if they are lost or stolen.

### What are ways to reduce risk associated with mobile devices and workers?

**Policies:** One of the best ways to reduce risk related to mobile devices is using policies, like: Acceptable Use, Personal Computers, BYOD/CYOD (Bring Your Own Device/Choose Your Own Device), and Education, Awareness, and Training.

**Process related to lost or stolen devices:** Typically, this involves notification of IT and security personnel as well as a means by which the device can be remotely wiped. Note that remotely wiping is dependent upon the device being connected to the internet and a savvy attacker can easily prevent this from happening.

### What is the best remote access security approach for mobile devices?

**Remote access security:** VPN and 2FA capabilities should be enabled by default, to prevent a mobile device from being used to connect to a remote network in an insecure manner (e.g., when a corporate employee is connecting to the company servers while located in a hotel, airport, or café during a business trip).

**Endpoint security:** Antivirus/malware, DLP, and similar MDM-provisioned software should be installed on mobile devices just like standard computing equipment. Additionally, the concept of hardening should be employed to minimize the potential attack surface of the devices.

**Application whitelisting:** Organizations should control which applications users may install on their mobile device through application whitelisting and not allow them to install anything not present on the approved application list.

## 3.5.4 OWASP Mobile Top 10

### CORE CONCEPTS

- OWASP Mobile Top 10
- The OWASP Mobile Security Testing Guide is a manual for mobile application security testing and reverse engineering for mobile security

**testers.**

The Open Web Application Security Project (OWASP) Foundation is an organization that is driven by community-led efforts dedicated to improving the security of software, including software and applications that run on mobile devices. Among OWASP's many substantial contributions to the security community are the globally recognized OWASP Top 10 and OWASP Mobile Top 10 lists that are based on data from a variety of sources like security vendors and consultancies, bug bounties, and numerous organizations located around the world. The key element collected in every case is the Common Weakness Enumeration (CWE) and associated software or hardware that contain the CWE. In addition to collected data, OWASP surveys members of the community to identify potential new categories for inclusion in the Top 10.

OWASP Mobile Top 10 adheres loosely to OWASP Top 10 methodology, with the focus and categories being on mobile applications. The recent OWASP Mobile Top 10 is listed in [Table 3-18](#).

<b>M1</b>	Improper Credential Usage
<b>M2</b>	Inadequate Supply Chain Security
<b>M3</b>	Insecure Authentication/Authorization
<b>M4</b>	Insufficient Input/Output Validation
<b>M5</b>	Insecure Communication
<b>M6</b>	Inadequate Privacy Controls
<b>M7</b>	Insufficient Binary Protections
<b>M8</b>	Security Misconfiguration
<b>M9</b>	Insecure Data Storage
<b>M10</b>	Insufficient Cryptography

Table 3-18: OWASP Mobile Top 10

### What are the top vulnerabilities on mobile devices?

For each identified risk, specific details about it can be found, including threat agents, attack vectors, security weakness, technical impacts, and business impacts, as well as ways to prevent or mitigate the risk. From the perspective of a security professional, this type of information is invaluable, especially

as the Top 10 list reflects a large cross section of global organizations and businesses and is updated according to industry feedback about attacks and vulnerabilities.

### What is the OWASP Mobile Security Testing Guide?

In addition to the valuable information provided in the OWASP Mobile Top 10 list, the OWASP Foundation has developed a security standard for mobile applications that helps with security testing and reverse engineering. It's called the Mobile Application Security Testing Guide (MASTG). Another interesting OWASP project is the Mobile Application Security Verification Standard (MASVS), which helps guide secure development and testing for mobile applications.

## 3.5.5 Distributed Systems

### CORE CONCEPTS

- **Distributed systems are systems that are spread out and can communicate with each other across a network. The internet is a great example of a distributed system.**
- **Distributed file systems are systems where files are spread across multiple hosts and made available via sharing across a network.**
- **Grid systems are interconnected systems that are usually working together to solve a specific and usually very complex problem.**

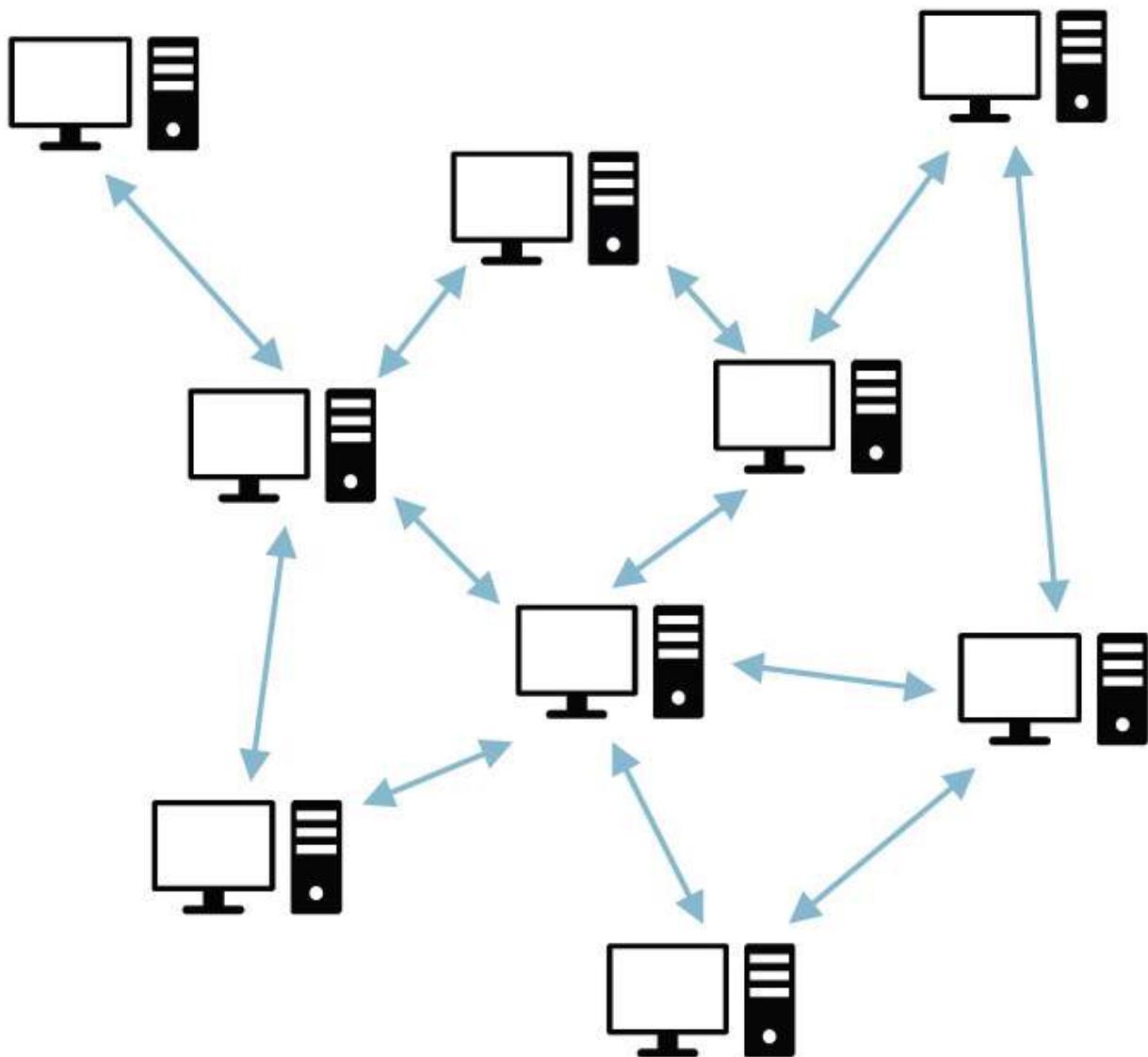


Figure 3-21: **Distributed Systems**

#### What is an underlying risk related to distributed file systems (DFS)?

Distributed systems are a number of different systems that are networked together and can communicate with each other as depicted in [Figure 3-21](#). A great example of the world's largest distributed system is the internet. A company network is an example of a distributed system. Although there is significant value in connecting the systems within an organization and then connecting the organization to the internet, there are also significant risks, such as providing a means for potential attackers to gain access to the corporate network and cause mayhem (data breaches, denial-of-service, ransomware, etc.).

Distributed file systems (DFS) take the concept of distributed systems a step further by allowing files to be hosted by multiple hosts and shared and accessed across a network. DFS software helps manage the files being hosted and presents them to users as if they're stored in one central location.

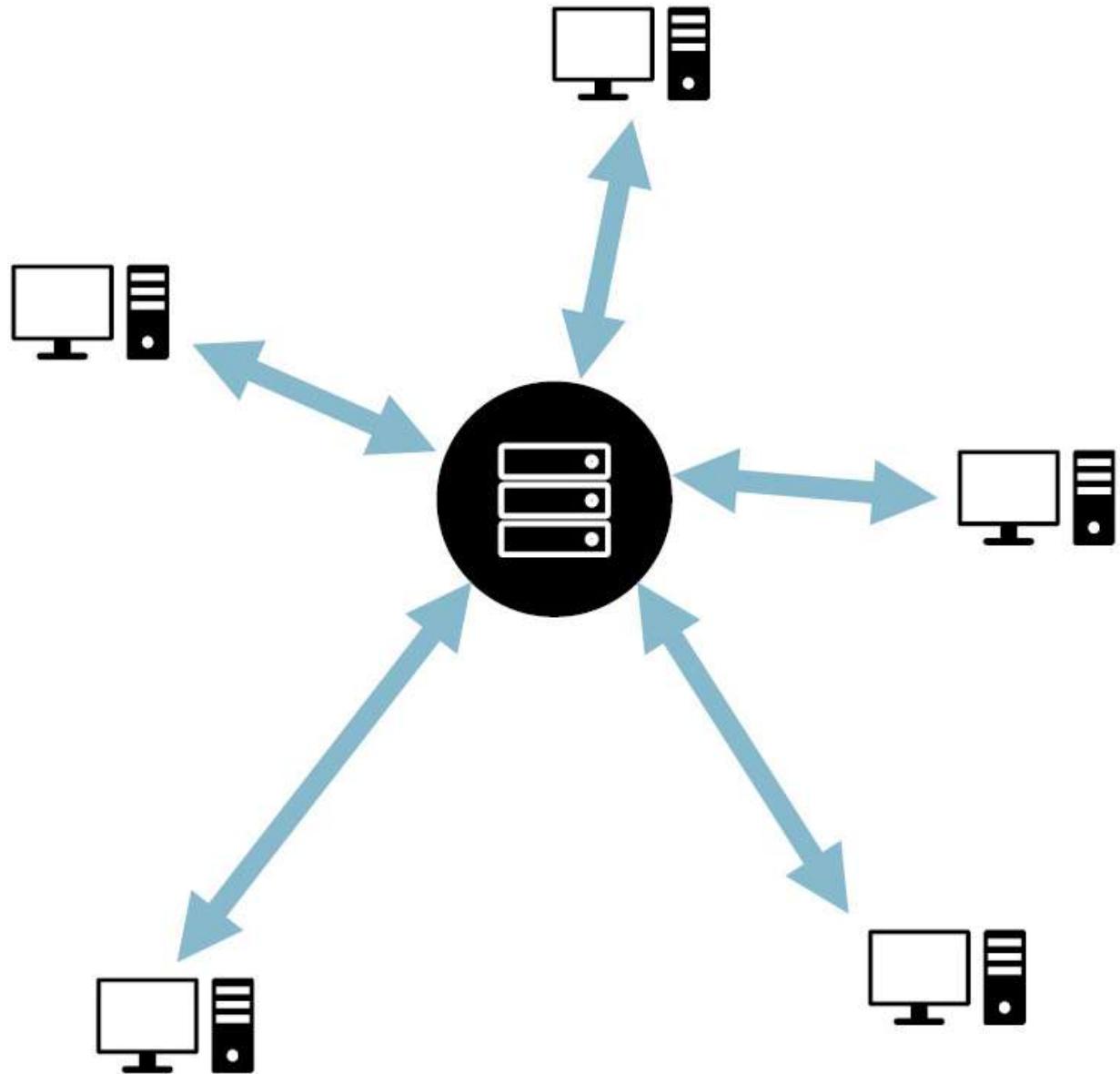


Figure 3-22: **Grid Computing**

## Grid Computing

Grid computing, depicted in [Figure 3-22](#), is like distributed systems as it still relates to systems that are connected together, but the thinking behind grid systems is that they're usually connected via a very high-speed connection to serve a greater purpose than simply passing the occasional email or file back and forth. Rather, grid systems are multiple systems working together to solve very complex problems

that require more computing power than one system can provide; so, a number of systems are interconnected into a grid and programmed to work in unison to solve difficult problems.

A very interesting example of grid computing is the Search for Extraterrestrial Intelligence (SETI). SETI is a non-profit organization with the stated mission of searching for extraterrestrial life—they were looking for aliens. As a non-profit, SETI does not have access to significant amounts of funding to support their mission. Many years ago, they started a very interesting program based upon short periods of time SETI researchers had access to radio telescopes. When a researcher was done with their research project, there might be a few hours where the radio telescope went unused before another researcher began using it. During the hours of unused telescope time, SETI would point the telescope at a portion of the sky and record significant amounts of data. Not surprisingly, SETI accumulated tons of data, but they had no way to easily process it because high-powered computers are very expensive. Within the data, SETI researchers were looking for signals or patterns that might indicate alien communications. With seemingly no solution to the large data-set analysis needed, SETI solved the problem very creatively. They created a screensaver called SETI at Home, and anybody could download and run it on their computer. When a computer was idle and the screensaver activated, it would download a small chunk of the radio telescope data and process it. While processing, the screensaver displayed a cool visualization depicting what was happening, and the idea quickly resonated with people around the world. Millions of people around the globe downloaded and installed SETI at Home on their home computer and processed these little chunks of data. And in doing so they essentially created the world's largest distributed grid computer.

## What's the security risk with grid computing?

Looking at the SETI at Home example, what happens if someone's computer sends inaccurate results? Does this skew a bigger subset of data? So, data integrity and data validation are relevant facets of the concept.

Likewise, inappropriate use of the grid computer is another. Here's an example of misuse: A group of Russian nuclear physicists were working in Siberia and had access to a mainframe system for simulating nuclear explosions and similar scenarios. After some time, they decided to stop using the mainframe for research and switched to cryptocurrency mining. Their new "research" was discovered, likely by a three-letter name Russian agency, which resulted in a permanent move somewhere even more remote than Siberia.

### 3.5.6 Inference and Aggregation

#### CORE CONCEPTS

- **Data warehouse**
- **Big data**
- **Data mining/analytics**
- **Inference and aggregation**

- Reduce the risk of unauthorized inference and aggregation

## Data Warehouse

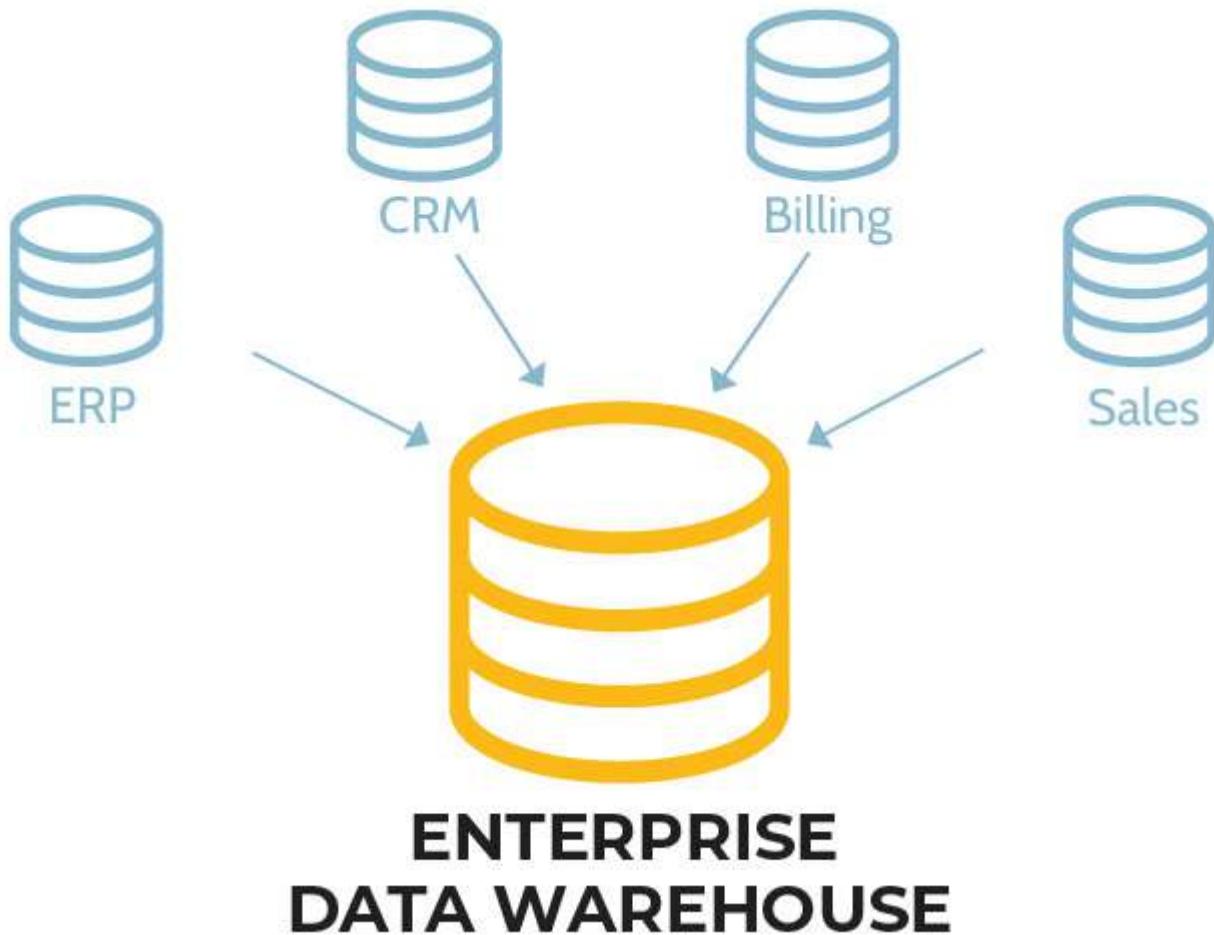


Figure 3-23: Data Warehouse

The idea behind a data warehouse, depicted in [Figure 3-23](#), is to perform data analytics from a number of different data sets, with the hope of identifying interesting bits of information. Data warehouses are not new and in fact have been around for decades. The problem with them relates to the desire to analyze data from multiple data sets that are stored in different systems; it's challenging to do this, because each data set essentially resides on its own island. A common term related to data warehouses is *data island*, and it's used in the form of a question: "Where are the data islands located?" As this question alludes, the premise of a data warehouse is that all the data from these islands is brought together in one central location. Once in one location, the data is much easier to analyze and to search for trends and other interesting nuggets of information.

## What are the security risks related to a data warehouse?

For a couple of reasons, it could be a single point of failure. The first relates to availability. If the data warehouse goes down, access to valuable data insights could be lost. The second relates to the fact that if someone gains unauthorized access to the data warehouses, they could have access to significant amounts of valuable information. So, in addition to some type of redundancy or a good backup plan, access control is an important topic to consider. When data is stored in multiple locations, fine-grained access control is easy. Only the finance people have access to finance data, HR people to HR data, and so on. When data from multiple locations is brought together in one location, managing access control becomes a much more complex issue.

## Big Data

Unlike data warehouses, which have been around for some time now, big data is relatively new and has become very popular. The gist of big data is this: data from many different locations is brought into a central repository to be analyzed. On the surface, this sounds very similar to a data warehouse. What's the difference? Three things: variety, volume, and velocity.

**Variety** means that data can be pulled from a number of different sources. In a data warehouse, only relational data can be stored, only data in a clean table format, with rows and columns. In big data, just about anything can be stored—a text file, an Excel file, a Word document, etc. This fact represents the variety that can be found within a big data repository.

**Volume** refers to the size of the data sets. With a data warehouse, storage is typically restricted to the storage capacity of a single system; with big data, storage spans multiple systems. Think about this in the context of Google, arguably the best-known big data creator and consumer. When Google first started, what did it do? It indexed the entire internet, and it did so through the creation and use of an open source tool called Hadoop. Google wanted to determine the location of everything on the internet, which resulted in an enormous data set. When a query for anything on the internet was made, Google wanted results to be returned in milliseconds. Google is so proud of query results that it displays the amount of time a search takes. Google invented a number of technologies that enabled them to store vast data sets across many servers and to be able to perform analytic searches across this massive data set very quickly. The technologies built by Google eventually ended up in Hadoop and similar tools. Google wasn't the only one to invent tools like Hadoop, but it's companies like Google that invented ways of dealing with these big data problems. This is the volume part; enormous volume sets can be stored in big data.

**Velocity** refers to the fact that data can be ingested and analyzed very rapidly in big data—even faster than is possible with data warehouses.

Examples of big data tools include Hadoop, MongoDB, and Tableau.

## Data Mining and Analytics

The primary driver behind data warehouses and big data is the desire to identify trends and other interesting insights. Through the analysis of seemingly disparate data, otherwise invisible relationships

and little nuggets of valuable information can be gleaned. These insights are typically referred to as *inference* and *aggregation*. Aggregation pulls data into one location. Inference tries to infer things; it tries to identify bits of information in the data. As the rest of this section and the examples will show, **inference, especially unauthorized inference, can create a significant risk to an organization.** Unauthorized inference can take information from the hands of key decision makers or secure systems and expose it to an entire organization, the competition, or even to the enemy.



Here's a real-life example that helps illustrate. A large retailer had been sending pregnancy-related advertisements and coupons to someone's teenage daughter without that person even knowing their daughter was pregnant. How did they know this fact? It was all about what she had purchased in the previous days and weeks. Basically, they were looking at her buying habits as they employ many data scientists who focus on identifying really interesting nuggets of information based upon consumer buying habits. They look at the massive data sets gathered from all their stores and can figure out trends and, in this case, that a particular consumer was interested in pregnancy-related items. Note that this isn't something illegal, as they were merely analyzing available consumer data. This event made national news, but they didn't stop their practices of mining and analyzing data in order to send targeted advertising; they simply grew more savvy in their approach. Now, instead of a full booklet of pregnancy-related items, a lawn mower, a pool table, and other non-pregnancy-related items are included in the advertisement to better disguise things.

Inference and aggregation can obviously cause problems for organizations. Another example where this can be an even greater issue is in a large bank. Most large banks have multiple divisions, including a brokerage arm—stock traders—as well as a mergers and acquisitions (M&A) arm, which focuses on helping companies raise money in the equities and debt markets, purchase companies, and so on. In this context, the bank's stock traders would love to know what the M&A folks are doing because they could then offer the best trade advice to their customers. But that's not allowed, as it would constitute insider trading. Banks must be very careful to prevent insider trading by preventing employees from inferring things they're not supposed to know. To summarize, [Table 3-19](#) denotes the definitions of aggregation and inference.

Aggregation	Inference
Collecting, gathering, or combining data for the purpose of statistical analysis	Deducing information from evidence and reasoning rather than from explicit statements

Table 3-19: Aggregation and Inference

**Understand the difference between aggregation and inference and how inference can be mitigated**

## Reduce Risk of Inference and Aggregation

One method to reduce the risk of unauthorized inference is using “polyinstantiation,” which allows information to exist in different classification levels for the purpose of preventing unauthorized inference and aggregation (note that more detail about this will be provided in Domain 8).

### 3.5.7 Industrial Control Systems (ICS)

#### CORE CONCEPTS

- **Industrial control system (ICS) is a general term used to describe control systems related to industrial processes and critical infrastructure.**
- **Three primary types of ICSs: Supervisory Control and Data Acquisition (SCADA), Distributed Control System (DCS), Programmable Logic Controller (PLC)**
- **Due to their inherent complexity and the things they help control/manage, ICS can be quite vulnerable to attack; the best way to reduce risk to ICS is to keep them offline—to “air gap” them from direct or indirect access to the internet.**

**Industrial control system (ICS) is a general term used to describe control systems (hardware and software) used in industrial processes and critical infrastructure.** ICSs are the computer systems and software that control things like power grids, nuclear power plants, automobile manufacturing plants, and similar systems. They are mission critical and very sophisticated with one of the inherent challenges being the specialized software they require. If the software for a power grid or nuclear power plant malfunctions, the consequences can be dire, and it may even result in loss of life. This type of software requires considerable oversight. However, because of the high level of customization an attempt to upgrade the OS, for example, may render the component non-functional. For example, if it was designed to run on Windows XP, an upgrade of that system to Windows 7 could cause the customized software to break. So, a huge challenge with ICS is they are often running on antiquated hardware and software. In addition, nobody wants to tamper with them out of fear that doing so could render them inoperable. As a result, these components that control very important and critical functions can be running on very insecure systems.

Operational technology (OT) is a broader term than ICS. It refers to the hardware and software technology used to monitor and control physical processes, devices, and industrial systems. ICS is a subset of OT that focuses on the control and automation of industrial processes.

## Reduce Risk in Industrial Control Systems

**Understand the risk associated with ICS and how best to reduce this risk**

One of the best ways to protect ICS is keeping them offline, also known as “air gapping” or creating an “air gap.” What this simply means is that ICS devices can communicate with each other, but the ICS network is not connected to the internet or even the corporate network in any way. So, even if someone does try to connect to these ICS systems from the internet or corporate network, they’ll be unable to do so.

## Patching Industrial Control Systems

### Understand the implications of patching ICS or alternative ways to mitigate risk if patching is not possible

Industrial control systems by their very nature are difficult to maintain, especially where security is concerned. Often patching of ICS has been avoided, as patching these critical systems may cause unintended consequences and downtime. However, the ubiquitous and far-reaching nature of modern networks (where organizational and ICS networks now often share many of the same pathways, and the use of “smart” technology further blurs the lines and increases potential attack surfaces) requires that ICS be patched when needed. Strong configuration management processes, good patch management and backup/archive plans, and so on should be in place and used when and where possible. When patching ICS systems is not possible (or not possible to the degree needed), additional mitigating steps can be taken to reduce the risk and impact of disruption of critical infrastructure:

- Implement nonstop logging and monitoring and anomaly detection systems to rapidly detect nefarious activities within ICS networks.
- Conduct regular vulnerability assessments of ICS networks, with particular focus on connections to the internet or direct connections to internet-connected systems, rogue devices, and plaintext authentication.
- Use VLANs and zoning techniques to mitigate or prevent an attacker from pivoting to other neighboring systems if the ICS is breached.

Additionally, privileged access management and privilege task automation tools can potentially be deployed to help manage risks associated with legacy systems often found within ICS.

The three major types of ICS that should be recognized are described in [Table 3-20](#).

SCADA (Supervisory Control and Data Acquisition)	DCS (Distributed Control System)	PLC (Programmable Logic Controller)
--	--	---

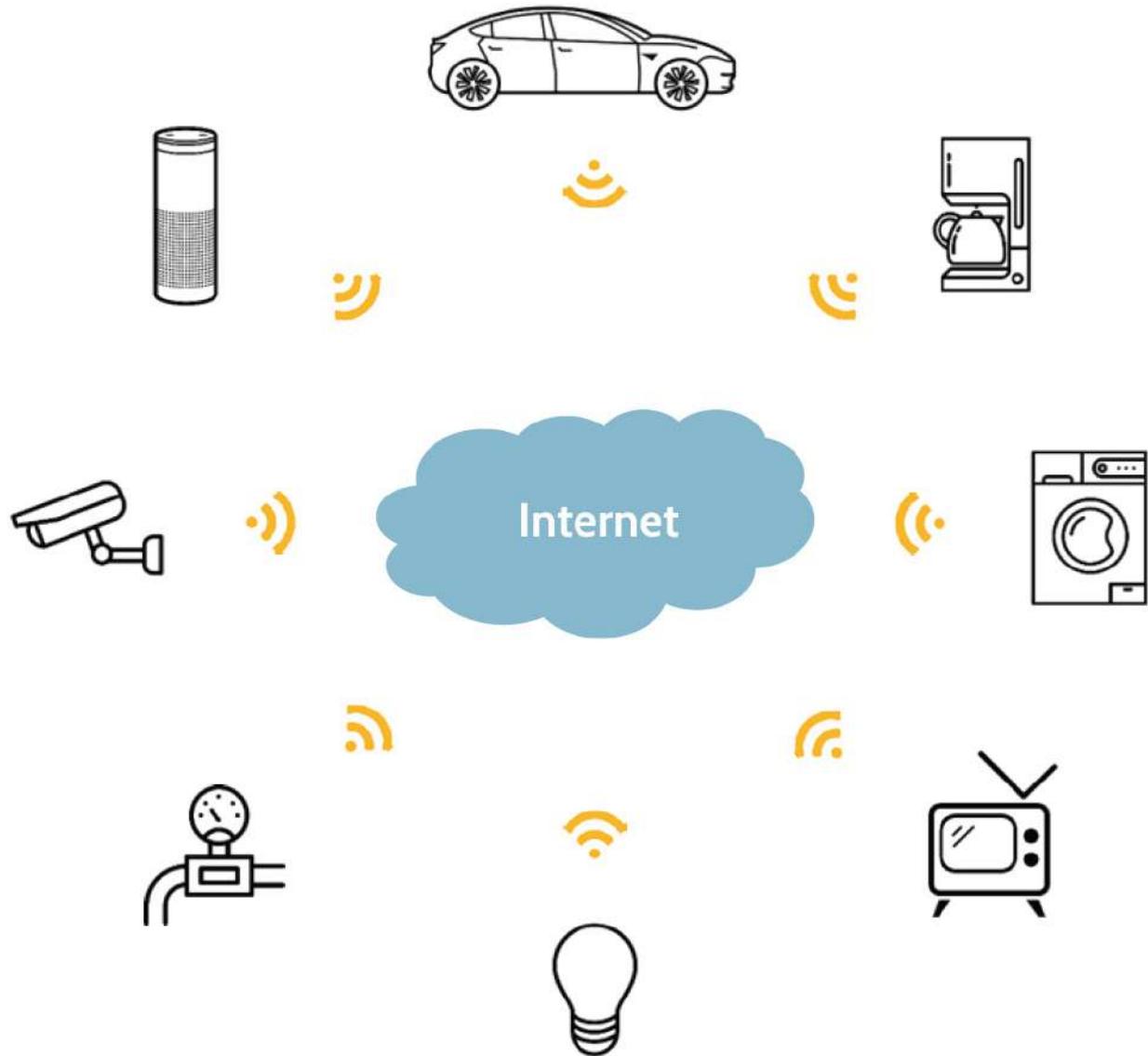
<ul style="list-style-type: none"> <li>■ System architecture that comprises computers, networking, and proprietary devices as well as graphical interfaces for management of the entire system</li> <li>■ Used to manage small and large-scale industrial, infrastructure, and facility processes</li> </ul>	<ul style="list-style-type: none"> <li>■ Process control system that monitors, controls, and gathers data from components like controllers, sensors, and other devices typically found in large processing facilities.</li> <li>■ Unlike SCADA, which includes local and remote management capabilities, DCS is typically controlled locally.</li> </ul>	<ul style="list-style-type: none"> <li>■ Industrial computer, specifically used for the control of manufacturing processes</li> <li>■ Key features include high reliability, ease of programming and diagnosis of process problems.</li> <li>■ Often networked with other PLC devices and SCADA systems</li> </ul>
--	--	--

Table 3-20: ICS Types

### 3.5.8 Internet of Things (IoT)

#### CORE CONCEPTS

- **Internet of Things (IoT) refers to all the devices, like home appliances, that are connected to the internet.**
- **IoT devices, by their nature, are risky. Reducing their risk involves making different purchase decisions, taking every precaution when installing and keeping the technology up to date.**



The term *Internet of Things (IoT)* refers to the multitude of devices that are connected to the internet and probably shouldn't be connected to it. IoT is the concept that tiny processing computers, the network cards that allow you to connect a device to the internet, have become so cheap that manufacturers are putting them in everything, including refrigerators, washing machines, dryers, cars, toasters, and so on. It seems that some type of network functionality is being installed in most similar devices these days. What's the risk here? Manufacturers are installing computer and networking technology into the things they build, but the technology being installed is mass-produced; the computer and networking technology is made very cheaply and contains little to no security. After a refrigerator is purchased, how often is built-in technology upgraded? How often is the firmware upgraded on a fridge? The answer to both questions is likely "Never." Nobody thinks about patching their toasters, so the issue persists and grows; very insecure devices are being installed in hardware with a refresh cycle of years. More to the point, very insecure devices are connected to home, and even business, networks. If somebody can remotely connect to one of these devices, they could potentially

gain a foothold inside your network and use this access to pivot to bigger targets, like personal and business computers. This is a very real security problem.

Let's briefly touch on the topic of distributed-denial-of-service (DDoS) attack in the context of IoT. A DDoS attack takes place when many computer systems are harnessed together to create what's called a botnet. The botnet can then be used for malicious activities, including those aimed at only one victim. Because so much malicious traffic from so many different sources is aimed at that victim, it is commonly going to be brought offline. In 2016, one of the largest DDoS attacks in history occurred. A bug was found in security cameras and a malicious actor chose to exploit it. Think about it—millions of security cameras have been sold and installed throughout the world; they've been installed in homes, businesses, and anywhere else people might want visibility. Many of these cameras included cheap, mass-produced, highly insecure technology components mentioned earlier; thus, the cameras were IoT devices with zero security. The malicious actor identified a significant vulnerability in the camera IoT technology and proceeded to install malware on millions of cameras around the world; a botnet of massive proportions was created and then used to perpetrate one of the largest DDoS attacks in history. Security cameras and massive DDoS; ironic to say the least, and this highlights the primary problem with IoT devices. They are very insecure and often used in devices that aren't replaced frequently, and owners won't think to upgrade the technology. This is the challenge, and fortunately companies like Microsoft are doing research in this area, specifically focused on building secure chips that can still be mass-produced inexpensively.

## Reduce Risk of Internet of Things (IoT)

Reducing the risk associated with IoT devices isn't easy. One way would be to not use IoT-enabled devices and equipment unless you absolutely must. If you do, be very thoughtful about their installation, especially where a home or corporate network is concerned. Make sure the technology remains up to date, connect them to a segregated part of the network, and ensure adequate protection is built around it and ensure you scan that network for vulnerabilities and mitigate those accordingly.

### 3.5.9 Cloud Service and Deployment Models

#### CORE CONCEPTS

- **Characteristics of cloud computing: on-demand self-service, broad network access, resource pooling, rapid elasticity and scalability, measured service, multitenancy**
- **Cloud service models: IaaS, PaaS, SaaS, CaaS, FaaS**
- **Cloud deployment models: Public, Private, Community, Hybrid**
- **Protection and privacy of data in the cloud should be carefully considered.**

## Cloud Computing

Cloud computing allows individuals and organizations to access and use computing resources (like servers, storage, databases, networking, software, and more) over the internet, on a pay-as-you-go basis. This enables users to access data and applications from anywhere, without having to maintain physical hardware and software infrastructure themselves. It's worth noting that it's also possible to operate a private cloud where the user of the private cloud owns and operates all the hardware and software.

Six defining characteristics of cloud computing are listed in [Table 3-21](#).

<b>On-demand self-service</b>	This means when particular resources are needed, they can be provisioned immediately and automatically. Whether the resource is additional storage, more CPUs, more RAM, and so on, a cloud consumer can provision or de-provision as they see fit.
<b>Broad network access</b>	This means the cloud can be accessed from anywhere, using various types of devices, like smartphones, tablets, laptops. In fact, most applications in use today are Software as a Service and are accessed through a web browser connected to the internet.
<b>Resource pooling</b>	Resource pooling relates to sharing the three primary sources of cloud computing (processors, disk space, and the network). They are almost never directly accessible because of typically being shared—pooled—among multiple users. So, resource pooling describes the relationship between the fundamental hardware that makes up the compute, storage, and network resources—the pool—and the multiple customers that utilize those resources. This is one of the key characteristics of cloud computing that points to significant value and economies of scale for consumers and cloud providers.
<b>Rapid elasticity and scalability</b>	Rapid elasticity/scalability relates to how quickly compute, storage, and network can be increased or decreased in the cloud. Resources can be rapidly provisioned and deprovisioned—usually automatically or with just a few clicks.
<b>Measured service</b>	Measured service means the cloud provider tracks resource usage very closely, to the point that a cloud customer only pays for the resources used, measured in very small increments—minutes, or even seconds.
<b>Multitenancy</b>	A final characteristic of cloud computing is multitenancy. Multitenancy means that everybody has access to the cloud—it's open to the public—and cloud resources could potentially be shared with anybody, including malicious third parties being present as tenants on a cloud server. From a security perspective, multitenancy implies significant risk, which places

responsibility on cloud providers to implement very strong security controls and isolation between clients.

Table 3-21: Main Cloud Computing Characteristics

Note that multitenancy is not always included among the characteristics of cloud computing because many organizations utilize a private cloud. By nature, a private cloud is only accessible by a single cloud consumer. A public cloud, on the other hand, does reflect multitenancy.

## Cloud Service Models

There are three primary service models used in the cloud, depicted in Figure 3-24.



Figure 3-24: Cloud Models

**Software as a Service (SaaS)** in essence provides access to an application which is rented—usually via a monthly/annual, subscriber-based fee—and the application is typically web-based. A great example of SaaS is Office 365/Exchange 365. Rather than hosting their own email server and dealing with applications like Word and Excel in house, many organizations license the use of these applications by paying Microsoft monthly. This is Software as a Service—simply an application that is hosted in the cloud and accessed through a web browser, usually via monthly subscription.

At the other end of the spectrum is **Infrastructure as a Service (IaaS)**. To understand IaaS, think of it almost like a virtual data center. What is housed in a real data center? Servers, networking equipment, firewalls, switches, routers, storage servers, IDS, IPS, database servers, email servers, and similar devices are found in data centers. With IaaS, all these devices and equipment can be found (in the virtual sense of course). Instead of a physical firewall, IaaS would offer a virtual firewall. Instead of a physical database server, a virtual database server is used to host databases. Essentially, an entire physical data center can be presented virtually through IaaS. Devices can be interconnected, and network segments created, just like in a physical data center. This is Infrastructure as a Service.

In between SaaS and IaaS sits another cloud service: **Platform as a Service (PaaS)**. Let's imagine a customer that wants a custom application. This customer searches the marketplace, but they're unable to find existing software that meets their needs; so, they decide to create their own application and build

it from the ground up. Rather than building a development environment too, PaaS provides the infrastructure and platform upon which the application can be developed, tested, and run. Eventually, once the application is live, it essentially looks and functions like SaaS.

Two additional cloud service models that are now pervasively used are **Containers as a Service (CaaS)** and **Function as a Service (FaaS)**. Relative to the predominant service models noted above, CaaS and FaaS fit in roughly as shown in [Figure 3-25](#).

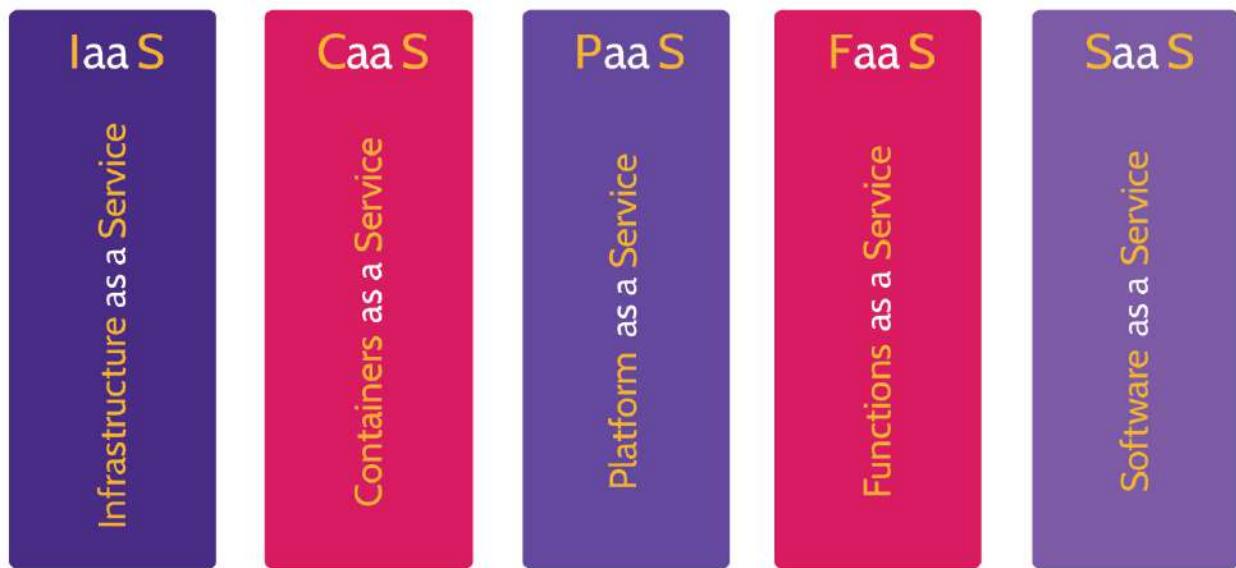


Figure 3-25: CaaS and FaaS added to IaaS, PaaS and SaaS

To understand **Containers as a Service (CaaS)**, it's first important to understand what is meant by the term container. A container is simply a package of software that contains all the components needed to run the software on any host system. Quite literally, because of this fact, a containerized application can be ported from one system to another and be up and running in a very short period of time. To make this point clearer, CaaS allows for multiple programming language stacks, like Ruby on Rails or node.js, to name a couple, to be deployed in one container. PaaS, on the other hand, requires multiple deployments, with each deployment explicitly supporting one or another stack. With the basic premise of a container being understood, CaaS automates the hosting and deployment of containerized software applications and allows development teams to focus on more important matters. In a world where software testing and updates previously took significantly longer, CaaS environments allow for agile DevOps (ideally SecDevOps) to become reality and to bring more value to customers and to the organizations utilizing this model.

To best understand **Function as a Service (FaaS)**, a couple of other terms should also be understood: serverless and microservices. The word *serverless* is often associated with information technology architectures and implies an architecture where developers don't have to worry about dealing with servers or other underlying infrastructure. They can just focus on their code. Similarly, microservices are very particular, self-contained services that provide specific business functionality. Due to their self-contained nature, they can be easily and quickly coupled to create stacks of microservices that support a business process or requirements, and they can be decoupled just as easily and quickly. With this

understanding, FaaS really describes serverless and the use of microservices to accomplish business goals inexpensively and quickly. Unlike PaaS, FaaS does not require supporting infrastructure to function. PaaS, at its core, requires servers and supporting infrastructure to be available, which costs money, even when idle. FaaS, on the other hand, only requires the underlying microservices to be present, and only when microservices are invoked are costs involved. As with other cloud-based services, FaaS can scale quickly to support growing and shrinking needs. AWS Lambda is one of the earliest examples of FaaS.

**Understand cloud service provider and cloud customer responsibilities, depending upon the cloud service model in use**

[Figure 3-26](#), the purple shading of boxes denotes cloud provider responsibility while the pink shading denotes customer responsibility.

In every case, in the public cloud, anything related to physical security, physical devices, underlying compute nodes and infrastructure, and hypervisors is going to be managed by the cloud service provider.

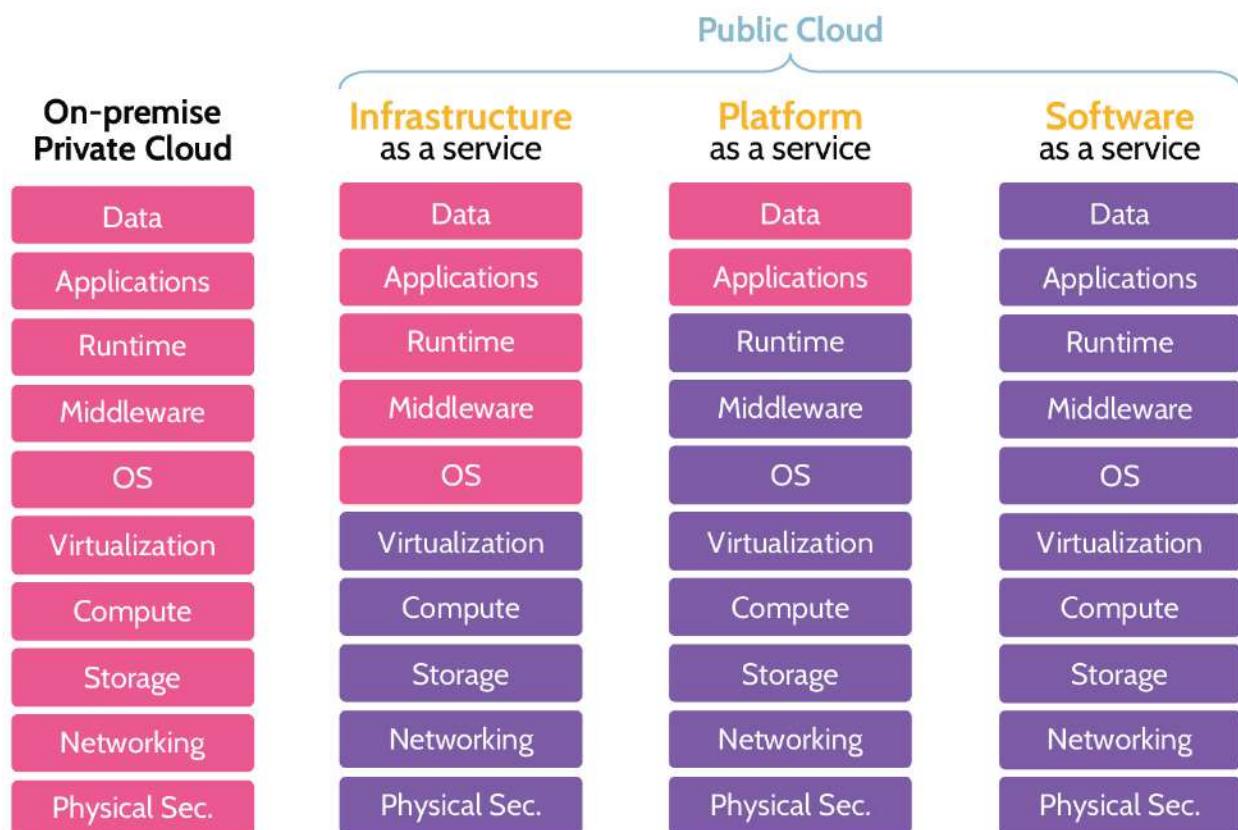


Figure 3-26: Responsibility between Client and Cloud Service Provider

Starting from the right, with **Software as a Service (SaaS)**, the purple shading implies that the cloud service provider is responsible for all the items noted in the boxes, so Data, Applications, Runtime, Middleware, and OS are all under the cloud provider's remit. This model provides the client peace of mind as they don't have to deal with any administration and just consume a service, e.g., Google Mail. With **Platform as a Service (PaaS)**, the cloud service provider is responsible for most of the items, but the client is now responsible for their applications and data. With **Infrastructure as a Service (IaaS)**, the client has the greatest degree of control over portions of the cloud environment. The client can create their own network, install their operating systems, and configure the environment exactly as needed.

However, it's important to note that even in cases where the cloud service provider appears to have the most responsibility over the cloud environment, **significant responsibility is still often shared between the cloud service provider and the cloud customer**, and this fact must be considered carefully. Especially where boundaries and therefore potential shared responsibilities exist, the cloud customer and cloud service provider must explicitly clarify who is responsible for what.

One example highlights this fact. In **SaaS**, access control is a shared responsibility. The service provider is responsible for creating the security kernel and all the security-related components that make up the environment, but the customer is responsible for creating user accounts, setting access permissions, and ongoing review and maintenance of those accounts.

Regardless of the nature of shared responsibility, one thing is always constant when talking about the relationship between the cloud service provider and the cloud customer: **the cloud customer is always accountable for their data and other assets existing in a cloud environment**.

## Cloud Deployment Models

Several cloud deployment models exist, as depicted in [Table 3-22](#), and these refer to how the cloud is deployed, what hardware it runs on, and where the hardware is located. Most of the deployment models are intuitive and easy to understand.

The first model is **public cloud**, and the name implies who can access it—everybody, the public. A public cloud is in the cloud service provider's data center and consumers are simply accessing it as a service (e.g., Gmail). It's accessible by anyone.

A **private cloud** is only accessible by a single customer, so it's private to that customer. If it's an on-premises private cloud, this means it's in the customer's own data center; but it can also be in a cloud service provider's data center, where a private, dedicated cloud is provided for use only by the customer. This would be considered an off-premises private cloud.

A **community cloud** is a cloud that is used by a group of users that share common needs or interests, like a group of hospitals, for example. One of the largest community clouds is GovCloud. It was built by AWS to be FedRAMP-compliant so that US Government agencies can use it.

The last cloud deployment model is a **hybrid cloud**. A hybrid cloud is any combination of the three previously mentioned, and it is usually a combination of a public and private cloud. For example, many

organizations will store their low-sensitivity data in the public cloud, and their high-sensitivity data in their own private cloud.

	<b>Infrastructure MANAGED by</b>	<b>Infrastructure OWNED by</b>	<b>Infrastructure LOCATED</b>	<b>ACCESSIBLE by</b>
<b>Public</b>	Third-Party Provider	Third-Party Provider	Off-Premises	Everyone (Untrusted)
<b>Private / Community</b>	Organization or Third-Party Provider	Organization or Third-Party Provider	On-Premises or Off-Premises	Trusted
<b>Hybrid</b>	Both: Organization and Third-Party Provider	Both: Organization and Third-Party Provider	Both: On-Premises and Off-Premises	Both: Trusted and Untrusted

Table 3-22: Cloud Deployment Models

## Protection and Privacy of Data in the Cloud

**What should be a primary concern of an organization considering a move to the cloud?**

Regardless of the cloud service or deployment model utilized, organizations should take every precaution to ensure that proprietary, personal, and other private information remain protected. In addition to implementing strong access controls, strong encryption practices should be used when and where necessary to properly secure this data. This is especially true when an organization makes the initial decision to move from legacy, on-premises infrastructure to that of a cloud provider. In cases like this, best practices indicate that data should be encrypted and secured locally and then migrated to the cloud.

### 3.5.10 Compute in the Cloud

#### CORE CONCEPTS

- A hypervisor, also known as a virtual machine manager/monitor (VMM), is software that allows multiple operating systems to share the resources of a single physical machine.

- A virtual machine (VM) resembles a computer, but everything is emulated using software.

As noted earlier, one of the characteristics of cloud computing is **resource pooling**, which describes the aggregation of **compute**, **network**, and **storage resources** for customer use. Cloud customers can access compute resources through:

- Virtual Machines (VM)
- Containers
- FaaS

## Hypervisors, Virtual Machines (VM), Containers, Serverless

### VMs | INSTANCES | GUESTS

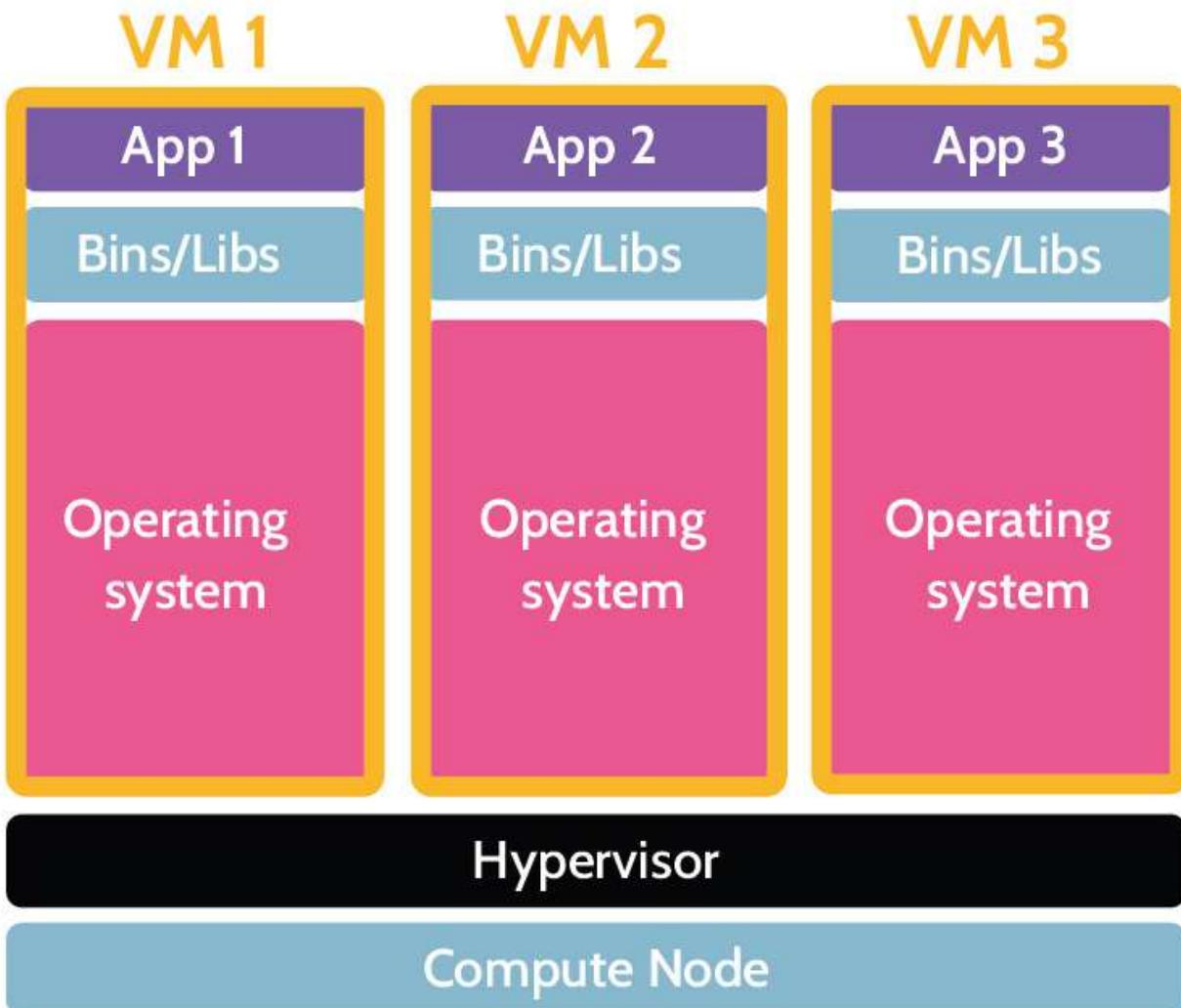


Figure 3-27: VM and Hypervisor

As illustrated in [Figure 3-27](#), a **hypervisor** is the software that runs directly on hardware or on an operating system and allows multiple operating systems to share the resources of a single physical machine, also known as a compute node. A hypervisor is often referred to as a **Virtual Machine Manager/Monitor (VMM)**, and allows administrators to manage virtual machines (VMs)—create, edit, and start/stop—as well as view VM performance and other statistics. Common hypervisor examples include Oracle VirtualBox and VMware Workstation.

**A virtual machine (VM) resembles a computer; however, everything is emulated.** Instead of an actual physical machine that contains a CPU, RAM, hard disk(s), and so on, everything is emulated using software. The result is a VM that can host an operating system and applications. In an ideal environment, specific functions would be segregated among individual virtual machines, and each virtual machine would be hardened and secured according to the value of the data being processed on the machine. This approach would make a potential attacker's job much more difficult, because multiple virtual machines would need to be attacked and compromised.

Virtual machines are commonly referred by a few different names: **Instances**, Guests and sometimes even Hosts.

### Effective and beneficial cloud environment design considerations

From a security perspective, virtual machines are very useful. Business functions can be isolated on a per-virtual-machine basis. In other words, rather than having numerous functions (services) on one machine, they can be separated, with each virtual machine supporting a specific function (e.g., a web server, or a database server, or an FTP server, etc.). This segregation and specialization allows each virtual machine to be very locked down and hardened to a specific function—greatly reducing the attack surface and the blast radius if a machine is compromised.

### The best point of attack to access multiple virtual machines

A useful capability of virtual machines is the ability to create a baseline image of a virtual machine. An **image** is essentially, a pre-built virtual machine ready to be deployed. Once an image has been created it is easy to spin up numerous virtual machines from the pre-built image.

Compromising the hypervisor would give an attacker access to the multiple virtual machines it controls, so considerable hardening should be enforced to avoid that.

## Virtual Machines versus Containers

Before looking more closely at containers, a quick side-by-side comparison of virtual machines and containers is useful to consider as illustrated in [Figure 3-28](#).

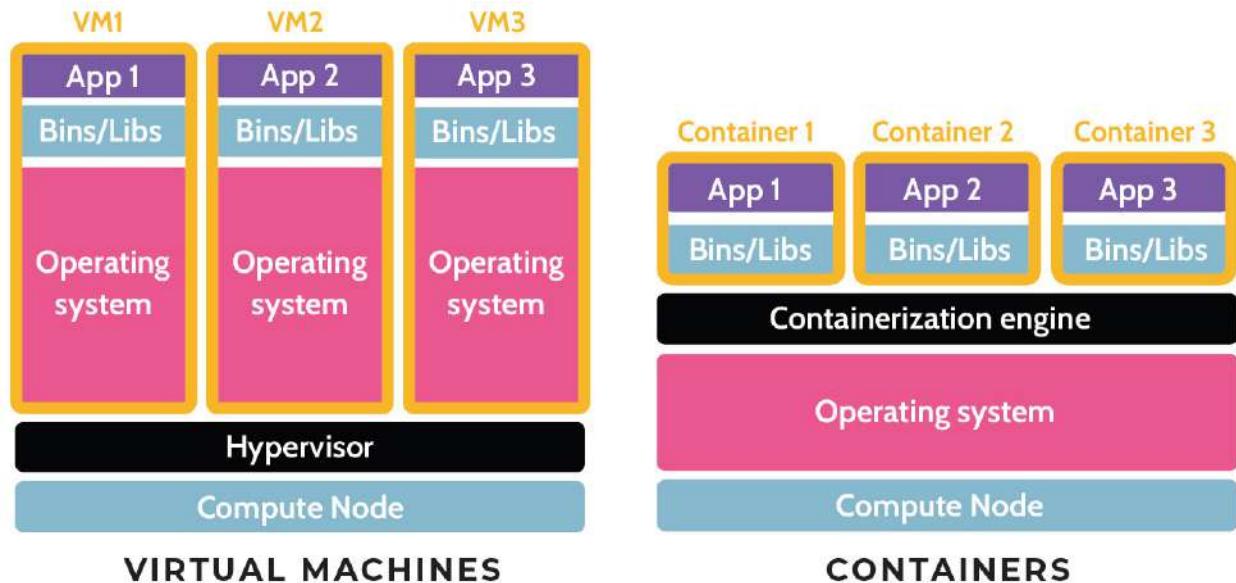


Figure 3-28: VMs vs. Containers

Virtual machines are quite “heavy” relative to containers. For every virtual machine being created, an accompanying operating system must also be installed. The hypervisor acts as the layer of abstraction that manages all the underlying physical infrastructure on behalf of each virtual machine. Containers, in comparison, are quite lightweight relative to virtual machines. Multiple containers can be supported by a single operating system, and the containerization engine acts as the layer of abstraction between the containers and the operating system. Relative to each other, containers are more efficient and portable than virtual machines. Virtual machines virtualize an entire computer system; containers only virtualize software and dependent supporting components.

Virtual machines and containers offer advantages and disadvantages relative to one another, some of which are noted in [Table 3-23](#).

	<b>Pros</b>	<b>Cons</b>
<b>Virtual Machines</b>	<ul style="list-style-type: none"> <li>■ Isolated security</li> <li>■ “Heavy” and effectively function as stand-alone computers and can be treated as such</li> </ul>	<ul style="list-style-type: none"> <li>■ As essentially stand-alone computers, working with VMs can take time in terms of modifying, building, and confirming the functionality of the VM image</li> <li>■ VMs can consume significant amounts of storage space on host systems, so initial planning and design of a VM environment must consider this fact</li> <li>■ Requires more administrative overhead</li> </ul>
<b>Containers</b>	<ul style="list-style-type: none"> <li>■ Lightweight and efficient, as they only include</li> </ul>	<ul style="list-style-type: none"> <li>■ Because containers in a given context share the same operating system and</li> </ul>

	<ul style="list-style-type: none"> <li>application software ■</li> <li>Highly portable, and many containers already exist and are available for download and use in development projects ■</li> <li>Allows developers to streamline focus on development efforts ■ Less overhead</li> </ul>	<ul style="list-style-type: none"> <li>underlying physical infrastructure, a threat to one container is potentially a threat to other containers or the shared resources.</li> </ul>
		<ul style="list-style-type: none"> <li>■ Due to many containers existing in the public domain for download and use, a malicious actor could exploit this fact</li> </ul>

Table 3-23: VM and Container Comparison

## Containers

As noted, containers are highly portable, self-contained applications, with an abstraction layer known as the containerization engine sharing and leveraging resources of that operating system on behalf of each container. As highlighted in [Figure 3-29](#), multiple containers can exist and operate in the context of one operating system. Each container supports one application and contains all the supporting binaries/libraries and other dependent components necessary for the application to run properly and to be easily ported to another system.

# CONTAINERS

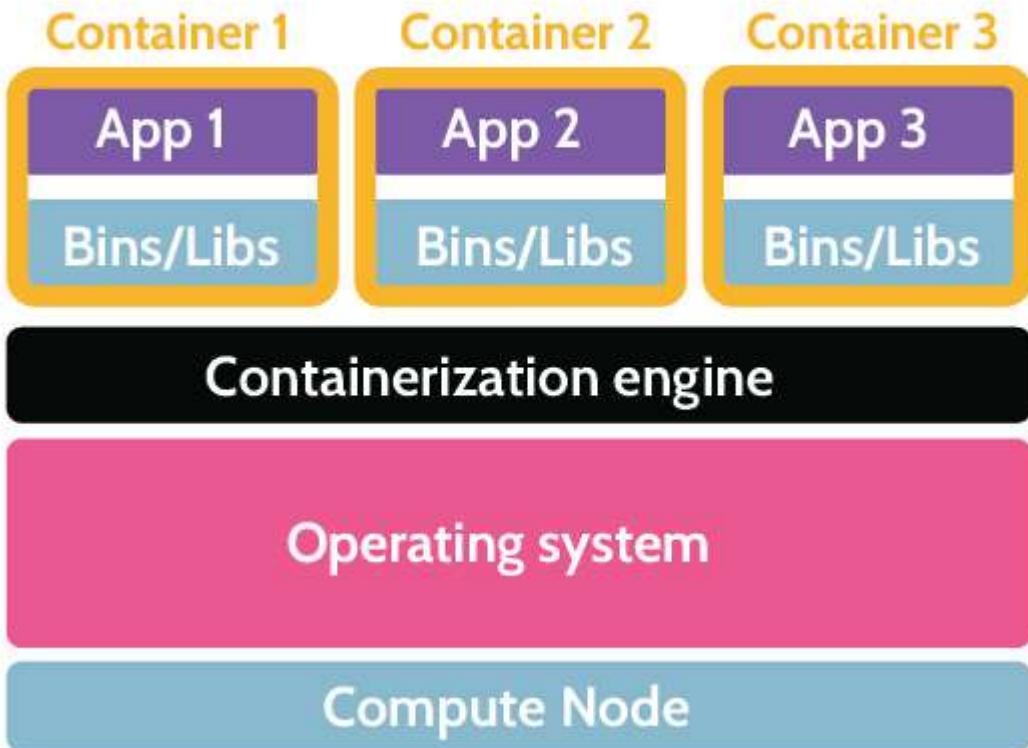


Figure 3-29: **Containers**

## Dividing Up Services

As shown in [Figure 3-30](#), all functionality of a monolithic application is wrapped together as a single unit, whereas with microservices and serverless, functionality is more defined and self-contained in smaller or individual units.

Monolithic applications typically comprise of a back-end database, an application and a user interface. Correspondingly, this implies a single large code base, and changes to an application may require updates to all three areas. In recent years, application development has trended toward the utilization of microservices, which has been further leveraged via serverless architectures found in cloud computing.

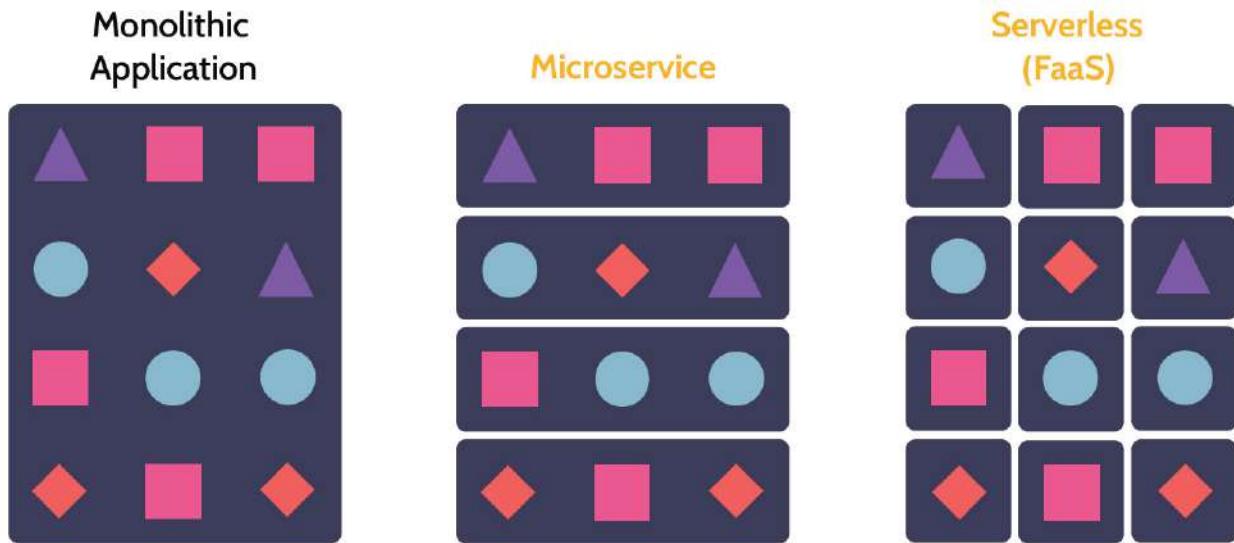


Figure 3-30: **Monolithic vs. Microservices and Serverless Computing**

## Microservices

A great way to understand a microservice is to imagine one piece of functionality found in a monolithic application operating as a stand-alone service or small set of related services—a microservice. Compared to a monolithic application that exists and operates as one unit, microservices exist and function as separate units that are loosely coupled via API calls. Due to the loosely coupled nature of these services, some of the disadvantages inherent with a monolithic application are mitigated, as each part of application functionality operates as a separate service. If one piece of functionality needs to be scaled, it can be updated and redeployed quickly. The same holds true if new functionality is needed—a new microservice can be built and deployed quickly. The fact that an application is composed of multiple, loosely coupled components allows for better overall understanding of the application, and functional components can be reused across multiple applications. Relative to monolithic application development projects, microservice components can be built more quickly and cost-effectively using much smaller development teams. At the same time that microservices mitigate many of the disadvantages found with monolithic applications, weaknesses of microservices include additional complexity, due to the distributed nature of the architecture. Connections between modules and databases particularly need to be considered, as does testing of components, as an issue with one service could potentially impact a multitude of other services due to the interdependent nature of microservice-based applications.

## Function as a Service (FaaS) / Serverless

The term *serverless* takes the basic premise of microservices—hyperfocused, independent pieces of functionality coupled together through APIs—and extends it to the cloud. With serverless, microservices are run in the cloud. AWS Lambda is a perfect example of a serverless environment. With serverless, as the name suggests, the compute service does not involve the provisioning or management of servers. This fact points immediately to cost savings. Rather, when the desired

functionality is needed, it can be invoked, and only the actual compute time required is charged. If the serverless functionality goes unused—for whatever reason—there is zero cost. Additionally, and like other potential benefits associated with cloud computing, serverless allows for high availability, scalability, and simple cost management, among others.

### 3.5.11 Cloud Forensics

#### CORE CONCEPTS

- Focus is on the forensic process in cloud computing environments
- Typically, more complex than on-premises forensic investigations
- Virtual disks and VM images are often analyzed as part of cloud forensics

With any type of investigation that involves on-premises computing equipment, the forensic process is generally straightforward and primarily revolves around securing the scene, not powering the equipment off or on (thus maintaining its original state), capturing data that may reside in volatile memory or storage areas, and making bit-for-bit copies of hard drives and other non-volatile storage devices. Where cloud forensics and investigations are concerned, the forensics process can become much more complex. Because public cloud environments involve multiple customers sharing the same physical infrastructure, including hard drives, physical access to the equipment and storage devices that may contain information relevant to the investigation is typically not possible or allowed.

#### What types of evidence a cloud forensics investigator might request

In these cases, rather than physical disks and systems, an investigator will most likely request copies—snapshots—of the virtual disk and VM images to obtain evidence and information pertinent to the investigation. A virtual disk is simply a virtual hard drive allocated to a customer from an actual physical hard drive. So, a physical hard drive in a system in a data center might have 1TB of available hard disk space, and one customer might be allocated 250GB of this space for their use. The 250GB of space—the virtual disk—would appear as a stand-alone hard drive to the customer, and it would only be available for their use. Put in bigger context, the virtual disk would be connected to a virtual machine (VM)—the system that appears as a stand-alone system to the organization and utilizes CPU, RAM, and other components of a physical system in a data center. As mentioned above, copies of virtual disks and VMs are known as snapshots. A snapshot is a “snap” of the state and data of a virtual disk or a virtual machine taken at a point in time. In essence, a snapshot is a backup of the disk or the machine. Best practices indicate that snapshot schedules should be set up as part of the host and virtual machine setup and configuration, though snapshots can also be taken on an as-needed basis. Because snapshots capture the state of a virtual machine at the time they’re taken, they can prove invaluable for the sake of an investigation. In addition to capturing data stored in non-volatile storage locations, snapshots can also capture evidence that may reside in volatile memory and similar locations on the virtual machine. As with other types of digital evidence, two bit-for-bit copies of a snapshot should be

created for purposes of forensic analysis, with the original snapshot and one copy being essentially locked up and untouched and only the second copy actually being examined.

<b>SaaS</b>	<ul style="list-style-type: none"><li>■ Consumer must rely entirely on CSP</li></ul>
<b>PaaS</b>	<ul style="list-style-type: none"><li>■ For underlying infrastructure, consumer must rely entirely on CSP</li><li>■ Consumer is responsible for any application layer code they deployed and application logging</li></ul>
<b>IaaS</b>	<ul style="list-style-type: none"><li>■ Consumers can perform forensic investigations on their VMs</li><li>■ Investigation of network traffic, access to snapshots of memory, or the creation of a hard disk images may require investigative support by the CSP</li></ul>

Table 3-24: **Forensic Data that Can Typically Be Captured by Service Model**

Table 3-24 shows the type of forensic evidence that can be acquired based on the cloud model being used. Cloud forensics presents significant additional challenges relative to those typically found with traditional digital forensics. In fact, NIST published a document in August 2020 entitled “NIST Cloud Computing Forensic Science Challenges” that summarized research in this area by members of the NIST Cloud Computing Forensic Science Working Group. Specifically, members of the research team identified nine challenge categories related to cloud forensics as outlined below:

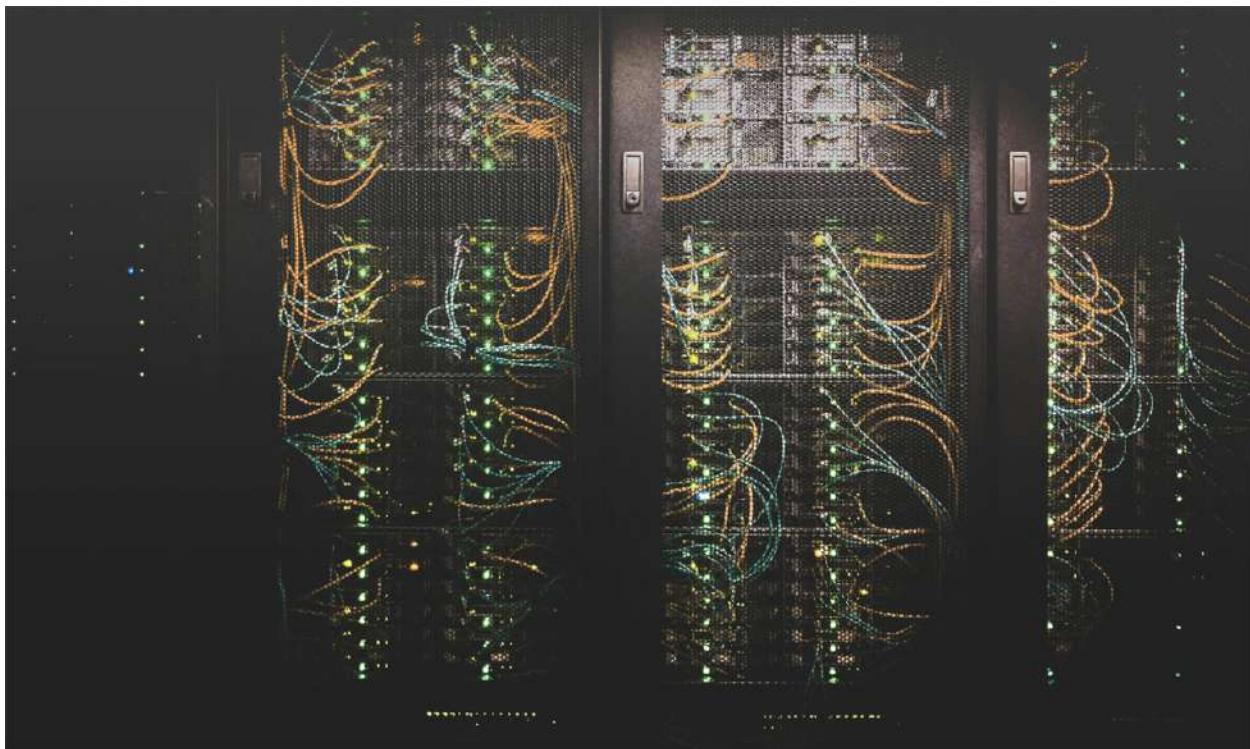
1. Architecture

2. Data collection
3. Analysis
4. Anti-forensics
5. Incident first responders
6. Role management
7. Legal
8. Standards

## 9. Training

Within each challenge category, the team further identified subcategories and specific challenges within each subcategory. For example, under the category Architecture there are several subcategories: Data Segregation, Multitenancy, and Provenance. Under the subcategory Data Segregation is a specific challenge noted as Potential evidence segregation. Under multitenancy, three specific challenges are

noted: Errors in cloud management portal configuration, potential evidence segregation, and boundaries. Looked at as a whole, sixteen subcategories exist under the nine challenge categories noted above, and within the sixteen subcategories sixty-five specific challenges are identified. Cloud forensics is a complex matter and one that no doubt will continue to pose challenges for years to come as related technologies continue to evolve.



### 3.5.12 Cloud Computing Roles

#### CORE CONCEPTS

- **Multiple computing roles relate to cloud computing: cloud consumer, cloud provider, cloud partner, cloud broker**
- **The cloud consumer is always accountable for their data stored in the cloud.**
- **Responsibility can be delegated to other cloud computing roles.**
- **Data controller = owner of data = cloud customer = accountable**
- **Data processor = processor of data = cloud provider or other agent of the customer = responsible**

## The role that is always accountable for data in the cloud

Along with the various cloud characteristics and deployment models described earlier, cloud computing involves a number of different roles, as listed in [Table 3-25](#), and it is critically important to have security in the cloud and to understand exactly who is doing what.

One of the major roles is **cloud service customer (or consumer)**. This is the person or organization purchasing cloud services. When making this purchase, the customer can't outsource accountability for any assets that are being entrusted with the cloud provider. The only thing that can be outsourced is some of the responsibility, and this is done through service level agreements (SLA) agreed upon and signed by the customer and provider. An SLA is a contract and is binding upon all parties involved, but regardless of the terms of the SLA, **accountability always remains with the asset owner**. Akin to a cloud service customer is **cloud service provider**. The provider is the organization that sells cloud services. Both role names and their meaning are intuitive.

**Cloud service broker** representatives provide service aggregation services to customers. For example, let's imagine a small or medium size company that wants to start moving to the cloud. If the organization wants to adopt the cloud to meet all its needs, it might end up contracting with many cloud service providers because different providers offer different services. For most organizations, managing all the necessary provider relationships would take a lot of work. A cloud service broker handles details like these. They work with individual cloud service providers, and put together packages of services, which they then sell to cloud service customers. They'll aggregate different services from different providers and then offer them as a package. In these cases, rather than having contracts with multiple providers, the cloud customer would only have one contract with the cloud service broker; and the cloud broker will have individual contracts with different providers.

<b>Cloud Service Customer/Consumer</b>	Individual or organization who is accessing cloud services
<b>Cloud Service Provider</b>	Organization that is providing cloud services/resources to consumers
<b>Cloud Service Partner</b>	Organization which supports either the cloud provider or customer (e.g., cloud auditor or cloud service broker)
<b>Broker</b>	Carrier, Architect, Administrator, Developer, Operator, Services Manager, Reseller, Data Subject, Owner, Controller, Processor, Steward

Table 3-25: Cloud Computing Roles

A term that relates to cloud service broker is service arbitrage. From a cloud service broker's perspective consider this: Is an individual negotiating and purchasing cloud services on their own going to get a price as good as a broker that goes to a provider on behalf of many customers? Essentially, the

broker is getting a volume discount relative to the stand-alone price, and they'll sell the service to their customers for more than what they paid, but less than what an individual customer would likely pay. The broker gets their percentage, the customer still gets a better price, and the provider gets a lot of customers. This is service arbitrage, and this is one way that cloud service brokers earn money.

### Understand the different cloud roles

Two other important roles are controller and processor. In the context of cloud services, the data controller is the cloud customer. The controller defines the rules by which data should be protected. The data processor, on the other hand, is the cloud provider. They're the party that is actually processing the data, based upon the rules defined by the data controller.

To simplify, controller = consumer; processor = cloud service provider.

## Accountability versus Responsibility

You may be wondering who maintains accountability when organizations outsource data, processes, and systems to the cloud and if accountability can be outsourced. The answer is an emphatic no! Accountability can never be delegated or outsourced as covered in detail in 1.3.2. As a reminder, accountability can't be outsourced or passed down, and it always remains with the owner. Responsibility, on the other hand, can be outsourced, and this often happens to a great extent when working with a cloud service provider. [Table 3-26](#) is the exact same table from 1.3.2 and is here as a reminder of the major differences between accountability and responsibility to help you solidify them in your mind.

Accountability	Responsibility
<b>Where the buck stops</b>	<b>The doer</b>
Have ultimate ownership, answerability, blameworthiness, and liability	In charge of task or event
Only one entity can be accountable	Multiple entities can be responsible
Sets rules and policies	Develops plans, makes things happen

Table 3-26: Accountability vs. Responsibility

### 3.5.13 Cloud Identities

#### CORE CONCEPTS

- **Third-party identity provider** is a trusted organization that manages user identities and related attributes for purposes of authentication and authorization.
- **Identity federation** involves protocols, standards, practices, and policies that support identity portability and trust relationships among unaffiliated resources and organizations.
- **SPML** enables the automation of adding users to multiple cloud services.
- **On-premise IAM solutions** include Microsoft Active Directory and LDAP based.
- **Cloud-based IAM solutions** include those offered by Amazon, Google, and many other cloud vendors.
- **Identity as a Service (IDaaS)** refers to cloud-based IAM services.

Identity and Access Management (IAM) in any context can be challenging, and especially so in the cloud, with one of the most significant challenges being that of provisioning users to multiple disparate resources spread across multiple cloud services. Security-related access control principles, like separation of duties, least privilege, and need to know still apply, therefore requiring streamlined and efficient IAM solutions.

For many years, most organizations have used on-premise IAM solutions such as Microsoft **Active Directory (AD)** and **Lightweight Directory Access Protocol (LDAP)**. Active Directory provides numerous functions, such as group and user management, permissions management, and control access to network resources. LDAP is an open and cross platform protocol used to authenticate to directory services, such as AD, and query them.

### Understand benefits and advantages of cloud-based IAM and FIM solutions over traditional IAM solutions

These traditional solutions served organizations well for years, but as cloud-based applications and the need to use them grew, so too did the pressure (and related challenges) to extend on-premise IAM capabilities outward. To resolve these challenges, **Identity as a Service (IDaaS)** solutions began to emerge that could seamlessly extend traditional IAM services to the Cloud or that could simply manage the IAM process outright. One of their primary advantages is that they provide a centralized, cloud-based system created by experts in the field, which is much easier to rely upon than a potentially decentralized, on-premise solution that requires additional resources to operate. Additionally, many IDaaS solutions offer automated account management and password management requirements are reduced. Utilizing IDaaS allows organizations to focus on their core competencies. At the same time, IDaaS solutions may be more expensive than on-premise solutions, but the trade-off of cost versus hours of labor and other resources required to maintain an on-premise IAM presence often results in substantial savings for organizations.

Federated identity, also known as federated identity management (FIM), operates in a similar context to IDaaS. At a high level, federated identity management involves protocols, standards, practices, and policies that support identity portability and trust relationships among disparate resources and organizations; essentially FIM extends the functionality of IDaaS to include multiple resources and organizations. Standards, protocols, and technologies that support FIM include Services Provisioning Markup Language (SPML), Security Assertion Markup Language (SAML), OAuth and OpenID, among others. [Table 3-27](#) provides a summary of the various identity technologies that can be used.

SPML is considered deprecated but still used and was one of the first federated standards to support and manage user access to many cloud-based services. Because SPML is a standard, organizations that utilize it are not locked into a solution, and automation of its capabilities opens the door to a broad spectrum of diverse web-based resources.

<b>SPML</b> <b>(Services Provisioning Markup Language)</b>	SPML is a deprecated XML-based, Organization for the Advancement of Structured Information Standards (OASIS) standard that was developed to allow cooperating users, resource owners, and service providers—the federation—to exchange information seamlessly for purposes of <b>provisioning</b> .
<b>SAML</b> <b>(Security Assertion Markup Language)</b>	SAML is an XML-based, OASIS standard that utilizes security tokens that contain assertions about a user. SAML facilitates service requests made by users to service providers in the form of requests to identity providers, which—if the user is authenticated/authorized—result in SAML assertions allowing the user access to the service.
<b>OAuth</b> <b>(Authorization)</b>	OAuth (authorization) is a Federated Identity Management (FIM) open-standard protocol that typically works in conjunction with OpenID (authentication). OAuth provides users and applications with “secure delegated access” via access tokens versus credentials. OAuth enables disparate resources to securely interact in a manner that ultimately allows a client to access data owned by a resource owner.

Table 3-27: **Identity Technologies**

### 3.5.14 Cloud Migration

#### CORE CONCEPTS

- Cloud migration involves benefits and risks that should be carefully considered
- One significant risk of cloud migration is vendor lock-in

**■ Security in the cloud should be understood thoroughly, and organizations should work closely with the cloud service provider to implement security that follows best practices.**

More and more organizations have either fully migrated to the cloud or are exploring options for migrating some or all their operations to the cloud. Benefits of such a move include shifting costs from a capitalization expense (CapEx) model, where networking and computer equipment is owned by the organization, to an operations expense (OpEx) model, where compute, storage, and networking costs are borne by the cloud service provider and paid for by the organization on an “as needed” basis. This shift, though not necessarily a huge cost-saver, can result in considerable efficiencies gained. Additionally, it’s in a cloud service provider’s best short- and long-term interest to provide reliable technology and support to its customers. This further shifts the load away from the organization so it can focus on core business activities.

Additionally, moving to the cloud makes applications, services, and data accessible from anywhere, using virtually any type of device, as long as an internet connection is available. This facilitates greater flexibility as well as opportunities for better collaboration among employees, vendors, customers, and other interested parties.

Migration to the cloud also facilitates the centralization of data, which can further facilitate safe storage and backup of data. Depending upon an organization’s needs, one of several cloud deployment models could be pursued.

Operating in the cloud requires high-speed access to the internet, potentially from anywhere in the world, as well as reliable backup and recovery options, to ensure ongoing availability of applications and data.

Depending on the cloud deployment model, with the exception of data and potentially of applications or services provided, organizations will have significantly less to zero control over the supporting infrastructure.

**Among Cloud migration risk, what is one of the most important things to consider?**

One of the most important considerations of cloud migration relates to vendor lock-in—the notion that once migrated, an organization is “stuck” with the cloud service provider and will be unable to move elsewhere. The possibility of this taking place requires an organization to perform significant due diligence and to analyze their needs in relation to what the cloud service provider is offering. One approach larger organizations take to mitigate vendor lock-in is to use multiple cloud service providers to support segments of their business.

Finally, security related to cloud migration must be carefully considered and addressed. Most cloud service providers provide robust security options, and it is imperative that an organization work closely

with the cloud provider to ensure best practices are being followed and implemented to secure the organization's valuable assets.

### 3.5.15 Edge Computing

Edge computing is a distributed computing approach that can reduce latency, speed up response times, and increase bandwidth availability. Instead of processing all of the data in a central data center, much of the processing is done closer to the source of the data, often on the devices themselves, or on a local server. This gives organizations faster, more timely insights, which can be beneficial in a range of different tasks. Edge computing also keeps significant amounts of traffic off the network and out of central data centers, which helps to reduce IT costs. In the cloud context, edge computing can bring down the costs of the cloud services that would otherwise have to transport, store and process this data.

Table 3.28 runs through some important concepts related to edge computing.

<b>Ingress</b>	Ingress traffic is traffic entering a network. In edge computing, ingress traffic is often generated by users who are accessing services that are hosted at the edge.
<b>Egress</b>	Egress traffic is traffic exiting a network. In the context of edge computing, this is generally data sent from services at the edge, either back to users, or to another network.
<b>Peering</b>	Peering is the interconnection between separate networks for exchanging traffic. This approach allows them to exchange traffic without going through the Internet. ISPs often have agreements between themselves to make it easier to send data to one another.

Table 3.28: Ingress, egress, and peering.

## Secure Access Service Edge

Secure access service edge (SASE), pronounced sassy, is a suite of technologies that is often looked upon as the future of wide area networks (WANs). It combines network security and wide area networking into a cloud-based service. It aims to get data and services as close to the end users as possible, while still maintaining robust security. It is an outgrowth of many other trends, such as cloud services, edge computing, and the increase in remote work, which requires new security approaches.

### 3.5.16 XSS and CSRF

#### CORE CONCEPTS

- As web-based applications continue to proliferate, so too do instances of cross-site scripting (XSS) attacks

- Two primary forms of cross-site scripting (XSS): stored/persistent XSS and reflected XSS
- An XSS attack involves a malicious script that is injected into a trusted website that a visitor's browser then downloads and executes
- A cross-site request forgery (CSRF) relies on persistence facilitated by cookies in browsers
- With XSS, the target of attack is the user's browser; with CSRF, the target of attack is the web server

The topic of assessing and mitigating vulnerabilities in web-based systems is important, because the prevalence of web-based applications only continues to grow. Web-based applications are used as a conduit between a client (e.g., a user's browser on their local machine) and an underlying information source (like a SQL database). They are becoming extremely pervasive, not only because of the growth of the cloud, but simply because this is how most organizations now deploy new systems in their environment. With this in mind, let's explore a few major web-based vulnerabilities.

## Cross-Site Scripting (XSS)

Cross-site scripting is seen most often in two forms:

- Stored/Persistent ■ Reflected/Nonpersistent **Stored/Persistent XSS**

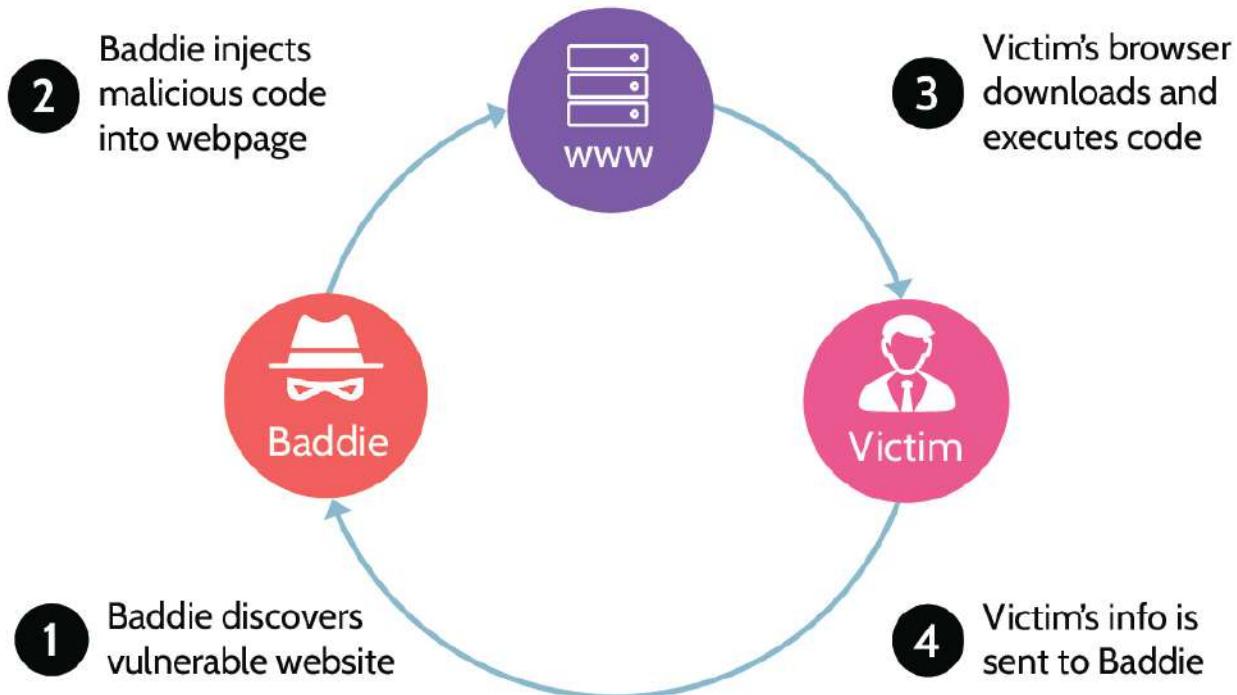


Figure 3-31: **Stored/Persistent XSS**

Figure 3-31 illustrates what stored XSS looks like. Imagine a user visits a website, like example.com, and sees that a friend has posted a picture of his family. The user comments on the picture, “Wow, what a nice-looking family!” What happens to that comment? It’s stored on the website’s servers, and anybody with permissions to view that picture can also see the comment. Now, let’s imagine that instead of writing a comment, the user writes JavaScript in the comment field. Like a regular comment, the JavaScript is stored on the website’s servers, and this time, instead of simply displaying the comment, the web browser of every person who views that picture is going to execute the malicious code that was inserted earlier. When the code executes, something malicious will take place every time that webpage is loaded. This is one form of cross-site scripting (XSS). It requires a vulnerable website that will accept this code. One of the great advantages of this attack is that it literally executes each time a victim visits the vulnerable webpage. The way that this can manifest, as depicted in Figure 3-31, is through the following sequence of steps:

1. **Attacker identifies a vulnerable website.**

The website must be vulnerable to XSS attacks for this to be successful. The attacker will typically use specific tools to examine underlying code related to pages on the website.

2. **Attacker injects malicious code into a page on the website.**

Based upon #1, the attacker can inject JavaScript on the website. This could be as simple as typing the code into a comment or a form field, with the result that the JavaScript is stored on the web server.

3. **When a victim visits the website, their browser downloads and executes the malicious code.**

With the JavaScript injected and stored on the website, every subsequent visitor’s browser will access the webpage and execute the malicious code.

**4. When the victim's browser executes the JavaScript, information from the victim will be sent to the attacker.**

Until the malicious code is cleared from the website, many site visitors could be impacted, and a savvy attacker might be able to gain username/password, banking, or other sensitive information because of the JavaScript executing in the browser.

Attackers might use XSS for any of a number of malicious reasons. For example, the JavaScript code could be used to:

- Send a copy of the victim's browser cookies, including session cookies, which could lead to session hijacking and taking over a user's active session to connect to the destination resource.

- Disclose files on the victim's computer system.
- Install malicious applications.

## Reflected/Nonpersistent XSS

Figure 3-32 shows how reflected cross-site scripting works.

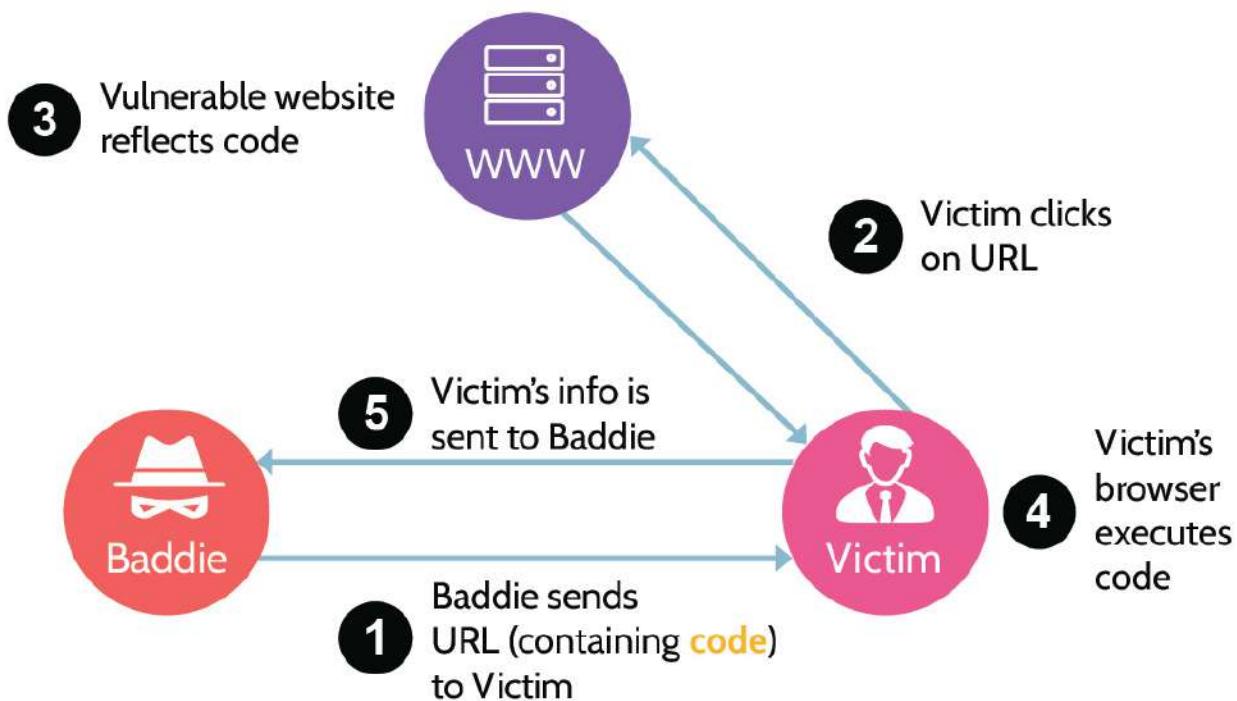


Figure 3-32: Reflected/Nonpersistent XSS

Imagine someone visits example.com, types “cat” in a search window, and selects “submit.” The webpage then returns the results of that search query, showing various pictures of cats. Now, let’s imagine that an attacker sends a malicious link to the cat lover, and they click on it. This time however, instead of searching for “cat,” the URL actually contains a search for a small bit of JavaScript code. So,

the cat lover thinks they’re searching for “cat,” but the URL contains malicious JavaScript code that the web browser is going to execute because of this.

Essentially what is happening in the background is the user is clicking on a URL, which directs them to a website, which then reflects malicious code that was sent to the website back to the victim. This can take place through the following sequence of steps: 1. **Attacker sends a malicious URL to a victim.** The URL containing malicious JavaScript code is sent to the victim (commonly via a phishing email).

**2. Victim clicks on URL.**

Because the URL looks legitimate (or through some other social engineering technique), the user clicks on the malicious link.

**3. The vulnerable website reflects malicious code to the victim.**

After clicking on the link, the URL that contains the JavaScript code is going to reflect that code back to the victim’s browser.

**4. The victim’s browser executes the reflected code.**

Because the malicious code has been reflected back to the victim, their browser is going to execute the JavaScript code.

**5. JavaScript code executes (i.e., sensitive data pertaining to the victim is sent to the attacker).**

As a result of the JavaScript executing, sensitive data from the victim will be transmitted to the attacker (or any other malicious action the code entails is performed).

It’s worth noting that the only person impacted in this scenario is the person who clicked on the malicious URL that was sent by the attacker. The next person to visit example.com and search for something will not receive the same JavaScript code. This is why it’s called “reflected,” because this attack only “reflects” back to the user who actually clicked on the malicious link. Anyone else who visits the same webpage will not be doing so with the same URL as the victim.

## Stored/Persistent and Reflected Cross-Site Scripting Takeaways

From the two examples mentioned above, a few important things should be noted.

- In the first example, malicious code is stored on the web server. This type of cross-site scripting is known as stored or persistent because the malicious code is stored on the server, and every user who visits that webpage is going to fall victim to that attack.
- In the second example, malicious code is reflected back to the victim via a carefully crafted URL. So, unlike the persistent cross-site scripting attack (which leverages multiple victims) reflected cross-site scripting targets one victim—the person who clicks on the provided crafted URL. Anyone else who visits the normal URL will not be impacted.

## Most common type of XSS attacks

Which is the most common type of attack? **Reflected** is by far the most common and often results when a link in a phishing email is clicked.

Table 3-29 provides a summary of the types of XSS.

<b>Persistent or Stored</b>	<ul style="list-style-type: none"><li>■ Injected code is stored on the server and embedded in the HTML page sent to all subsequent visitors (victims).</li><li>■ Stored XSS is <b>persistent</b>.</li></ul>
<b>Reflected</b>	<ul style="list-style-type: none"><li>■ Injected code is passed to a vulnerable server via URL and reflected to the victim.</li><li>■ Reflected XSS is the <b>most common</b> form of XSS.</li></ul>
<b>DOM-based</b>	<ul style="list-style-type: none"><li>■ Client-side Document Object Model (DOM) environment is modified, and malicious code injected.</li><li>■ Can be either persistent or reflected.</li><li>■ DOM-based XSS is intentionally not covered in detail, as this type of XSS attack is far more rare and unlikely to be covered on the exam.</li></ul>

Table 3-29: Summary of XSS Types

## What is the best way to prevent XSS attacks?

Regardless of the XSS type, one of the best ways to prevent them is using server-side input validation. Unlike client-side input validation, which could be easily manipulated by an attacker, server-side validation would remove the possibility of manipulation of the data. In addition, a web application firewall (WAF) would be another great way to prevent XSS attacks.

## Cross-Site Request Forgery (CSRF)

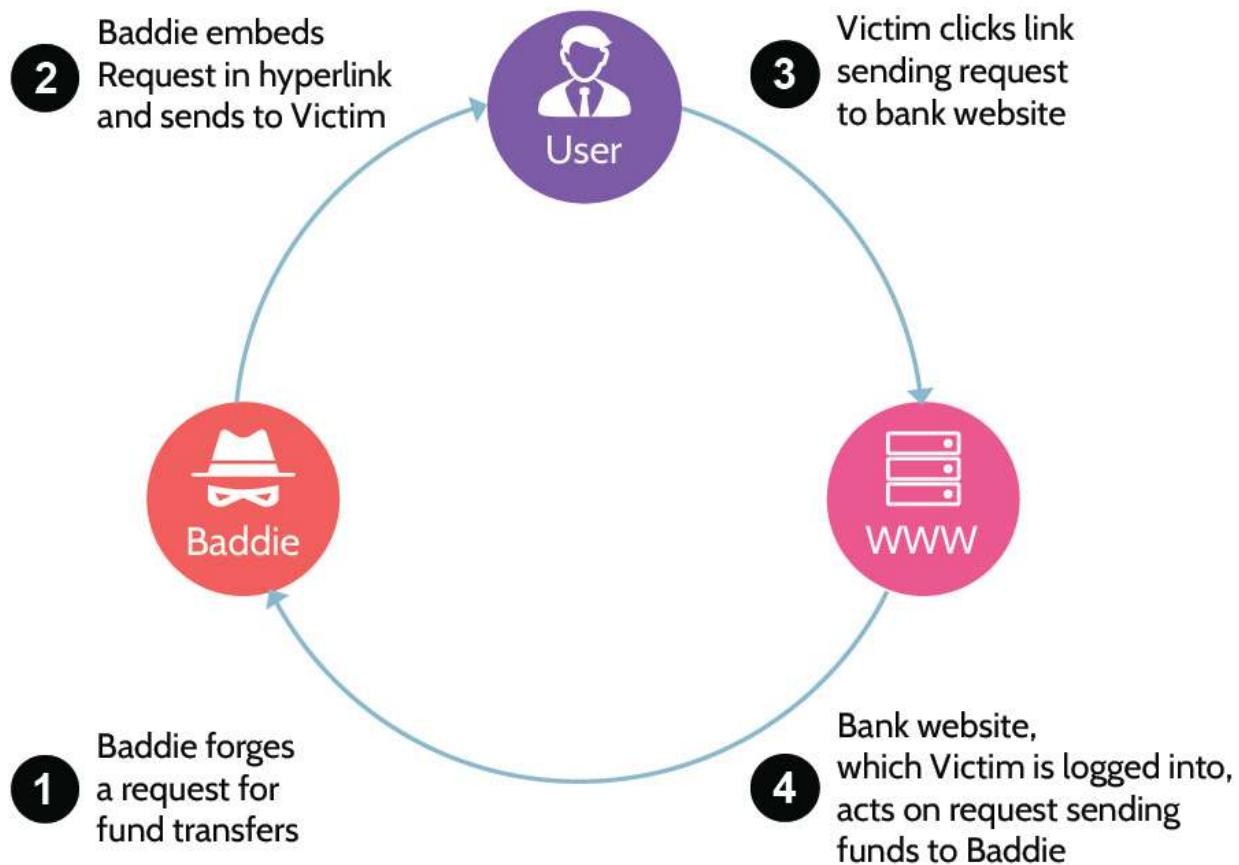


Figure 3-33: CSRF Attack

The success of a cross-site request forgery attack is predicated upon the concept of “**persistence**” that cookies in browsers facilitate. For example, when a user visits a website frequently, they often tick a box during the login process that says, “Keep me logged in for x number of days.” This information is stored in a cookie that is stored in the user’s browser. Then, when the user visits the website, e.g., their online banking portal, they’re automatically logged in because the cookie sends information to the web server that identifies the user as a legitimate system user. The attack is illustrated in [Figure 3-33](#) and follows the steps below:

1. **Attacker forges a request (e.g., a funds transfer request)**

The attacker crafts a forged request to make it look legitimate so the victim is fooled into executing it.

2. **Attacker embeds forged request in a hyperlink and sends URL to the victim (e.g., via a phishing email).**

The forged request could be sent via email, SMS, or another way that appears valid to the victim.

3. **Victim clicks link, sending the request to a legitimate entity, like their bank.**

Through social engineering or another means, the attacker can entice the victim into clicking

on the link.

#### 4. The legitimate entity, which the victim is logged into, acts on the request as requested by the attacker.

Because the victim is already logged into the legitimate entity's online portal (e.g., their bank), the funds transfer request appears valid. As a result, a malicious action can be taken, like transferring funds to the attacker's account, as the bank's web server considers this a legitimate action that is performed by the victim.

With this in mind, the attacker often has a window of time within which to operate, because the CSRF attack does not require the victim to be logged to their bank at the time of the attack. Rather, the persistence of the connection that is facilitated through the cookie can allow the attack to be successful days or even weeks later. By expiring cookies and session tokens more frequently, an attacker's window of opportunity can be considerably reduced.

### Who or what are the target of XSS and CSRF attacks?

Another important item to note is that even though the target of the attack appears to be the victim, in fact, **the target of the attack is always the server**. The victim is merely the vector used by the attacker to exploit the vulnerable server.

## XSS versus CSRF

Looking at XSS and CSRF side by side, some key differences exist as highlighted in [Table 3-30](#). **With XSS, the target of attack is the user's browser; with CSRF, the target of attack is the web server.**

XSS	CSRF
■ Unwanted action performed on the user's browser	■ Unwanted action performed on a trusted website
■ User's browser (client) runs malicious JavaScript code	■ Website (server) executes a command from trusted user's browser
■ <b>User's browser is exploited</b>	■ <b>Web server is exploited</b>

Table 3-30: XSS vs CSRF

## 3.5.17 SQL Injection

### CORE CONCEPTS

- **Structured Query Language (SQL) is the language used for communicating with databases.**
- **SQL Injection is a method of attack that utilizes SQL code and commands.**
- **Input validation is the best method to prevent SQL Injection attacks from being successful.**

Understanding SQL Injection should start with an understanding of **Structured Query Language (SQL)**. SQL is a language used for communicating with databases.

### Be able to recognize an example of an SQL Injection attack

**SQL Injection is a method of attack that utilizes SQL commands and can be used for modification, corruption, insertion, or deletion of data in a database.**

The screenshot shows a 'User Login' form with two fields: 'Username:' and 'Password:'. The 'Username:' field contains the value 'aaa' OR 1=1 --'. The 'Password:' field contains the value 'bbb'.

```
SELECT * FROM users WHERE username = 'aaa' OR 1=1 --' AND password = 'bbb'
```

```
SELECT * FROM users WHERE FALSE OR TRUE --' AND password = 'bbb'
```

```
SELECT * FROM users WHERE TRUE
```

The screenshot shows a 'User Login' form with a single message 'User Authenticated' displayed below it.

Figure 3-34: SQL injection

Figure 3-34 illustrates what SQL Injection looks like. In this case, imagine a webserver with a database residing behind it. Further imagine that the website associated with this configuration is a dynamic website, meaning that web pages can be created dynamically using data from the database, based upon user requests and interaction with the website.

Due to the dynamic nature of these websites, a persistent connection to the database is required, but a web user should never be able to directly interact with the back-end database. However, SQL Injection makes that possible.

A simple login screen is used so when a person enters their username and password, the database will be queried for the corresponding information, and if it is valid, the user should be authenticated. Using SQL Injection, however, neither a correct nor incorrect username is entered into the “Username” field; rather, a bit of SQL code is entered as shown in [Figure 3-34](#).

The first part of that—**aaa**—is just text and could be replaced by any other text, same as the entry in the password field. However, everything else following that in the username field (‘ **OR 1=1 --** ) constitutes the SQL Injection string.

Once this information is entered, the web server will formulate the request into SQL code and send it to the database server, asking if this username and password exist in the database. The first SQL statement below the login box shows how this request will be perceived by the back-end SQL database. Because of the apostrophe (‘) at the end of **aaa**, the database server treats **aaa** as the end of the username and then searches for a username **aaa**, which probably does not exist. Next, **OR 1=1** is treated as a SQL Statement, which when analyzed yields “true.” In essence the interpreter executes a logical OR query, which is true if either of the conditions accompanying it are considered true. **aaa** doesn’t exist (resulting in a false state); however, 1 always equals 1, so that returns a true state.

Finally, within SQL, the use of “--” is used to signify that everything that follows it is a comment and would be ignored by the SQL interpreter.

### What is the best way to mitigate a SQL injection attack?

So, the result is that based on the above SQL command, the attacker can successfully authenticate and gain access to the system behind the login screen. This example highlights one very important thing: the web server passed unvalidated information directly to the database server.

Unvalidated data should never be passed directly from a web server to a database server. **In other words, user input should always be validated, sanitized, or otherwise made to conform to expected formatting standards.** Additionally, the use of things like **prepared statement/parameterized queries** and **stored procedures** can also help protect against SQL Injection attacks. In short, why would you need a -- or = character to be present in a field storing someone’s name? The answer is you wouldn’t. So, input validation can help you clear the input from any characters that shouldn’t be passed on to the back-end SQL database.

The best way to understand a **prepared statement or parameterized query** is to think of a template of SQL code, where variables are used and passed to the query later. The separation helps prevent the intent of a query from being changed, regardless of the variable entered through user input or other means.

**Stored procedures** essentially operate under the same premise as prepared statements, with the biggest difference being that stored procedures are defined and stored in the database itself and then invoked in the application.

## SQL Commands

The SQL commands shown in the table below do not need to be memorized. In fact, a listing of all SQL commands would be significantly longer. These are just examples of some of the most common commands for your awareness—so you can recognize what SQL commands look like.

<b>CREATE</b>	<b>SELECT</b>	<b>GRANT</b>	<b>COMMIT</b>
<b>ALTER</b>	<b>INSERT</b>	<b>REVOKE</b>	<b>ROLLBACK</b>
<b>DROP</b>	<b>UPDATE</b>		<b>SAVEPOINT</b>
<b>TRUNCATE</b>	<b>DELETE</b>		
<b>RENAME</b>	<b>MERGE</b>		
	<b>LOCK TABLE</b>		

## SQL Code Examples

Table 3-31 shows you what SQL code looks like—again so you can recognize SQL code. Note that none of these code snippets would be used for SQL injection.

### Be able to recognize SQL commands and codes

<b>SELECT * FROM users;</b>	This command would return all of the data stored in the “users” table.
<b>INSERT INTO users (userID, password) VALUES (rob, Pass123);</b>	This command would insert a new record in the “users” table that contains a userID of “rob” with a corresponding password of “Pass123.”
<b>DROP accountsReceivable;</b>	This command essentially works like “delete” and results in deleting the table named “accountsReceivable” from the database. Let’s hope a working copy of the database backup is handy or else it may become very interesting for the database administrator.

Table 3-31: SQL Code Examples

### 3.5.18 Input Validation

#### CORE CONCEPTS

- **No input validation can lead to numerous web application vulnerabilities being exploited.**
- **Server-side input validation reduces web-based vulnerabilities and the risk of XSS and SQL Injection attacks from succeeding.**
- **Whitelist input validation only allows acceptable input.**

**What is the best way to mitigate web-based vulnerabilities and what types of attacks can be mitigated or prevented**

**Server-side input validation**—checking the contents of input fields—is one of the best ways to prevent XSS and SQL Injection attacks from succeeding. By validating data in an input field on the server side and only allowing data that meets input requirements, SQL code, and commands used in injection attacks can be prevented from running. In addition to standard input validation, which involves clearing input of invalid codes, characters, and commands, another form known as whitelist validation is often used. **Allow list (Whitelist) input validation** only allows acceptable input that consists of very well-defined characteristics, e.g., numbers, character, or both, size, or length, to name a few formats and standards.

Another approach is deny list (blacklist) input validation where malicious characters can be discarded as they are considered signs of an attack, i.e., if the = or - characters are met in a “First Name” field, they can be safely discarded as a person’s first name wouldn’t need to include = or -.

**What is the risk associated with client-side input validation?**

Contrary to server-side input validation, **client-side input validation**—because it’s done on the client side—may be bypassed and effectively rendered useless.

#### Lack of Input Validation

The more tightly controlled and managed the input, the more secure the application and environment. On the other end of the spectrum, no input validation can lead to serious negative consequences, as numerous web application attacks may be possible.

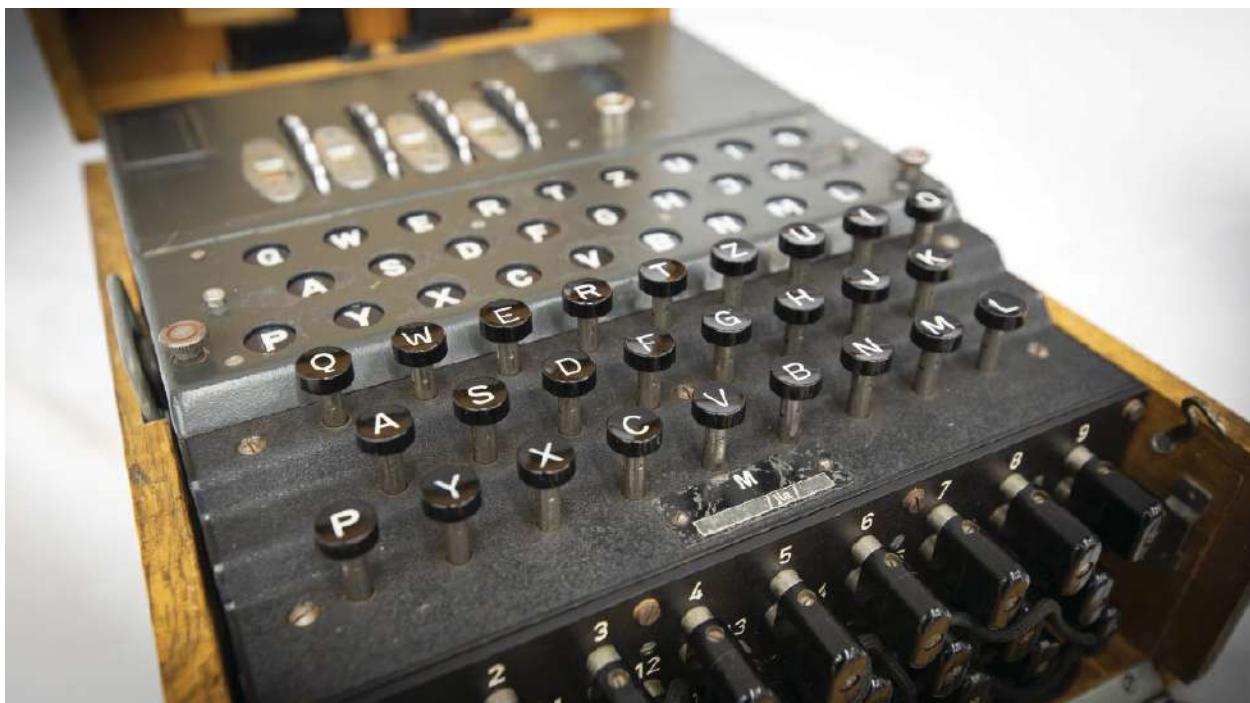
## Reduce Risk of Web-Based Vulnerabilities

To reduce the risk of web-based vulnerabilities, hardening steps, like those mentioned in [section 3.5.2](#), should be followed. Note that these may vary depending on organizational requirements as well as the systems being used.

In its simplest form, the goal of hardening is to reduce the potential attack surface of a system, in other words, to reduce risk. The hardening process could involve one or a combination of the following:

- Utilization of a manufacturer's product guide, specifically portions related to security or hardening
- Best practice guidance specific to the organization's industry
- Information gleaned from online sources

With information gained from sources like those noted above, an organization can make informed decisions and customize hardening steps specifically to their environment and security requirements. At the same time, an organization should document their hardening process and keep it updated for purposes of application to newly deployed systems as well as compliance with existing production systems.



## **3.6 Select and determine cryptographic solutions**

### **3.6.1 Introduction to Cryptography**

#### **CORE CONCEPTS**

- The history of cryptography spans approximately four thousand years
- The most critical aspect of cryptography is key management
- Cryptographic systems can provide up to five services: confidentiality, integrity, authenticity, nonrepudiation, and access control
- Cryptography is used extensively and is often all around us in many different contexts

#### **What is the foundation of cryptography?**

The word *cryptography* is derived from two Greek words—*crypto* and *graphia*; *crypto* means “secret” or “covert” and *graphia* means “writing.” So, the foundation and meaning of cryptography is “secret writing”—creating a cipher. Cryptography has existed for thousands of years; some of the earliest examples point to the Egyptians and their use of hieroglyphs. The meanings of the glyphs were often kept secret to give a very small segment of the population an advantage over the general public. Germany has long been a pioneer of cryptography, with one of the best-known examples being the Enigma machine, which was invented

in the 1930s. Even by today's standards, the Enigma machine is still an excellent encryption tool if used correctly with good key management.

Regardless of the encryption method or tool, the most important aspect of cryptography is **key management**; secrecy of the key is paramount to maintaining the effectiveness of any cryptographic system.

[Table 3-32](#) shows a very brief and high-level overview of the evolution of cryptography, which in itself is not very brief.

<b>Manual</b>	<p>The history of cryptography spans approximately four thousand years, and much time was spent in the manual era of cryptography, where simple rearrangements of letters served to create ciphertext. The <b>Caesar Cipher</b> is a perfect example of this type of cryptography.</p> <p>Language and reading comprehension abilities allowed this type of cryptography to be successfully used for years by kings, queens, and other rulers, like Julius Caesar.</p>
<b>Mechanical</b>	<p>As manual cryptography can be a very slow and tedious process, advancements were made that moved it to the mechanical age. Machines were developed that allowed cryptography to be done much more quickly and easily. The Spartan Scytale, perfected by the Greeks, involved the use of rods of various lengths and parchment or papyrus wrapped around a given rod from one end to the other, with the message being written along the length of the writing surface. Once unwrapped, letters would appear randomly everywhere. The only way to read the message would be through use of a rod with the exact</p>

	same dimensions as the source rod and the writing surface positioned exactly the same way.
<b>Electro-mechanical</b>	As the availability of electricity became more widespread, the mechanical era of cryptography matured into the electro-mechanical era, and devices like the well-known Enigma machine and the lesser-known Japanese cipher machines known as Red and Purple were developed and utilized with great success during World War II.
<b>Electronic</b>	Most current cryptography is electronic, meaning it's driven by software applications better known as cryptosystems. Most cryptosystems support several cryptographic algorithms, and as long as the same system or algorithm is available on each side, two people can communicate securely. A well-known crypto system is PGP, and common algorithms include DES, AES, and RSA.
<b>Quantum</b>	The quantum era of cryptography is still in the experimental stages, but quantum techniques could be used to break public-key algorithms, as well as to securely distribute keys.

Table 3-32: Cryptography Evolution

**What are the five services that can be achieved using cryptography?**

With any cryptographic system, one (or a combination of services) denoted in [Table 3-33](#) can be achieved.

<b>Confidentiality</b>	Confidentiality helps prevent unauthorized disclosure of information and to make data available to only those authorized to view it.
<b>Integrity</b>	Integrity ensures that information has not been manipulated or changed by unauthorized individuals without our knowledge; it helps identify unauthorized or unexpected changes to data.
<b>Authenticity</b>	Authenticity allows verification that a message came from a particular sender.
<b>Nonrepudiation</b>	Nonrepudiation prevents someone from denying prior actions. There are two flavors of nonrepudiation: <ul style="list-style-type: none"> <li>■ Nonrepudiation of <b>Origin</b>: the sender cannot deny that they sent a specific message.</li> <li>■ Nonrepudiation of <b>Delivery</b>: the receiver cannot deny that they received a specific message.</li> </ul>
<b>Access Control</b>	Cryptography enables a form of access control; by controlling the distribution of ciphertext and the corresponding decryption key to only certain people, control over the decryption and therefore access to data can also be controlled.

Table 3-33: **Cryptography Services**

## Everyday Uses of Cryptography



Whether people realize it or not, cryptography is all around us and used every day in multiple ways. When an online purchase is made, cryptography helps ensure the security and privacy of a customer's credit card details and personal information. When a mobile device or computer downloads security updates from Google, Apple, Microsoft, or another software vendor, cryptography ensures the integrity of the files. Criminals often turn to cryptography to hide communications from law enforcement and government agencies. Organizations that sell movies, video games, music, and similar types of consumable entertainment use cryptography for purposes of digital rights management and antipiracy. Other uses of cryptography include secure electronic voting, digitally signing documents, defensible

data destruction in the cloud, cryptocurrencies, and the list goes on and continues to grow.

### 3.6.2 Cryptographic Terminology

#### CORE CONCEPTS

- **Cryptography involves its own nomenclature.**
- **Important terms to be familiar with include: initialization vector (IV)/nonce, confusion, diffusion, avalanche, key space**

At its core, cryptography is the study and practice of securing communications to prevent attackers from reading or manipulating information.

[Figure 3-35](#) shows how cryptography systems work.

Plaintext (in this case “CISSP is awesome”) is provided as input into a cryptosystem, and a cryptographic algorithm transforms it into ciphertext. The only way this ciphertext can be transformed back into plaintext by a recipient is through use of a compatible cryptosystem and the same cryptographic algorithm.



**Figure 3-35: Cryptosystem Operation**

**Table 3-34:** Cryptography Terms contains a variety of terms relating to cryptography that you will need to be aware of.

<b>Plaintext</b>	Plaintext, also known as cleartext, is simply data that is readable by anyone.
<b>Encrypt/ Encryption</b>	Encryption is the process of converting plaintext into ciphertext using a cryptographic algorithm and a key/crypto variable.
<b>Key/Crypto Variable</b>	A crypto variable is also referred to as a <b>key</b> . When a given piece of plaintext is encrypted with a key, the key determines how the algorithm processes the plaintext to produce ciphertext. Once plaintext has been encrypted with a key, the only way to decrypt the ciphertext is with the appropriate key. The encryption and decryption processes are depicted in <a href="#">Figure 3-36</a> .
<b>Decrypt/ Decryption</b>	Decryption is the process of <b>turning ciphertext back into plaintext</b> using a cryptographic algorithm and a key.
<b>Key Clustering</b>	Key clustering describes what happens when <b>two different keys generate the same ciphertext from the same plaintext</b> . This is something that should be avoided, and effective cryptographic algorithms are

	<p>designed to minimize or, ideally, eliminate key clustering. Key clustering is bad, because if the two different keys can decrypt the same ciphertext, a brute-force attack can be performed twice as fast, as there will be two different keys that can decrypt the ciphertext. In other words, key clustering effectively reduces your key space (number of keys possible) in half.</p>
<b>Work Factor</b>	<p>Work factor is an <b>estimated amount of time or effort required by an attacker to break a cryptosystem</b>. The higher the work factor, the more secure the cryptosystem.</p>
<b>Initialization Vector (IV)/Nonce</b>	<p><b>Understand what an initialization vector/nonce is, how it is used, and potential weaknesses with it</b></p> <p>An initialization vector (IV), or nonce, is a <b>random number</b> that is used in conjunction with the key and fed into a cryptographic algorithm when encrypting plaintext. IVs should only be used once in any session and are used to <b>prevent patterns</b> in the resulting ciphertext. In other words, the same plaintext can be fed into a cryptographic algorithm, and even though the same key is used, if a different IV is used, the resulting ciphertext will be different, thus avoiding patterns. When decrypting ciphertext, the same IV used to encrypt the plaintext must be used. If the length of the IV is too short, the resulting ciphertext might be vulnerable to being deciphered. This was the case with Wired Equivalent Privacy (WEP) protocol and effectively rendered WEP useless.</p>
<b>Confusion</b>	<p>Effective cryptographic algorithms should demonstrate some key properties. The first property is known as confusion, which focuses on hiding the relationship between <b>the key</b> and the resulting ciphertext. The</p>

	confusion property suggests that if one bit of the <b>key</b> is changed, then about half of the bits in the ciphertext should change.
<b>Diffusion</b>	Diffusion follows similar thinking as confusion, but is focused on the <b>plaintext</b> . It suggests that if a single bit of the <b>plaintext</b> is changed, then approximately half of the bits in the ciphertext should change. The diffusion property focuses on hiding the relationship between the <b>plaintext</b> and the ciphertext.
<b>Avalanche</b>	To determine the security and effectiveness of an algorithm, the avalanche effect should be studied. The avalanche effect looks at the <b>degree of confusion and diffusion</b> that an algorithm provides. The ideal case is that a single bit change to either the key (confusion) or to the plaintext (diffusion) will result in at least a 50 percent change in the ciphertext. <a href="#">Figure 3-37</a> depicts confusion, diffusion, and avalanche.

Table 3-34: Cryptography Terms

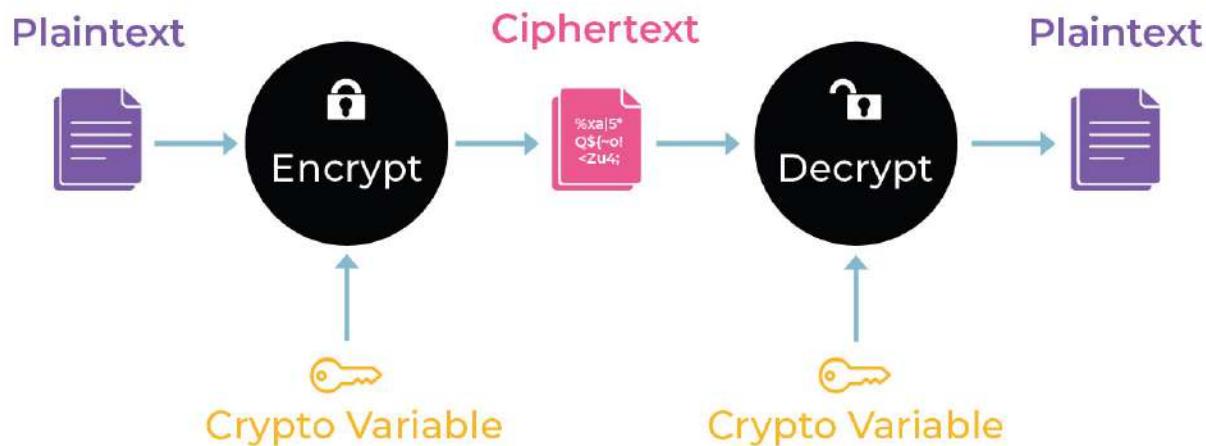


Figure 3-36: Encryption/Decryption

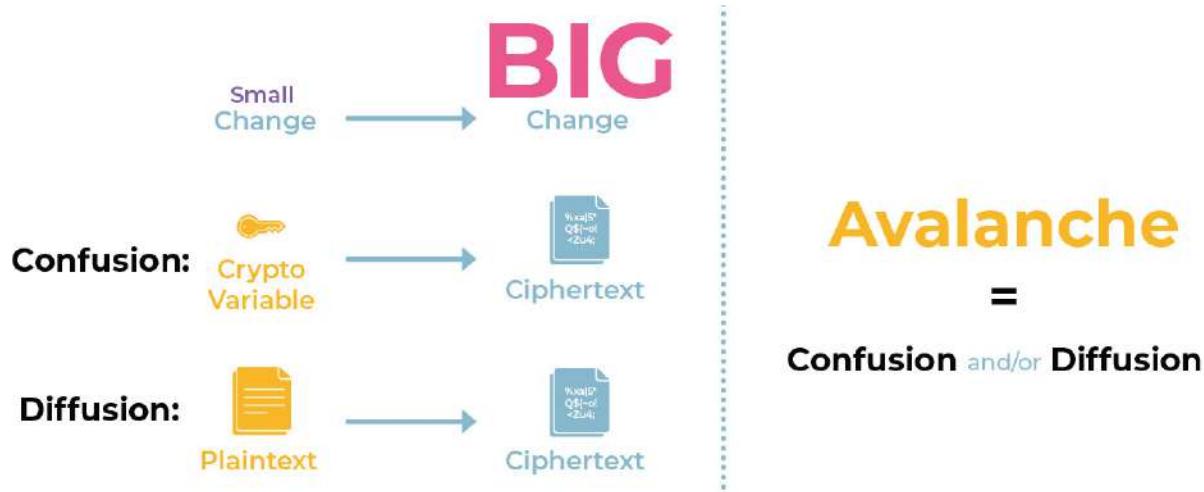
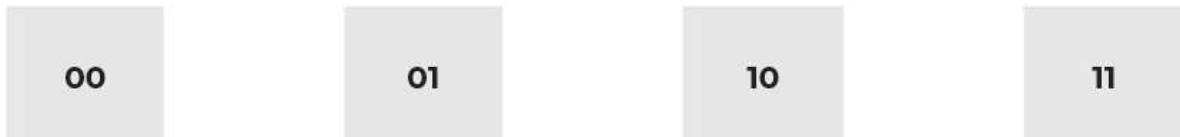


Figure 3-37: Confusion, Diffusion, and Avalanche

## Key Space

The term *key space* refers to the unique number of keys that is available based on the length of the key. For example, a 2-bit key has a total of four possible, or unique, keys:



Even with the best algorithm, the means to creating strong and effective encryption is to have strong keys.

Data Encryption Standard (DES) uses a 56-bit key, which equates to  $2^{56}$  unique keys, or 72,000,000,000,000,000 (15 zeros, 72 quadrillion) unique keys. This is a really large number of keys, but modern computers can brute-force a 56-bit key in a matter of anywhere from a few hours to

several days. The amount of time needed to break a key is also known as the *work factor* (as already highlighted in [Table 3-34](#)).

To significantly increase the work factor, most symmetric keys in use today are 128- or 256-bit keys, while RSA keys are moving toward 2048 bits and above as computing advances.

### 3.6.3 Substitution and Transposition

#### CORE CONCEPTS

- Encryption involves methods known as substitution and transposition.
- Encryption is accomplished through the manipulation of bits—1s and 0s—via synchronous or asynchronous means.
- Patterns must be avoided.
- When implemented and used correctly, one-time pads are the only unbreakable cipher systems.
- Bits are encrypted/decrypted as stream ciphers or block ciphers.

## Methods of Encryption

The strongest encryption methods use substitution and transposition as outlined in [Table 3-35](#). If only one round of each method is used, breaking the encryption is fairly easy; thus, with every good method, multiple rounds of substitution and transposition are used. 3-DES, for example,

performs forty-eight rounds of substitution and transposition, which provides strong encryption.

Substitution	Transposition
Characters are replaced with a different character <b>GUBBINS &gt; JXEELQV</b>	The order of characters is rearranged <b>GUBBINS &gt; BINBUGS</b>

Table 3-35: Substitution and Transposition

## Substitution

Substitution, shown in [Figure 3-38](#), is a method of encryption where every plaintext character is replaced/substituted with a different character to create ciphertext. In the example, all G characters are replaced with J, all B characters are replaced with E, and so on, based on a given key. In this case, GUBBINS is the plaintext, and based upon the key used (a simple shift of three letters to the left), the corresponding ciphertext is JXEELQV.

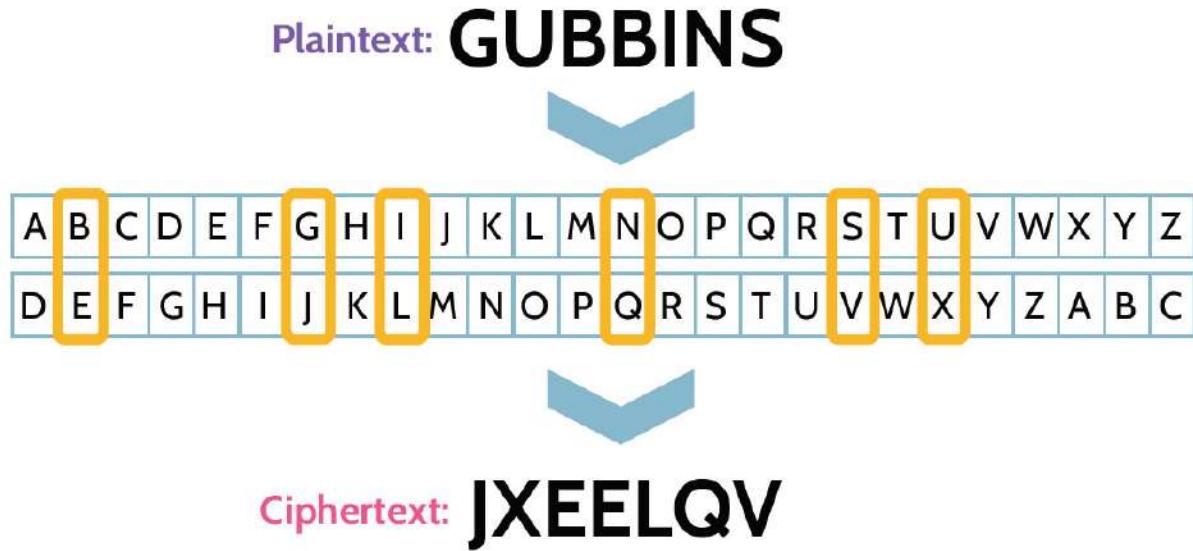


Figure 3-38: Substitution Operation

## Transposition

Transposition, shown in [Figure 3-39](#), is a method of encryption where every plaintext character is shifted around/rearranged based on a given key. Using the prior example, based upon the shifting of characters, the plaintext GUBBINS becomes the ciphertext BINBUGS.

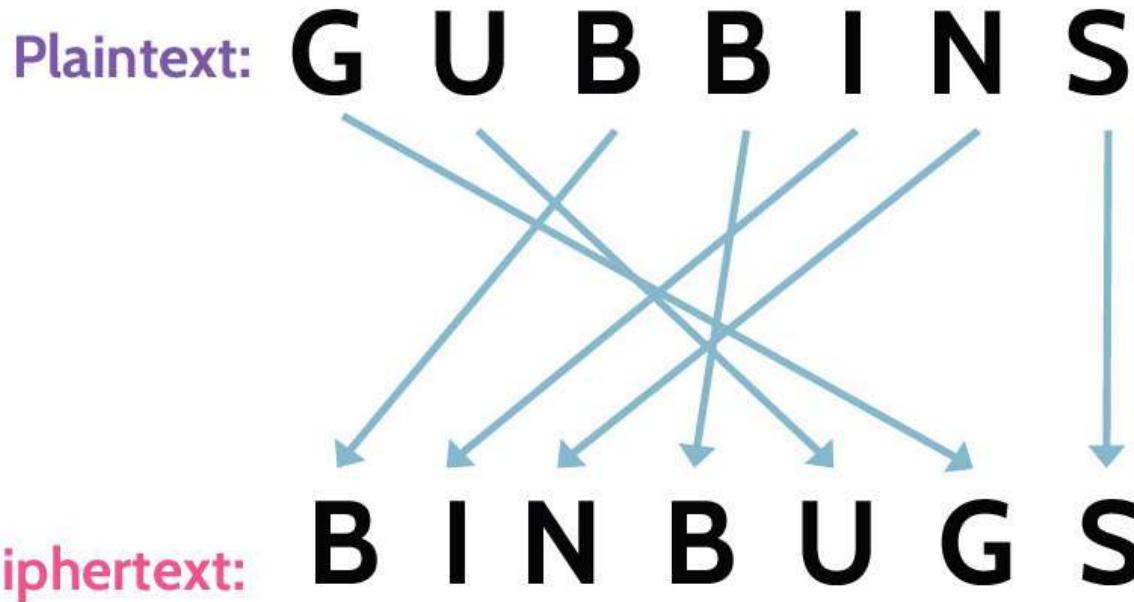


Figure 3-39: Transposition Operation

**Understand the inherent weakness of simple substitution and transposition ciphers**

In each example, it should be evident that simple substitutions and transpositions do not hide patterns effectively. The word GUBBINS was specifically used to illustrate this fact. The letter *B* is used twice in GUBBINS, and the resulting ciphertext shows a pattern that reflects this fact. Ultimately, patterns make deciphering the word much easier. In cryptography, patterns should be avoided at all costs, because they significantly undermine the end goal.

## Rail Fence (Zigzag)

Another simple form of transposition is called the *Rail Fence (zigzag) cipher* and is illustrated in [Figure 3-40](#).

Basically, the text is transposed by writing it in a table where each row represents a rail, following a zigzag pattern.

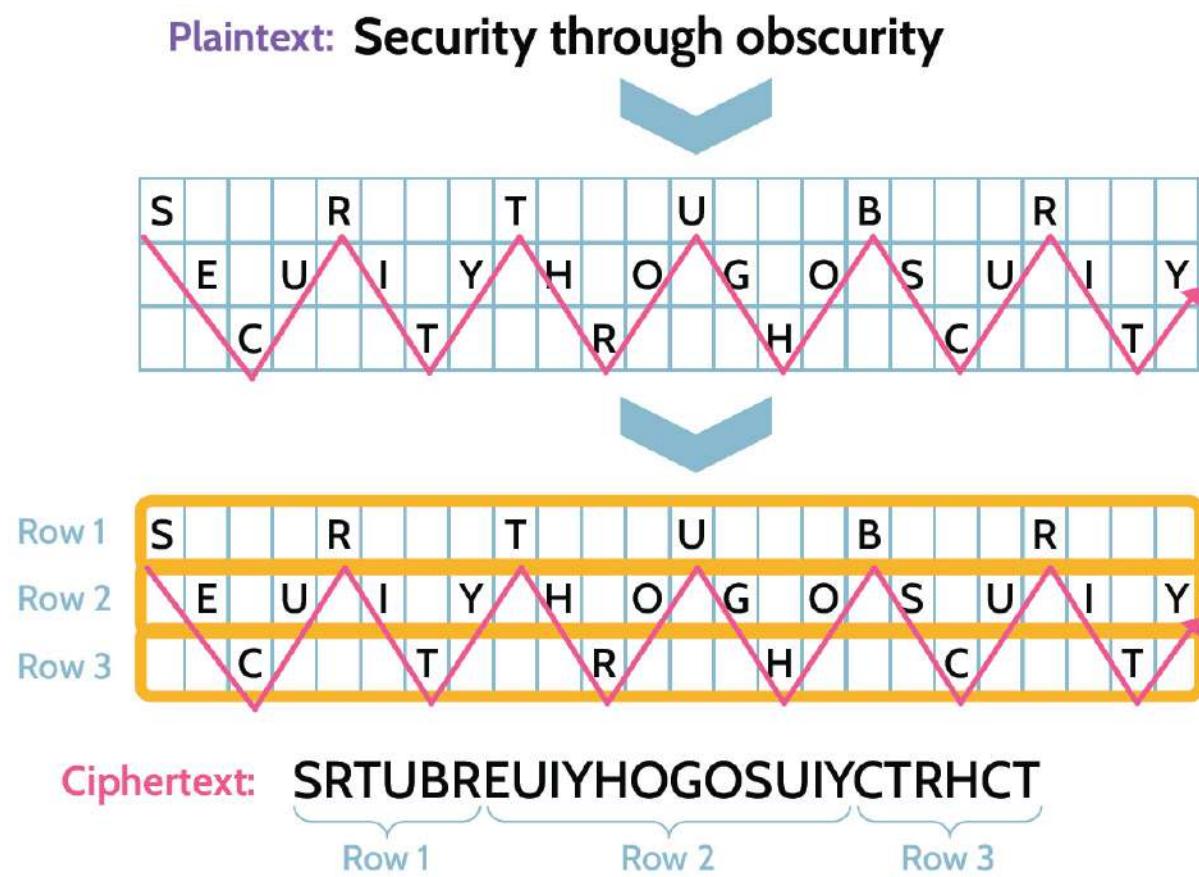


Figure 3-40: Rail Fence Operation

Despite the seemingly more secure-looking output, this form of transposition may still result in patterns being recognizable.

Similarly, transposition ciphers can also be employed using a columnar or a diagonal approach as the key, with the output changing based upon the method used. That is denoted in [Figure 3-41](#).

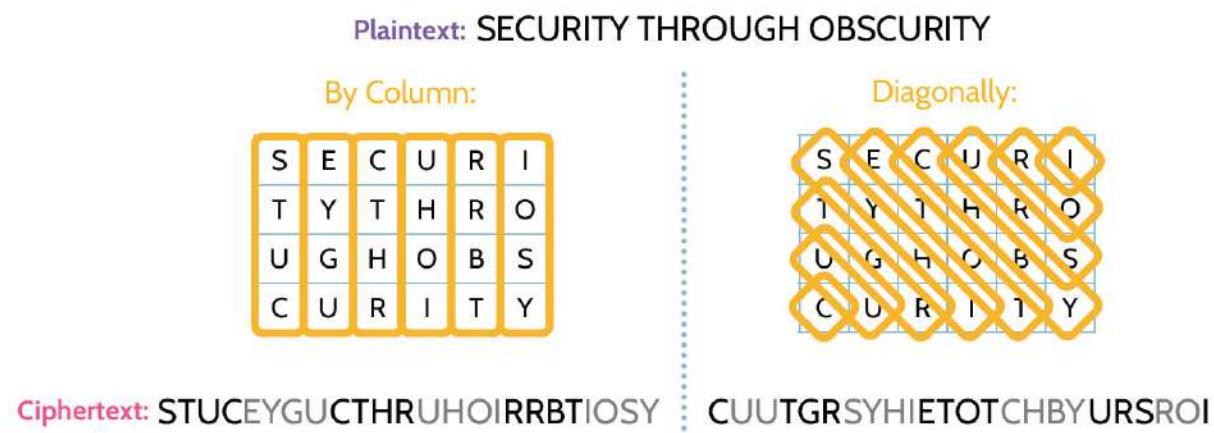


Figure 3-41: **Columnar and Diagonal Transposition Ciphers**

As highlighted earlier, simple transposition is being used, which means that patterns may still exist in the resulting ciphertext.

### Synchronous versus Asynchronous

The examples above could lead a person to believe that cryptography involves the manipulation of actual characters. In fact, cryptography involves the manipulation of bits—1s and 0s—that represent those characters. Cryptography takes the bits that represent, e.g., GUBBINS and manipulates them (using, e.g., substitution and transposition) to create ciphertext.

The bits are manipulated via synchronous or asynchronous methods. Synchronous involves working with bits synchronized through some type of timing mechanism, for example, a clock while encryption/decryption takes place immediately. Asynchronous involves working with collections of bits, and the input is typically dictated by the user or some other element that requires input. A comparison of the two methods is also provided in [Table 3-36](#).

Synchronous	Asynchronous
<ul style="list-style-type: none"><li>■ A timing element is involved</li><li>■ Encryption/decryption requests are performed immediately</li></ul>	<ul style="list-style-type: none"><li>■ Dictated by some other element or entity that requires input</li><li>■ Encryption/decryption requests are processed in batches (queued)</li></ul>

Table 3-36: **Synchronous vs. Asynchronous Encryption**

### Repeating Patterns Must Be Avoided

As already mentioned, several times, one of the most important things with any cryptography implementation is the avoidance of patterns. Patterns can severely weaken cryptography, and numerous failures point to key management issues that lead to the emergence of numerous patterns.

To further illustrate, using the English language as an example, common letter usage and patterns exist.

- Most common letter in the English language is **E**
- Most common three-letter word in the English language is **the**
- Most common four-letter word in the English language is **that**

Using this information, a cryptanalyst could examine ciphertext and count all the letters. The one used most often likely would represent *E*. Similarly, the most commonly used three- and four-letter combinations in the ciphertext would likely represent the words *the* and *that*. In the latter case, the word *that* begins and ends with the same letter: *T*. Thus, ciphertext that includes a four-letter combination beginning and ending with the same letter is very likely the word *that*.

The activity of trying to determine keys based upon letter usage patterns is known as **frequency analysis**. Individuals who perform this activity are typically well-trained linguists who deeply understand the language and language statistics related to whatever language is used for encryption.

## Substitution Patterns in Monoalphabetic Ciphers

Worth reiterating is that simple substitution and transposition does not hide patterns in monoalphabetic ciphers. Frequency analysis can easily detect patterns in them, as denoted in [Figure 3-42](#), which can then lead to determination of the key.

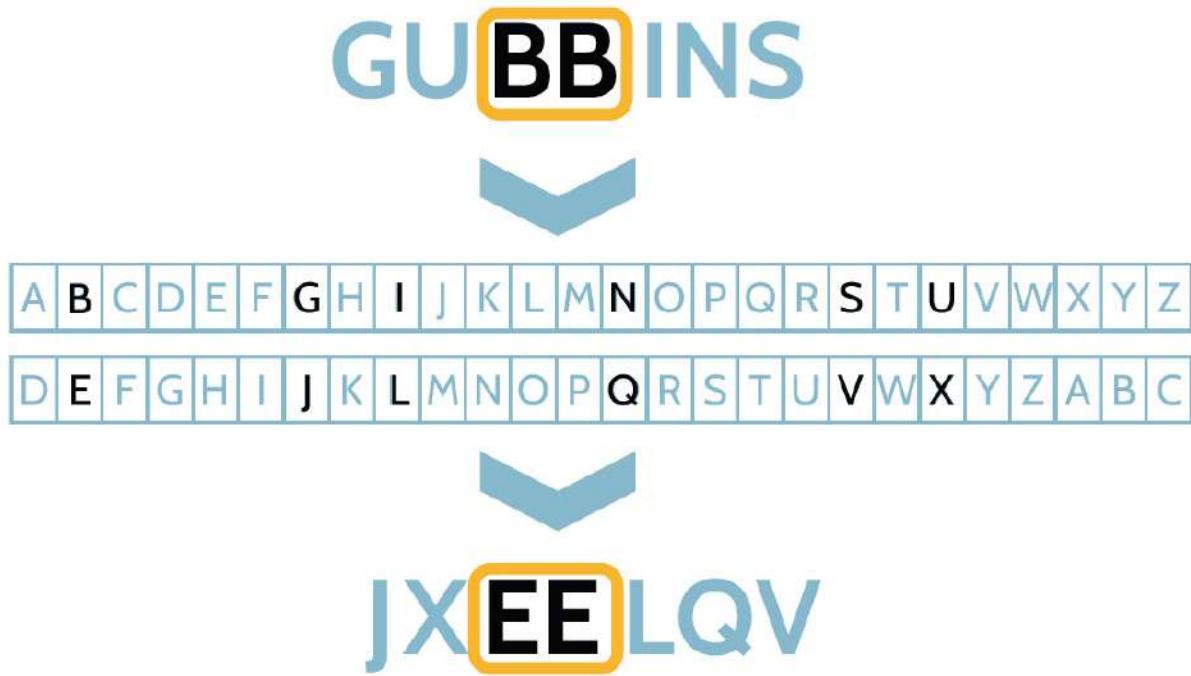


Figure 3-42: Patterns Indicating Encryption Weaknesses in Monoalphabetic Ciphers

## Substitution—Polyalphabetic Ciphers

By using polyalphabetic ciphers, frequency analysis becomes much more difficult, because patterns are reduced significantly.

The prefix *poly* means *many*, so with polyalphabetic ciphers, multiple alphabets are created and used. [Figure 3-43](#) shows an example of this. The standard alphabet and order can be noted along with the same alphabet in nonstandard format four times. In row 1, the first letter is *Z*, in row 2, the first letter is *Y*, in row 3, the first letter is *X*, and in row 4, the first letter is *W*. Each row number is part of the key, which in this example is 4312.

Based upon the key 4312, the word GUBBINS becomes CRAZEKR. The first letter *G* corresponds to *C* in row 4, the second letter *U* corresponds to *R* in row 3, and so on and so forth. In the case of letter *I*, key usage starts over at row 4, so *I* corresponds to *E*, *N* corresponds to *K* in row 3, *S* corresponds to *R* in row 1. If the word were longer, or a phrase were involved, key usage would repeat in the same manner until the word or message was encrypted.

As can be seen, in this example, no patterns exist. Though *B* is used twice in the plaintext, it becomes *A* and *Z* in the ciphertext because a polyalphabetic cipher was utilized.

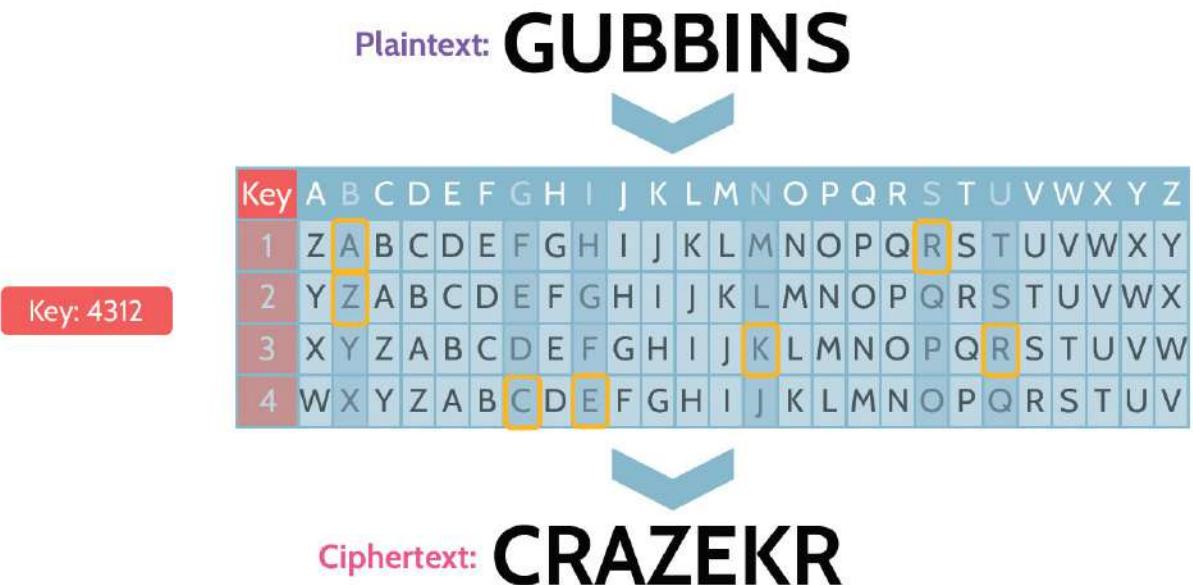


Figure 3-43: Lack of Patterns in Polyalphabetic Ciphers

### Substitution—Running Key Ciphers

Another cipher that hides patterns very well is known as the running key cipher, which has been used throughout history and especially so after World War II, when distrust between countries was extremely high.

To utilize the running key cipher, the same “book” must exist at both ends of the communication channel. For example, imagine a spy located in one country who needs to communicate with a government located in another country. The spy and the government would both need the same book.

When the spy wishes to communicate with their government, the message that needs to be encrypted is

combined with text from the book that appears in a specific location in the book. For example, the spy and government would both know that the starting point is page 32, paragraph 1, sentence 1, word 5. With this knowledge, the secret message is combined with the corresponding text to create ciphertext. The way the letters are combined is via the numeric equivalent of each letter as shown in [Figure 3-44](#). In this case, the numeric equivalent of each letter in the word *GUBBINS* is combined with the corresponding numeric equivalent of each letter of the text that corresponds with the predetermined starting point from the book, in this case *THEQUIC*.

So, the value of *G*—7—is combined with the value of *T*—20, which yields 27. As noted in the illustration below, the English alphabet contains only twenty-six letters, so in cases where a combined number is greater than twenty-six, the formula  $(\text{combined number}) - 26 = n$  should be followed. With the first letter,  $27 - 26 = 1$ , so the first ciphertext letter is A.

For the second letter,  $U - 21 + H - 8$  yields 29. Using the formula,  $29 - 26 = 3$ , so the second ciphertext letter is C. Following the same method, the third ciphertext letter is G, because *B* (2) and *E* (5) = 7, which corresponds to *G*.

**Values of plaintext:**

G U B B I N S

+

**Values of running cipher:**

(The quick brown fox...)

T H E Q U I C

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26

Add values of plaintext + running cipher:

$$\begin{array}{r} \text{G: 7} \\ + \text{T: 20} \\ \hline 27 - 26 = 1 \end{array}$$

Lookup value is ciphertext: 1 = A

Figure 3-44: Running Key Cipher Operation

Because a book is used, a very large number of possible keys exist, and each subsequent communication would simply require each party knowing the starting point for purposes of encryption/decryption. As long as a previously used key is not reused, this method of communication can remain very secure. For this reason, spies and their home offices often utilized what's known as a code book. That is a book that has a predetermined starting point acting as the encryption/decryption key for each message. Once a key was utilized, that page in the book would be destroyed, so the key could not be used again.

## **Substitution—One-Time Pads**

With a one-time pad, after every message is encrypted, the key is changed and never reused. Additionally, the key length with one-time pads is always the same length. When implemented and used correctly, **one-time pads are the only unbreakable cipher systems.**

## **Stream versus Block Ciphers**

All symmetric and asymmetric algorithms in cryptography work with bits, not letters. Once a message has been turned into bits, two options exist with regards to how those bits are encrypted and decrypted.

In one case, bits can be worked on one bit at a time as a stream; in the other, bits can be worked on in collections, or blocks of data. So the two types of ciphers that exist are known as **stream ciphers** and **block ciphers**. Block ciphers are more versatile than stream ciphers, but this versatility comes at the cost of speed and code size. We tend to favor stream ciphers in low-end devices, and in circumstances where we want to maximize efficiency.

Stream ciphers deal with one bit at a time and are faster, as opposed to block ciphers which work on one block and then must wait momentarily to work on the next block, until

all information is encrypted. Stream and block ciphers are summarized in [Table 3-37](#).

Stream	Block
Encrypt/decrypt data <b>one bit at a time</b>	Encrypt/decrypt <b>blocks of bits at a time</b> (for example, AES has a 128-bit block size)

Table 3-37: **Stream vs. Block Ciphers**

## Stream Ciphers

When considering stream ciphers a bit further, on the surface it doesn't sound like much can happen with a bit when it is either a 0 or a 1 and can only remain a 0 or a 1. In other words, encryption of only 0 or 1 leaves little to no options. So a bit of creativity needs to be employed to enhance the encryption process.

Look at [Figure 3-45](#).

- **Plaintext** bits that need to be encrypted are combined with bits generated by a **keystream generator**, which is seeded by a crypto variable.
  
- The bits are combined using a logical operation called **“exclusive or,” or XOR**, which helps create the

needed confusion and diffusion to make a stream cipher secure.

- In the XOR process,  $0 + 0 = 0$ ,  $0 + 1 = 1$ ,  $1 + 0 = 1$ ,  $1 + 1 = 0$  (in other words, the output will be 1 if only one of the inputs is 1).
- The result from each XOR operation becomes the ciphertext.

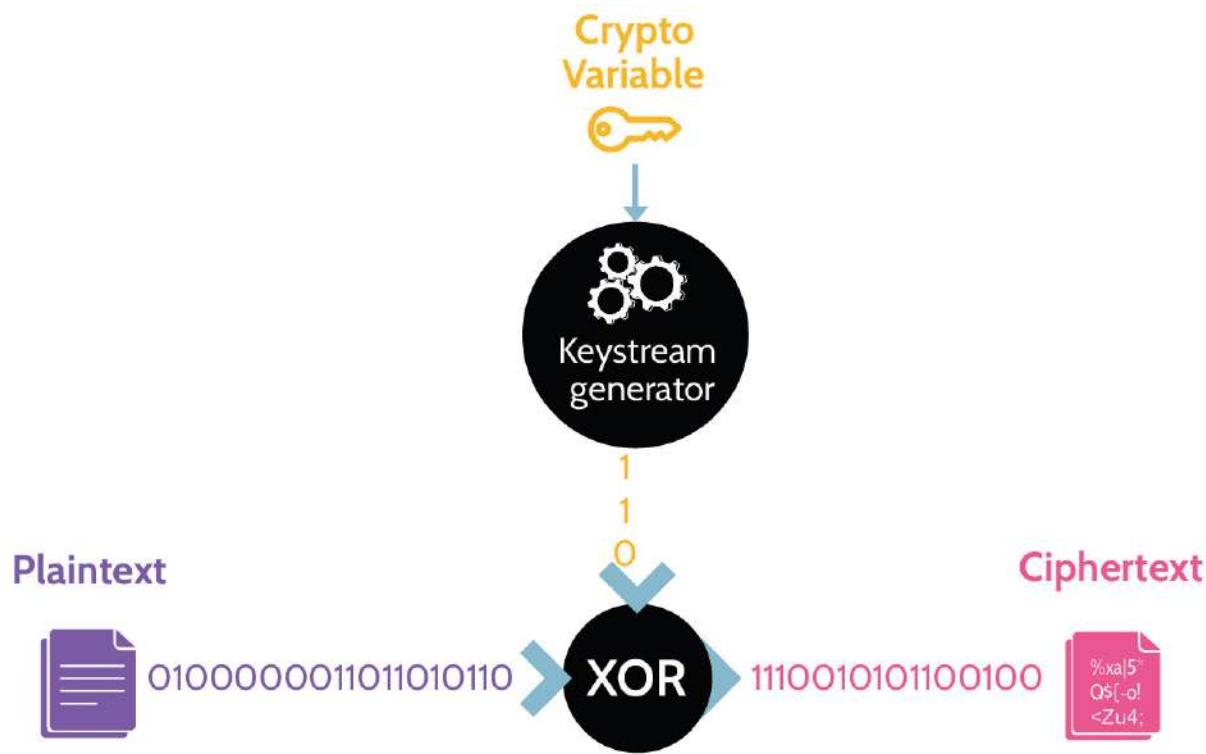


Figure 3-45: Stream Cipher Operation

**The most commonly used stream cipher is Rivest Cipher 4 (RC4).**

## Block Ciphers

Block ciphers encrypt data in chunks that we call blocks. As an example, AES has a 128-bit block size.

As [Figure 3-46](#) illustrates, instead of bits being encrypted one at a time, they're encrypted in blocks.

- Plaintext blocks, like the letters GUB and BIN, are processed by a block cipher that has been seeded by a crypto variable.
- The output of each operation results in chunks of ciphertext, JXE, ELQ, and the S as V.
- Each block is processed individually.

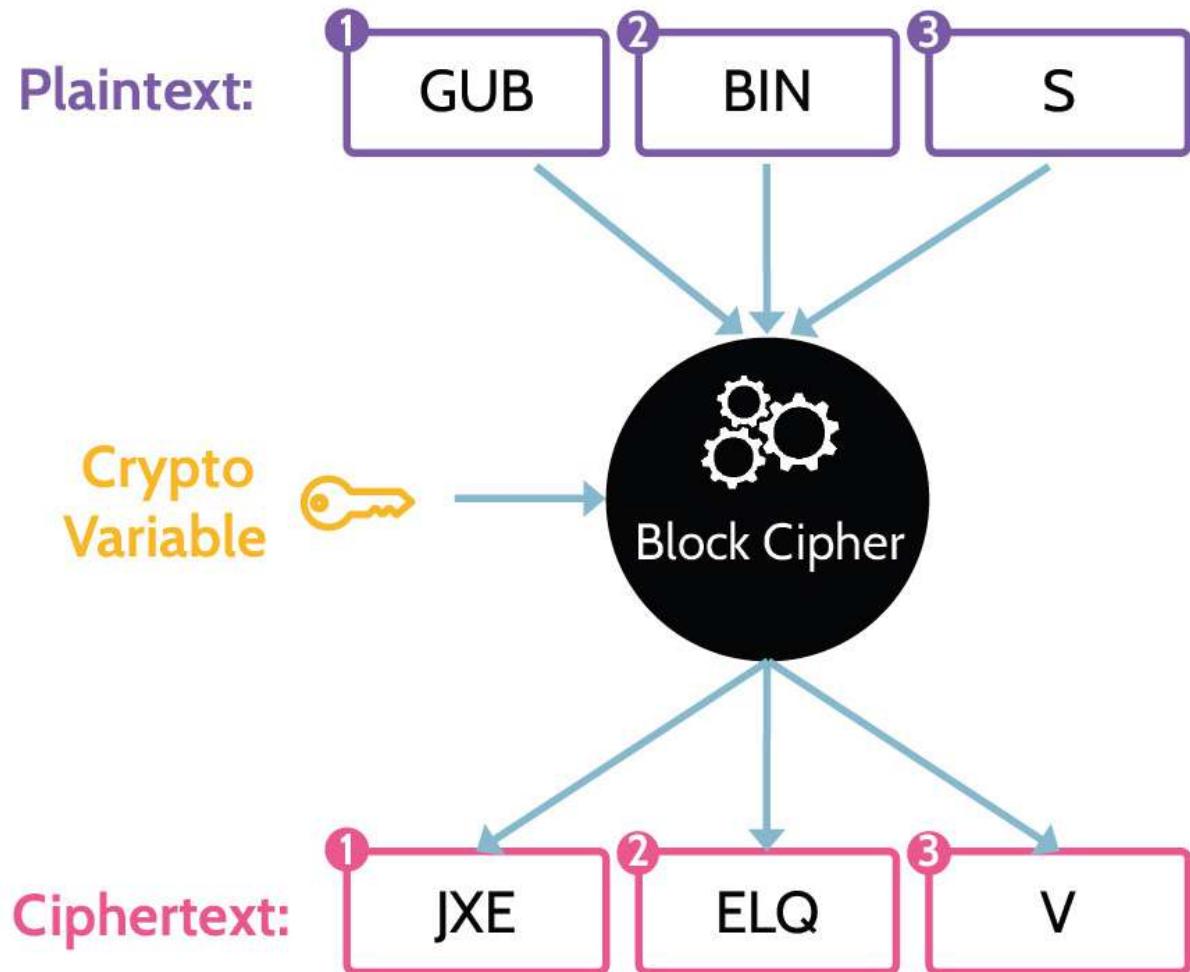


Figure 3-46: Block Cipher Operation

### Symmetric Block Modes

**Understand which mode of symmetric cipher is faster/slower**

In cryptography, **stream ciphers provide a clear speed advantage over block mode ciphers, because they work with one bit at a time as opposed to block ciphers that**

## **need to fill blocks and do creative operations with those**

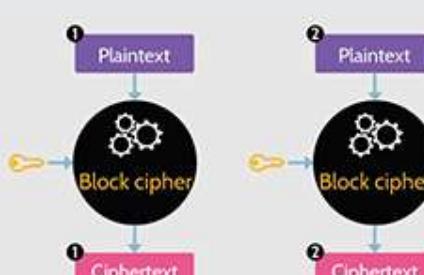
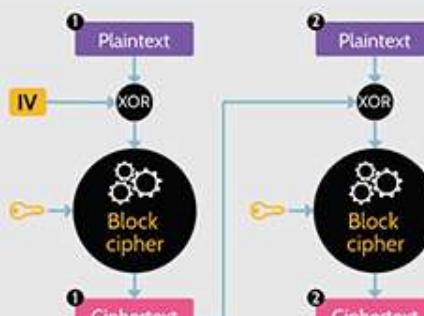
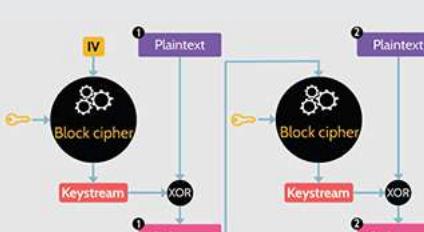
**blocks.** This fact that stream ciphers work with one bit at a time makes them especially suitable for encryption across networks and at the hardware level. However, block mode ciphers also provide an advantage, namely, that they have a high diffusion rate and are very resistant to tampering.

Any block mode cipher, like DES or AES, for example, will support the five modes listed in [Table 3-38](#). Specifics worth noting about each mode are noted in the “**Characteristics**” column. Depending on the need, it might make sense to use one block mode over another. For example, for the encryption of short, nonrepeatable amounts of characters or numbers, ECB works very well, because it is very fast. However, for something like email or text messaging, or anything that might include the same text or numbers, ECB is not very secure. Due to its lack of use of an IV, the same text would result in the same ciphertext being produced, and patterns would emerge, making it susceptible to cryptanalytic attacks and successful deciphering.

For email and longer messages, CBC, CFB, and OFB all work well, because each cipher mode employs an IV as part of the encryption process.

From a speed and security standpoint, however, CTR is likely the best mode to use for longer messages. CTR uses a

counter, which is a random initial number (an initialization vector) that is incremented by 1, 2, 3, and so on during each subsequent encryption of blocks. CTR is fast, and due to use of a counter, patterns in the ciphertext do not exist. As a result, it is the most commonly used block mode cipher.

Name	Function	Characteristics
<b>Electronic Codebook (ECB)</b>	 <pre> graph LR     subgraph ECB [Electronic Codebook (ECB)]         direction TB         ECB[Block cipher] -- "1" --&gt; ECB[Block cipher]         ECB[Block cipher] -- "1" --&gt; ECB[Ciphertext]         ECB[Block cipher] -- "2" --&gt; ECB[Block cipher]         ECB[Block cipher] -- "2" --&gt; ECB[Ciphertext]         ECB[Ciphertext] --- ECB[Ciphertext]     end </pre>	<ul style="list-style-type: none"> <li>Least secure mode (no IV) but fastest.</li> <li>Should only be used for short bits of random text that do not repeat, for example, with PIN codes.</li> </ul>
<b>Cipher Block Chaining (CBC)</b>	 <pre> graph LR     subgraph CBC [Cipher Block Chaining (CBC)]         direction TB         ECB[Block cipher] -- "1" --&gt; ECB[Block cipher]         ECB[Block cipher] -- "1" --&gt; ECB[XOR]         ECB[XOR] --- ECB[XOR]         ECB[XOR] --- ECB[Ciphertext]         ECB[Block cipher] -- "2" --&gt; ECB[Block cipher]         ECB[Block cipher] -- "2" --&gt; ECB[XOR]         ECB[XOR] --- ECB[XOR]         ECB[XOR] --- ECB[Ciphertext]         ECB[XOR] --- ECB[XOR]     end </pre>	<ul style="list-style-type: none"> <li>Uses an IV</li> <li>Good for things like email</li> </ul>
<b>Cipher Feedback (CFB)</b>	 <pre> graph LR     subgraph CFB [Cipher Feedback (CFB)]         direction TB         ECB[Block cipher] -- "1" --&gt; ECB[Block cipher]         ECB[Block cipher] -- "1" --&gt; ECB[XOR]         ECB[XOR] --- ECB[XOR]         ECB[XOR] --- ECB[Ciphertext]         ECB[XOR] --- ECB[Keystream]         ECB[Block cipher] -- "2" --&gt; ECB[Block cipher]         ECB[Block cipher] -- "2" --&gt; ECB[XOR]         ECB[XOR] --- ECB[XOR]         ECB[XOR] --- ECB[Ciphertext]         ECB[XOR] --- ECB[Keystream]     end </pre>	<ul style="list-style-type: none"> <li>Uses an IV</li> <li>Good for things like email</li> </ul>

<b>Output Feedback (OFB)</b>		<ul style="list-style-type: none"> <li>■ Uses an IV</li> <li>■ Good for things like email</li> </ul>
<b>Counter (CTR)</b>		<ul style="list-style-type: none"> <li>■ Uses a counter (a random number), which serves a similar purpose as an IV</li> <li>■ Almost the most secure, and <b>fast. Best balance of speed and security.</b></li> <li>■ <b>Most commonly used mode</b> of block cipher</li> </ul>

Table 3-38: Block Cipher Modes

### 3.6.4 Steganography and Null Ciphers

#### CORE CONCEPTS

- Steganography is hiding information of a particular type within something else (like a sound file hidden in a picture).
- A null cipher involves hiding a plaintext message within other plaintext.

**Understand the premise underlying steganography and null ciphers**

Steganography and null ciphers are both related to cryptography and used quite extensively. **Steganography** refers to concealing a message within something else, and there are different ways to do this. For example, physical steganography could be as simple as writing a password on a sticky note and hiding the note under a keyboard. Modern steganography implies that technology-driven techniques are used, like hiding a message within an image file, or a music file, or utilizing slack space on a hard drive. **Slack space** is the leftover storage that exists when a file does not need all the space it has been allocated.

As an example of steganography, another way to hide something within something else is using a null cipher. A **null cipher** hides a message by embedding the plaintext message within other plaintext or noncipher materials. For example, the first letter of each word in a sentence or paragraph could spell the secret message; see [Figure 3-47](#).

Salad embers can ultimately reinvigorate  
incessant throat yodelers

Through high rising oceans urchins go hither

Only Brad seemed confident upwind regarding  
Italy's third yacht

## = Security Through Obscurity

Figure 3-47: Null Cypher Example

Steganography and null ciphers are summarized in [Table 3-39](#).

Steganography	Null Cipher
Plaintext is hidden <b>within something else</b> (e.g., a picture)	Plaintext is <b>mixed with a large amount of nonciphertext</b>

Table 3-39: Stenography and Null Ciphers

### 3.6.5 Symmetric Cryptography

#### CORE CONCEPTS

- **Symmetric key cryptography is fast.**
- **Key distribution and scalability are major disadvantages.**
- **Out-of-band communication can facilitate key distribution.**

■ Know symmetric algorithms from weakest (DES) to strongest (AES).

## Understand advantages and disadvantages of symmetric cryptography

Symmetric key cryptography is extremely fast and can encrypt massive amounts of data. In the context of networks—trusted and untrusted—where significant amounts of data need to be encrypted/decrypted quickly, symmetric cryptography tends to be the best solution.

However, key distribution is a glaring and inherent weakness, especially if the parties involved in communication are separated by any amount of distance.

**Out-of-band communication can be used to overcome this weakness**, but this itself is not necessarily the best or most feasible solution. Out-of-band implies that normal communication channels are not utilized to exchange a key, so internet-based solutions would not qualify. Rather, a meeting in person, a voice call, or perhaps an SMS message could be utilized for the purpose of key exchange. In addition to the key distribution problem associated with symmetric cryptography, a problem with scalability also exists. **Scalability** refers to the number of symmetric keys that would be required to support secure communications among a large group of users. For example, if two people

want to communicate with each other, only one key is required. However, if three people want to securely communicate with each other, three keys would be required. Four people would require six keys. With each new person added, the number of keys required for secure communication doubles; so, the key requirement grows exponentially. A simple formula can be used to determine the number of keys required for a given number of people to securely communicate with each other:  **$n * (n-1) / 2 =$**  **number of keys**

$$n = \text{number of people}$$

For example, for 1000 people,  $1000 * (1000-1) / 2 = 499,500$  keys. That's a tremendous number of keys, and it underscores the key distribution management problem that exists with symmetric cryptography. [Table 3-40](#) shows a summary of advantages and disadvantages of symmetric cryptography.

Advantages	Disadvantages
■ Fast/efficient ■ Strong	■ Key distribution ■ Scalability ■ No authenticity, integrity, or nonrepudiation

**Table 3-40: Advantages and Disadvantages of Symmetric Cryptography**

A number of symmetric key algorithms exist, and several of them are popular and heavily used. For example, Data Encryption Standard (DES), which uses an algorithm called **Data Encryption Algorithm (DEA)** is still used, though its key length of 56 bits leaves it susceptible to brute-force attacks. Knowing the risks associated with shorter key lengths, other algorithms have been developed over the years, including the **International Data Encryption Algorithm (IDEA)**, with its key length of 128 bits—the first symmetric algorithm to use this length. In addition to its longer key length, IDEA is significant, because PGP supported it first. PGP is a cryptosystem that became available in the early 1990s that offers the five services that cryptography can support.

Even with the strength of IDEA due to its 128-bit key length, it never became popular. In fact, the problem with DES's 56-bit key length was addressed by development of 2-DES (Double DES) and 3-DES (Triple DES), which make it more secure and will be discussed further below.

### Rank symmetric algorithms from weakest to strongest

As can also be seen in [Table 3.41](#), several Rivest cipher (RC) algorithms exist (attributed to Ron Rivest). One of Rivest's

cipher algorithms—Rivest Cipher 4 (RC4)—is a stream cipher, while others like RC5 and RC6 are block ciphers.

Regardless of the algorithm, longer key lengths equate to larger key spaces, which provides better overall key strength and security against brute-force attacks.

Strength	Name	Key Length (Bits)	Block Length (Bits)
	RC2-40	<b>40</b>	<b>64</b>
	DES	<b>56</b>	<b>64</b>
	RC5-64/16/7	<b>56</b>	<b>128</b>
	RC5-64/16/10	<b>80</b>	<b>128</b>
	Skipjack	<b>80</b>	<b>64</b>
	RC2-128	<b>128</b>	<b>8</b>
	RC5-64/12/16	<b>128</b>	<b>128</b>
	IDEA	<b>128</b>	<b>64</b>

	Blowfish	<b>128</b>	<b>64</b>
	<b>3DES</b>	<b>168 = 112</b>	<b>64</b>
 <b>Very Strong</b>	RC5-64/12/32	<b>256</b>	<b>128</b>
	Twofish	<b>256</b>	<b>128</b>
	RC6 (derived from RC5)	<b>256</b>	<b>128</b>
	<b>Rijndael (AES)</b>	<b>128, 192, or 256</b>	<b>128</b>

Table 3-41: Symmetric Algorithms

## DES/3-DES

DES, 2-DES, and 3-DES all share the same basis, a 56-bit key, sixteen rounds of substitution and transposition, and a 64-bit block size. From a confusion and diffusion perspective, based upon the multiple rounds of substitution and transposition, DES is one of the best algorithms available. As a result of this strength, the cryptography industry has found ways to extend the life of DES by developing variations known as 2-DES and 3-DES. 2-DES works by doubling the number of keys used. So instead of using a

single 56-bit key, two 56-bit keys are used. 3-DES works similarly but does not take its name from the number of keys used but rather the fact that three iterations of the algorithm always take place. In fact, 3-DES can use two or three keys. Its characteristics have been summarized in

**Table 3-42.**

<b>56-bit key</b>
<b>16 rounds of substitution and transposition</b>
<b>64-bit block size</b>
2-DES—susceptible to meet-in-the-middle attack
3-DES—effective key length is 112 bits

**Table 3-42: DES Characteristics**

### Understand why the effective key length of 3-DES (Triple DES) is 112

One thing worth mentioning about DES involves an attack known as “meet-in-the-middle” and pertains to 2-DES and 3-DES. Remember, 2-DES uses two 56-bit keys, so the key length is 112 bits; 3-DES uses three 56-bit keys, so the key length is 168 bits. However, because of the way the meet-in-

the-middle attack works, which effectively removes 56 bits of strength, the effective key length of 2-DES is reduced to 56 bits and 3-DES is reduced to 112 bits. Without getting into the technical specifics, a meet-in-the-middle attack works by attacking both ends of the key space and working toward the middle. Once the middle is reached, both keys are known. Take 2-DES for example. In order to encrypt a plaintext with it, like the word CISSP, it first is encrypted with key1 (producing, e.g., 2873!@dOIUD), and that is then encrypted with key2 (producing, e.g., NBDJ029845!@). The attacker will try to generate all possible key1 combinations ( $2^{56}$ ) and all key2 combinations ( $2^{56}$ ) so they can:

1. Take the plaintext (CISSP) and encrypt it with numerous key1 values (generated by all combinations in the  $2^{56}$  key space)
2. Take the final ciphertext (NBDJ029845!@) and try to decrypt it with numerous key2 values (generated by all combinations in the  $2^{56}$  key space).

3. When the output of #1 matches that of #2, the attacker can successfully decrypt the plaintext.

As a result, 2-DES is really no more secure than DES, and it is not used. 3-DES was considered stronger, with an effective key length of 112 bits, but it has since been disallowed by NIST. The current NIST standard now is AES256.

## Rijndael/Advanced Encryption Standard (AES)

As was noted earlier, Rijndael—better known as AES—was the winner of a US government-sponsored competition.

AES is considered a variable key size algorithm, which means key sizes of 128, 192, and 256 are all supported, while the block size is always 128 bits. An interesting side note is that Rijndael supports block sizes of 128, 192, and 256, but the US government did not adopt the extra block sizes.

## The ChaCha family

The ChaCha family of algorithms are variants of the Salsa family, which are all ciphers developed by Daniel J. Bernstein. ChaCha8 is a 256-bit stream cipher that's based on the 8-round Salsa20/8 algorithm. The design improves the diffusion for each round, which aims to increase the resistance against cryptanalysis. It also lowers the time per round. There is also ChaCha12 and ChaCha20, with 12 and 20 rounds, respectively. On many systems, ChaCha20 can even run faster than AES. Organizations like Cloudflare and Google are now offering support for TLS cipher suites that are based on ChaCha20-Poly1305 AEAD. This combines ChaCha20 with the Poly1305 hash family. The resulting authenticated encryption with associated data (AEAD) allows us to encrypt data as well as verify its integrity and authenticity.

## Out-of-Band Key Distribution

### Understand why out-of-band key distribution is necessary

With symmetric key cryptography, one of the biggest challenges is key distribution, because the sender and receiver of an encrypted message must have the same key. Sending the key via the same communication channel used to send the message itself is ineffective, because the key could easily be intercepted and used to read the encrypted message as well as any further encrypted communication.

So, some type of out-of-band—and ideally more secure—key distribution method must be employed to share the key, as also highlighted in [Figure 3-48](#).

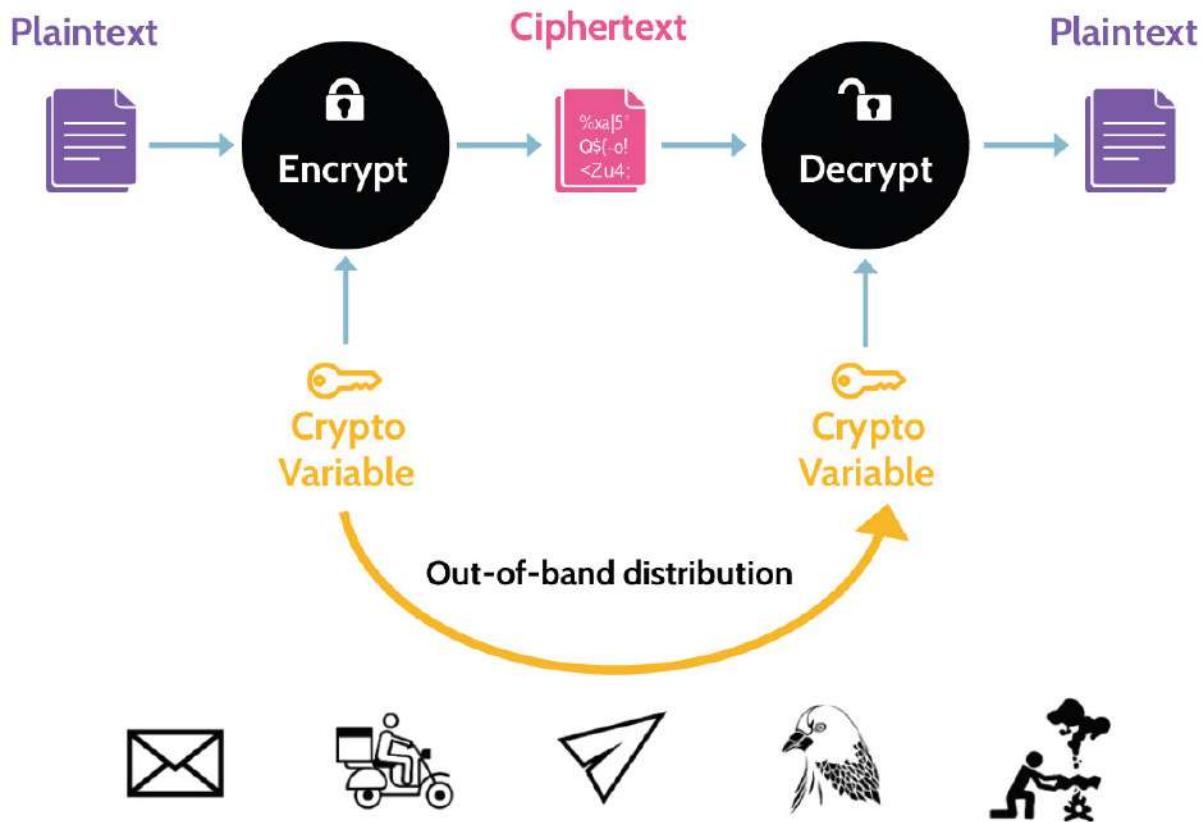


Figure 3-48: Out of Band Distribution

Out-of-band distribution might mean the two parties meet someplace and exchange the key, sending a letter, having a phone call, or some other means by which the key can be shared.

### 3.6.6 Asymmetric Cryptography

#### CORE CONCEPTS

- **Asymmetric cryptography solves the key exchange problem associated with symmetric cryptography.**

- Enables digital signatures, digital certificates, authenticity, and nonrepudiation (of origin and delivery)
- Utilizes key pairs consisting of a public key and a private key
- Two primary types of hard math problems: factoring and discrete logarithms
- Popular asymmetric algorithms include RSA (uses factoring) and Elliptic Curve (ECC, uses discrete logarithms)

## Understand advantages and disadvantages of asymmetric cryptography and how these compare to symmetric cryptography

As discussion about symmetric cryptography explained, it is very fast, but the fundamental problem with key distribution is significant. As a result, cryptologists explored ways to solve the key distribution problem, and in the 1970s two of them—Diffie and Hellman—developed an idea they called “**public key cryptography**.” They premised their idea on the fact that symmetric key cryptography requires the sender and receiver to share the same key, and they asked “What if everybody had two keys that were related to each other mathematically, with the mathematical linkage being such that if something is encrypted with one key it can never be decrypted using the same key? Rather, the other key would need to be used.” This way, one of the keys can be shared with everybody for communication purposes, and

this key can be known as the **public key**. The other key would be retained by the owner and be known as the **private key**; it would not be shared with anybody else. Then, if anybody wants to send the owner a secret message, they can encrypt the message using the public key, and because only the mathematically linked key—the private key—can be used to open the message, nobody else would be able to read it.

So the problem with key distribution is solved, because the only key that ever needs to be shared with anybody is the public key, and it can be shared via any and every means available.

Because the linkage between the public and private **key pair** is mathematically based, it's critical that very complex mathematics be utilized to prevent someone from looking at a public key and computing the private key relationship. Thus, because of the mathematical relationship between a public and private key pair, asymmetric cryptography is significantly slower than symmetric cryptography. Furthermore, as processors become faster, asymmetric algorithms need to be strengthened, which means asymmetric cryptography becomes even slower. A perfect example of this is Rivest, Shamir, and Adleman (RSA). RSA is the most commonly used asymmetric algorithm, and its strength has had to continue to increase to mitigate

advancements in processing technology. While its strength has increased, its functionality has slowed due to the increased mathematical complexity required to keep the key pair secure.

In addition to solving the key distribution and scalability issues associated with symmetric cryptography, asymmetric cryptography provides some other significant benefits, namely in the form of digital signatures, authenticity (also referred to as proof of origin), and other services. Because of the key pair relationship and the idea that the private key should remain secure and private, if somebody wants to prove to somebody else that a certain message was sent by them, they can encrypt the message using their private key. Anybody with the public key could decrypt and read the message, but this might not matter as much as proving that the sender of the message is who they say they are. This is known as authenticity or proof of origin.

Looking at symmetric and asymmetric cryptography together, it's clear that each offers advantages and disadvantages; yet the disadvantage of one is usually found as an advantage with the other. Thus, both types of cryptography are often used together in what's known as hybrid cryptography, or hybrid mode, which is what SSL/TLS uses. See [Table 3-43](#).

Advantages	Disadvantages
<ul style="list-style-type: none"> <li>■ Solves key exchange problem ■ Enables digital signatures and other services, like authenticity (proof of origin), confidentiality, and access control ■ Solves scalability</li> </ul>	<ul style="list-style-type: none"> <li>■ Significantly slower ■ Requires large key sizes</li> </ul>

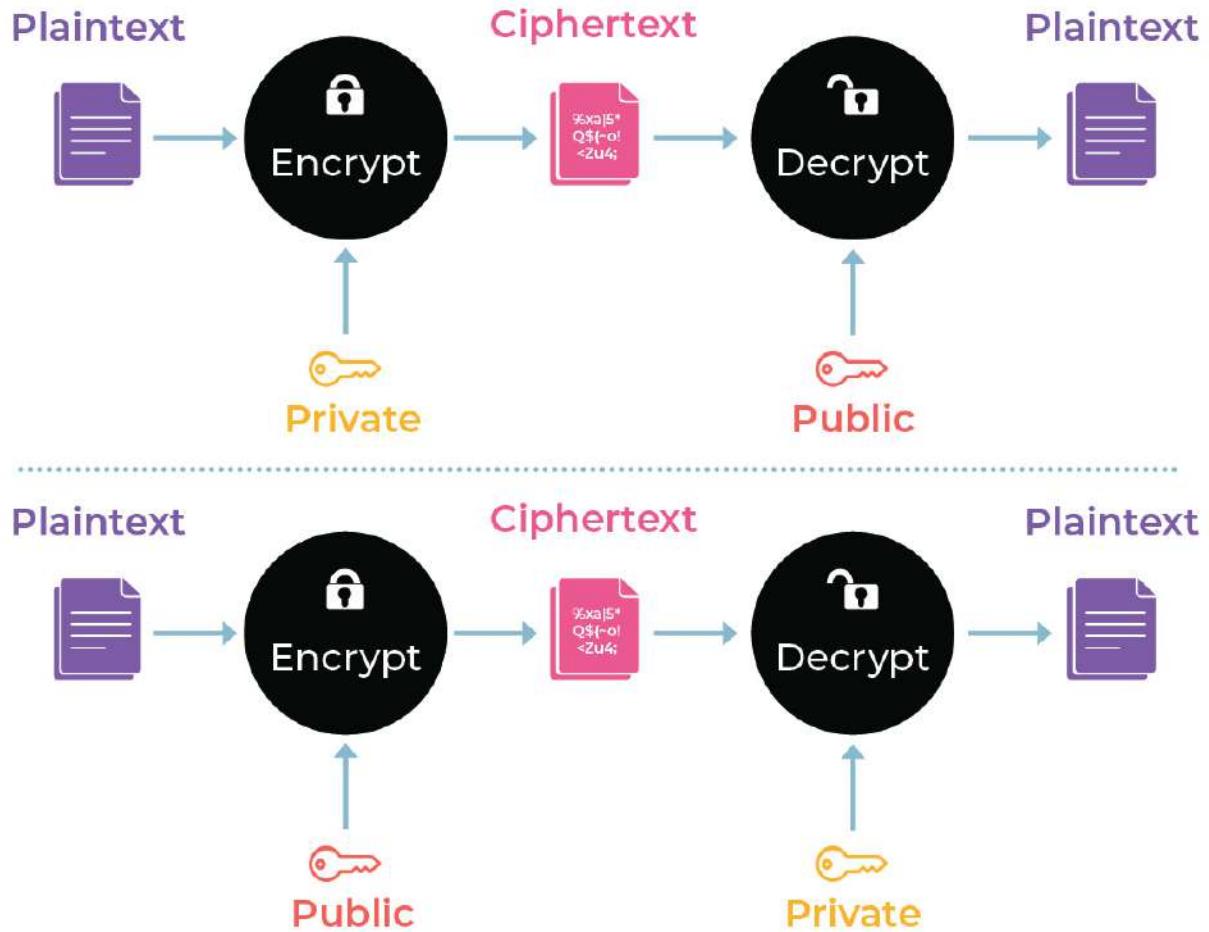
**Table 3-43: Asymmetric Cryptography Advantages and Disadvantages**

[Figure 3-49](#) shows two uses of asymmetric cryptography. The top part shows how it addresses **proof of origin**, while the bottom part shows how to address **confidentiality**.

To obtain confidentiality, anybody who wants to send you a secret message can encrypt the plaintext message using your public key. Remember, anybody else having access to the public key cannot decrypt that message; only the holder of the private key can decrypt it. Thus, when you receive the message, you'll use your private key to decrypt and read it, which you only have access to.

Whenever confidentiality is desired, the public key of the receiver should be used to encrypt the message; then, only the receiver's private key can be used to decrypt the message.

To obtain authenticity, or proof of origin—identify with certainty who a message came from—a sender should encrypt the message using the sender's private key. Anybody with the sender's public key can decrypt the message and therefore know without a doubt who sent the message, as the sender is the only person having access to their private key. Though this example does not provide confidentiality, it does provide authenticity or proof of origin —knowing the source of the message.



**Figure 3-49: Using Asymmetric Cryptography for Proof of Origin and Confidentiality**

A good example might include situations where software patches are concerned. When Microsoft publishes patches, end users want to know with certainty that the patch is issued by Microsoft. If Microsoft encrypts the patch with their private key, all end users with Microsoft's public key can decrypt it and know with confidence that the patch they're applying to their servers is a valid vendor patch.

## Hard Math Problems

Two hard mathematics problems are still primarily used for key generation: factoring and discrete logarithms as denoted in [Figure 3-50](#). Note that for each of the examples below, very small numbers are being used, but in reality, cryptography would use significantly larger numbers.

The idea behind **factoring** is that multiplication can be done very quickly and easily—take two large prime numbers and multiply them to come up with a result. However, with only the result on hand, it is very difficult to determine two numbers that were multiplied together to produce that. If two significantly long prime numbers are multiplied together, determining those numbers based upon only the result could take several years. This is the type of math used by RSA for key generation and helps explain why RSA is so effective for purposes of cryptography.

With **discrete logarithms**, a different type of mathematics is being used by all the other asymmetric algorithms, like Elliptic Curve (ECC) and Diffie–Hellman, to name a couple. In this case, any prime number is raised to the power of another prime number to determine a result. As with factoring, this process can be performed very quickly and easily. However, with only the result on hand, it can be extremely difficult to determine what was the prime

number that was raised to the power of another prime number to produce that result.

It should be noted at this point that mention of the use of prime numbers is very specific. A prime number is special in that it can only be divided by itself or by 1. Thus, when factoring is used, there is only one solution to the problem.

Finally, a third hard math problem exists—the Knapsack problem—but it is not used anymore, because attacks have been identified that can break any algorithm that uses it. In other words, attacks can solve the Knapsack problem, which has effectively made any algorithm that uses it insecure and useless. Deprecated algorithms that use the Knapsack problem include Chor Rivest Knapsack and Merkle Hellman Knapsack algorithms.

To summarize:

- Factoring and discrete log asymmetric algorithms depend on using **very large prime numbers**.
- When using such large numbers, it is **very difficult to work backward to determine the original integers**.

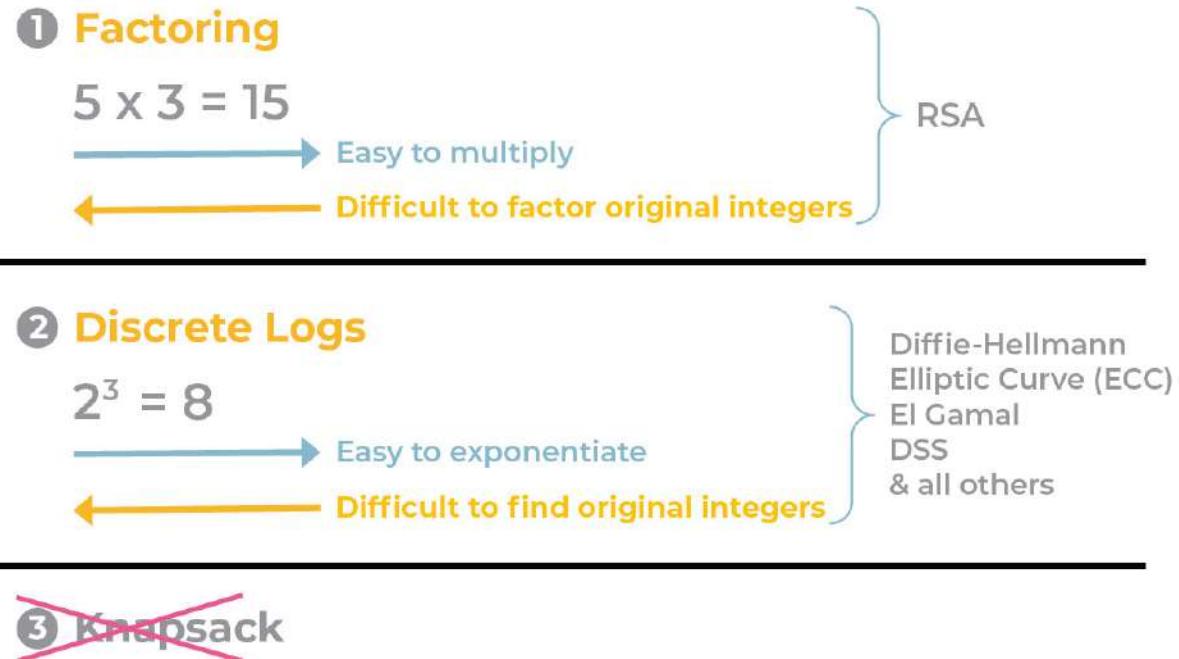


Figure 3-50: **Factoring and Discrete Logarithms Operation**

## Asymmetric Algorithms

### Understand an advantage ECC has over RSA

Table 3-44 contains some of the most popular asymmetric algorithms in use today. Despite having been around since the late 1970s, **RSA**, for example, is still extensively used, because it continues to provide exceptional security. Despite efforts, no significant weaknesses have been identified. This points to the way that hard mathematical problems can be used so effectively and successfully.

**Elliptic Curve (ECC)** was developed in the early 2000s, and it improved upon RSA by allowing for the use of shorter keys to achieve the same level of security. This improvement came because of ECC using discrete logarithm mathematics for key generation instead of factoring. As a result, ECC is faster and more efficient. ECC is particularly useful in situations where bandwidth, computational power, and storage capacity might be limited, e.g., on mobile phones.

**Understand the two primary types of hard math problems used with asymmetric cryptography and which type is used**

Like RSA, the **Diffie–Hellman Key Exchange** was developed shortly before RSA, and it too continues to provide very effective encryption services, though it is almost exclusively used today for the exchange of symmetric keys between parties.

Remember that RSA uses factoring of large prime numbers for key generation, while ECC and Diffie–Hellman both use discrete logarithms.

## Quantum Key Distribution

Quantum key distribution is another emerging technology that aims to solve the key distribution problem. Out-of-band channels aren't always available, and many of our current public-key algorithms are at risk if quantum computers become practical (but NIST is currently working on public-key algorithms that will be safe in a post-quantum world). Describing the physics behind quantum-key exchange is a little outside of our wheelhouse, but you may have heard about how even the act of observing a quantum system ends up changing the system itself. If we take this principle and apply it to key exchange, then in a quantum system, **anyone observing a key exchange—or intercepting a key exchange—ends up impacting it.** Therefore, if anyone is eavesdropping on quantum key distribution, the two parties exchanging keys will know about it. However, quantum key distribution is still fairly experimental at this stage and it has a number of challenges to overcome before we see it implemented widely

Rivest, Shamir, and Adleman (RSA)	Uses <b>factoring</b> mathematics for key generation.
Elliptic Curve (ECC)	Uses <b>discrete logarithm</b> mathematics for key generation. ECC uses shorter keys than RSA to achieve the same level of security, which means ECC is faster and more efficient.

### Diffie–Hellman Key Exchange

Uses **discrete logarithm** mathematics for key generation is **primarily used for the exchange of symmetric keys between parties.**

Table 3-44: Common Asymmetric Algorithms

### 3.6.7 Hybrid Key Exchange

#### CORE CONCEPTS

- Diffie–Hellman Key Exchange (uses discrete logarithms) is an asymmetric algorithm used primarily for symmetric key exchange.
- Hybrid cryptography blends the advantage of symmetric cryptography—extremely fast—with the advantage of asymmetric cryptography—solves the key distribution problem.

### Diffie–Hellman Key Exchange Protocol

#### The value and use of Diffie-Hellman Key Exchange Protocol

Referring to the early discussion about symmetric and asymmetric key cryptography, it was made clear that symmetric key cryptography is the best, and really only, thing to use when speed and bulk processing are required. It is the only type of cryptography that can host the speeds required for being able to encrypt and decrypt fast enough for the applications that require cryptography today. For

example, with a VPN, only symmetric key cryptography can be used to efficiently support the amount of data traversing the network. The fact that symmetric key cryptography is used means that the same key needs to be on each end of the connection, and this presents a challenge—securely communicating the key between a remote user and the corporate network. This challenge is resolved through a bit of mathematical magic, and this method is used by all VPN solutions to generate the same secret at each end for purposes of encryption and decryption during each VPN session. In fact, this is why these keys are called session keys.

**Session keys** are symmetric keys that are used for specific sessions—they are only used for one session, and when a new session is started a new session key will be generated. If a given VPN session ends for any reason, the establishment of a new VPN session would include the negotiation of a new symmetric key. For this key to be agreed to, without sending it across the network where it could be intercepted, the Diffie–Hellman Key Exchange Protocol would be utilized. The way this protocol works is described in [Figure 3-51](#).

Let's imagine that Alice wants to communicate from a remote location with Bob, who is located at the corporate office.

1. Alice's and Bob's systems each generate a random number, 7 and 3, in this example. The numbers are essentially a secret, as they don't know each other's numbers.
2. On each end, the random number is multiplied by the same number, in this example 2. As a result, Alice's number is now 14 and Bob's is 6. Keep in mind that with a real scenario, the actual numbers used would be much more complex, as would be any applied mathematical computations, which would actually entail one-way operations—in other words, operations that could not be easily reversed by an attacker. The numbers here are simply to help illustrate the concept.
3. At this point, Alice and Bob each have numbers related to each other mathematically. Alice's and Bob's systems send the numbers to each other; Alice's system sends 14 to Bob's system, and Bob's sends 6 to Alice's.
4. At this point, they each relate the number they received back to the original random number through a multiplication operation. So, Alice multiplies the number she received (6) with her original number (7), and Bob does the same ( $14 \times 3$ ).

5. Quick math shows the result of the mathematical operation to be 42—the key—on both sides.

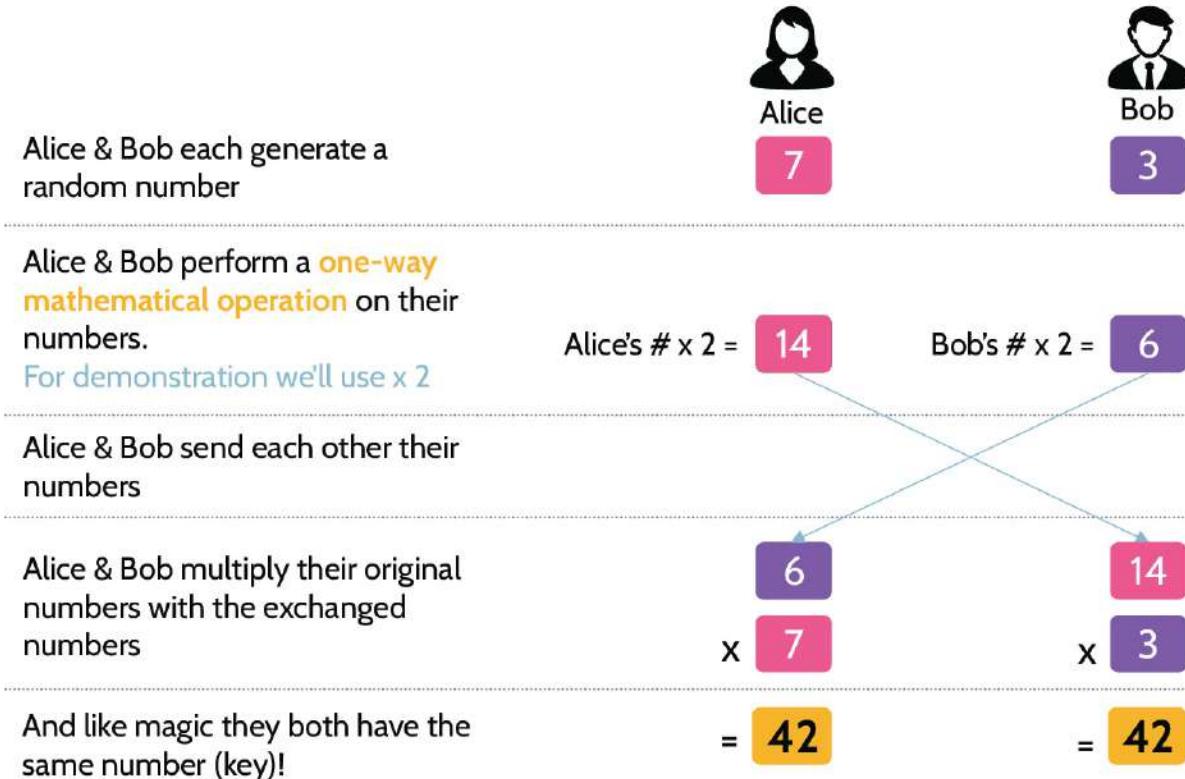


Figure 3-51: Diffie–Hellman Operation

Though simple, this example illustrates how a bit of mathematical magic serves to generate the same secret at both ends, which really represents the symmetric session key that can be used for encryption and decryption purposes. This process explains why Diffie–Hellman is referred to as a key management protocol.

## Hybrid Cryptography

Hybrid cryptography was mentioned earlier in the context of advantages offered by symmetric and asymmetric cryptography. In each case, the advantage of one often counters the disadvantage of the other, which implies that the use of both types in a hybrid fashion can be particularly effective. Remember that symmetric cryptography is extremely fast, but key distribution is a problem that asymmetric cryptography solves but in a much slower manner.

Hybrid cryptography solutions employ the advantages of symmetric and asymmetric cryptography. Symmetric algorithms are used for bulk processing and speed—for anything that requires frequent encryption and decryption and where both need to be done very quickly. Asymmetric algorithms are used to exchange symmetric keys. Additionally, most hybrid solutions incorporate hashing algorithms for purposes of integrity, the ability to create digital signatures for purposes of nonrepudiation, and other features.

[Figure 3-52](#) illustrates how a simple hybrid solution works. In this case, Alice wants to send Bob a very large message. As a result, she can only use symmetric cryptography, and let's assume she chooses to use 3-DES and picks one of the keys that it uses. Alice knows that for Bob to be able to decrypt

the message, he's going to need the exact same symmetric key used by Alice to encrypt the message.

To share the symmetric key securely with Bob, Alice knows that she can encrypt it with Bob's public key and send this to Bob. Because Bob keeps his private key secure and only known to himself, he is the only person who can decrypt the symmetric key sent by Alice. Once decrypted, Bob will then have the same session key, which will allow him to quickly decrypt and read Alice's very large message.

Though simple, this is a great example of hybrid cryptography—a combination of symmetric and asymmetric cryptography.

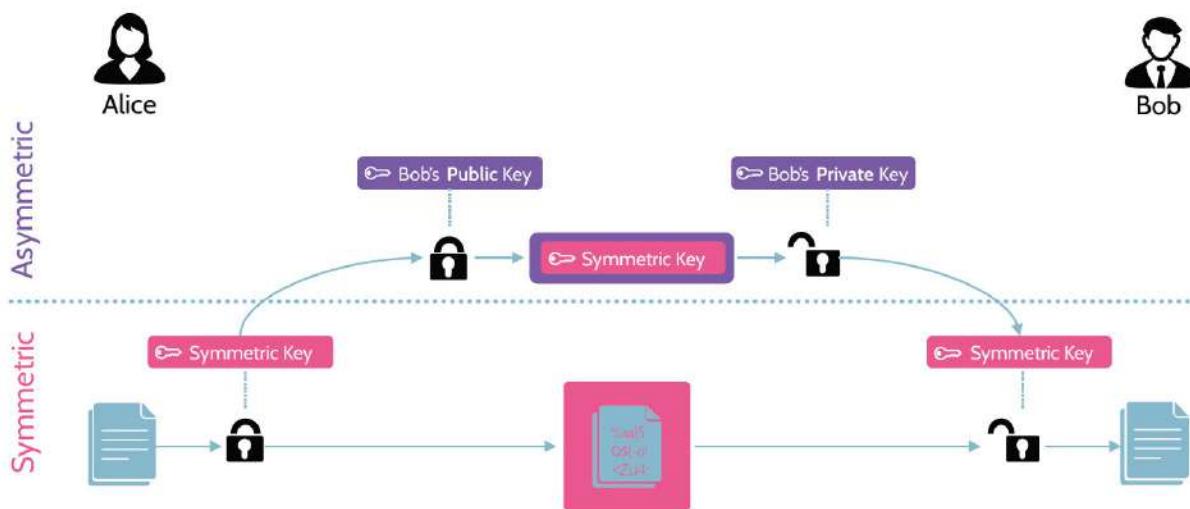


Figure 3-52: Hybrid Cryptography Operation

### 3.6.8 Message Integrity Controls

#### CORE CONCEPTS

- Message integrity checks (MIC) help to ensure the integrity of a message between the time it is created and the time it is read.
- A MIC works by creating a representation of the message, which is sent with the message.
- Message integrity checks are based upon math, some more complex—and therefore more effective—than others.
- The use of simple math can result in a collision, meaning two different messages can result in the same representation.
- Hashing is very effective as a MIC and works the same way, regardless of the length of input; the result is always a fixed length digest, based on the hashing algorithm used.
- The birthday paradox best illustrates how collisions occur and why they should be avoided to maintain integrity.

As a refresher, it has been mentioned that there are five core services that can be provided by cryptography:

- Confidentiality
- Integrity
- Authenticity (proof of origin)
- Nonrepudiation
- Access control

In the context of security and cryptography, integrity means something is intact and unchanged from its original state. **Message integrity checks (MICs)** are designed to ensure that messages remain unchanged from the time of creation to the time they're read. Changes to a message can happen

intentionally or unintentionally, and MICs can help identify when changes have occurred, regardless of the cause. Note that message integrity controls and message integrity checks can be used interchangeably.

From a cryptographic perspective, if one person sends another person a message, something needs to be done to the message first to create a representation of it. Then, when the message is sent, the representation of the message is also sent. The recipient then takes the message, applies the same integrity algorithm as the sender, and compares the resulting representation of the message with the representation included by the sender with the original message. If the two representations match, integrity of the message is confirmed.

Figure 3-53 shows several types of MICs. Each control is based upon mathematics, which differ between each of them. Cyclical redundancy check (CRC), checksum, and Hash Message Authentication Code (HMAC) controls tend to use very simple math, while the controls based on hashing algorithms use much more complex math. The simple math used by these basic message integrity controls is very susceptible to something known as a collision, meaning two different messages can result in the same representation of each message. Collisions are described in more detail below. **Hashing algorithms, on the other hand, are much more sensitive to small bit changes and much more resistant to collisions, and they are therefore much more effective as integrity control mechanisms.**

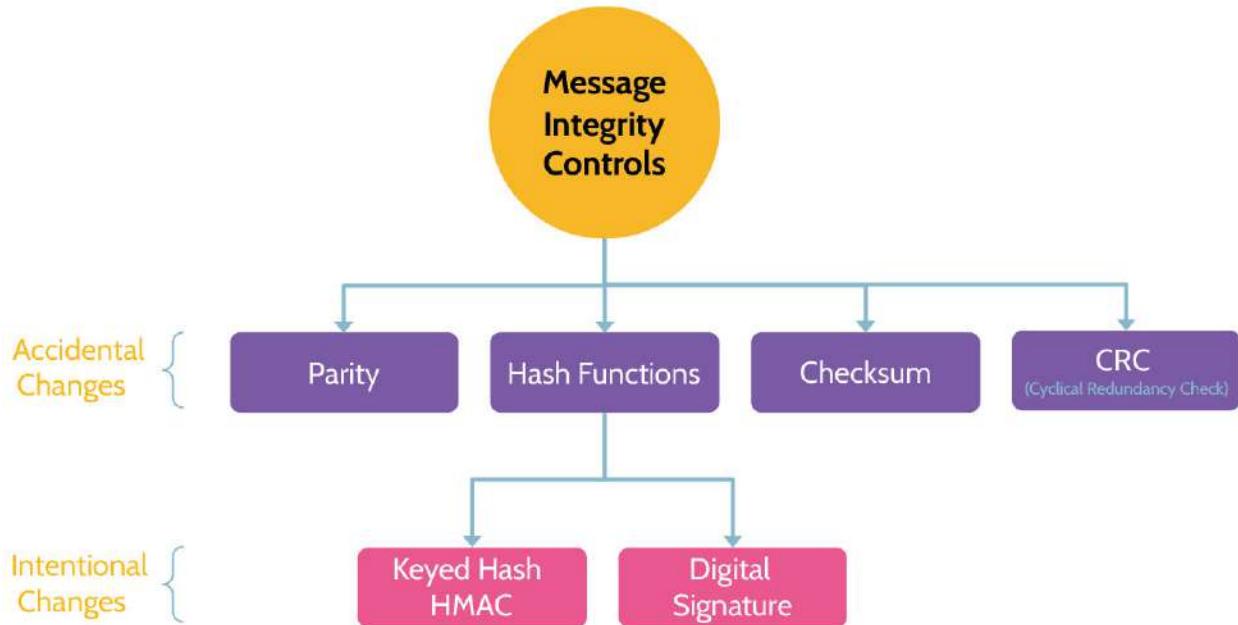


Figure 3-53: Message Integrity Controls

## Hashing

The key elements that make hashing algorithms so effective are outlined in [Table 3-45](#).

<b>Fixed length digest</b>	<p>A message of any length can be hashed, resulting in a fixed length message digest. For example, a 20-terabyte hard drive or a single sentence email could be run through the same hashing algorithm and the results will be digests (hash values) of exactly the same length. The length of digests for common hashing algorithms are listed below. <b>Any length input always equals the same length output.</b></p>
<b>One-way</b>	<p>Hashing relies on one-way mathematical functions that generate an output – a fixed length message digest (hash value) – that represent the input, but it is NOT possible to go backwards and figure out the original input. In short, <b>it is not possible to determine the input of a hashing algorithm by inspecting the output.</b></p>

<b>Deterministic</b>	If the same message is hashed twice using the same algorithm, the message digest will be exactly the same in each case. <b>The same input always equals the same output.</b>
<b>Calculated on entire message</b>	For a message digest to be trustworthy, it must be calculated on the entire message, not simply a portion of it.
<b>Uniformly distributed</b>	A good hashing function should map the inputs as evenly as possible over its output range. In other words, there should be a roughly equal probability of any hash value being generated.
<b>Collision resistant</b>	A collision takes place when two different input values generate the same output. A good hashing algorithm should generate a completely different output even if only a single bit of the input is changed. A hashing algorithm would not be useful to prove integrity if the input could be modified and exactly the same hash value was generated on the original input and the modified input. <b>It should be very hard to find two inputs that hash to the same output.</b>

Table 3-45: Hashing Algorithm Key Properties

## Hashing Algorithms

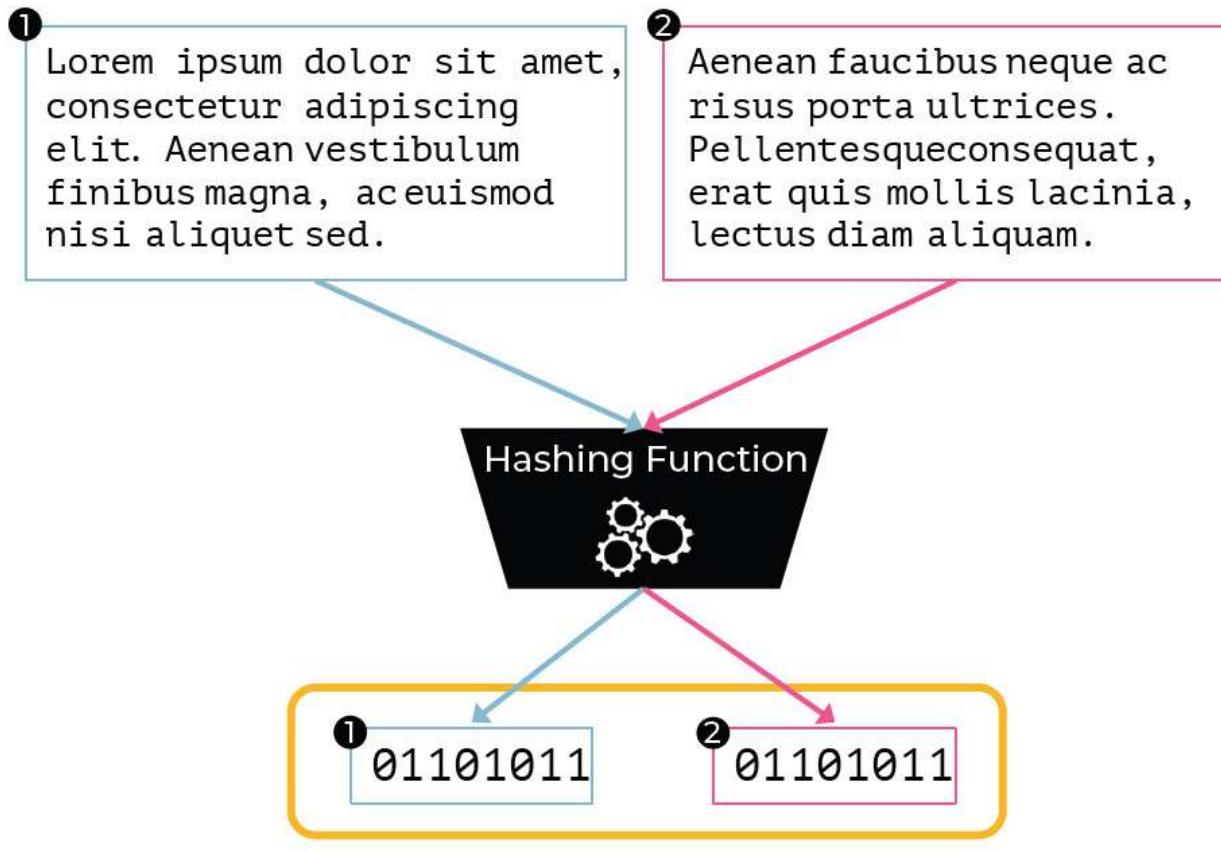
Several of the most popular hashing algorithms are noted below with the length of their outputs.

- **MD5:** 128-bit digest ■ **SHA-1:** 160-bit digest ■ **SHA-2:** 224/256/384/512-bit digests ■ **SHA-3:** 224/256/384/512-bit digests
- MD5 message digests are always 128 bits in length. SHA-1 digests are always 160 bits in length. In the case of SHA-2 and SHA-3, the message digest length is determined by the version used. The longer the digest, the less opportunity for collisions.

## Collisions

Collisions occur when the hash values—digests—from two different inputs are the same. As a result, integrity of a message can't be confirmed. In the case of sensitive communication between two parties, imagine if a man in the middle was able to modify a message in such a way that the digest of the original message and modified message were the same. This could potentially lead to serious consequences, as the receiving party would think the message is uncorrupted.

[Figure 3-54](#) illustrates how a collision occurs. Two different messages exist (#1 and #2), which have different content and length. When they're run through the same hashing algorithm, however, they each result in the same message digest. This is a collision.



The hash values (digests) from two different inputs are the same

Figure 3-54: Collision Example

## Birthday Attack

What's the best way to explain collisions? Something called the “birthday paradox” offers great insight.

As more people enter a room, the chance of any two people having the same birthday (same month and day of birth but not down to the year) grows exponentially. With two people, there's a 1 out of 365 chance of a match. If one more person is added, the probability rises exponentially. Every time one

person is added, the probability doubles. By the time twenty-three people are in the room, there's a 50 percent chance that at least two people share the same birthday. To reach 99 percent, only 60 people need to be in the room.

This same concept and mathematics apply to attacking hashing algorithms, where the goal is to identify collisions. In other words, the goal is to find different messages that produce the exact same digest, meaning a collision is then present. This is referred to as the “birthday attack.”

### 3.6.9 Digital Signatures

#### CORE CONCEPTS

- **Digital signatures provide three services: integrity, authenticity, nonrepudiation.**
- **Digital signature uses include having the same legal significance as a written signature, code signing to verify the integrity and authenticity of software, and nonrepudiation (of origin and delivery).**

Before diving too deeply into the topic of digital signatures, it's worth thinking about why they are so beneficial and necessary. As discussed, hashing is used to provide **integrity**. However, attaching a hash value to some data, such as an email, that is transmitted across a network is insufficient to prove integrity. Imagine sending an important, unencrypted email across the internet and only attaching the hash digest of the message to prove the integrity of the email. Now, further, imagine that a man-in-the-middle (MITM) intercepted the email before it reached the intended recipient. After reading the message, the MITM altered the contents of the email, removed the original hash digest, calculated, and attached a new digest

(based upon the altered message), and sent the email on to the original intended recipient. Upon receiving the message, the recipient would calculate the hash digest of the message, see that it matches with the attached hash digest, and falsely assume that the email has integrity.

This is a huge problem that, thankfully, can be easily resolved with digital signatures, which also provide two additional extremely useful services: authenticity and nonrepudiation.

Digital signatures provide a means by which communication between two parties can be assured to be authentic, have integrity, and be unable to be repudiated later. Specifically, digital signatures provide three services: integrity, authenticity, and nonrepudiation. More details about each service can be found in [Table 3-46](#).

<b>Integrity</b>	To confirm the integrity of a digital signature, the receiver calculates the hash value of the message sent by the sender and then compares this value to the hash value attached in the original message. If the two values match, the receiver knows the integrity of the message that was sent is intact.
<b>Authenticity</b>	Authenticity (or proof of origin) is achieved and confirmed very easily by the sending party encrypting the hash value of the message with their private key and the receiving party being able to decrypt the message with the sender's public key. As soon as the message can be successfully decrypted, authenticity has been achieved and confirmed.
<b>Nonrepudiation</b>	If integrity and authenticity are both achieved, nonrepudiation of origin and delivery has also been achieved. This simply means that the sender cannot deny they sent the message—the receiver knows the sender sent the message (by virtue of use of the sender's private key) and the message

has not been altered in any way (by virtue of the hash values matching)—and the receiver cannot deny receiving the message.

Table 3-46: Digital Signature Services

## Creating Digital Signatures

The process of creating a digital signature is quite easy and fundamentally involves two steps as also shown in [Figure 3-55](#): 1. The sender hashes the message, which produces a fixed length message digest.

2. The sender encrypts the hash value with the **sender's private key**.

Note that because a digital signature is basically just an encrypted hash value with an output of 128/192/256/512 bits, digital signatures are very small in size.

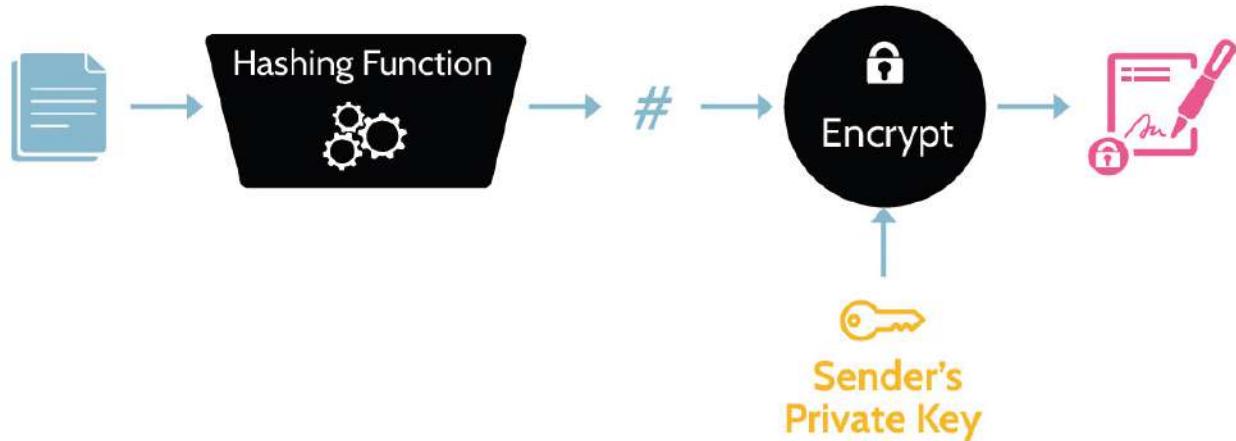


Figure 3-55: Digital Signature Creation

## Using Digital Signatures

Figure 3-56 shows how Alice and Bob might use digital signatures as part of their communication in order to achieve integrity, authenticity, and nonrepudiation.

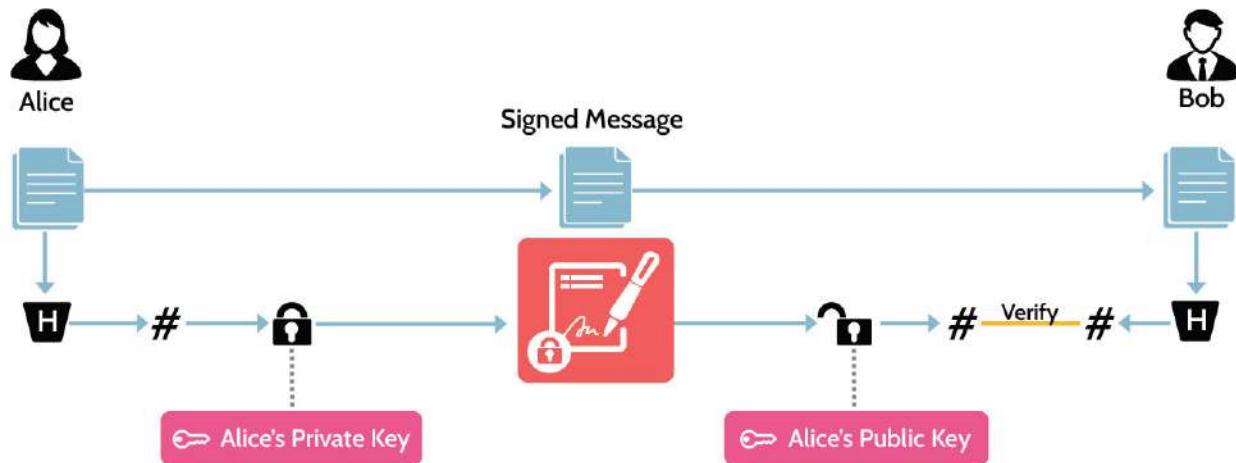


Figure 3-56: Digital Signature Creation

Alice wishes to send a message to Bob, and she desires for it to have integrity, authenticity, and nonrepudiation. To that end, she follows these steps: 1. She writes her message, and she hashes it, thus creating a fixed length message digest.

2. She encrypts the hash value (the message digest) using her private key, now producing a digital signature (similarly to what was depicted in Figure 3-55).
3. She attaches the digital signature to the email and sends it to Bob.

Do note that although Alice has produced a digital signature and sent it as part of the message to Bob, if anybody intercepts that, they'll be able to read the message. A digital signature does not provide confidentiality—anybody could potentially read the message, as that content hasn't been encrypted.

Once Bob receives the message from Alice, he needs to reverse the process by the following steps: 1. Decrypt the digital signature with Alice's public key. Once he's able to do that, authenticity has been proven. Bob knows the message came from Alice.

2. Hash the message he received (using the same hash algorithm used by Alice).
3. Compare the hash value he calculated in step #2 with the hash value received from Alice. If the values match, Bob will know that the message was not altered in transit.

With authenticity and integrity verified, nonrepudiation has also been verified. Alice cannot deny that she sent the message.

## Uses of Digital Signatures

### How digital signatures might be used

Digital signatures can be used in numerous ways. They are commonly used for document signing. In fact, compared to traditional signatures that can potentially be forged, a digital signature could not easily be duplicated,

because this would require access to the sender's private key. These days, even a whole house can be rented just by electronically signing documents provided by the real estate agent.

Another common usage of digital signatures **is code signing**, which allows for the authenticity and integrity of code to be verified. For example, when downloading and installing an operating system update on an iPhone, it is important to know that the software update is from Apple (authenticity) and the update wasn't modified in transit (integrity). This can be achieved by Apple creating and attaching a digital signature to a software update. An iPhone will decrypt the digital signature attached to an Apple software update with Apple's public key—if the digital signature decrypts with Apple's public key, that proves authenticity. Next, the iPhone will hash the software update and compare the hash value generated to the hash value contained in the digital signature—if the hash values match, that proves integrity.

Another example of code signing is for a developer to generate and attach a digital signature to code they are uploading to a code repository, and thus the authenticity and integrity of code can be verified.

### 3.6.10 Digital Certificates

#### CORE CONCEPTS

- **Digital certificates bind an individual to their public key.**
- **All certificate authorities conform to the X.509 certificate standard.**
- **The “root of trust” or “trust anchor” is the foundation of all digital certificates and is represented by a root certificate authority.**

- Digital certificate best practices suggest that public/private key pairs be periodically replaced, which means the associated digital certificate is also replaced.
- When a private key has been compromised, a digital certificate should be revoked by the issuing certificate authority.
- With certificate pinning, when a certificate from a web server is trusted, each subsequent visit to the site does not include a request for a new copy of the certificate.

## How Can We Be Certain We Have Someone's Public Key?

### What does a digital certificate contain?

To confirm with certainty someone's public key, a copy of their digital certificate is needed. Digital certificates bind individuals to their public keys. The issuing **Certificate Authority (CA)** signs an individual certificate with the CA's private key, thereby ensuring the integrity and validity of the certificate and public key being issued.

The process of creating a digital certificate is outlined in [Figure 3-57](#).

1. Alice generates a standardized file that contains information about herself, her company, and her public key.
2. Alice sends the file to a certificate authority, like Entrust, GoDaddy, Network Solutions, or Comodo, to name a few.

3. The certificate authority will first confirm Alice's identity and other information (thus performing identity proofing).
4. The CA will encrypt the information Alice provided with the CA's private key, which creates Alice's digital certificate.

Who can decrypt a given CA's digital certificate? Anybody who has the CA's public key can decrypt the digital certificate, and virtually every web browser and computer operating system include the public keys of the major global certificate authorities. Thus, virtually everybody can decrypt one of the big CA's digital certificates.

When exchanging public keys with others, therefore, the best practice is to do so by sending them a copy of the digital certificate that contains the public key. Doing so ensures that an attacker in the middle, even if they're able to intercept the digital certificate, can't replace the owner's public key with the attacker's public key. Only a certificate authority can modify a digital certificate, because the CA's private key is needed to do so.

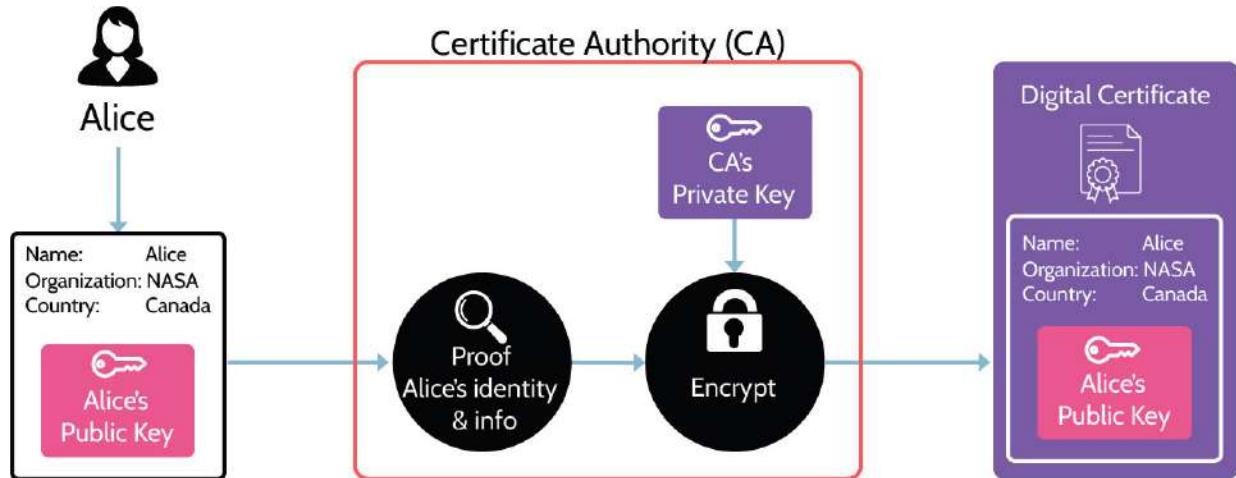


Figure 3-57: Digital Certificate Creation

## X.509—Digital Certificate Standard

### What is the digital certificate standard?

For purposes of interoperability and ease of understanding by browsers, each CA needs to create certificates that are consistent in their formats. As such, all certificate authorities conform to the X.509 certificate standard, which contains fields like certificate version, serial number, encryption algorithm, issuing CA, validity period, and public key value.

## Root of Trust

### Understand the relationship between root, intermediate, and issuing CAs

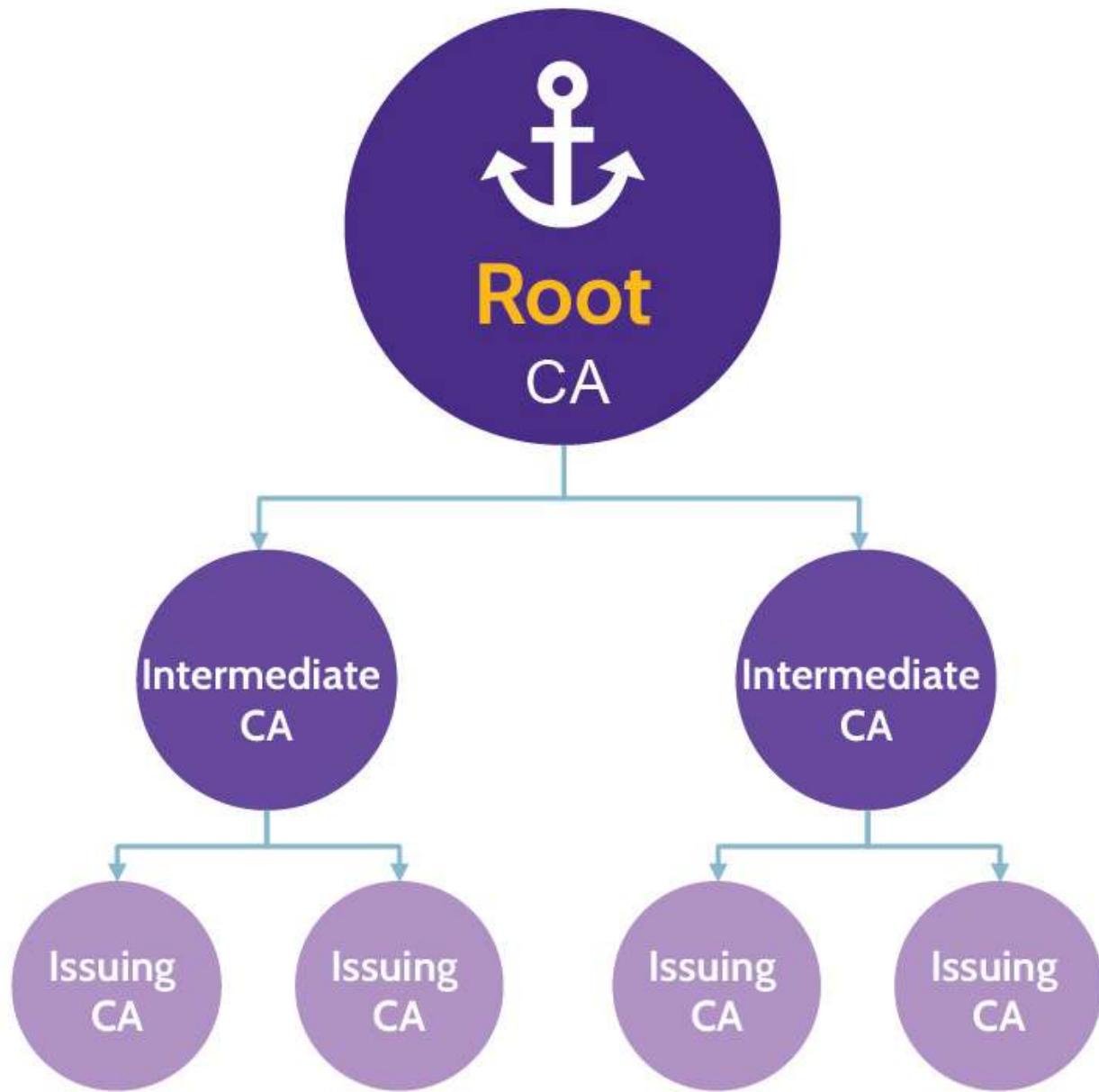


Figure 3-58: CA Hierarchy

Root of trust, or the **trust anchor**, is the foundation of digital certificates' integrity. In the case of every digital certificate, ultimately a root CA's key is used to sign the certificate, but oftentimes intermediary CAs (Subordinate

CAs) act as proxies and sign and issue digital certificates on behalf of the root CA as seen in [Figure 3-58](#).

From a security perspective, the reliability of the entire system is dependent on the security of the root CA's private key. If this key were ever compromised, significant and far-reaching global damage could take place, and the entire system could fall apart. The best protection against something like this is to keep the root CA offline or inaccessible and to use intermediate CAs to issue certificates.

The Root CA self-signs its certificate, and it is then used to sign subordinate certificates, usually known as intermediate CAs. **Intermediate CAs** can also sign certificates, shown as **Issuing CAs**. These issuing CAs would be the ones used to sign entity-level CAs—those used by organizations when applying for a digital certificate. If one of these CAs is compromised, then the damage can be contained by revoking only the certificates issued by them, which would be far less than those that would be signed by a root CA if it was constantly kept online.

[Figure 3-59](#) shows an example of the root CA, or chain of trust, related to Amazon's digital certificate. That's signed by a subordinate CA (DigiCert Global CA G2), which in turn is signed by the root CA, DigiCert Global Root G2.

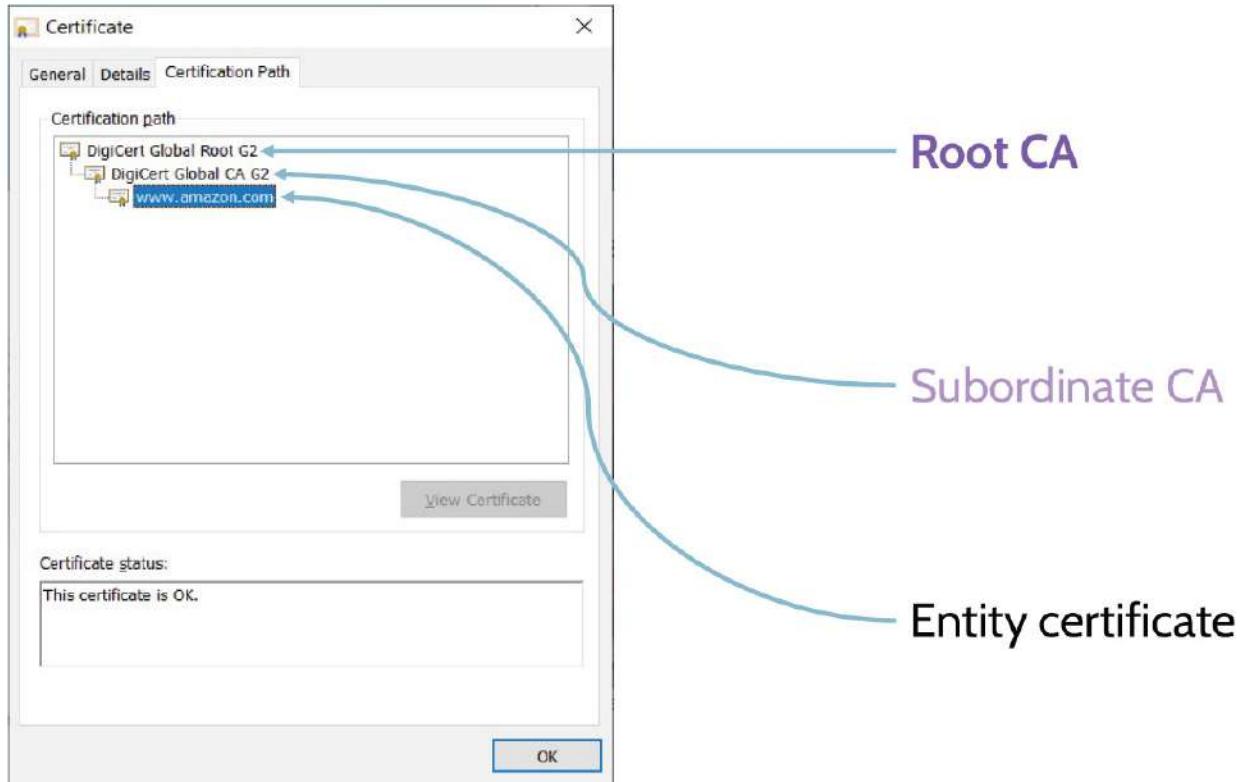


Figure 3-59: Amazon's Digital Certificate

## Digital Certificate Replacement and Revocation

A best practice related to key management is periodic replacement or rotation of keys, and the same applies to management of digital certificates. If a public/private key pair is being replaced, this means that the accompanying digital certificate would also need to be replaced. As part of normal business, digital certificates are issued with an expiration date, and a given certificate should be replaced on or before the expiry date in order to maintain continuity of operations.

However, in cases where a private key may have been compromised, all trust in the associated public key should cease, and the certificate revocation

process should be followed. Certificate revocation involves contacting the issuing CA, explaining the situation, and asking that the digital certificate associated with the key pair be revoked.

Of the two situations, revocation is much worse, because it implies that a private key has been compromised and nobody should trust the public key.

Digital certificate replacement and revocation are summarized in [Table 3-47](#).

Replacement	Revocation
Regular replacement of <b>expired</b> certificates	Replacement of certificate when associated private key has been <b>compromised</b>

Table 3-47: Digital Certificate Replacement and Revocation

## Revocation Confirmation Methods

**Understand how the revocation status of a certificate can be checked and the difference between CRL and OCSP**

Continuing the discussion from above, once a digital certificate has been revoked by a CA, there are two primary ways to confirm the revocation. The older method for checking involves what's known as the **Certificate Revocation List (CRL)**. In this case, an organization would contact a CA and ask if a particular certificate has been revoked. In turn, the CA would send the organization a list of all revoked certificates from the CA. This list

can be quite extensive. Once received, the organization's cryptosystem would search the entire list to determine if the certificate in question has, in fact, been revoked. This is not an efficient process, as it requires a large volume of data to be often transmitted to be able to identify revoked certificates.

The better, newer method involves **Online Certificate Status Protocol (OCSP)**. With this method, an organization's system will query the CA, asking if a particular certificate has been revoked, and the CA will reply with a simple yes or no.

This said, even when a certificate has been revoked and a browser indicates the same, users will oftentimes ignore warnings and proceed to a website that may be malicious or compromised. User awareness is still a key component of protecting an organization.

CRL and OCSP are summarized in [Table 3-48](#).

Certificate Revocation List (CRL)	Online Certificate Status Protocol (OCSP)
Client downloads and searches list of serial numbers of all revoked certificates from the CA	Client queries CA for revocation status of specific certificate serial number

**Table 3-48: Revocation Confirmation Methods**



## Certificate Life Cycle

A certificate's life cycle includes a number of distinct phases presented in [Table 3-49](#).

<b>Enrollment</b>	To be issued a digital certificate, an entity must first submit a request for certificate to a certificate authority (CA). This request is usually in the form of a CSR, or certificate signing request, that requires certain identifying fields to be completed for the process to move forward. As part of this process, the requesting entity also generates a public/private key pair. The private key is usually stored in the entity's local certificate store, and the public key is included as part of the CSR.
<b>Issuance</b>	Upon receiving a valid CSR, a registration authority follows a process known as <b>identity proofing</b> , where the validity of information included in the entity's CSR is confirmed. This process is typically quite thorough, as the CA wants to ensure the legitimacy of the entity requesting the certificate.  After completing this process, the CSR will be effectively signed by the root of trust (root CA) private key, following the X.509 standard, and the

	<p>certificate will be issued by an intermediate/issuing CA.</p> <p>Upon issuance, the entity will store the digital certificate in their certificate store.</p>
<b>Validation</b>	<p>When a certificate is invoked, as part of web browsing or ecommerce transactions, the <b>validity of the certificate</b> is typically automatically confirmed with the issuing CA. This process specifically confirms if a certificate has been revoked or is expired, and if either case exists, a warning is issued, and many browsers today will block access to a website or prevent the transaction from proceeding.</p> <p>Validity of a certificate can also be performed manually by checking the CA's certificate revocation list (CRL), a list of digital certificates that have been revoked by the issuing CA) or by using the online certificate status protocol (OCSP), an internet protocol that enables automated lookup of the status of an issuing CA's digital certificates.</p>
<b>Revocation</b>	<p>For any of a number of reasons, a certificate might become invalid and need to be revoked. For example, through improper key management or malicious activity, an entity's private key might become compromised, or the enrollment process could involve incorrect validation of information and issuance. In cases like these and others, a certificate will be revoked and added to the issuing CA's revocation list, which can be queried to verify a certificate.</p>
<b>Renewal</b>	<p>All certificates are issued with an expiration date, typically in increments of twelve months from the time of issuance. Prior to expiration, an entity will receive a notice of expiration, at which time the entity can <b>renew the certificate</b>. If an expiration date passes, a warning will be issued to any users who visit the site where the expired certificate is in use. Relative to issuance, renewal of a certificate is a relatively simple process that typically involves confirming the information included as part of the original CSR.</p>

**Table 3-49: Certificate's Life Cycle**

## Certificate Pinning

### What is certificate pinning?

When a user visits a website, like example.com, the web server sends the server's certificate to the browser by virtue of the browser requesting it. However, if a malicious actor is sitting between the web server and the user, the malicious actor might send a spoofed certificate to the user when the browser makes the request. Certificate pinning offers a means by which this possibility can be avoided.

With certificate pinning, when a certificate from a web server is trusted, each subsequent visit to the site does not include a request for a new copy of the certificate. There are two primary ways to accomplish certificate pinning: 1. When an application is first created, by coding the certificate into the application itself. Thus, if the certificate is already pinned to the application, there is no need to request a copy of the certificate from the server.

2. The very first time a website is visited, the certificate obtained from the initial request is pinned to the browser, and no subsequent requests are required during future visits to the same site.

Fundamentally, certificate pinning involves no key distribution and thus alleviates any concerns relating to it.

### 3.6.11 Public Key Infrastructure

#### CORE CONCEPTS

- **Public key infrastructure (PKI) is the basis for keys to be distributed and owners of public keys to be verified.**
- **The standard used to create all digital certificates is X.509.**
- **PKI consists of several components: certificate authority (CA), registration authority (RA), intermediate/issuing CA, certificate DB, certificate store.**
- **The root of trust in any PKI is the CA, which ultimately issues certificates.**

#### Understand the major components of PKI and the certificate life cycle

Public key infrastructure (PKI) is the entire suite of technology systems that allows keys to be distributed and owners of public keys to be verified.

[Figure 3-60](#) highlights the major components of PKI, which have also been summarized in [Table 3-50](#).

In the left pane, you see Alice, who desires to have her own public and private key pair. She generates this pair of keys using an application on her computer and stores them in her machine's certificate store. It's especially important that her private key be kept very secure, because Alice is the only person who should ever have access to it. Looking at the lower right pane, we see Bob as the relying party with whom Alice wants to communicate. To

communicate with Bob, Alice must send him her public key. However, if she simply sends her public key, there's a risk: her key could be intercepted and replaced with an attacker's public key. Instead, Alice will send her digital certificate.

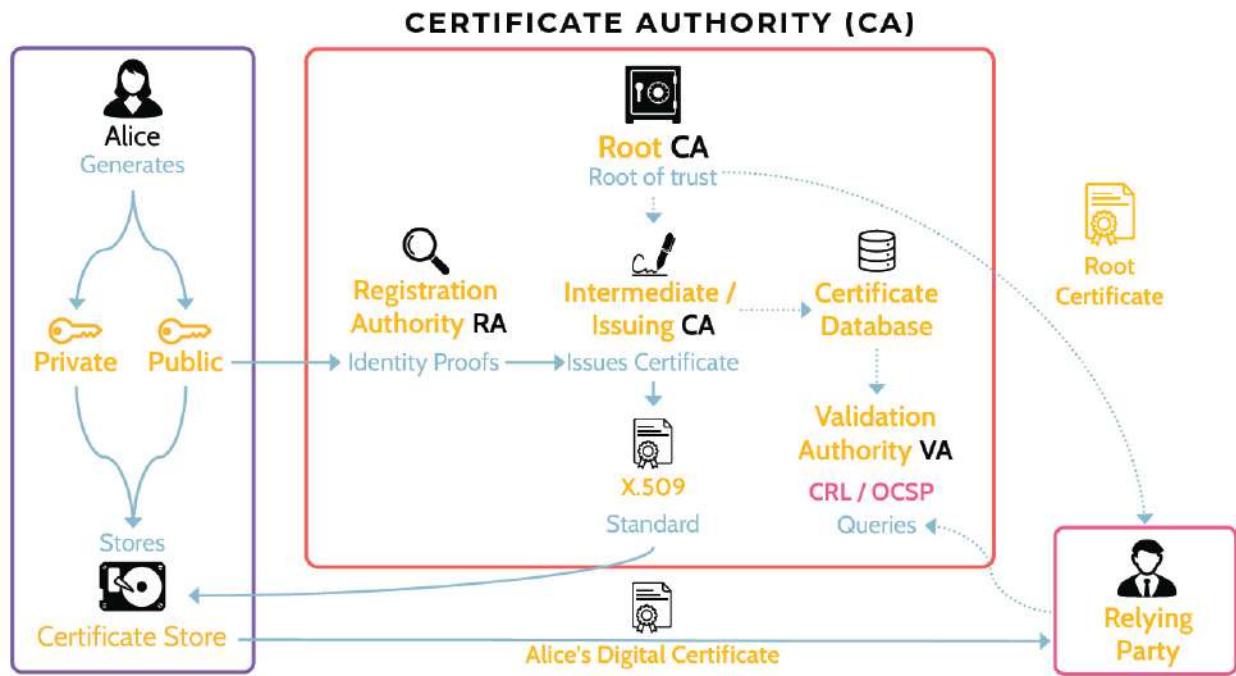


Figure 3-60: PKI Operation

To obtain a digital certificate that contains her public key, Alice must utilize a certificate authority (CA). A **Registration Authority (RA)** will ask Alice for some identifying information as well as a copy of her public key and it will “**identity proof**” Alice to confirm she is, in fact, Alice. After this step, Alice’s public key will be signed by the root CA’s private key. The root CA is known as the root of trust, but the signing process is usually facilitated through an intermediary known as the intermediate or issuing CA. Once signed, the intermediate/issuing CA issues the certificate.

Once issued, Alice can store a copy of the certificate in her Certificate Store, and she can send a copy of the certificate to Bob. As time passes, Bob might want to confirm that Alice's certificate is still valid or learn if it's been revoked. To do this, Bob can query the Validation Authority (VA) entity of the certificate authority via one of two prementioned protocols, CRL or OCSP. Each certificated authority maintains a large database, known as the **Certificate Database**, of all the certificates they have issued, and which have been revoked.

Finally, for Bob to decrypt Alice's certificate, he's going to need the root CA's public key (which can be obtained from the root CA's certificate), which is typically preinstalled in commonly used operating systems and web browsers.

<b>Certificate Authority (CA)</b>	Root of trust
<b>Registration Authority (RA)</b>	Identity proofs on behalf of CA
<b>Intermediate / Issuing CA</b>	Issues certificates on behalf of CA
<b>Certificate DB</b>	List of certificates issued by CA and revocation list
<b>Certificate Store</b>	Repository of certificates and user's private key on user's computer

Table 3-50: **PKI Components**

Note that without a PKI it is still possible to encrypt and send data, but you cannot verify the identities of the other participating parties. In other words,

without a PKI, it's impossible to entirely trust the digital identity of another entity or person.

### 3.6.12 Key Management

#### CORE CONCEPTS

- Proper key management is paramount to the security of any cryptographic system.
- Kerckhoffs' principle
- Key management activities: generation/creation, distribution, storage, change/rotation, disposition/destruction, recovery

As already mentioned, key management is incredibly important. If a key is secure, the underlying cryptographic system is secure. In other words, an attacker can know the ciphertext, the algorithm, the IV, and everything else about the system, and if the key remains secure, the system is secure. This concept was formalized by Auguste Kerckhoffs, a Dutch cryptographer, in the nineteenth century and is known as Kerckhoffs' principle.

#### Kerckhoffs' Principle

##### Understand key management activities

Kerckhoffs' principle states that *a cryptosystem should be secure even if everything about the system, except the key, is public knowledge.*

[Table 3-51](#) contains numerous key management activities, which are also explained in more detail later.

<b>Key Creation / Generation</b>	The key generation/creation process includes the following attributes: <ul style="list-style-type: none"><li>■ fully automated process, because a manual process would likely lead to patterns being present</li><li>■ keys are randomly chosen from the entire available key space, which helps avoid patterns, and pseudorandom number generators are typically employed for this very purpose</li><li>■ asymmetric keys are much longer than symmetric keys</li></ul>
<b>Key Distribution</b>	Key distribution is the practice of securely distributing keys. Methods used could include: <ul style="list-style-type: none"><li>■ out-of-band distribution (although this is not very efficient)</li><li>■ key wrapping using key encrypting keys (KEK)</li></ul>
<b>Key Storage</b>	Key storage is one of the most critical—if not the most critical—aspects of securing a cryptographic system. Two types of systems are utilized for key storage: <ul style="list-style-type: none"><li>■ Trusted Platform Module (TPM)</li><li>■ Hardware Security Module (HSM)</li></ul>
<b>Key Change / Rotation</b>	Key change/rotation refers to how often encryption keys should be replaced.
<b>Key Destruction / Disposition</b>	Key disposition refers to how keys are handled, especially in instances where data in the cloud is concerned. Two primary methods of key disposition/destruction are most often used: <ul style="list-style-type: none"><li>■ crypto shredding</li><li>■ key destruction</li></ul>
<b>Key Recovery</b>	Key recovery refers to techniques used to recover a key. Three primary techniques exist: split-knowledge, dual control, and key escrow.

**Table 3-51: Key Management Activities**

## Key Creation/Generation

Key creation/generation focuses on creating keys. Ideally, this process should be fully automated. In other words, people should not attempt to generate their own keys in order to avoid a significant weakness: patterns. Humans are not very adept at creating true randomness, and anything that contains patterns in cryptography is prone to being deciphered very quickly. Thus, the use of computer systems can help avoid this issue, though even computer systems use what are known as pseudorandom number generators, which are not truly random numbers. However, for purposes of cryptography, a pseudo randomly generated number will not likely demonstrate a pattern for too many years to count.

In addition to utilizing an automated key creation process, keys should be randomly chosen from the entire key space. For example, DES has a key of 56 bits, which gives it a key space of  $(2^{56})$  or approximately 72 quadrillion keys. When choosing a key from that space, it should be chosen in a truly random manner instead of in order of creation or incrementally.

## Key Distribution

After keys have been generated, they need to be distributed, and two primary methods of key distribution exist: out-of-band and key wrapping.

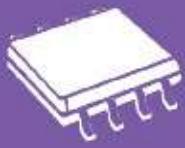
Out-of-band refers to the practice of using a different way to communicate the key than the method being utilized to exchange messages. In other words, if people are communicating via the internet, out-of-band could be any of the following, to name a few examples:

- in person
- phone call
- letter
- Key

wrapping, better known as key encryption keys (KEK), operates just as the name suggests. A key is “wrapped” inside another key. For example, imagine two people (Amanda and John) know they’ll be communicating quite a bit over the course of a week, and they want to use a symmetric key to encrypt their communications. Additionally, for extra security, they want to change the symmetric key every hour. Obviously, many keys will be required to meet this requirement. Something like Diffie–Hellman Key Exchange could be used to share the keys, but this would require quite a bit of work. A better solution would be to generate the number of keys needed, about two hundred for this example so extra keys are also available, and then use Diffie–Hellman to generate the same symmetric key on each side. Then this one symmetric key can be used to “wrap” and exchange the two hundred keys. This is key wrapping—many keys are encrypted with another key.

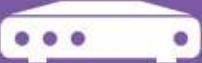
## Key Storage

By now, it should be clear that key storage is one of the most critical—if not the most critical—aspects of securing a cryptographic system. Two types of systems are utilized for key storage: the Trusted Platform Module (TPM) and Hardware Security Module (HSM). [Table 3-52](#) summarizes their characteristics.

 <b>Trusted Platform Module (TPM)</b>
---

A TPM is a tiny microchip installed on the motherboard of laptops and servers that stores encryption keys for the system on which it is installed. In other words, a given system’s certificate store is storing the keys in the TPM module.

This is important, especially as it relates to full drive encryption, which means the entire hard drive is encrypted. If the keys were stored on the hard drive and somebody stole the drive, they would have access to the

	<p>ciphertext—the encrypted hard drive—as well as the key to decrypt it. This explains why a TPM module is installed on the motherboard and essentially operates as a self-contained unit for the sake of cryptographic operations related to the device on which it is installed.</p> <p>To summarize, <i>a TPM is a secure computer chip (crypto processor) built into machines (e.g., laptops), which stores encryption keys and certificates.</i></p>
 <p><b>Hardware Security Module (HSM)</b></p>	<p>Unlike a TPM, which stores encryption keys for a single device, a Hardware Security Module (HSM) stores encryption keys for an entire organization. An HSM essentially looks like a server and is connected to a network, though it is a very hardened device. It's locked down so tightly because its sole purpose is to generate and store encryption keys for an organization.</p> <p>To summarize, <i>a HSM is a physical device, typically connected to an organization's network, specifically built to securely store and manage encryption keys for the organization.</i></p>

**Table 3-52: Key Storage Techniques**

## Key Rotation

The concept of *key rotation* refers to how often encryption keys should be replaced. The value of the asset is what commonly dictates the frequency of key rotation. The more valuable the asset, the more frequently the key should be rotated.

## Key Recovery

There are three main methods to perform key recovery as noted in [Table 3-53](#).

<b>Split Knowledge</b>	<p>Could be as simple as writing the key out on a piece of paper, cutting the paper in half, or into thirds, and then giving the other pieces to other people. Knowledge of the key is split among two or more parties.</p>
<b>Dual Control</b>	<p>Imagine a scenario that involves a nuclear missile. To launch it, two keys must be turned at the same time. Have you ever seen <i>Crimson Tide</i>? Both the submarine captain and the executive officer needed to turn their launch keys for a missile launch to take place. So, there's at least two different controls that must happen at the same time. With encryption keys, this is often implemented using tiny vaults. The backup encryption keys are stored in a vault, which can only be unlocked via the presence of two individuals. This is a form of dual control</p>
<b>Key Escrow</b>	<p>The term <i>escrow</i> refers to a trusted third party, usually in the context of significant financial transactions. Anyone who has bought or sold a house has likely dealt with escrow, because the seller doesn't want to give the keys to the buyer until money is in the bank, and the buyer doesn't want to give the money until the keys are in hand. So, a trusted third-party acts on behalf of both parties and accepts the keys and the money on behalf of each party. Once in possession of both, it hands the keys to the buyer and the money to the seller.</p> <p>Key escrow means encryption keys are stored with a trusted third party. It's very commonly used with cloud computing. In some countries, encryption keys must be shared with the government in order to conduct business there.</p>

**Table 3-53: Key Recovery Methods**

## Key Disposition

Key disposition, or key destruction, is a very important matter. Let's examine this in the context of cloud, where data is actually stored across multiple servers and hard drives. Imagine that some of this data is sensitive, like PII. Now imagine that the organization storing this data desires to move to a

different cloud service provider. After moving the data, say, from AWS to Azure, privacy laws dictate that the data previously stored on AWS must be securely destroyed. Secure destruction means the organization can prove the data has, in fact, been removed from AWS. The best way to securely destroy data is to physically destroy the media used to store the data. This points to things like shredding, melting, or otherwise rendering every part of a hard drive unusable.

However, in this case, simply asking Amazon to delete the data is not enough. You can't be sure the vendor will do that, especially when other customers have data on those same hard drives and that data is likely spread across numerous drives.

**Crypto shredding** is another key disposition method that can be used, especially if the data is stored in the cloud. You can choose a strong encryption algorithm and encrypt that data and then destroy the key, which would mean that the data is effectively destroyed. Nobody can read it, because it's been strongly encrypted, and the decryption key has been destroyed.

### 3.6.13 S/MIME

#### CORE CONCEPTS

- **S/MIME is a standard for public key encryption and provides security services for digital messaging applications.**
- **S/MIME requires the establishment or utilization of a public key infrastructure (PKI) in order to work properly.**

Secure/Multipurpose Internet Mail Extensions—better known as S/MIME, is a standard for public key encryption and provides a number of security services for digital messaging applications. Basic security services offered by S/MIME are:

- Authentication
- Nonrepudiation of origin
- Message integrity

■ Confidentiality S/MIME also offers optional security services, including:

- Signed receipts
- Security labels
- Secure mailing lists
- Extended method of identifying the signer's certificate(s)

S/MIME's popularity grew out of necessity, as internet email has evolved from a simple platform capable of handling text-based messages to a much more complex platform capable of handling digital images, files, sound clips, and other forms of multimedia. This evolution of internet email technology coincided with a shift in design and use, from a relatively small community of trusted colleagues at universities and government agencies to the global community. Due to the nature of internet email usage during the early years, the need for security was minimal, and it was not designed into solutions.

As usage expanded to include millions of users around the world as well as different types of digital information beyond simple text-based messages, Multipurpose Internet Mail Extensions (MIME) was employed. MIME does not address security issues, but security features were developed and added to MIME to create S/MIME.

S/MIME adds features to email messaging, including:

- Digital signatures for authentication of sender
- Encryption for message privacy
- Hashing for message integrity and nonrepudiation of origin To use these features, a public key infrastructure (PKI) must be in use to support senders and recipients of S/MIME messages.

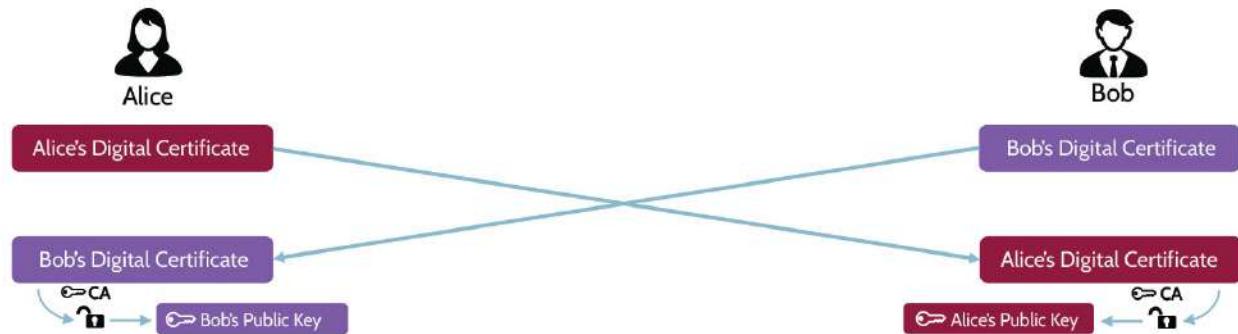
## Putting it All Together

Let's now walk through an example that shows all five services of cryptography in use and demonstrates where it's best to use symmetric cryptography vs. asymmetric cryptography, and digital signatures vs. digital certificates. In this example, Alice and Bob have never communicated before, and Alice wants to send a very large file to Bob. In sending this file, Alice wants to achieve:

- Confidentiality,
- Integrity,
- Authenticity (Proof of Origin),
- Nonrepudiation of origin and nonrepudiation of delivery,
- and Access control

**Step 1.** Alice and Bob need a copy of each other's public keys. But recall, they can't just send each other their public keys, because they couldn't verify

they had received each other's public key. Therefore, the first thing Alice and Bob exchange are their digital certificates to give each other a verifiable copy of their public key, as depicted in [Figure 3-61](#).



**Figure 3-61: Verifying Public Keys**

**Step 2.** Alice wants to send a very large document to Bob, and she wants it to remain *confidential*. Thus, she needs to encrypt the file, and because it is a very large file, she needs to use a fast and efficient encryption method. Symmetric key cryptography provides the solution. She needs to encrypt the file using a symmetric encryption algorithm, such as AES. She now faces the Achilles' heel of symmetric algorithms: key distribution. She needs to somehow securely transmit a symmetric encryption key to Bob.

Alice selects a symmetric encryption algorithm and generates a symmetric encryption key. To send this symmetric encryption key with confidentiality to Bob, she encrypts the symmetric key with Bob's public key. This ensures that only Bob's private key can be used to decrypt the copy of the symmetric key and solves the problem of key distribution. Alice can send the encrypted symmetric key to Bob, and the only person in the world that should be able to decrypt is Bob with his private key, as depicted in [Figure 3-62](#).



**Figure 3-62: Symmetric Key Distribution**

**Step 3.** Alice can now encrypt the file with the symmetric algorithm she selected—the same symmetric key she sent to Bob. Encrypting the plaintext of the file generates a large file of ciphertext that Alice sends to Bob. Bob can then decrypt the file using the symmetric encryption key he received from Alice in Step 2, as depicted in [Figure 3-63](#).

By encrypting the file and sending Bob the ciphertext and the symmetric key in Step 2, Alice has achieved two services of cryptography here in Step 3: 1. **Confidentiality:** the large file was sent encrypted, thus providing confidentiality 2. **Access control:** By Alice choosing who she sends the

symmetric key to and the ciphertext, she can control who can decrypt and thus access the file. This is a form of access control as she is allowing Bob to access the encrypted large file.

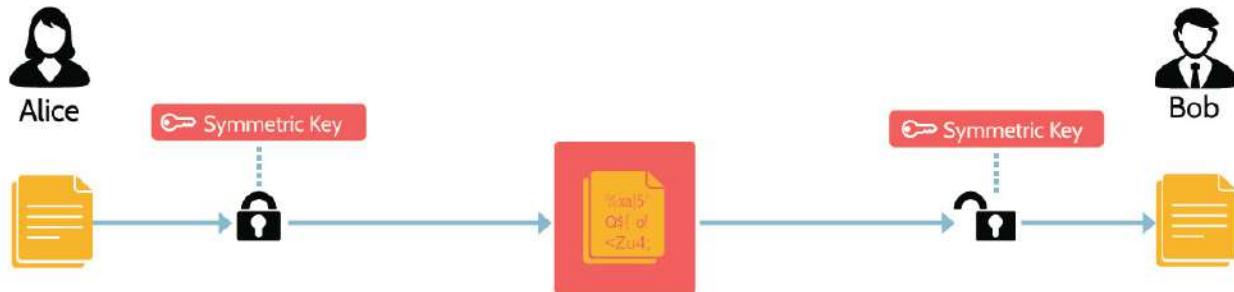


Figure 3-63: Symmetric Encryption and Decryption

**Step 4.** Alice also wants Bob to be able to verify the integrity and authenticity of the file, and she wants nonrepudiation of origin. To achieve these three services, Alice needs to generate a digital signature for the file. To accomplish this, she first hashes the file and encrypts this hash value with her private key, thus generating her digital signature for the file. Next, she sends her digital signature to Bob. Upon receiving Alice's digital signature, he needs to decrypt it with Alice's public key. If Alice's digital signature decrypts with her public key, that proves **authenticity, otherwise referred to as “proof of origin.”** Only Alice could have created her digital signature with her private key, and this proves it came from her.

When Bob decrypts Alice's digital signature, he gets a copy of the hash value that Alice generated for the file. Bob now needs to hash the file that he received and compare the hash value from Alice's digital signature to the hash value that he just calculated. If they match, that proves **integrity**. Bob now knows he received exactly the same file that Alice sent.

Additionally, Bob now has nonrepudiation of origin for the file—Alice cannot deny she sent the file. Bob knows the file came from Alice via the **authenticity**, and he knows it was exactly the file Alice sent via the **integrity** and thus **nonrepudiation of origin**, as depicted in Figure 3-64.

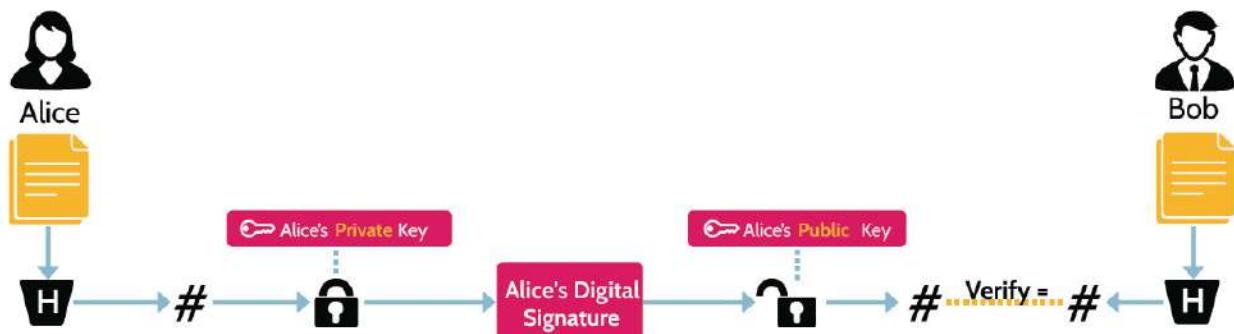


Figure 3-64: Alice's Digital Signature

**Step 5.** The final cryptographic service that Alice wants to achieve is nonrepudiation of delivery. She wants to know that Bob received exactly the file she sent, and she doesn't want him to be able to deny that he received the exact same file. To achieve nonrepudiation of delivery, Bob will need to create and send his own digital signature for the file back to Alice. To create his digital signature, Bob encrypts the hash value for the file he calculated in Step 4 with his private key. He then sends his digital signature to Alice. Alice decrypts Bob's digital signature with Bob's public key and gets a copy of the hash value that Bob generated for the file. If Bob's digital signature decrypts with his public key, that proves **authenticity (proof of origin)**—Alice knows the digital signature came from Bob. The final step is for Alice to compare the hash value from Bob with the hash value for the file she calculated in Step 4. If the hash values match, then she knows Bob received exactly the same file she sent—**nonrepudiation of delivery**, as depicted in [Figure 3-65](#).

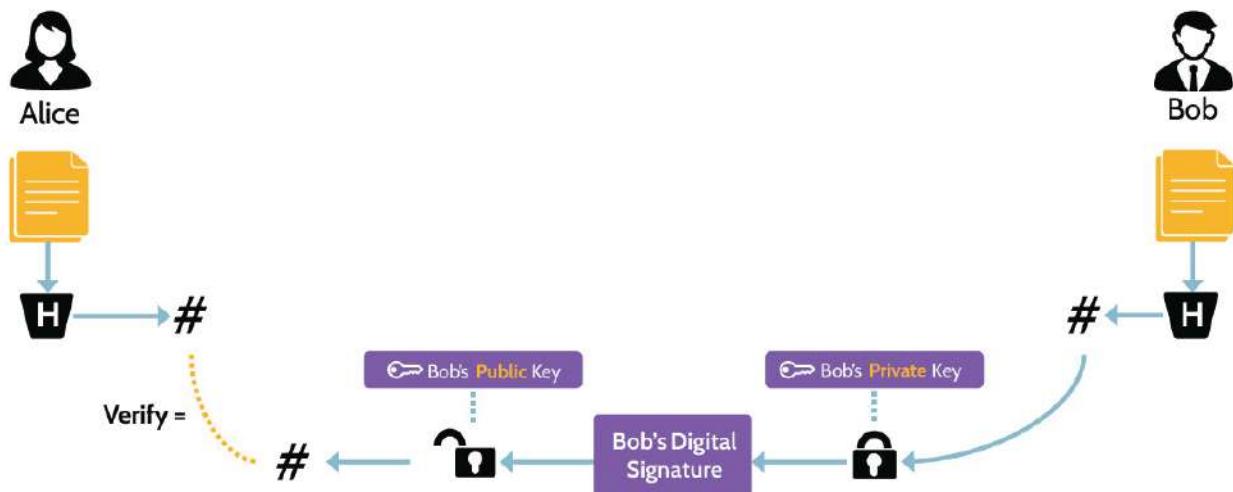


Figure 3-65: **Bob's Digital Signature**

[Figure 3-66](#) illustrates all these steps together to achieve all five services of cryptography: confidentiality, integrity, authenticity (also referred to as proof of origin), nonrepudiation (of origin and delivery), and access control.

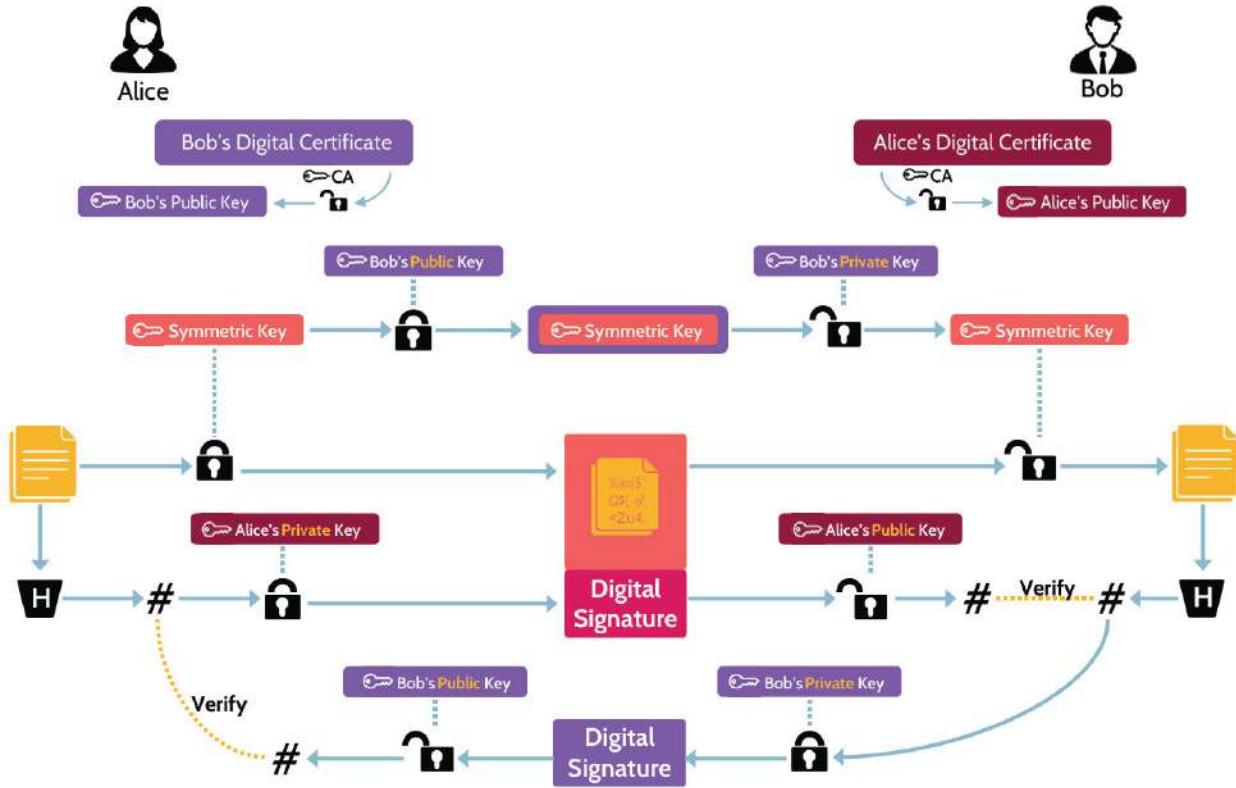


Figure 3-66: Five Services of Cryptography Depicted

## 3.7 Understand methods of cryptanalytic attacks

### 3.7.1 Cryptanalysis

#### CORE CONCEPTS

- Cryptanalysis is a multidisciplinary science.
- Two primary types of cryptanalysis attacks exist: cryptanalytic attacks and cryptographic attacks.
- The main purpose of cryptanalysis is to deduce or figure out the key (since in cryptography, everything is known except for the key).

Cryptanalysis is the science of:

- Cracking codes ■ Decoding secrets ■ Breaking cryptographic protocols ■ Finding and correcting weaknesses in encryption algorithms ■ Finding or deducing the key

types of cryptanalysis attacks can be defined, as also summarized in [Table 3-54](#), which are going to be analyzed later.

Cryptanalytic Attacks	Cryptographic Attacks
<ul style="list-style-type: none"><li>■ Ciphertext only ■ Known plaintext ■ Chosen plaintext ■ Chosen ciphertext ■ Linear and differential ■ Factoring</li></ul>	<ul style="list-style-type: none"><li>■ Man-in-the-middle ■ Replay ■ Temporary files ■ Implementation ■ Side-channel ■ Dictionary attack ■ Rainbow tables ■ Birthday ■ Social engineering</li></ul>

Table 3-54: Cryptanalysis Attack Types

### 3.7.2 Cryptanalytic Attacks Overview

#### CORE CONCEPTS

- The primary goal of a cryptanalytic attack is to determine the key.
- Brute-force attack involves trying every possible key until the correct key is identified; typically not effective unless the key length is very short (56 bits or less).
- Cryptanalytic attacks: ciphertext only, known plaintext, chosen plaintext, chosen ciphertext.
- Linear and differential attacks use complicated math to deduce the key; linear cryptanalysis uses a known-plaintext approach and differential cryptanalysis uses a form of chosen-plaintext attack.
- Factoring attacks attempt to factor a very large prime number, to determine the private key, and are utilized against the RSA algorithm (which uses factoring as the underlying hard mathematical problem).

#### What is the primary goal of cryptanalytic attacks?

The primary goal of any cryptanalytic attack is to *determine the encryption key*.

#### Brute-Force Attack

Brute-force attacks are quite simple. The attacker tries every possible key until the correct key is identified. While simple to execute, brute-force attacks are not very effective, because any key length of a reasonable size will increase the attack time exponentially. [Table 3-55](#) illustrates this concept. For clarity, the term *key length* also refers to the number of bits in a key; so, key length = bits. Similarly, the