CompTIA Security+

EXAM NUMBER: SY0-701

1.0 General Security Concepts

CompTIA Security+ SY0-701

Fundamental Security Concepts

- Information Security
- CIA Triad
- Essential Terminology

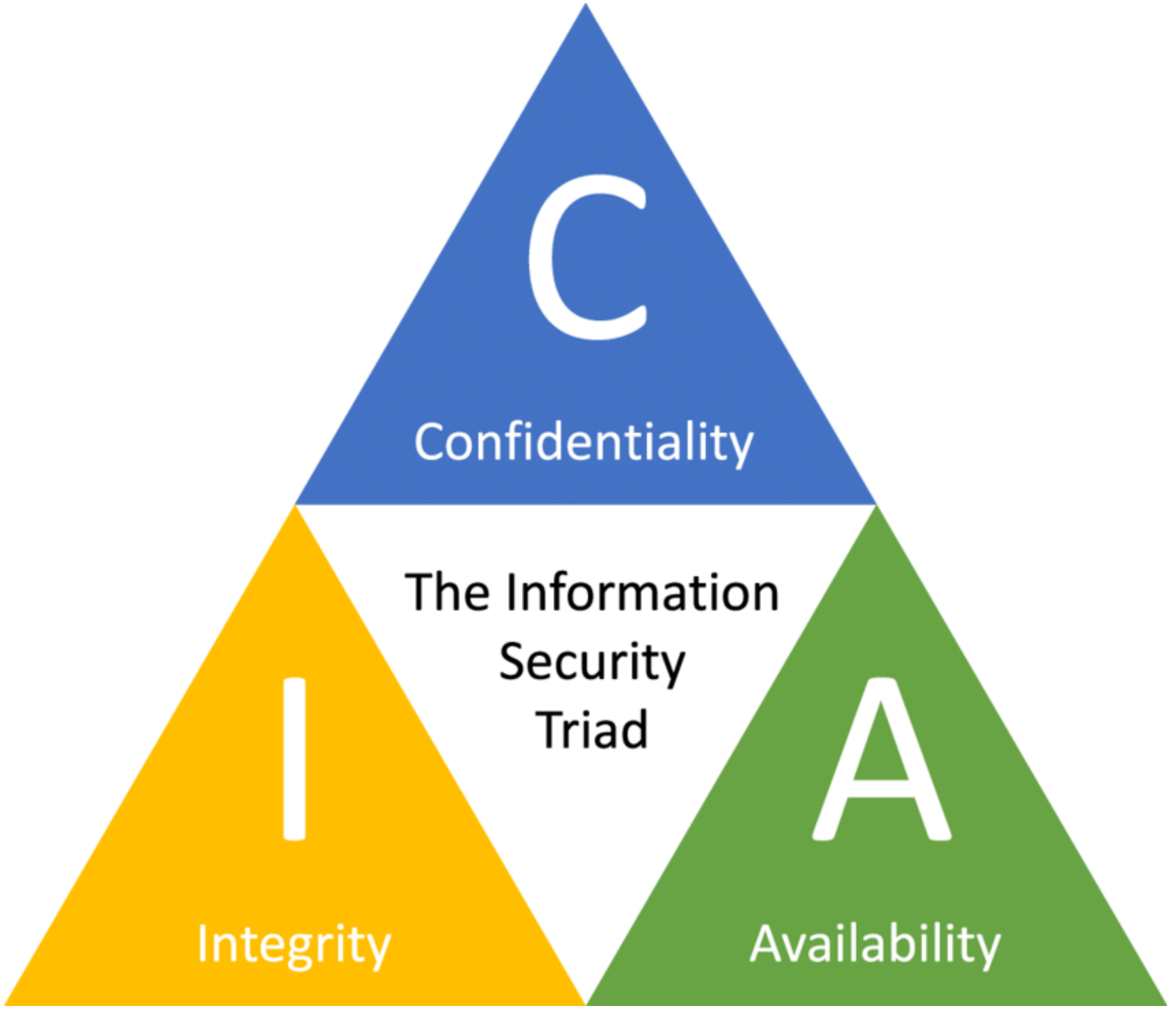
What is Information Security?

- The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction
- The goal is to provide confidentiality, integrity, and availability of systems and data

CIA Triad Example

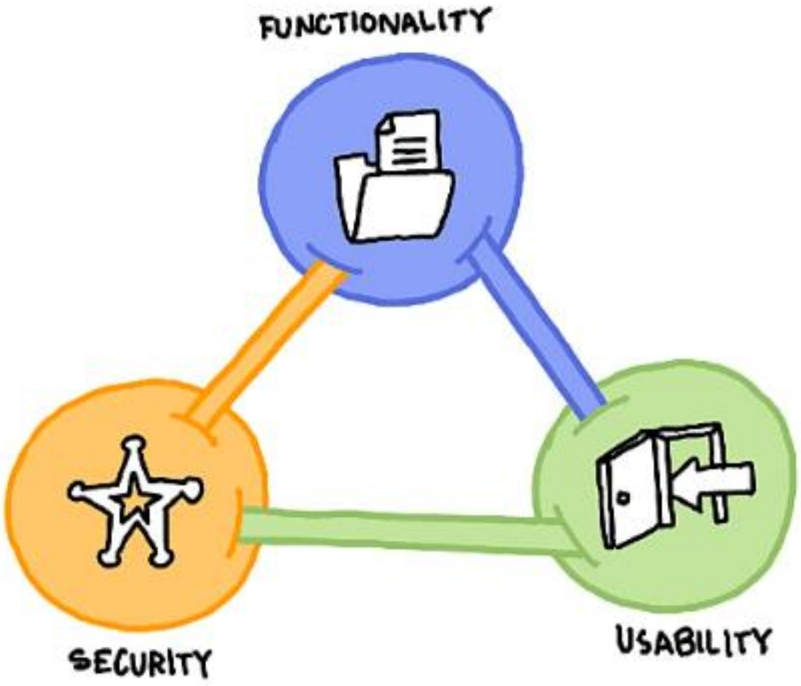
- **Confidentiality**
  - Protects data and systems from unauthorized access or disclosure
  - Typically implemented through encryption, access controls and permissions
- **Integrity**
  - Ensures data and systems are accurate, consistent, and protected from unauthorized modification, deletion, or corruption
  - Typically implemented through digital signatures
- **Availability**
  - Ensures that data and systems can still be accessed and used by authorized users and systems when needed
  - Typically implemented through high availability measures such as redundancy

CIA Triad Example



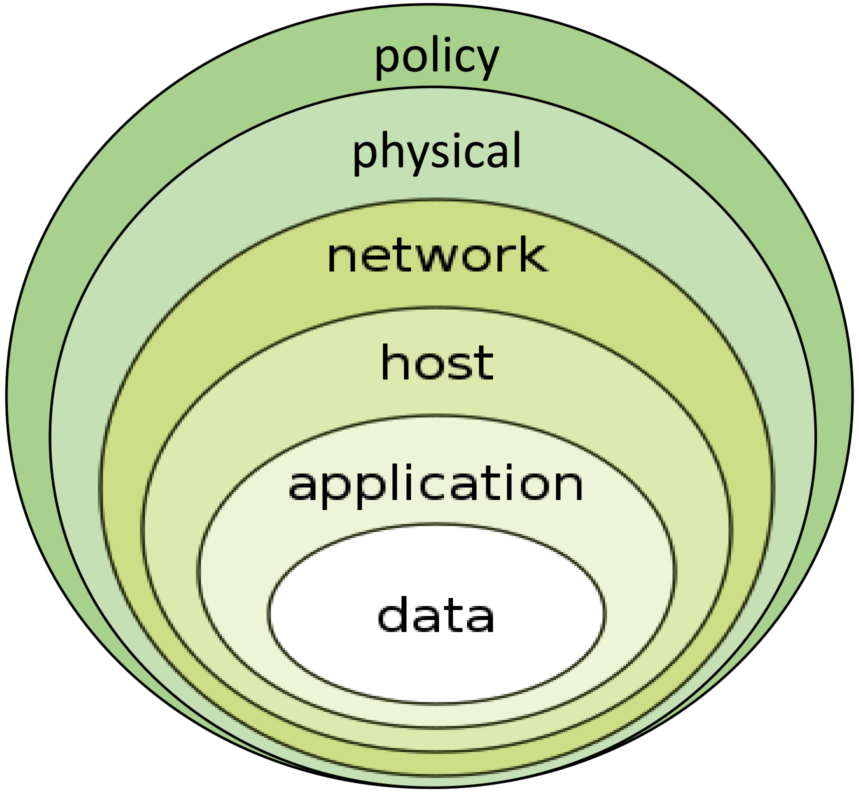
Security, Functionality, Usability

- These attributes are interlocked
- Security is at odds with nearly every other organizational process
- Increasing security usually requires decreasing functionality and usability
- You need to find an acceptable balance between these three



Defense in Depth

- Multiple layers of security controls
- Provides redundancy in the event of a control failure



Essential Terminology

Term	Definition
Vulnerability	A weakness or flaw in a system
Threat	Anything that can potentially violate the security of a system or organization
Exploit	An actual mechanism for taking advantage of a vulnerability
Payload	The part of an exploit that actually damages the system or steals the information
Zero-day attack	An attack that occurs before a vendor is aware of a flaw or is able to provide a patch for that flaw
Control	Any policy, process, or technology set in place to reduce risk
Mitigation	Any action or control used to minimize damage in the event of a negative event

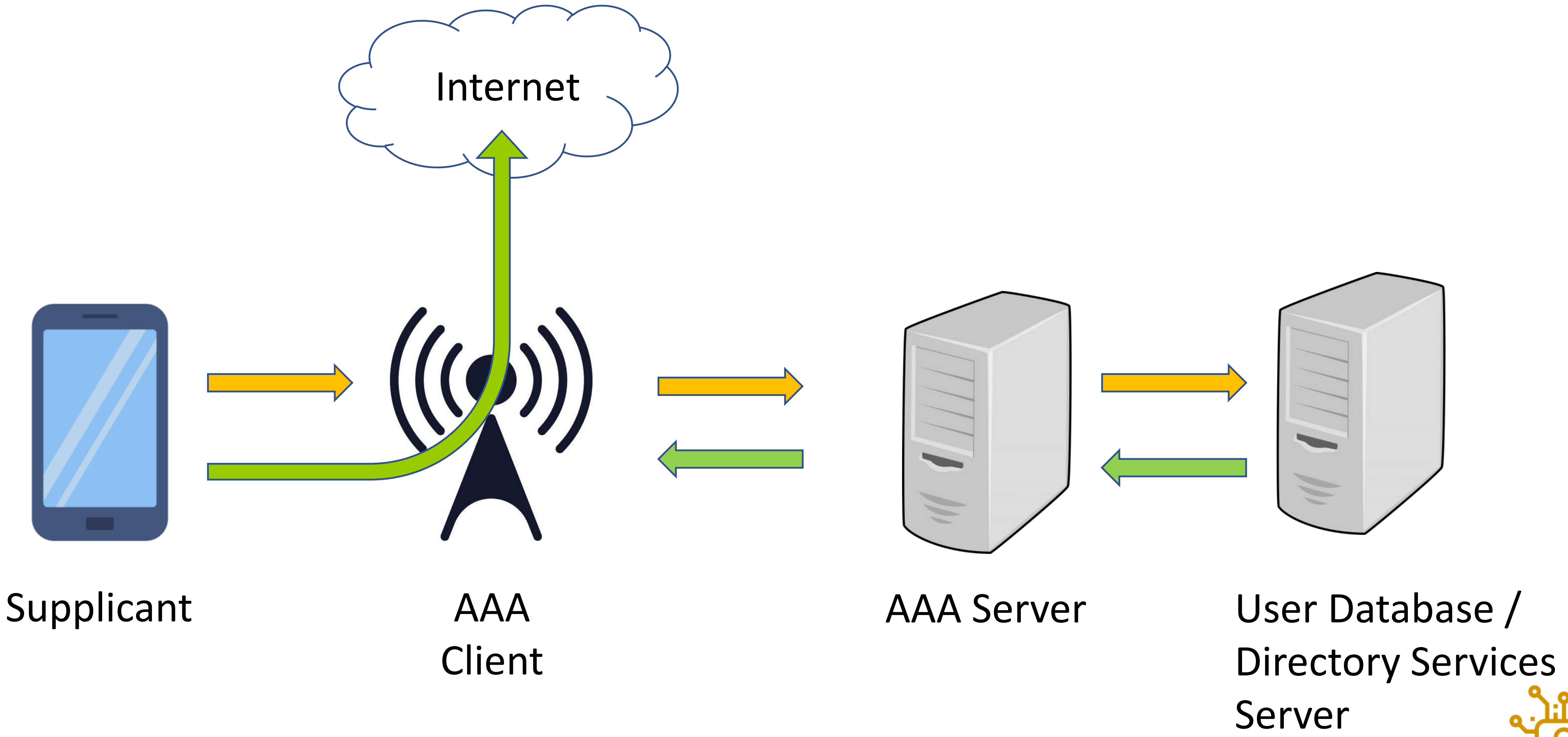
Essential Terminology (cont'd)

Term	Definition
Non-repudiation	• A security concept that prevents the denial of involvement or responsibility • Usually accomplished by applying a digital signature to a documents • Provides evidence of the origin and content of the message or document • Useful for legal, financial, or contractual purposes
Principle of Least Privilege	• A security concept in which users or systems are granted the minimum level of access or permissions that they need to perform their tasks, and nothing more
Accountability	Ensures that responsible parties are held liable for actions they have taken
Authenticity	The proven fact that something is legitimate or real
Gap analysis	• A thorough analysis of an organization's security defenses • Used to identify "gaps" between the current state of security and the desired state • The goal is to reduce the attack surface to prevent breaches

Authentication, Authorization, Accounting (AAA)

- Architectural framework to provide, enforce, and audit access to a network or compute resources
- Authenticates and authorizes the user based on user’s credentials
- Authentication requests are forwarded by wireless access points, network switches, and other connection points to a central AAA server
- The AAA server then typically forwards authentication requests to a directory service server
- Common AAA protocols include:
  - RADIUS
  - TACACS
  - TACACS+

AAA Example



Internet

Supplicant

AAA Client

AAA Server

User Database / Directory Services Server

Question

- Your healthcare organization wants to deploy a web application that allows individuals to digitally report health emergencies.
- Which part of the CIA triad should their application developers focus on?
- **Availability**

Question #2

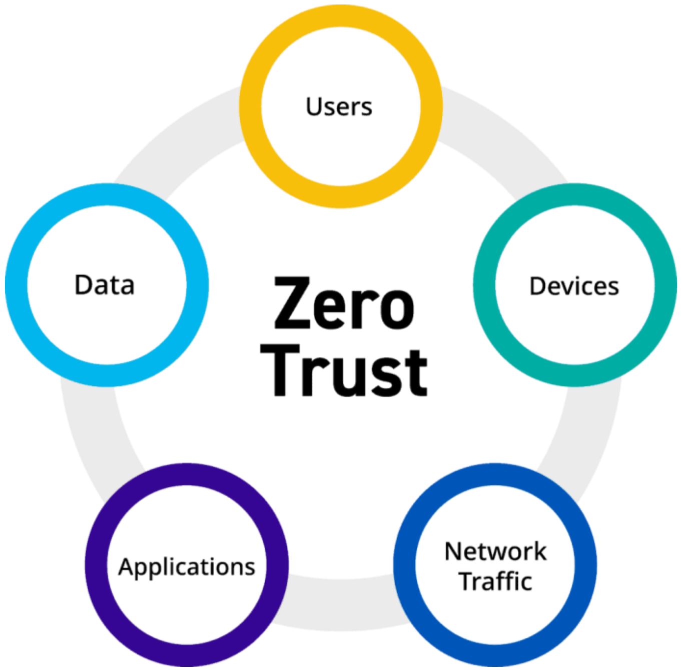
- You want to put permissions on a human resources file share to follow the principle of least privilege.
- What part of the CIA triad would this support?
- **Confidentiality**

Zero Trust

- Principles
- Architecture
- Techniques

Zero Trust

- A security strategy
- Never trust, always verify
- Based on the idea that nothing is completely secure:
  - Not people, devices, apps or processes



You can't separate the "good guys" from the "bad guys"

Three Principles of Zero Trust

1. Give the least privileged access necessary — nothing more
    - Preferably only at the moment the access is required (Just-in-Time access)
2. Access privileges must be constantly reauthorized
    - Nothing (subject, resource, system, app, network, etc.) is to be implicitly trusted
    - Assume that a connection can become compromised at any moment
3. Continuously monitor

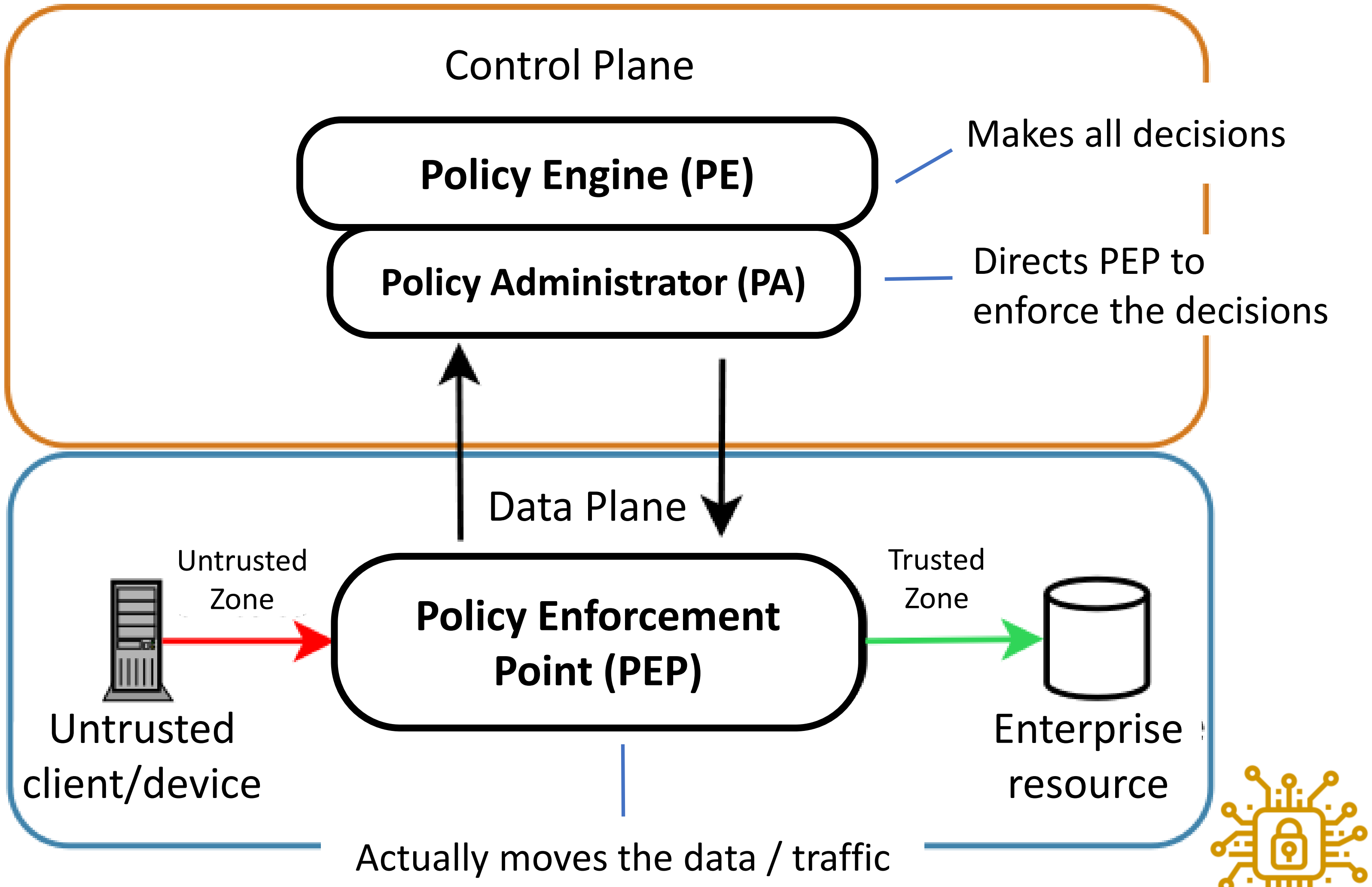
Zero Trust Fundamental Assertions

- The network is always assumed to be compromised and hostile.
- External and internal threats exist on the network at all times.
- Treat all hosts as if they are Internet-facing.
- Treat every connection attempt as a breach.
- Network locality is not sufficient for deciding trust in a network.
  - We no longer depend on a secure perimeter
- Every device, user, and network flow must be authenticated and authorized.
- Policies must be dynamic and calculated from as many sources of data as possible.

Zero Trust Architecture

Supporting Components

- Data Access Policy
- Identity Management
- PKI
- SIEM
- Industry Compliance
- Activity Logs
- CDM System
- Threat Intelligence



Control Plane

Policy Engine (PE)

Makes all decisions

Policy Administrator (PA)

Directs PEP to enforce the decisions

Data Plane

Untrusted Zone

Policy Enforcement Point (PEP)

Trusted Zone

Untrusted client/device

Enterprise resource

Actually moves the data / traffic

Tenets of Implementing Zero Trust

- All enterprise systems are considered resources.
- The enterprise ensures all owned systems are in their most secure state possible.
- All communication is done in a secure manner regardless of network location.
- Access to individual enterprise resources is granted on a per-connection basis.
- User authentication is dynamic and strictly enforced before access.
- Access to resources is determined by policy, including the observable state of user, system, and environment.

Some Techniques Used to Implement Zero Trust

- Policies
- Strong access controls
- Just-in-time access
- Secure network zones
- Micro-segmentation
- Data and system isolation
- Granular segmentation for users, devices, and applications
- Separation of duties

Some Techniques Used to Implement Zero Trust (cont’d)

- Endpoint Protection
- Multifactor authentication
- Conditional access
- Zero-knowledge cloud data encryption
- User and Entity Behavior Analytics (EUBA)
- Privileged Identity Management (PIM)
- Extended Detection and Response

Question

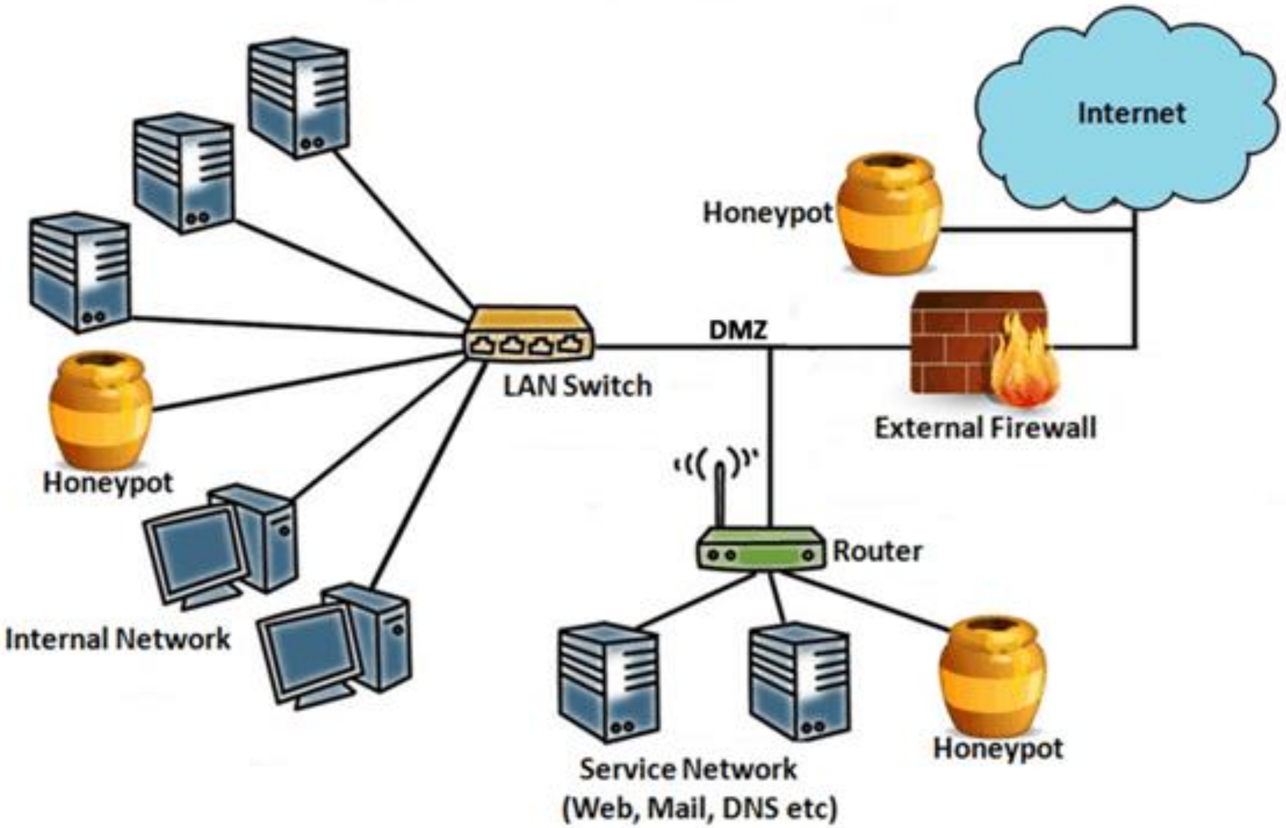
- In terms of network security, what is the key assumption of Zero Trust?
- **That your network is already compromised.**

Deception and Disruption

- Technologies
- Honey Assets
- Honey Tokens

Deception and Disruption Technology

- Honeypot
- Honeynet
- Honey file
- Honey token



Honey Assets

Asset	Description
Honeypot	• A decoy intended to look like a legitimate, vulnerable system • Deployed next to your genuine digital assets • Designed to make a would-be attacker waste their time, while you collect information about the attack
Honeynet	• A decoy network that contains one or more honeypots • Looks like a real network with multiple systems • Typically hosted on one or just a few servers
Honey file	• A fake file located on a network file share or server • Designed to detect attackers who are accessing and potentially stealing data
Honey token AKA honey credentials, canary traps or canary tokens	• Fake data deployed such that no legitimate user would have a need to access it • When accessed, an alert fires indicating an attack is in progress • May actively collect information such as IP addresses, unique browser fingerprints, or system information to help identify the attacker • Might include an executable file or a hidden linked image within a document that, upon running, extracts data and sends it to the security team

Benefits of Honey Assets

- You can gather information about attackers and their tactics without impacting production servers
  - The methods they use to move around the network
  - How they evade other detection techniques
  - If they are an insider threat, an outside attacker, or an advanced persistent threat that is already in the network
  - Use honey tokens as breadcrumbs/lures to lead an attacker away from actual assets to other decoys
- You can adapt cyber defenses accordingly

Honeytoken Examples

- **Credentials**
  - Dummy usernames, passwords, API keys, access tokens and other credentials that are planted across various applications and systems
  - If someone tries to use them, the security team knows an attack is underway
  - Attempted use is logged, revealing information about the attacker
- **Database entries**
  - Fake, seemingly high-value database records, including customer or employee credentials and financial information
  - Can be interspersed throughout a real database
  - The data should remain dormant. If accessed, it indicates malicious activity.
- **Documents**
  - Dummy Word documents, Excel files, PDFs and other documents that appear to contain sensitive information
  - If accessed, can fire an alert

Honeytoken Examples (cont’d)

- **Email addresses**
  - If an inactive decoy email address starts receiving spam or phishing emails, it indicates that attackers found it via an intrusion or insider threat
  - Email headers can be used to track the spammer/cyber criminal
- **Executable files**
  - Software programs that, when triggered, can automatically collect identifying information, such as threat actor’s names and IP addresses.
  - Can be hidden as a trojan horse inside some other file
  - Might not be effective if attackers have their own cyber defense measures in place.
- **Web beacons**
  - A hidden digital object, such as a transparent image or a single tracking pixel, that links to a unique URL
  - If an attacker opens a file that contains a web beacon, it automatically and surreptitiously initiates a server request that alerts the security team and potentially provides information about the threat actor

Question

- Which of the following can you use to identify potential attacker activities without affecting your production servers?
- Honey pot
- Video surveillance
- Zero Trust
- Geofencing

Question

- What can you use to identify potential attacker activities without affecting your production servers?
- **Honey pot**
- Video surveillance
- Zero Trust
- Geofencing

Question #2

- A server administrator places a document named password.txt on the desktop of an administrator account on the server.
- What type of honey asset is this?
- **A honey file**

Security Controls

- Control Categories
- Control Types

Control Categories

Security Control Categories

Category	Description	Examples
Physical	• Tangible mechanisms designed to deter or prevent unauthorized access to rooms, equipment, documents, and other items • Largely implemented by systems, as opposed to people • Designed to protect the physical environment of your information system	Lights, cameras, motion detectors/sensors, access badges, walls/fences, bollards, access control vestibules (mantraps), turnstiles, locks, alarms, disposal tools such as document and hard drive shredders, guards, dogs
Administrative / Managerial	• Procedures and policies that inform people on how the business is to be run and how day to day operations are to be conducted • Can be enforced through management policing, physical and technical means	Security policies and procedures, guidelines, formal change-management procedures, personnel background checks, training, software bug bounties, engaging a security audit team

Security Control Categories (cont’d)

Category	Description	Examples
Technical	Any measures taken to protect assets and reduce risk via technological means	IDS/IPS, firewall, anti-virus software, data classification solutions, encryption, authentication protocols, access control systems
Operational	Security controls that primarily are implemented and executed by people (as opposed to systems)	Organizational culture, day-to-day procedures designed to protect data and systems, awareness training, procedures for personnel management, incident response

Physical Security Examples

Question

- Name two ways to ensure only authorized personnel can access a secure facility
- **Badge access**
- **Access control vestibule**

Question #2

- You’re concerned about hackers potentially entering a facility and plugging in a remotely accessible Kali Linux box
- What should be the first line of defense against such an attack?
- **Physical controls such as badge access and access control vestibules**

Question #3

- In what control category is a turnstile?
- **Physical**

Question #4

- What can you use to keep a car from ramming into the front of the building?
- **Bollard**

Question #5

- Which security control category does an acceptable use policy best represent?
- **Administrative**

Control Types

Security Control Types

Control Type	Description	Examples
Preventive	• Designed to prevent errors, irregularities or undesirable events from occurring in the first place • Make it difficult or impossible for a bad actor to carry out the threat • Most security controls are preventive	Fences, gates, locks, authentication, logical access controls, encryption, firewalls, segregation of duties, employee screening and training
Detective	• Designed to detect and promptly correct undesirable events (errors, irregularities, violations, intrusions) that have already occurred.	Audits, intrusion detection, cameras, motion sensors, anti-virus, IDS/IPS, mandatory vacations, job rotation

Security Control Types (cont’d)

Control Type	Description	Examples
Deterrent	Designed to discourage would-be attackers or malicious insiders	Door locks, laptop locks, lighting, CCTV cameras, suspensions, fines, warning signs, lights, high fences, guards, dogs, logon banners
Mitigating/ Recovery	Designed to minimize the impact of a security incident	System isolation, repair, restore operations, fire suppression

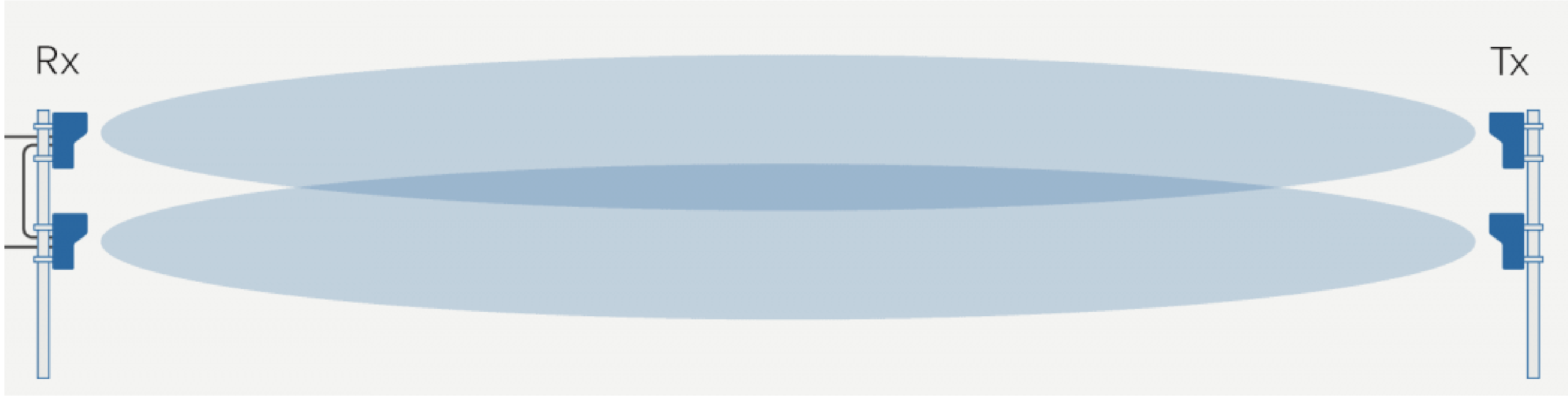
Security Control Types (cont'd)

Control Type	Description	Examples
Compensating	• Alternative fixes to cover any gaps in the other control types • Also referred to as "workarounds." • Designed to mitigate the risk associated with exceptions made to a security policy	You have a medical instrument that uses an older, vulnerable operating system. The instrument cannot be updated, so you keep it unplugged from the network to protect it from remote attack
Corrective	• Seeks to reverse/remediate a security incident after it occurs	• Restoring backups after a ransomware attack
Directive	• Designed to establish desired outcomes • Used to guide the execution of security within an organization	• Policies, procedures, standards, guidelines, laws, and regulations, mandatory settings or requirements

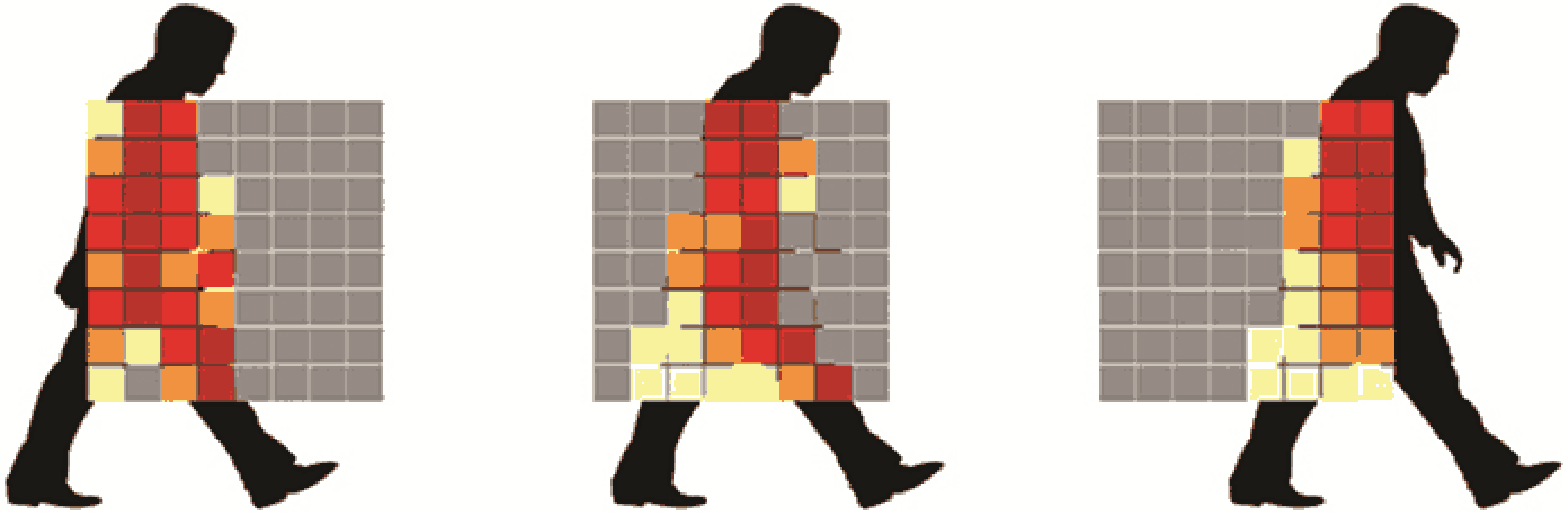
Intrusion Detection Sensor Types

Type	Description
Microwave	• Sends waves of electromagnetic energy between transmit and receive pairs • If an intruder enters, the energy is interrupted, and the sensor generates an alarm
Pressure	• Detects the weight of a person or object • If an intruder steps on a pressure mat, the change in surface weight activates an alarm
Infrared	• Beam – Invisible light beam from transmitter to receiver • If interrupted, sets off an alarm • Passive – Able to detect different levels of infrared radiation • Any significant change in IR sets off the alarm
Ultrasonic	• A protected area is flooded with an oval pattern of sound waves • As the sound waves bounce off objects, they reflect a signal back to a receiver • Any movement in the protected area will cause a change in the reflected pattern, activating the alarm • Sound waves are in a frequency range that is above the capacity of the human ear • Susceptible to false alarm due to air turbulence

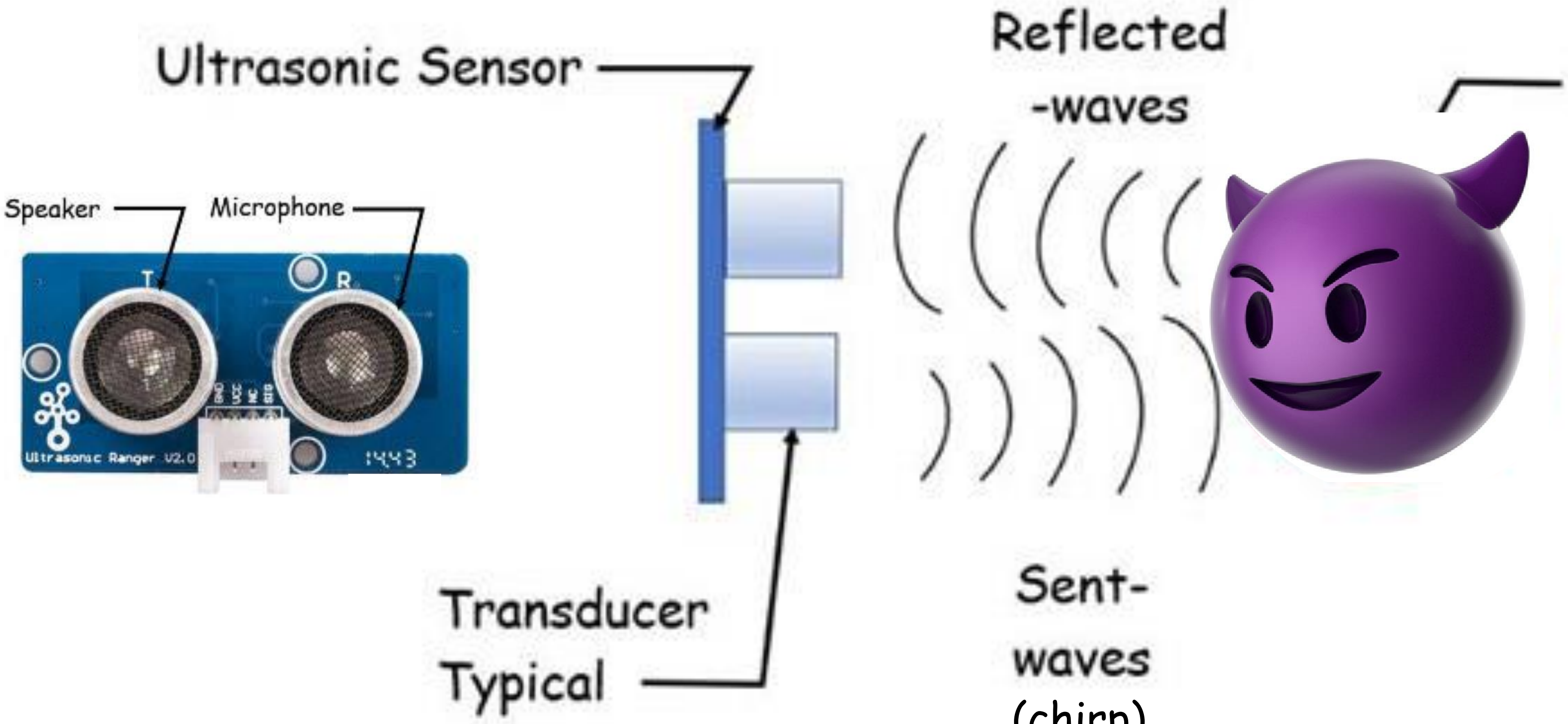
Microwave Sensor Examples



Passive Infrared Sensor Examples



Ultrasonic Sensor Example



(chirp)

Intruder changes the chirp response

Question

- Your data center requires entry and exit through multiple access points:
  - a lobby, an access control vestibule, doors leading to the server room itself and a caged area that contains the hardware
- Which control type is described in this scenario?
- **Preventive**

Question #2

- You are reviewing log files after a recent ransomware attack.
- Which control type are you using?
- **Detective**

Question #3

- You have a legacy Linux system that does not support many security controls.
- You implement a host-based firewall to allow connections from only specific internal IP addresses.
- Which control type are you using?
- **Compensating**

Change Management and Security

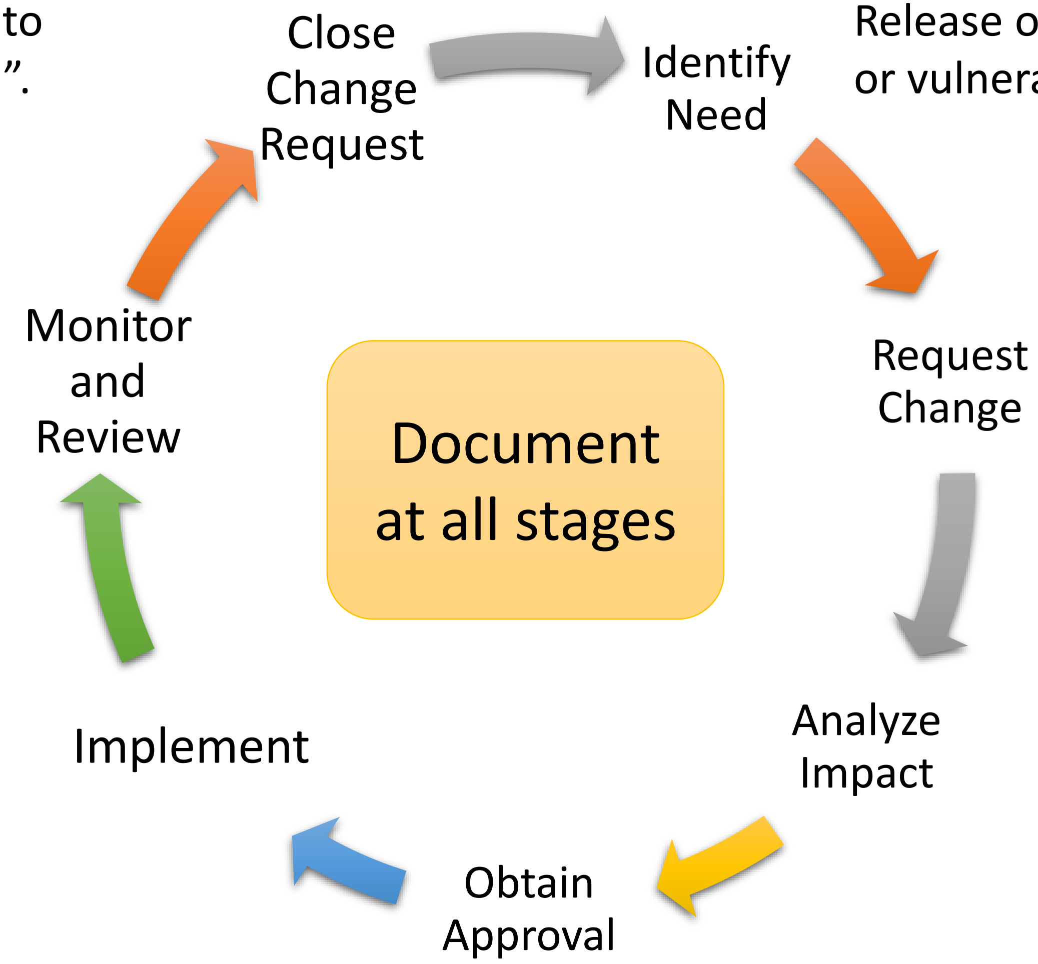
- Change Management
- Documentation
- Version Control

Change Management

- A structured process for updating the network, devices, and software
  - All change management procedures must adhere to this process
- Helps you make informed decisions to reduce unauthorized, failed, and emergency changes
  - The goal is to implement changes with minimal impact and risks
  - You can assess, prioritize, and schedule changes with input from your change advisory board
- Use change management software to help streamline the process

In IT, change management is sometimes referred to as “change control”

Change Management Process



Identify Need

Release of new security advisory, patch, or vulnerability assessment results.

Request Change

Purpose, scope, impact, benefits, roles, responsibilities

Analyze Impact

Feasibility, risks, costs, dependencies, alternatives, contingency plans

Obtain Approval

Authorization to proceed

Implement

Execute according to plan and schedule.
Verify results and outcomes.

Monitor and Review

Measure performance and effectiveness.
Identify issues for review, gaps for improvement

Close Change Request

Update status to “Implemented”.

Document at all stages

Impact of Change Management on Security

- Change management minimizes the rate at which security risks occur
  - Promotes standards, process improvement, reduces complexity and risk, and provides sanity in complex environments
  - Ensures that the right person can access the right information at the right time
- By creating a thoughtful and intentional plan, security practitioners can:
  - Make longer-lasting changes
  - Increase adoption and awareness throughout the company
  - Reduce overall security risks

Technical Considerations During Change Management

- You can use allow and deny lists to set your baseline of approved applications, operating systems, hardware, and configurations
  - Use change control to formally update your baseline
- Changes during the maintenance window often require system or application restarts and scheduled downtime
  - Users might not be able to access services during the maintenance window, or might be restricted to limited activities until the change is finalized
  - You can minimize the impact by notifying users of scheduled downtime
- Changes to a system can adversely impact a dependent system
  - You might not even be aware of the impact until after the fact
  - It’s important to conduct tests, include all stakeholders when performing an impact analysis, and have a backout plan ready
- Legacy systems are usually intolerant of security updates
  - You will need to put compensating controls in place to protect these systems

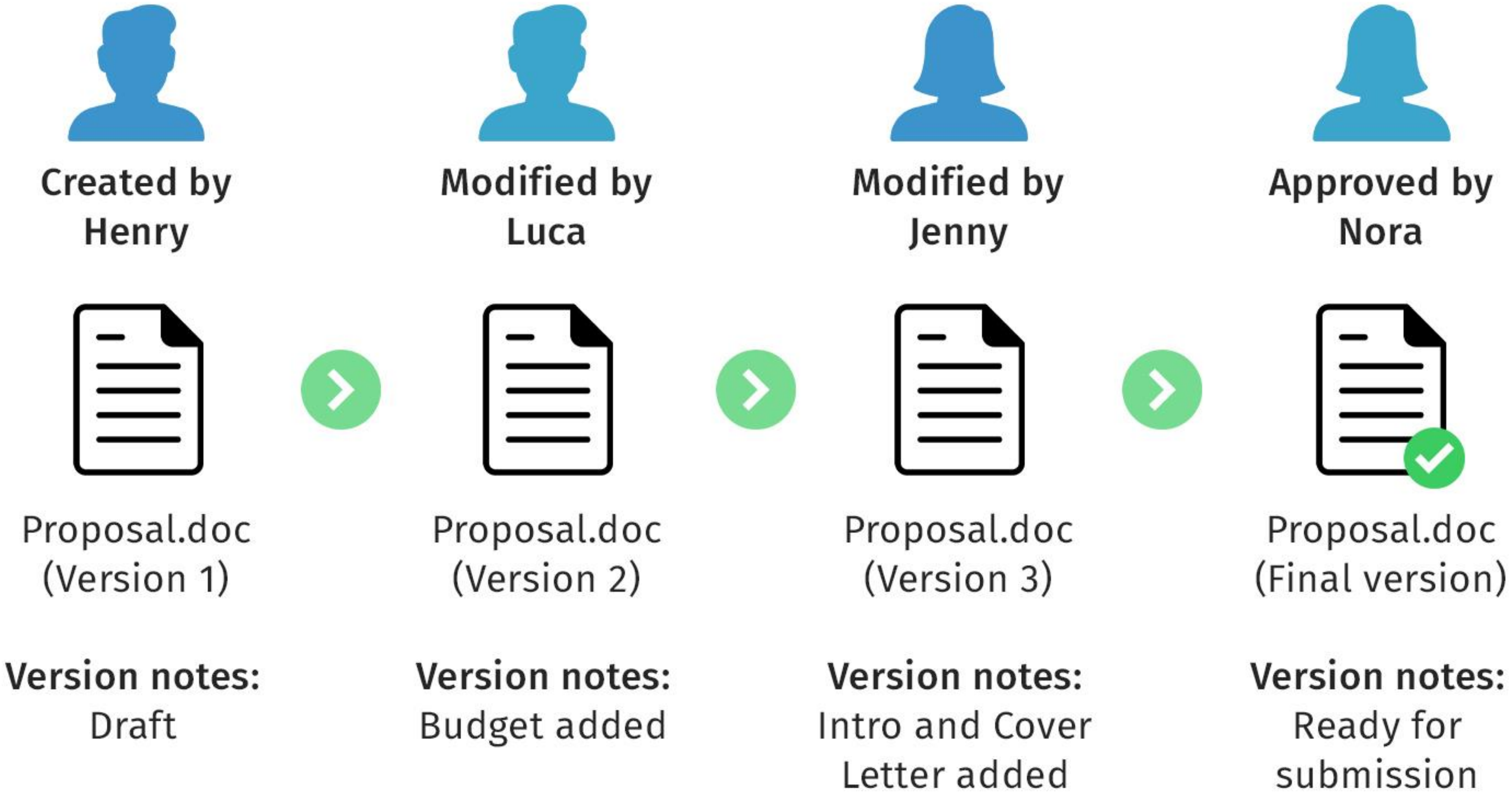
Documentation

- All policies, procedures, and changes should be well-documented.
- Documentation should be easily available to those who need it.
- Consider keeping both (searchable) electronic and hard copies.
- Documentation should be reviewed and updated regularly.

Version Control

- AKA source control or revision control.
- Typically applied to documents, procedure manuals/SOPs, and software development.
- Tracks and documents changes to the item.
- Allows developers and users to stay synchronized with the current version.
- Serves as a safety net to protect source code from irreparable harm.
- Should include:
  - a consistent format for major and minor revisions
  - the date and author(s) of the revision
  - a list or highlighting of changes from the previous version
- Copies of all previous versions, forks, and branches should be saved and made easily accessible in case the team needs to refer to/fall back on them.

Version Control Example



Created by Henry

Proposal.doc (Version 1)

Version notes: Draft

Modified by Luca

Proposal.doc (Version 2)

Version notes: Budget added

Modified by Jenny

Proposal.doc (Version 3)

Version notes: Intro and Cover Letter added

Approved by Nora

Proposal.doc (Final version)

Version notes: Ready for submission

Question #1

- You need to apply a high-priority patch to a production system.
- What should you do first?
- **Create a change control request.**

Question #2

- You want to ensure systems are available, and any maintenance will minimizes business impact.
- What would you use to specify a set period of time to perform changes to an operational system?
- **Scheduled downtime / maintenance window**

Question #3

- What should you adhere to when setting up a new set of firewall rules?
- **Change management procedure.**

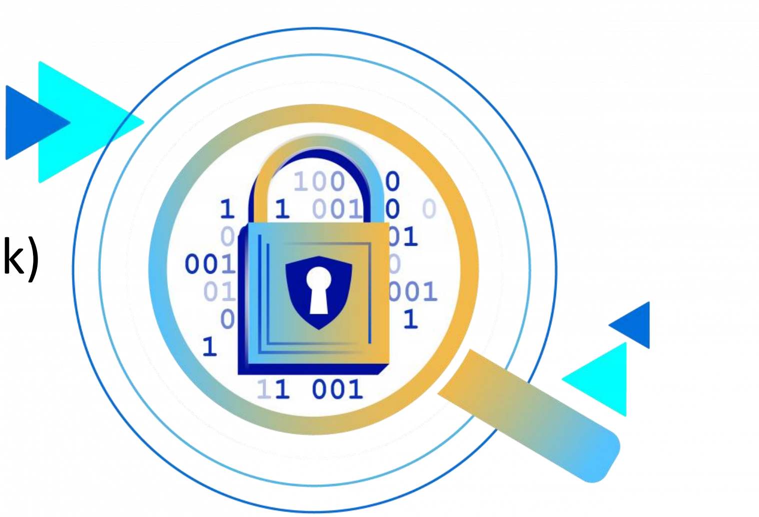
Cryptography Basics

- Cryptography
- Symmetric Encryption

Cryptography Basics

Cryptography

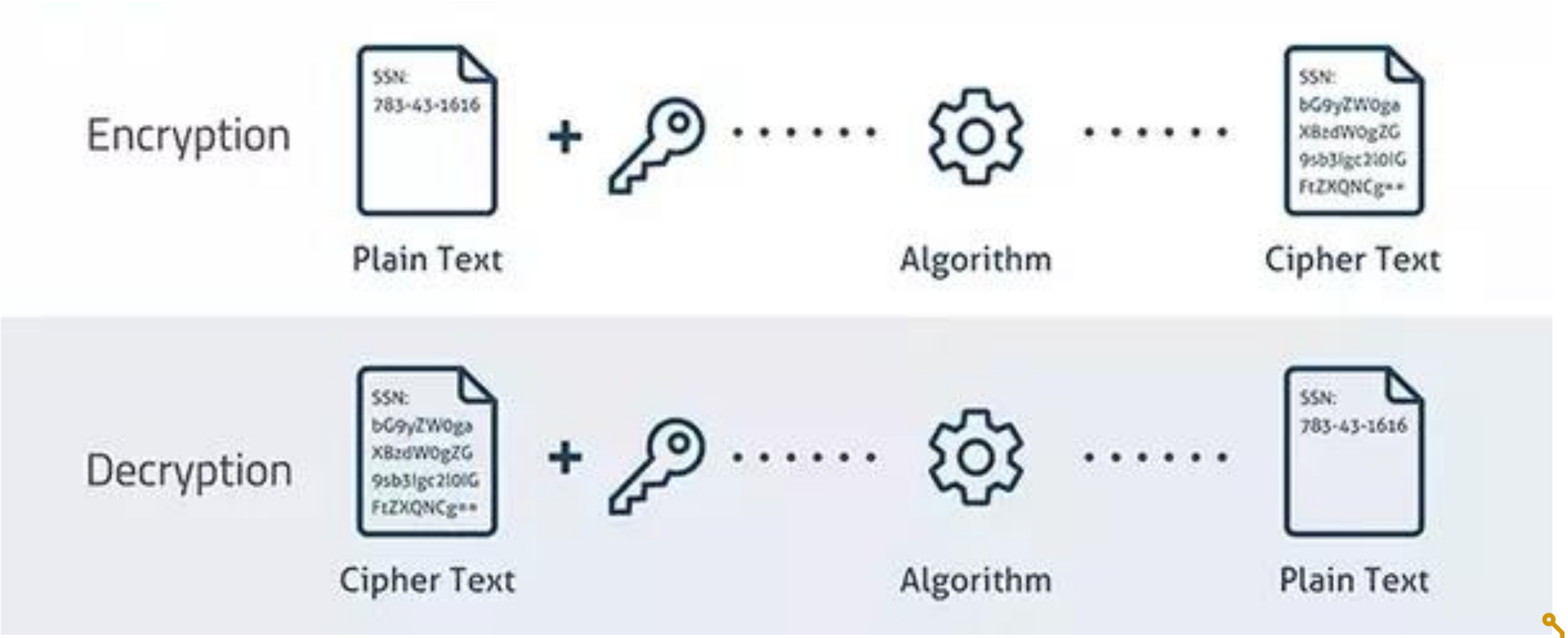
- The process of converting data in its original form (such as ordinary plain text) into something unintelligible, and vice-versa.
- When encrypted, the data can be safely stored, used, or transmitted across a network.
- Even if the data is stolen or intercepted, the attacker cannot read it.
- Used to protect data confidentiality.
- Can protect data in any of its three states:
  - At rest (stored on storage media)
  - In transit (actively being transmitted across a network)
  - In use (loaded in RAM or being processed on a CPU)



Components of Cryptography

- Unencrypted data (“plain text”)
  - Data in its original form
- Algorithm (cipher)
  - Mathematical formula for scrambling the data
- Key
  - Introduces an “unknown” variable into the scrambling formula
  - Anything that can be reduced to a number:
    - String of characters / password / PIN
    - Biometric information
    - Swipe or drawing pattern
  - Longer key = stronger encryption
- Ciphertext (encrypted data)

Cryptography Example



Cipher

- AKA algorithm
- A mathematical formula for scrambling data
- Block cipher
  - Data is encrypted in fixed-size blocks (typically 64 bits)
  - Plain text is converted into cipher text one block at a time
  - Often some output from one encrypted block is added to the encryption of the next block
  - Good for large amounts of data
    - E.g. files, data at rest
- Stream cipher
  - Data encrypted in a continuous stream
  - Uses XOR to encrypt data one bit, byte, or character at a time
  - Typically faster than block ciphers
  - Requires fewer resources and less complex circuitry
  - Good for real-time communications

Types of Encryption

- Symmetric Encryption
  - Uses the same key for both encryption and decryption
- Asymmetric Encryption
  - Uses one key for encryption and a different key for decryption
- Hashing
  - One way encryption
  - Fixed length output for any length input
  - No key
  - Meant for data integrity
  - Data is not encrypted
  - Hashed output accompanies the data for anyone to verify

Question

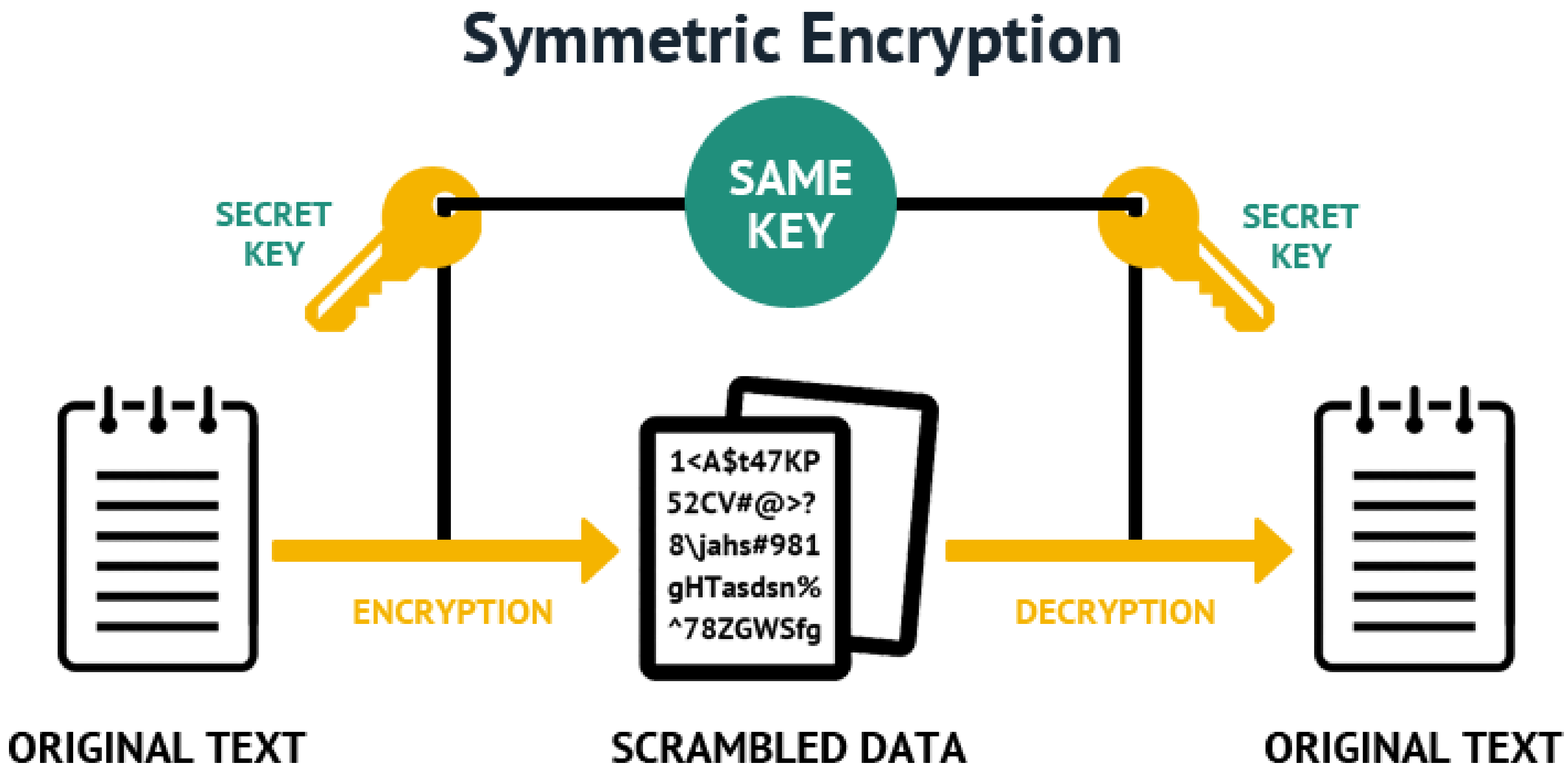
- What makes an encryption key strong?
- **Its length**

Symmetric Encryption

Symmetric Encryption

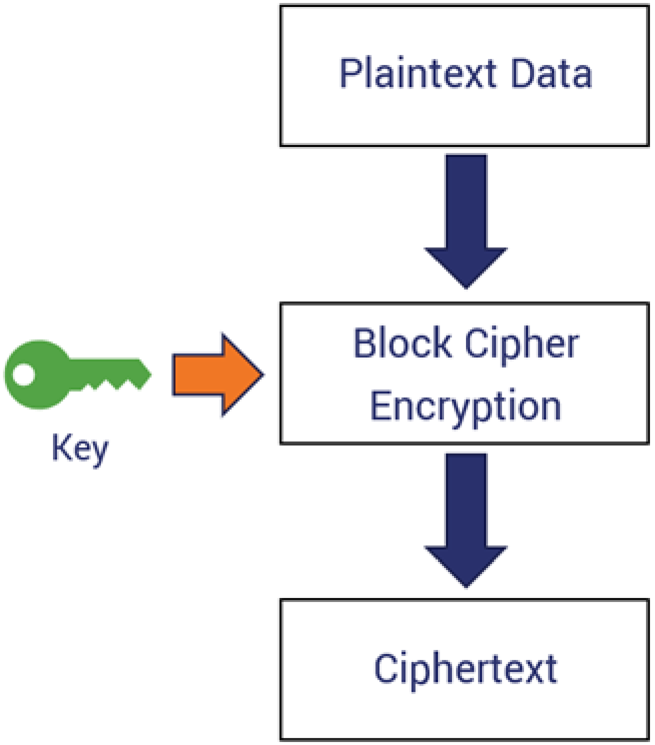
- Same key is used to encrypt and decrypt
- Used extensively to protect data at rest
- Provides confidentiality
- Excellent for bulk data encryption
  - Fast, with good performance
- Single key is a risk:
  - You must share the key in advance
  - If the key is compromised, all files are at risk of loss of confidentiality

Symmetric Encryption Example



Block Cipher Symmetric Algorithm

- Excellent for encrypting whole files
- Also used by WPA2 WI-FI encryption
- Encrypts data in fixed-size blocks of (typically) either 64 or 128 bits
- The bits in a block can be encrypted simultaneously
  - The choice of block size does not directly affect the strength of encryption scheme
- In some implementations, some of the output of one block is then fed into the encryption of the next block
  - Makes it very hard to reverse-engineer
  - Will propagate errors
- The strength of the cipher depends upon the key length



Common Block Ciphers

- DES
  - Archetypal 64-bit block cipher
  - Transforms fixed-length blocks of plaintext into ciphertext bit strings of equal length
  - Inherently weak with current technology
  - 64-bit key (effectively only 56 bits of security)
  - Has already been broken
- 3DES
  - DES process repeated 3 times with 3 keys to increase encryption strength
- AES (the current US government standard)
  - Symmetric-key algorithm designed to secure unclassified, sensitive U.S. government documents
  - Iterated block cipher designed to keep doing the same operation repeatedly
  - Block size of 128 bits
  - AES key sizes:
    - 128 for AES-128
    - 192 for AES-192
    - 256 for AES-256

Common Block Ciphers (cont’d)

- Blowfish
  - 64 bit block cipher
  - 32 – 448 bit key length
  - Faster than DES
- Twofish
  - 128 bit block cipher
  - 128 – 256 bit key length
- RC2, RC5, RC6
  - 64 – 128 bit block cipher
  - Each iteration has increased the key size
  - RC6 supports 2040 bit keys

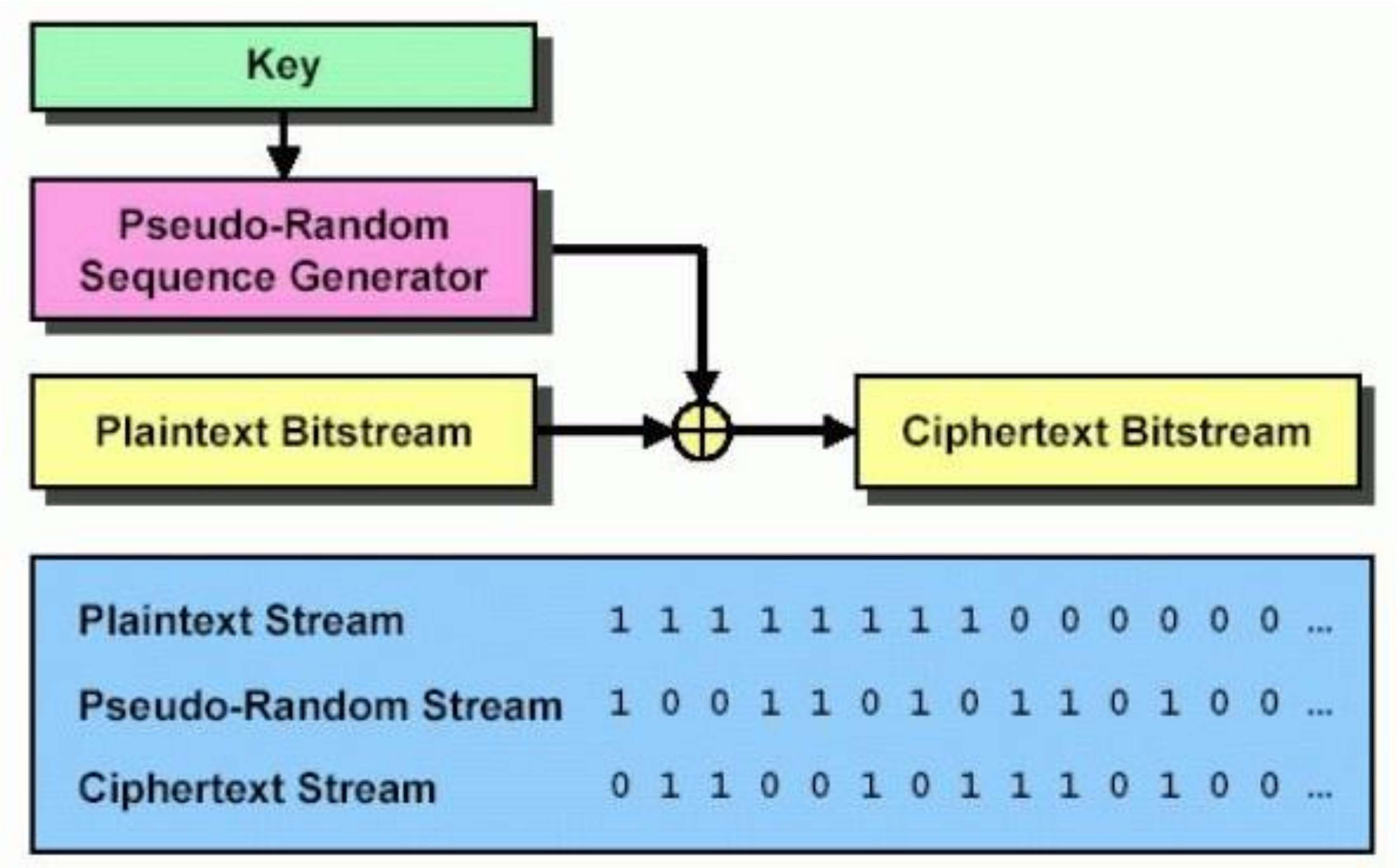
Stream Cipher Symmetric Algorithm

- Excellent for encrypting a continuous stream of data such as real-time voice/video/multimedia
- Also used by WEP and WPA Wi-Fi encryption
- Processes data one bit at a time
  - Does not divide the data into discrete blocks
- At the transmitting end, XOR each bit of:
  - Your plaintext continuous data stream + a key stream
  - The key stream starts with the key as its “seed”, and then continues with a pseudo-random generated sequence
- At the receiving end, use the same symmetric key and XOR to decrypt
- Often faster than block ciphers
- Also useful when transmission errors are likely to occur
  - Has little or no error propagation

XOR

- Exclusive OR
- A simple mathematical function used to encrypt data in a stream cipher
- Two input bits are compared to produce a 3rd output bit
- IF the two bits are the same, the output = 0
- IF the two bits are different, the output = 1

Stream Cipher Example



Plaintext Stream	1	1	1	1	1	1	1	1	0	0	0	0	0	0	...
Pseudo-Random Stream	1	0	0	1	1	0	1	0	1	1	0	1	0	0	...
Ciphertext Stream	0	1	1	0	0	1	0	1	1	1	0	1	0	0	...

Common Stream Ciphers

- RC4
  - Popular stream cipher
  - Used in Wi-Fi WEP
  - Key length 40 – 2048 bits
- ChaCha20
  - Quickly replacing RC4
  - Google QUIC protocol, used in HTTPS/3
  - FreeBSD arc4random random number generator
- PKZIP
  - File archive/compression program that uses a proprietary stream cipher to encrypt files

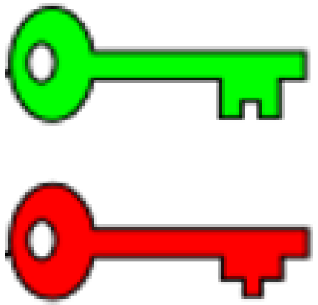
Question

- Which cipher type would you use to encrypt an email?
- **Symmetric Block Cipher**

Asymmetric Encryption

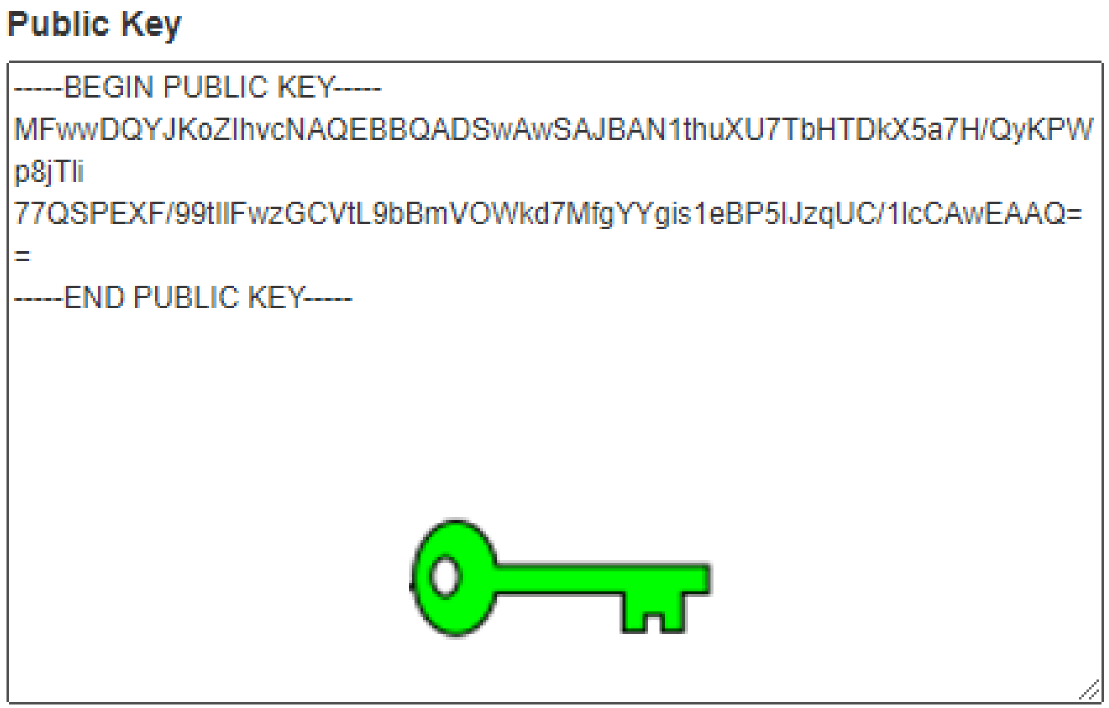
- Asymmetric Key Pair
- Diffie-Hellman
- ECC

Asymmetric Encryption



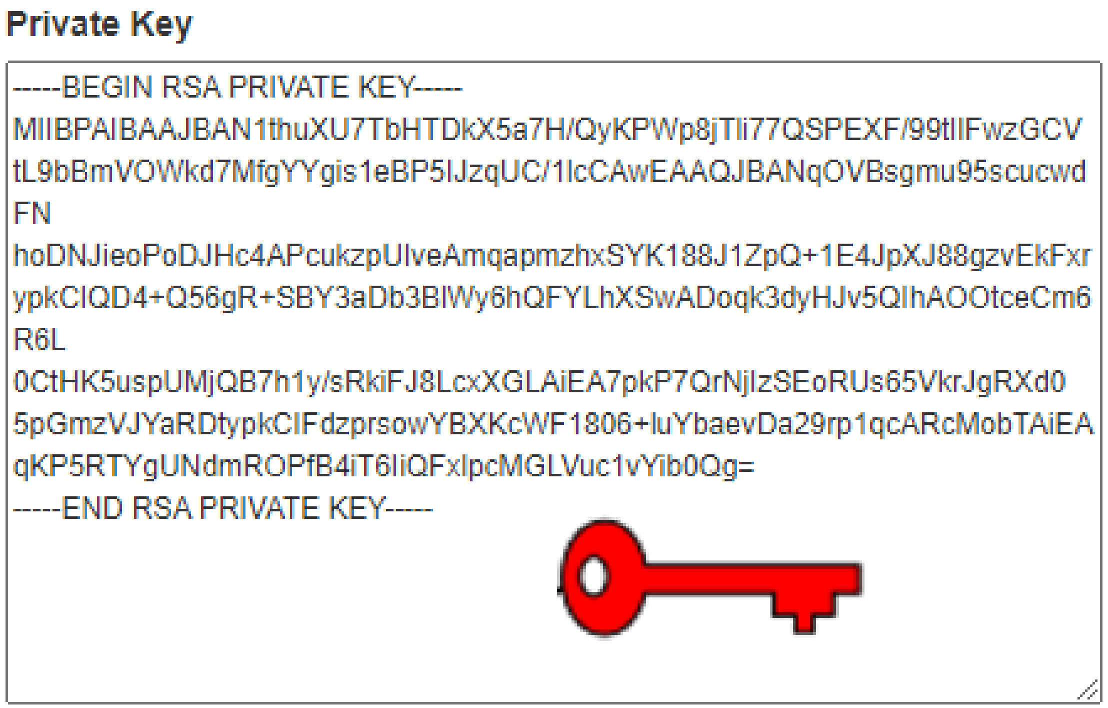
- Uses two mathematically related keys
  - Encrypt with public key
  - Decrypt with private key
- Provides confidentiality and integrity
- You can request (or create your own) public/private key pair
- Freely give away your public key to anyone
- Carefully guard the private key
  - Never let anyone else have access to it
  - Keep it stored in encrypted format

Asymmetric Key Pair Example



Public Key

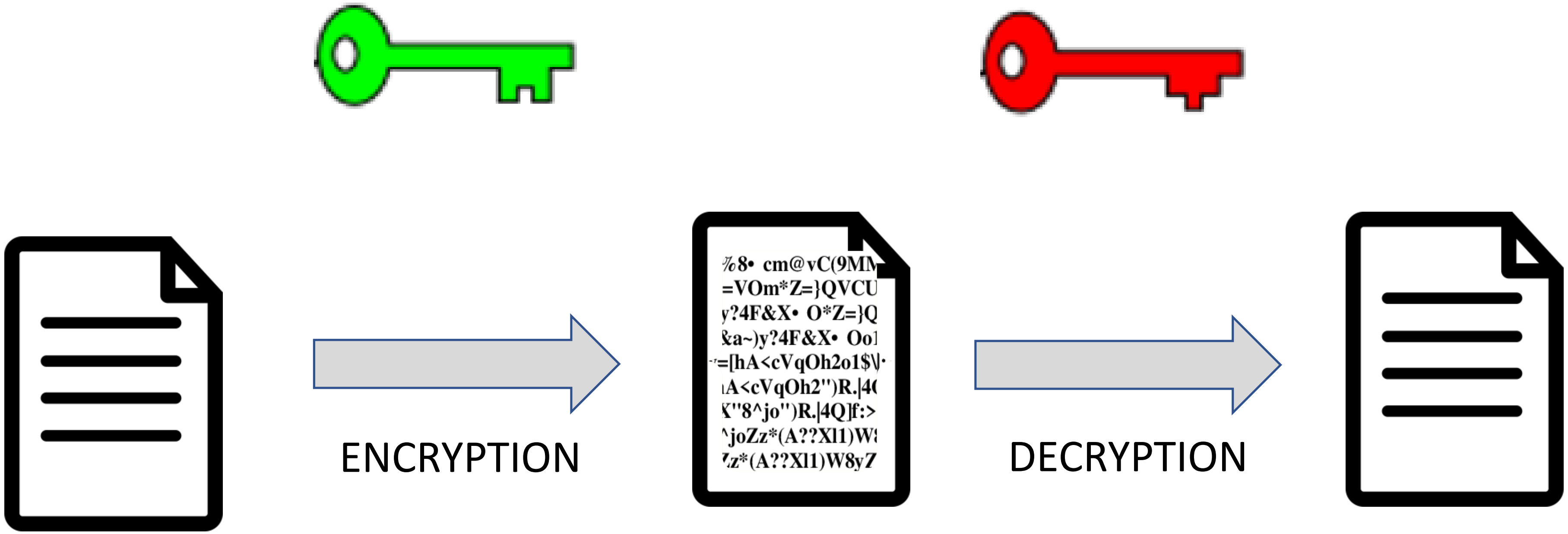
```
-----BEGIN PUBLIC KEY-----
MFwwDQYJKoZIhvcNAQEBBQADSwAwSAJBAN1thuXU7TbHTDkX5a7H/QyKPW
p8jTli
77QSPEXF/99tlIFwzGCVtL9bBmVOWkd7MfgYYgis1eBP5lJzqUC/1lcCAwEAAQ=
=
-----END PUBLIC KEY-----
```



Private Key

```
-----BEGIN RSA PRIVATE KEY-----
MIIBPAIBAAJBAN1thuXU7TbHTDkX5a7H/QyKPWp8jTli77QSPEXF/99tlIFwzGCV
tL9bBmVOWkd7MfgYYgis1eBP5lJzqUC/1lcCAwEAAQJBANqOVBsgmu95scucwd
FN
hoDNJieoPoDJHc4APcukzpUlveAmqapmzhxSYK188J1ZpQ+1E4JpXJ88gzvEkFxr
ypkClQD4+Q56gR+SBY3aDb3BlWy6hQFYLhXSwADoqk3dyHJv5QlhAOOtceCm6
R6L
0CtHK5uspUMjQB7h1y/sRkiFJ8LcxXGLAiEA7pkP7QrNjlzSEoRUs65VkrJgRXd0
5pGmzVJYaRDtypkClFdzprsowYBXKcWF1806+luYbaevDa29rp1qcARcMobTAiEA
qKP5RTYgUNdmROPfB4iT6IiQFxlpcMGLVuc1vYib0Qg=
-----END RSA PRIVATE KEY-----
```

Asymmetric Encryption Example



Public Key

Private Key

ENCRYPTION

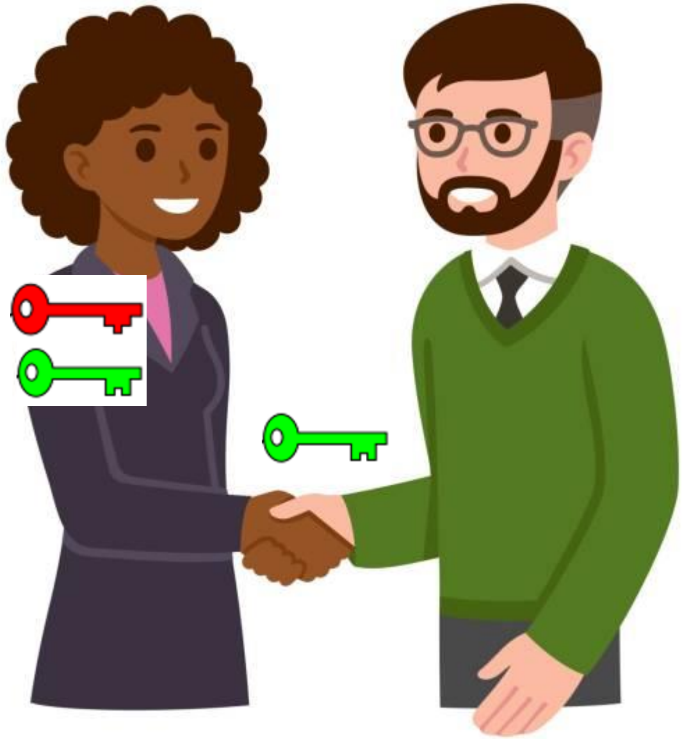
DECRYPTION

ORIGINAL DATA

SCRAMBLED DATA

ORIGINAL DATA

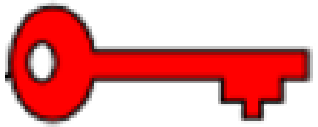
Asymmetric Encryption Example #2



- Alice has an asymmetric key pair
- She gives Bob a copy of her public key
  - Bob uses Alice’s public key to send her an encrypted message
  - Alice uses her private key to decrypt
- Alice can also use her private key to digitally sign messages
  - Bob can use her public key to verify the signature

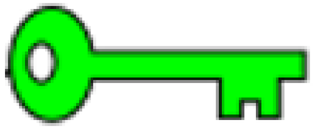
When to Use Which Key?

- You use your own **PRIVATE** key to:



  - Decrypt data
  - Digitally sign data

- Someone else will use your **PUBLIC** key to:



  - Encrypt data that only you can decrypt
  - Verify your digital signature

Asymmetric Algorithms

- RSA
  - De facto Internet encryption standard
  - Used in certificates
  - Based on the practical difficulty of factoring the product of two large prime numbers, the factoring problem
- Diffie-Hellmann
  - Protocol for exchanging asymmetric keys
  - Diffie-Hellman (DH) groups determine the strength of the key used in the key exchange process
- Elliptic Curve Cryptography (ECC)
  - Based on the algebraic structure of elliptic curves over finite fields
  - Can achieve the same level of security provided while using a shorter key length
    - An ECC algorithm using a 256-bit key length is just as strong as an RSA or Diffie-Hellman algorithm using a 3072-bit key length
  - Good for devices that have lower computing power
    - Smart cards
    - Mobile devices
  - Also used by WPA3 WI-FI encryption

Diffie-Hellmann Key Exchange

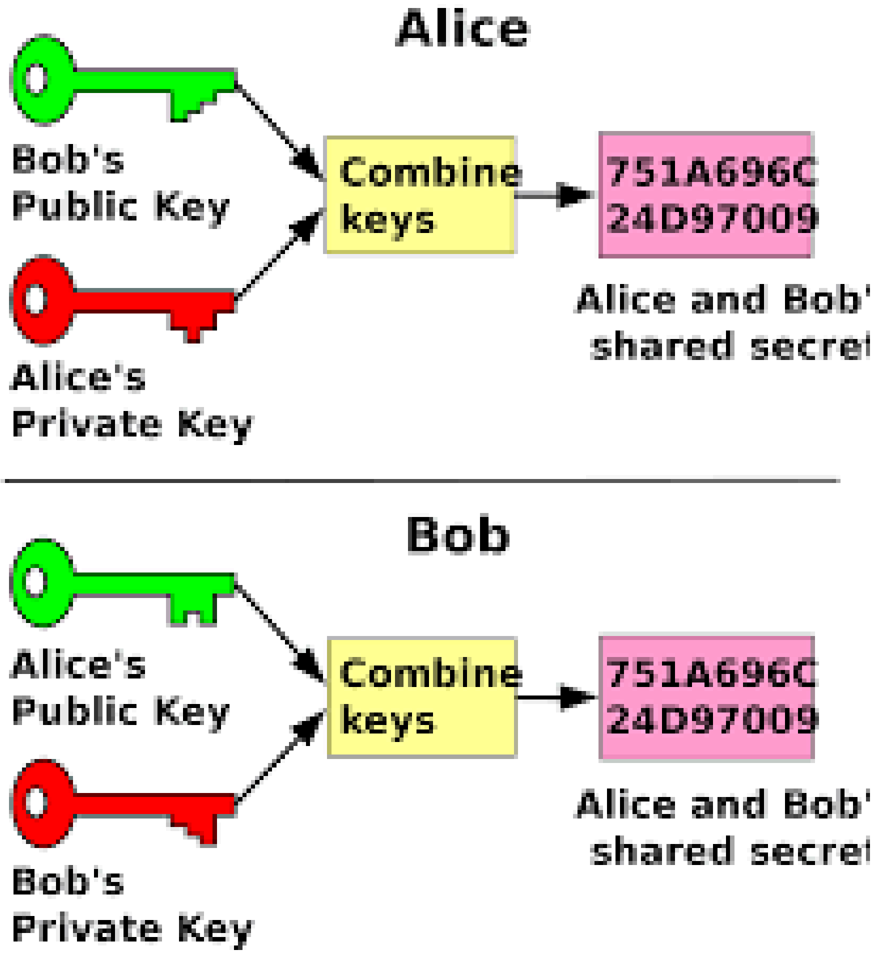
- Protocol for automatically exchanging public keys
- The first widely used method of safely developing and exchanging keys over an insecure channel
- Largely replaced by RSA, which has its own key exchange algorithm and can digitally sign certificates
- Diffie-Hellman Groups are used to determine the strength of the key used in the Diffie-Hellman key exchange process
  - Higher Diffie-Hellman Group numbers are more secure
  - But higher groups also require additional CPU power
- Commonly used DH Groups:
  - DH Group 1: 768-bit group
  - DH Group 2: 1024-bit group
  - DH Group 5: 1536-bit group
  - DH Group 14: 2048-bit group
  - DH Group 15: 3072-bit group

Higher DH group number = stronger key

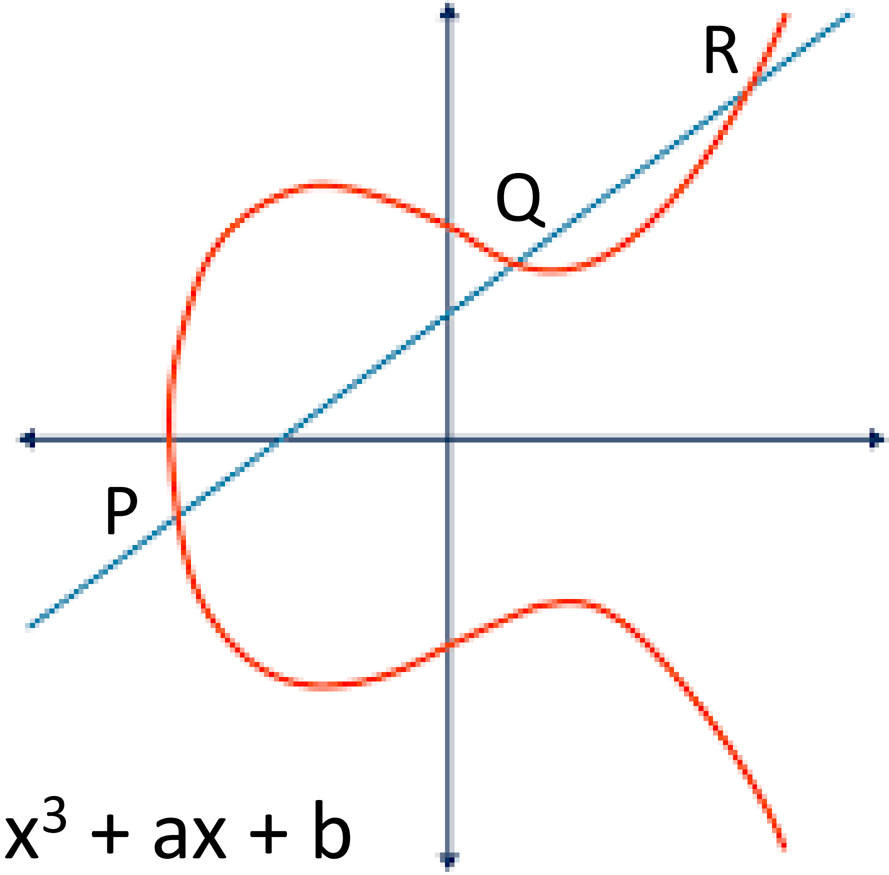
Diffie-Hellman Example

- Alice and Bob trade public keys
- Each side combines their private key with the other's public key to create the same shared secret
- The shared secret is used as a temporary session key to encrypt communications

Alice and Bob's shared secret



ECC Example



$y = x^3 + ax + b$

Symmetric vs Asymmetric Encryption

Encryption Type	Description
Symmetric	• Single key to encrypt/decrypt • Fast • Well-suited for encrypting large blocks of data
Asymmetric	• Key pair • Public key encrypts • Private key decrypts • Well-suited for encrypting small amounts of data such as a symmetric key • Often used to create a temporary session key

Question

- What key do you need to encrypt something only Moo can read?
- **Moo’s public key**

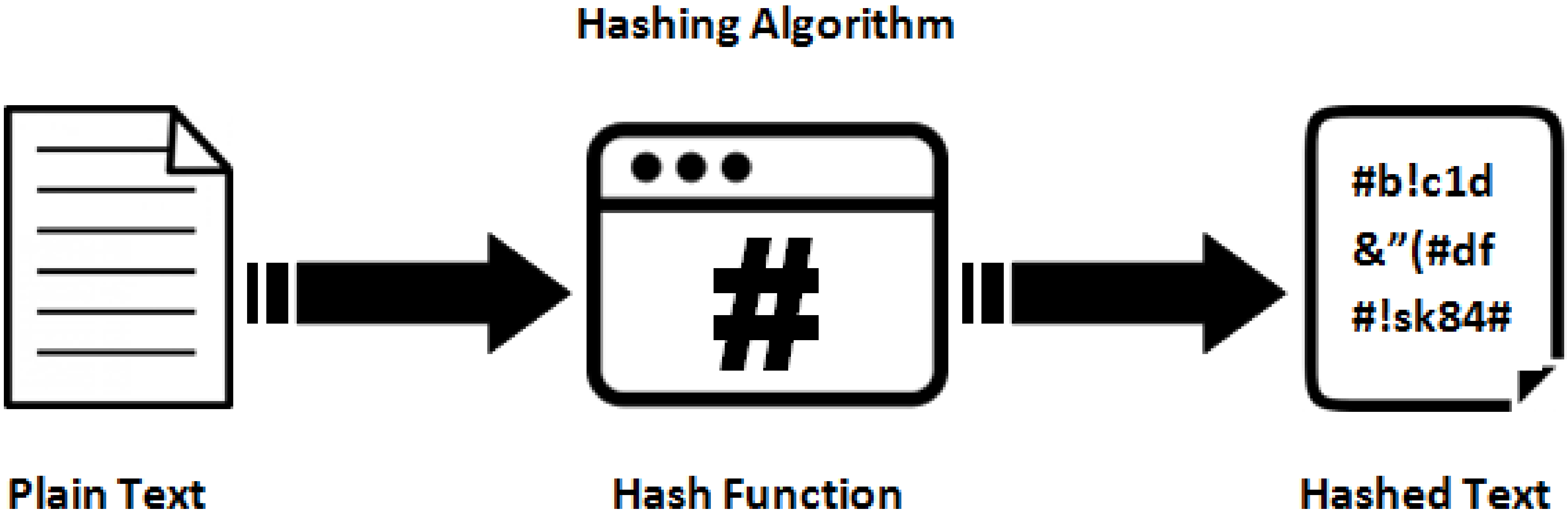
Hashing

- Hashes
- Salting

Hashing

- A function that takes data of any size or type, and creates from it a fixed-length string of characters
- Creates a one-way “encryption”
- Does not require a key
- Does not modify the original file/data
- The values returned by a hash function are called hash values, hash codes, digests, or simply hashes
- The slightest change to the input dramatically changes the output
- Typically used to:
  - securely store passwords
  - accompany files or packets to guarantee their integrity

Hashing Example



Hashing Algorithm

Plain Text

Hash Function

Hashed Text

Requirements of an Effective Hashing Algorithm

- Computationally infeasible to decrypt
- Resistant to collisions
  - Two different inputs must not create the same output

A collision attack is an attempt to find two input strings of a hash function that produce the same hash result.

Popular Hashing Algorithms

- Message Digest MD2/MD4/MD5 – 128 bit
- Secure Hash Algorithm
    - SHA-1 – 160 bit
    - SHA-2:
        - SHA-256
        - SHA-384
        - SHA-512
    - SHA-3
        - The latest version of SHA
        - Same hash lengths as SHA-2
        - Internal structure is significantly different
        - Currently the strongest hashing algorithm
- RIPEMD – 160 bit

Salting

- The process of adding a random string to the beginning or end of the input text prior to hashing or encryption
- Adds randomness and length to a password
  - Makes it harder to crack or guess
- Mostly used to keep passwords safe during storage
  - Can also be used with other types of data
- Each stored password is given a different salt
- The salt must be stored along with the password hash
- When a user logs on:
  - The same salt is added to the password they enter
  - The password + salt is run through the same hashing algorithm
  - The result is compared to the stored hash
  - If the result is the same, then the user has entered the correct password

Salting Example

Password	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz	p4s5w3rdz
Salt	-	-	et52ed	ye5sf8
Hash	f4c31aa	f4c31aa	lvn49sa	z32i6t0

Question

- Which of the following is used to add extra complexity before using a one-way data transformation algorithm?
- **Salting**

Question #2

- What can be used by an authentication application to validate a user's credentials without the need to store the actual password?
- **Hashing**

Digital Certificates

- Certificate Concepts
- Digital Signatures

Digital Certificate

- A public key on a document
  - Includes some metadata about the key
  - The file that contains the certificate typically has the extension .cer or .der
- Issued by a certification authority to a user, device, or service account
- When first issued, will be accompanied by the related private key
  - The private key itself is typically encrypted with a password to protect it
  - The file containing a private keys typically has the extension .pfx or .pvk

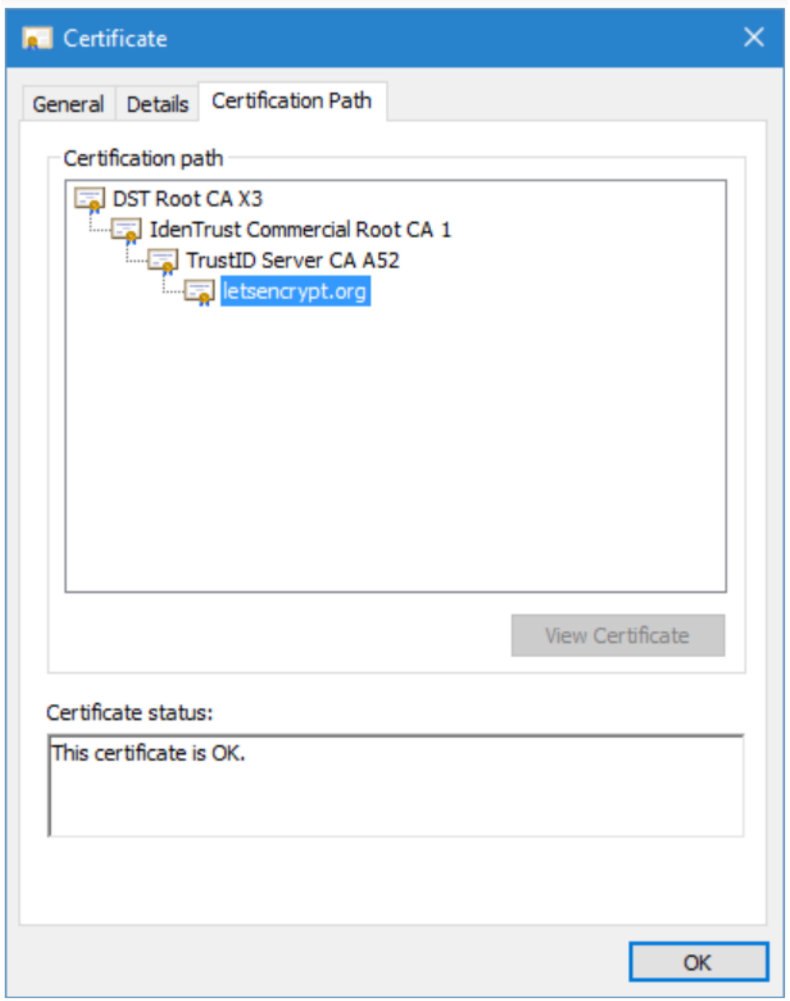
Certificate Concepts

Concept	Description
Certificate authority (CA)	A service that issues certificates
Certificate revocation list (CRL)	A list of certificates that were administratively invalidated (revoked) before they expired
Online Certificate Status Protocol (OCSP)	• Used by a nearby server that contains copies of the CRL • Distributes the burden of validating certificates • Can quickly respond to client CRL requests
Third-party	A certificate issued by a trusted certificate authority
Root of trust	• Highly reliable hardware, firmware, and software • Used to generate and protect root and CA keys

Certificate Concepts (cont’d)

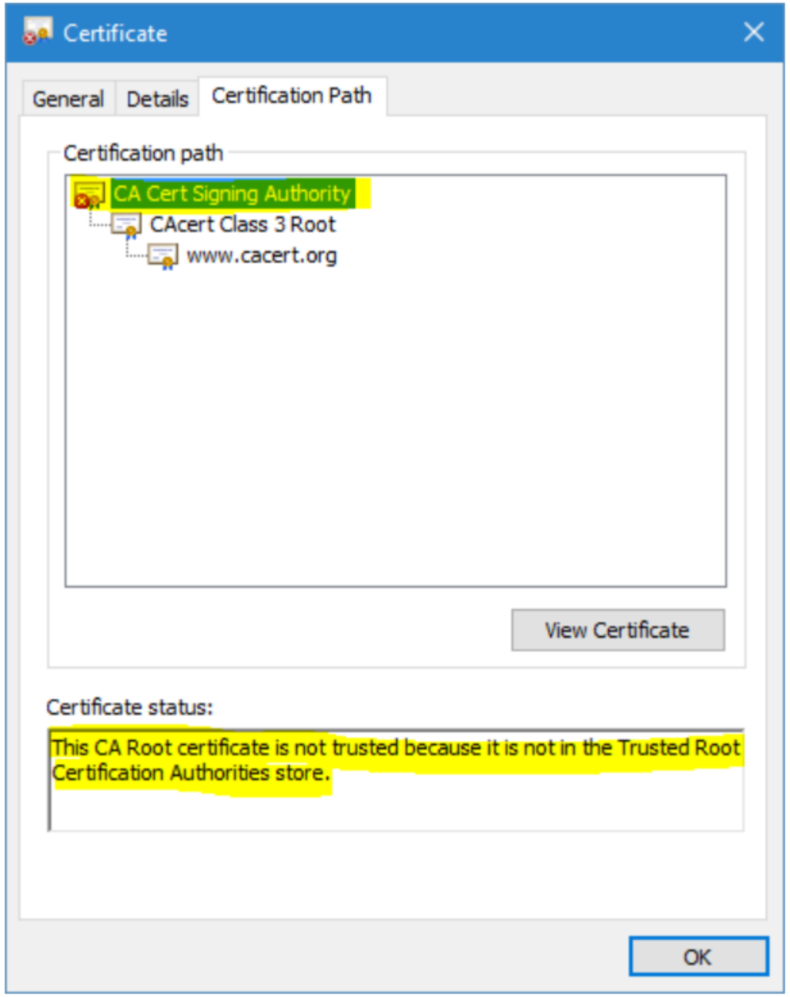
Concept	Description
Self-signed	• A certificate that was created locally, rather than by a CA • Meant for internal use • Created for convenience when a trusted CA is not required
Certificate signing request (CSR)	A message sent to the CA requesting a digital certificate Request files typically have an extension of .csr
Wildcard certificate	A certificate that includes any possible subdomains or host names under a parent domain The wildcard is represented by an asterisk (*)

Certification Path Examples



DST Root CA X3
IdenTrust Commercial Root CA 1
TrustID Server CA A52
letsencrypt.org

This certificate is OK.

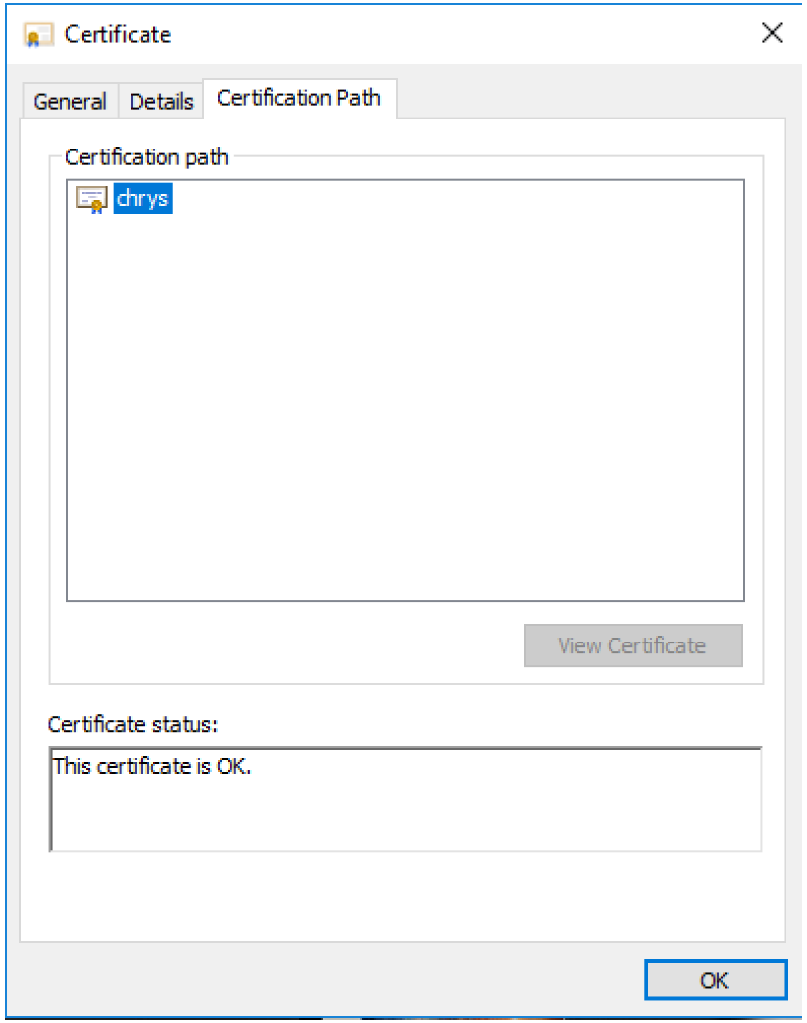


CA Cert Signing Authority
CAcert Class 3 Root
www.cacert.org

This CA Root certificate is not trusted because it is not in the Trusted Root Certification Authorities store.

Self-signed Certificate

- User creates public/private key pair using any available tool
- User self-signs certificate (public key) with their private key
- Public keys are traded
- A temporary symmetric session key is created
- The session key is protected by our public keys, which can only be decrypted by our private keys



Certification path

chrys

Certificate status:

This certificate is OK.

Question

- A certificate vendor notified you that recently invalidated certificates may need to be updated.
- What should you use to determine whether the certificates installed on your company's machines need to be updated?
- **CRL**

Question #2

- What can you use to speed up answering client CRL requests?
- **Online Certificate Status Protocol (OCSP)**

Question #3

- You want to simplify the certificate management process.
- You have a single domain with several dozen subdomains, all of which are publicly accessible on the Internet.
- What type of certificate should you use?
- **Wildcard**

Question #4

- What is used to verify that a certificate has not been revoked?
- **Certificate Revocation List (CRL)**

Question #5

- Which file extension is used to request a digital certificate?
- **.csr**

Question #6

- Name two file extensions that are used by certificates (public keys)
- **.cer, .der**

Question #7

- A security engineer at an offline government facility is concerned about the validity of an SSL certificate.
- They want to perform the fastest check with the least delay to determine if the certificate has been revoked
- What would BEST meet their requirement?
- **OSCP**

Question #8

- Name two file extensions are used by a certificate’s private key
- **.pfx, .pvk**

Question #9

- You need to generate a server certificate to be used for secure communications.
- What should you do first?
- **Generate a CSR**

Question #10

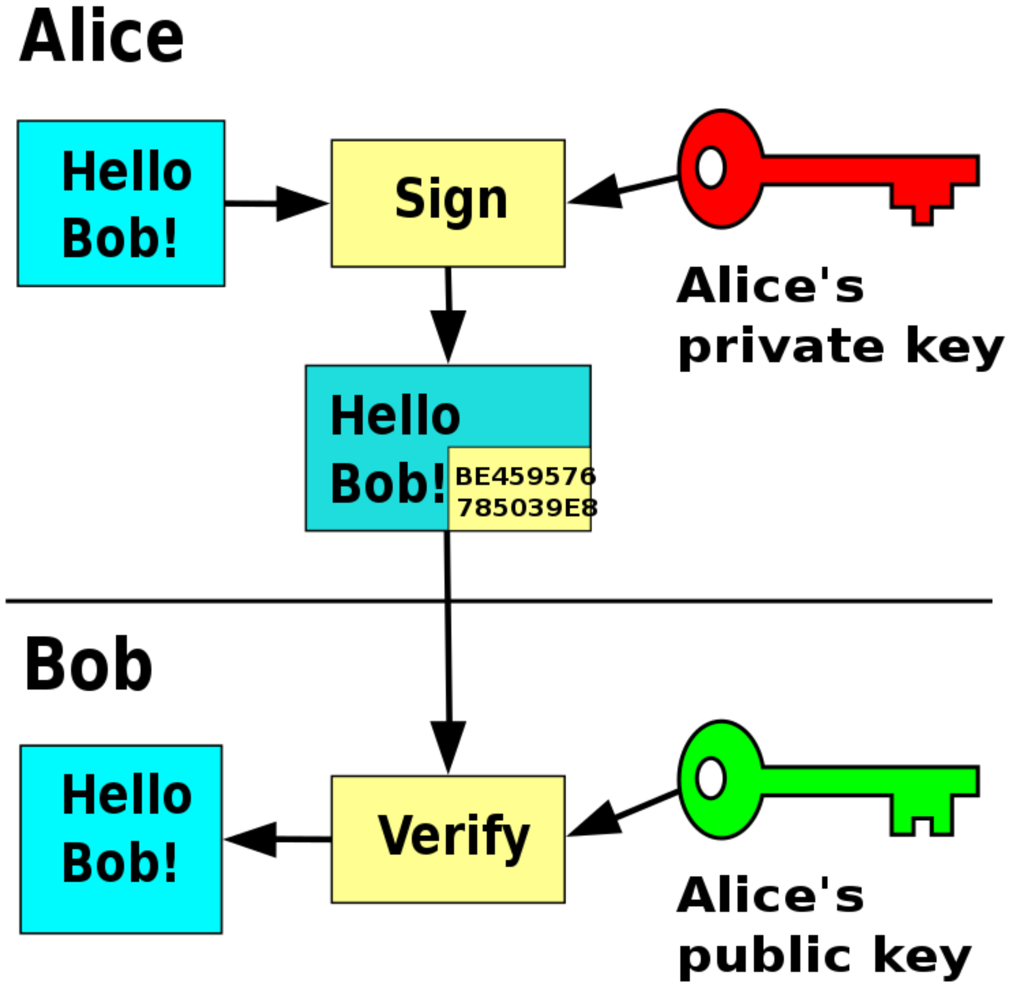
- What type of certificate would help reduce administrative effort?
- **Wildcard certificate**

Digital Signatures

Digital Signature

- Uses asymmetric cryptography
- Simulates security properties of a written signature in digital form
- Can be applied to documents, code, files, network packets, certificates, and other data
  - Attached to the original; does not encrypt the original
  - Can be verified using the associated public key
- Must be unforgeable and authentic
- Proves the integrity and identity of the data it signs
  - Uses both the signer's private key and a hash
- Provides non-repudiation
  - Since it uses the signer's private key, the signer cannot disavow it
  - You can be legally liable for documents that contain your digital signature

Digital Signature Example



Digital Signature Schemes

- RSA
- Digital Signature Algorithm (DSA)
  - Specific by FIPS 186-2
  - Used to generate and verify digital signatures for unclassified, sensitive applications
- Commercial online services:
  - Adobe Sign
  - DocuSign

Question

- What allows for the attribution of messages to individuals?
- **Non-repudiation**

Question #2

- Which two cryptographic concepts do you utilize when implementing non-repudiation?
- **Private key, hashing**

Question #3

- What is assured when a user signs an email using a private key?
- **Non-repudiation**

Public Key Infrastructure

- PKI Components
- PKI Process
- Certificate Authorities
- Key Escrow

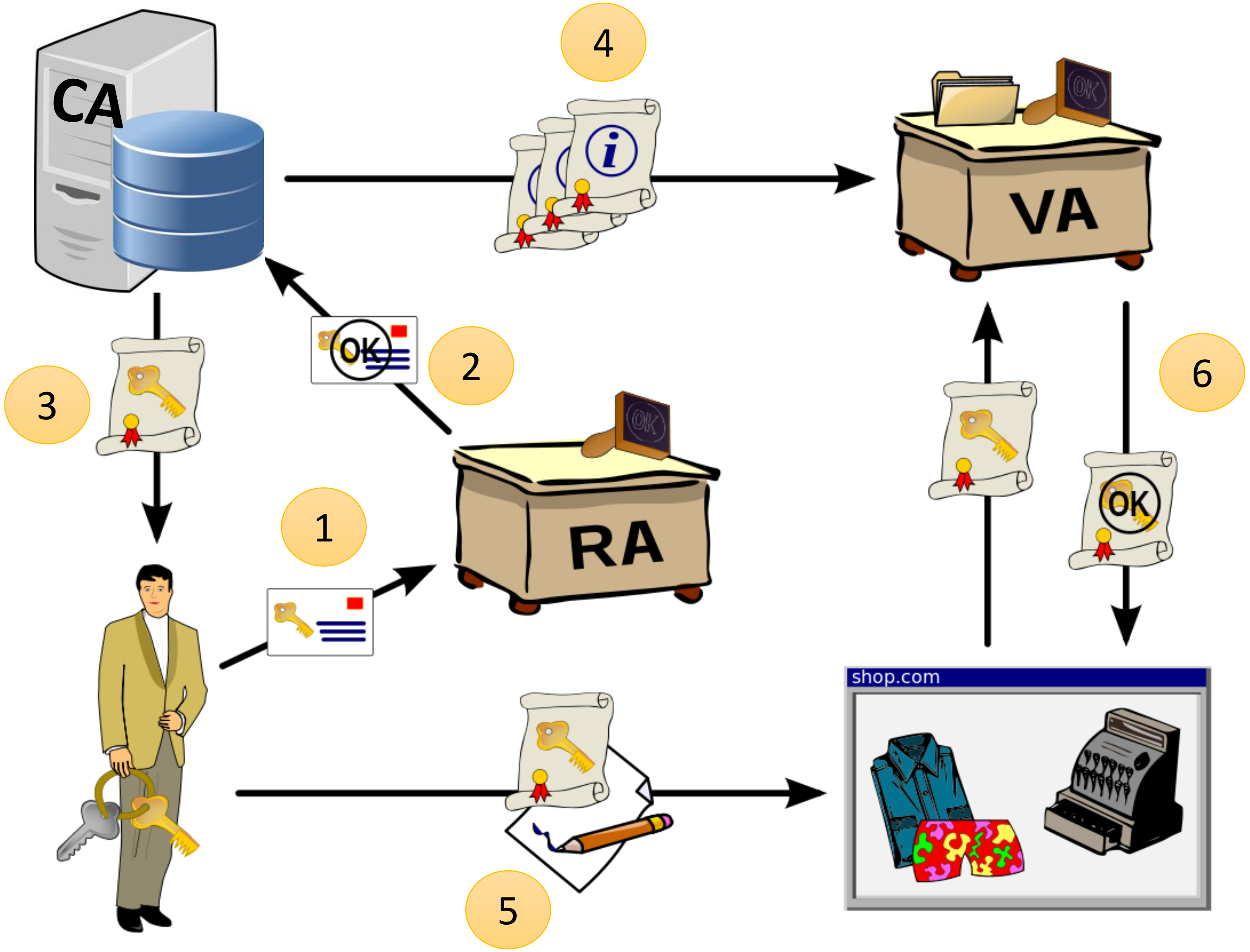
Public Key Infrastructure (PKI)

- PKI is an arrangement that ”binds” public keys with respective identities of entities
  - People, organizations, devices, services
- PKI is a set of roles, policies, hardware, software and procedures
  - Used to create, manage, distribute, use, store and revoke digital certificates and manage public-key encryption
- Used to facilitate the secure electronic transfer of information for a range of network activities including:
  - e-commerce, Internet banking, confidential email
- PKI is required for activities where:
  - Simple passwords are an inadequate authentication method
  - More rigorous proof is required to confirm the identity of the parties involved in the communication
  - The information being transferred needs to be validated

PKI Components

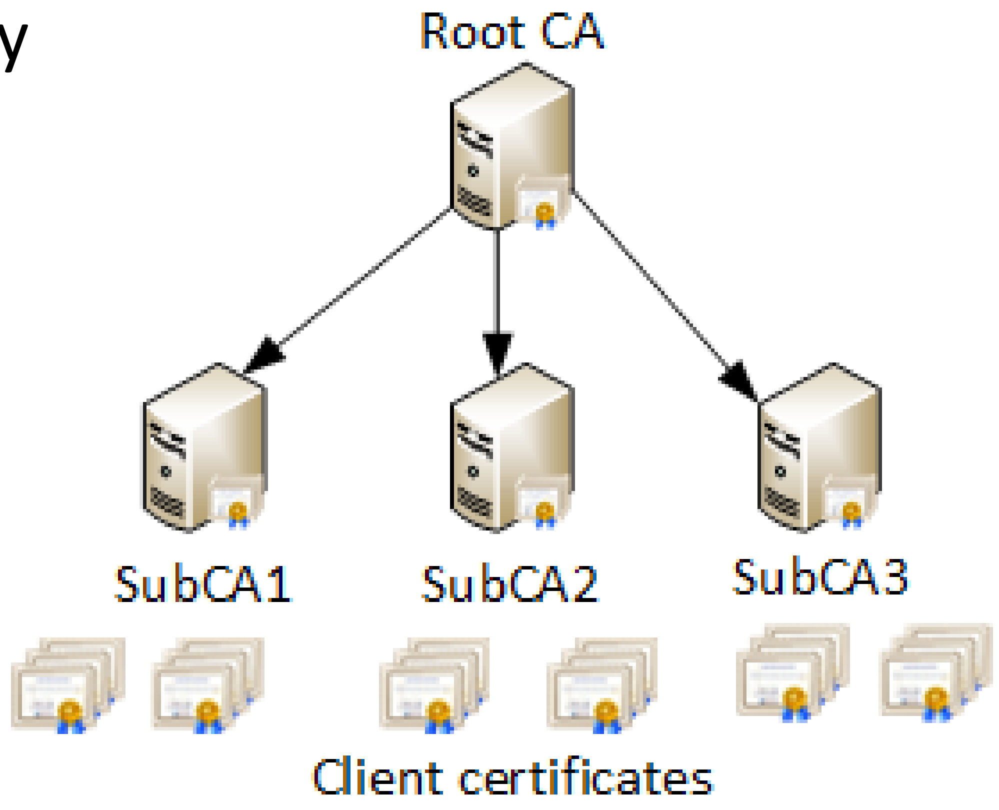
- Certificate Authority (CA)
  - AKA Certification Authority
  - A service that registers and issues certificates
  - May be automated or manual
- Registration Authority
  - A role that may be delegated by a CA to assure valid and correct registration
  - Responsible for accepting requests for digital certificates and authenticating/verifying the entity making the request
- Validation Authority
  - Validates the identity of an entity bearing a certificate
- Certificates
  - A document issued by the CA
  - Contains the issued public key
  - Is accompanied by a private key

PKI Process



Certificate Authority Hierarchy

- The Root CA is the highest authority
  - It self-signs its own root certificate
- It issues certificates to digitally sign subordinate CAs
- The subordinate CAs issue certificates to users and devices



Popular Public Certification Authorities

- VeriSign
- Digicert
- Godaddy
- Microsoft
- COMODO
- Norton Symantec
- Thawte
- Entrust

Implications of Implementing Your Own CA

- Many organizations implement their own internal PKI system, including root and subordinate Certificate Authorities
- These internal CAs are typically referred to as “Enterprise CAs”
- Certificates issued by an Enterprise CA will be unknown and therefore untrusted by the general public
- You will have to yourself distribute the certificate of your Enterprise Root CA to all of your users, devices, and services
    - Can be done automatically using various tools such as Active Directory Group Policy

Key Escrow

- A special component of PKI
- The private key is held in escrow, or stored, by a third party
- Helps protect the private key from unauthorized access or compromise
- The private key can be retrieved if necessary
- Keys held in escrow can also be divided into parts
    - Each part is stored by a different entity
    - All parts must be retrieved and put together to recreate the private key
    - This reduces the risk of fraud and collusion

Example Implementations of Key Escrow

- Microsoft Active Directory
  - The directory service stores a copy of the private key of every certificate issued to domain users and devices.
  - The private key can be accessed for recovery purposes by the user who received the certificate, or an authorized administrator.
- Public Root CA
  - Public Root Certification Authorities will take the private key of their self-signed certificate and place it in special offline storage.
  - The private key might even be divided into parts, with each part being stored by a different entity.

Question

- You have a public root certification authority that is issuing certificates to e-commerce sites.
- What can you use to protect the root certificate private key from compromise?
- **Key escrow**

Data and Keys

- Data States
- Key Stretching
- Perfect Forward Secrecy

Data States

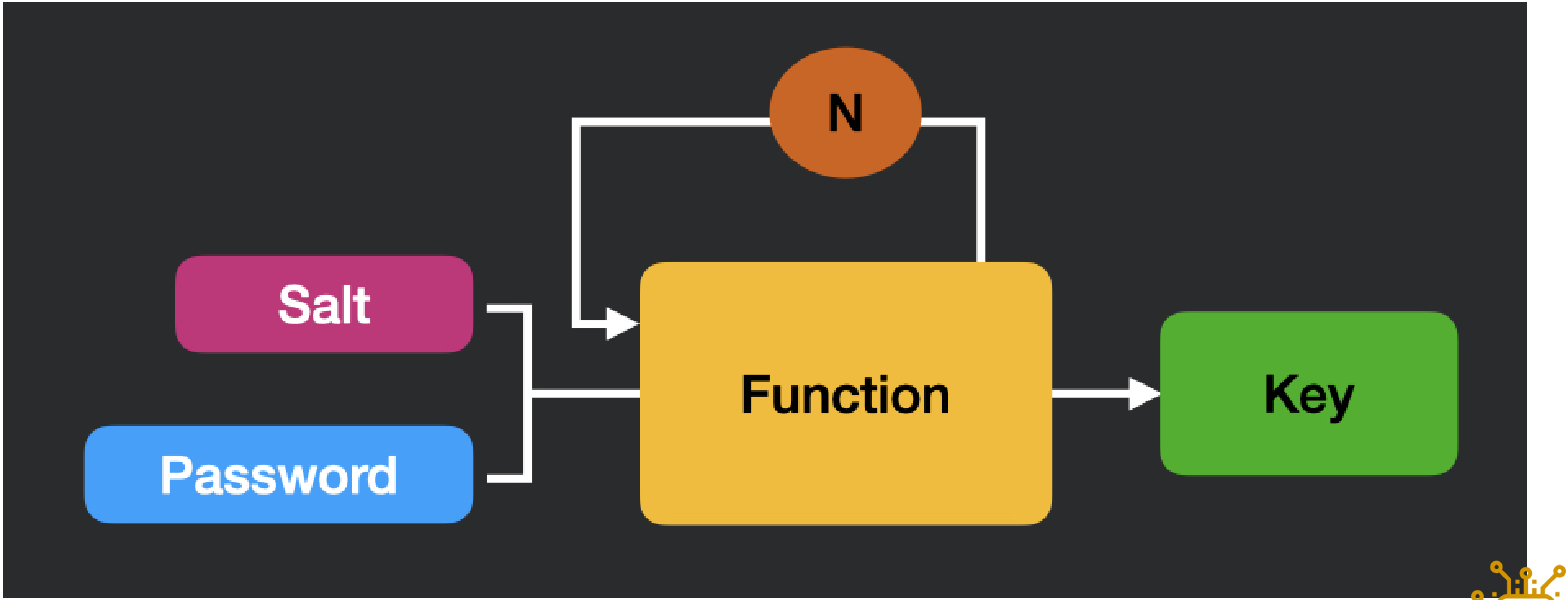
- Data at Rest
  - Stored on a hard drive, USB stick, CD/DVD, or any other type of electronic storage medium
  - Data can be encrypted at any level including full-disk, partition, file, volume, database, record, string of text, image, etc.
- Data in Transit
  - Data is actively being transmitted on a network
- Data in Use
  - Data is loaded into memory
  - Is being, or will shortly be, processed by the CPU

You can encrypt data in any of these states to increase confidentiality and trust

Key Stretching

- The practice of transforming a “weak” key (password) into something that is computationally harder to crack
  - Makes the key longer and more random
- Requires a key-stretching algorithm such as bcrypt, PBKDF2, scrypt, or Argon2
- Common approaches to key stretching include:
  - Sending a salted key through multiple rounds of hashing
  - Apply a block cipher to the key repeatedly in a loop
- 7zip, PGP, WPA, and WPA2 all use key stretching

Key Stretching Example



Perfect Forward Secrecy

- AKA forward secrecy (FS)
- Refers to an encryption system that changes the keys used to encrypt and decrypt information frequently and automatically
- Ensures that even if the most recent key is hacked, a minimal amount of sensitive data is exposed
- Loss or theft of one decryption key does not compromise any additional sensitive information—including additional keys
    - Even if one session key is compromised, it cannot be used to decrypt other sessions
- Web pages, calling apps, and messaging apps all use encryption tools with perfect forward secrecy
    - Switch their keys as often as each call or message in a conversation, or every reload of an encrypted web page

Question

- A bank insists all of its vendors must prevent data loss on stolen laptops.
- What encryption strategy is the bank requiring?
- **Encryption for data at rest**

Question #2

- If a current private key is compromised, what would ensure it cannot be used to decrypt all historical data?
- **Perfect forward secrecy**

Question #3

- You have an encrypted thumb drive with data on it.
- As you carry the thumb drive to another office, what state is the data in (at rest, in transit, in use)?
- **At rest**

Question #4

- What technique is used to make a key longer and harder to crack?
- **Key stretching**

Question #5

- What technique is used to ensure that if a key is compromised, only a minimal amount of data is exposed?
- **PFS**

Question #6

- You regularly perform backups of your critical servers
- You can’t afford to send the backup tapes to an off-site vendor for long-term storage and archiving
- Instead, you store the backup tapes in a safe in your office
- Your manager wants to take the tapes home in her briefcase every night
- What can she do to secure those tapes while in transit?
- **Encrypt the backup tapes**
- **For good measure, have her carry them in a lockbox and not just her briefcase**

In this scenario, the data is still considered to be “at rest”.
Even though someone is physically carrying the storage media to another location, the data itself is not being transmitted across a network where it can be intercepted by a sniffer

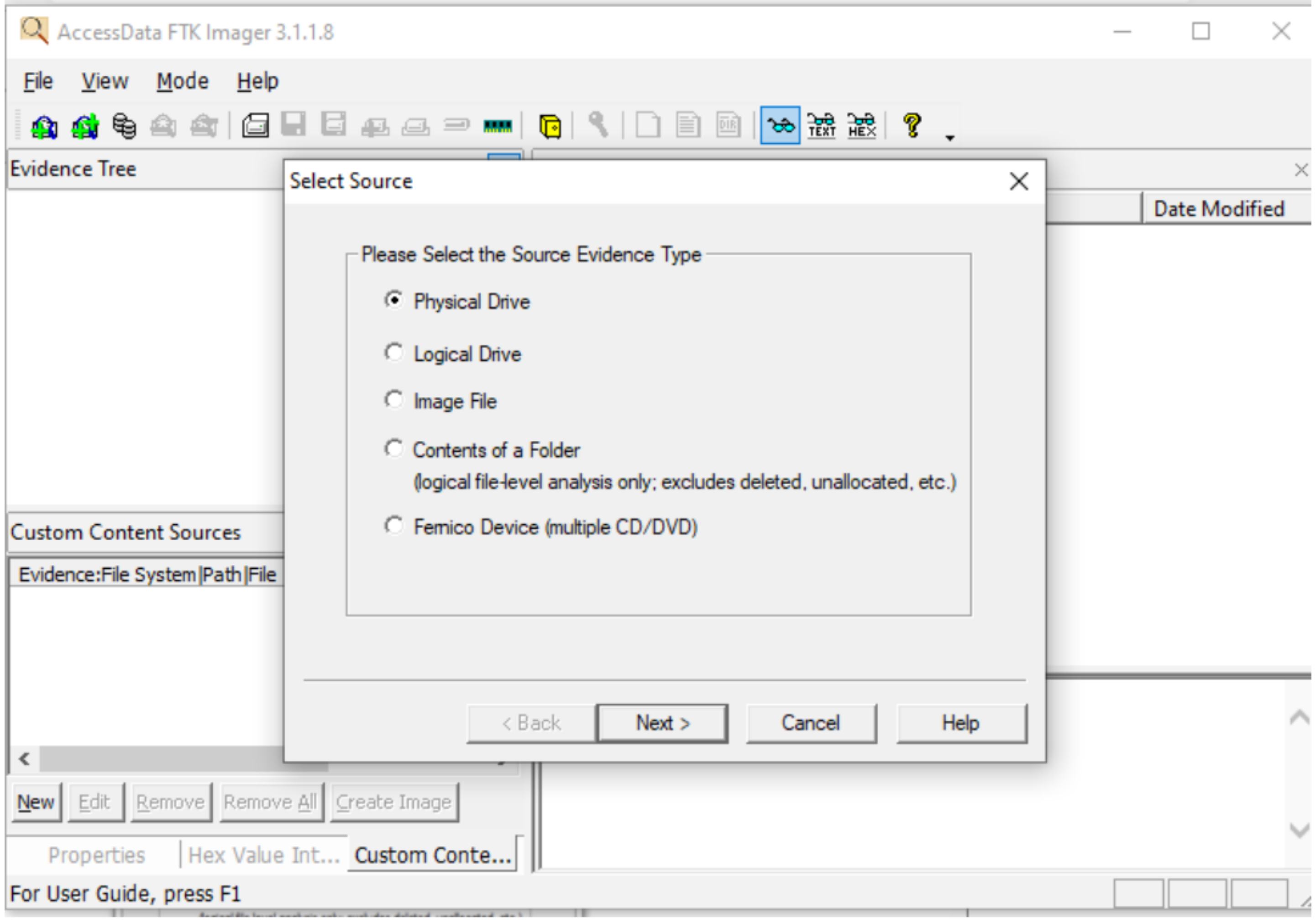
Crypto Implementations

- Hashing
- Disk Copies
- Protocols
- TPM
- SED
- HSM
- KMS
- Secure Enclave

The Role of Hashing in Cyber Forensics

- The first thing that must be done after acquiring a forensic disk image is to:
  - Create a hash digest of the source drive and destination image file
  - Ensure they are identical
- A critical step in the presentation of evidence will be to prove:
  - Analysis has been performed on an identical image to the data present on the physical media
  - Neither data set has been tampered with
- The standard means of proving this is to create a cryptographic hash (fingerprint) of the disk contents and any derivative images made from it
- When comparing hash values, you need to use the same algorithm used to create the reference value

Forensic Disk Copy Examples



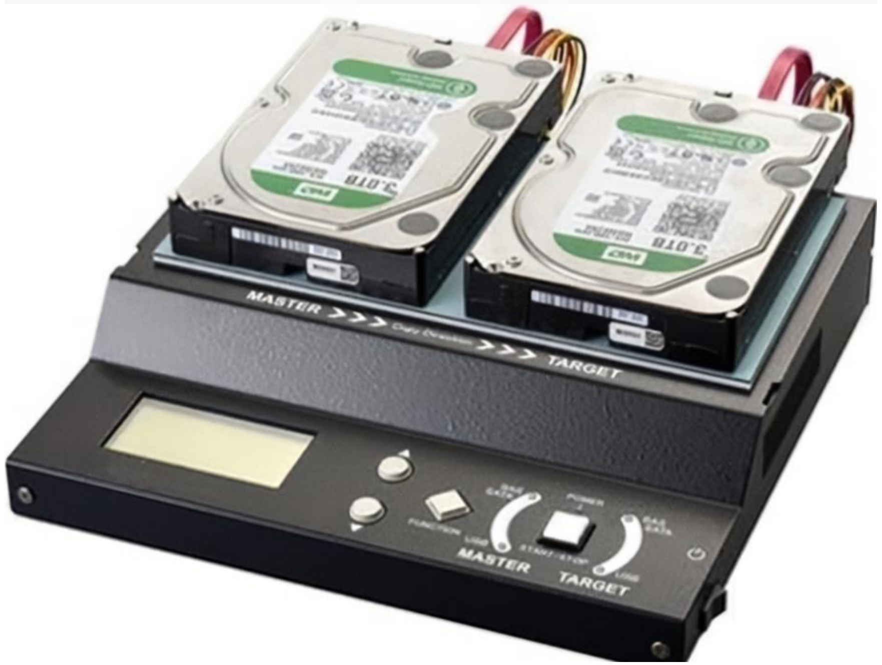
AccessData FTK Imager 3.1.1.8
File View Mode Help
Evidence Tree
Date Modified
Custom Content Sources
Evidence:File System|Path|File
New Edit Remove Remove All Create Image
Properties Hex Value Int... Custom Conte...
For User Guide, press F1

Select Source

Please Select the Source Evidence Type

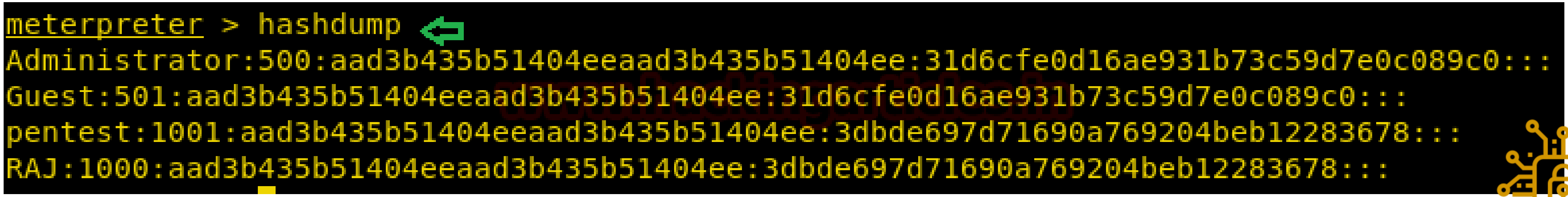
- ● Physical Drive
- ○ Logical Drive
- ○ Image File
- ○ Contents of a Folder
  (logical file-level analysis only; excludes deleted, unallocated, etc.)
- ○ Fernico Device (multiple CD/DVD)

< Back
Next >
Cancel
Help



Pass-the-hash Attack

- A hacking technique that allows an attacker to authenticate without knowing the password
  - The username and password are not entered via a normal login screen
  - Instead, the password *hash* is provided over the network using a special app
- Used when a password is too difficult to crack
- Requires the attacker to obtain the password hash ahead of time
- Hashes can be dumped from memory using tools such as:
  - Mimikatz, psexec, Metasploit meterpreter, fgdump, pwdump, cachedump, etc.



```
meterpreter > hashdump
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
pentest:1001:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
RAJ:1000:aad3b435b51404eeaad3b435b51404ee:3dbde697d71690a769204beb12283678:::
```

Protocols that Use Asymmetric Cryptography

- SSL/TLS
- S/MIME
- PGP/GPG
- SSH
- Internet Key Exchange (IKE) for IPSEC

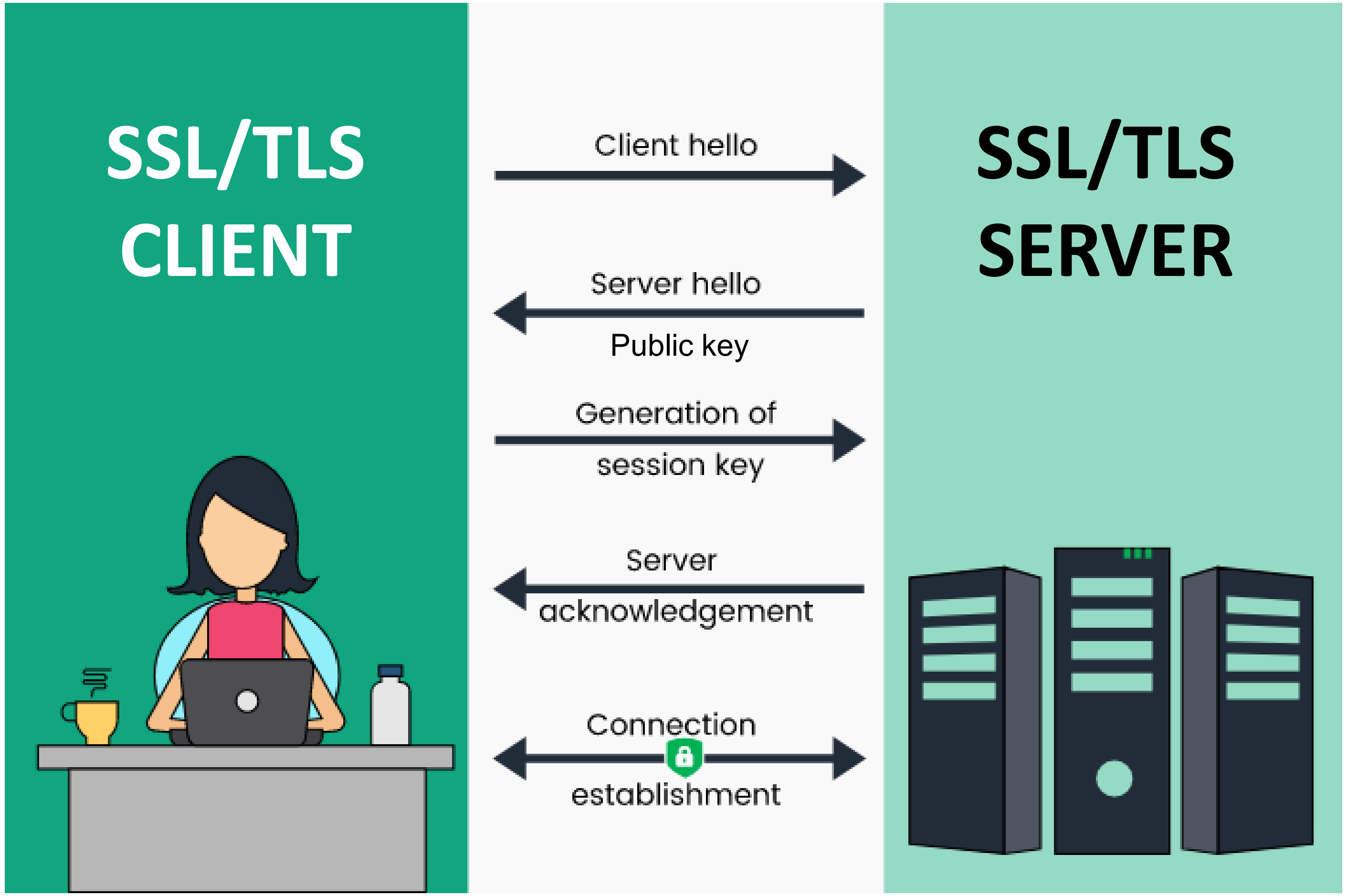
Secure Sockets Layer (SSL)

- Protocol that establishes a secure connection between a client and server
- Used to secure confidentiality and integrity of data transmissions over the Internet
  - Particularly used by HTTPS to encrypt web traffic
  - Server proves its identity to the client
  - Server provides its public key to client
- Allows a client and server to:
  - Authenticate each other
  - Choose an encryption algorithm
  - Exchange public keys
  - Create a temporary session key
- Uses RSA asymmetric encryption
- Last version was SSL 3.0
- Has been replaced by TLS
- No longer considered secure
- Most modern browsers no longer support SSL

Transport Layer Security (TLS)

- The successor to SSL
- Fixes SSL security vulnerabilities
- Uses stronger encryption algorithms
- Can work over different ports
- More standardized
  - Can support emerging encryption algorithms
- Currently at version 1.3

SSL/TLS Session Example



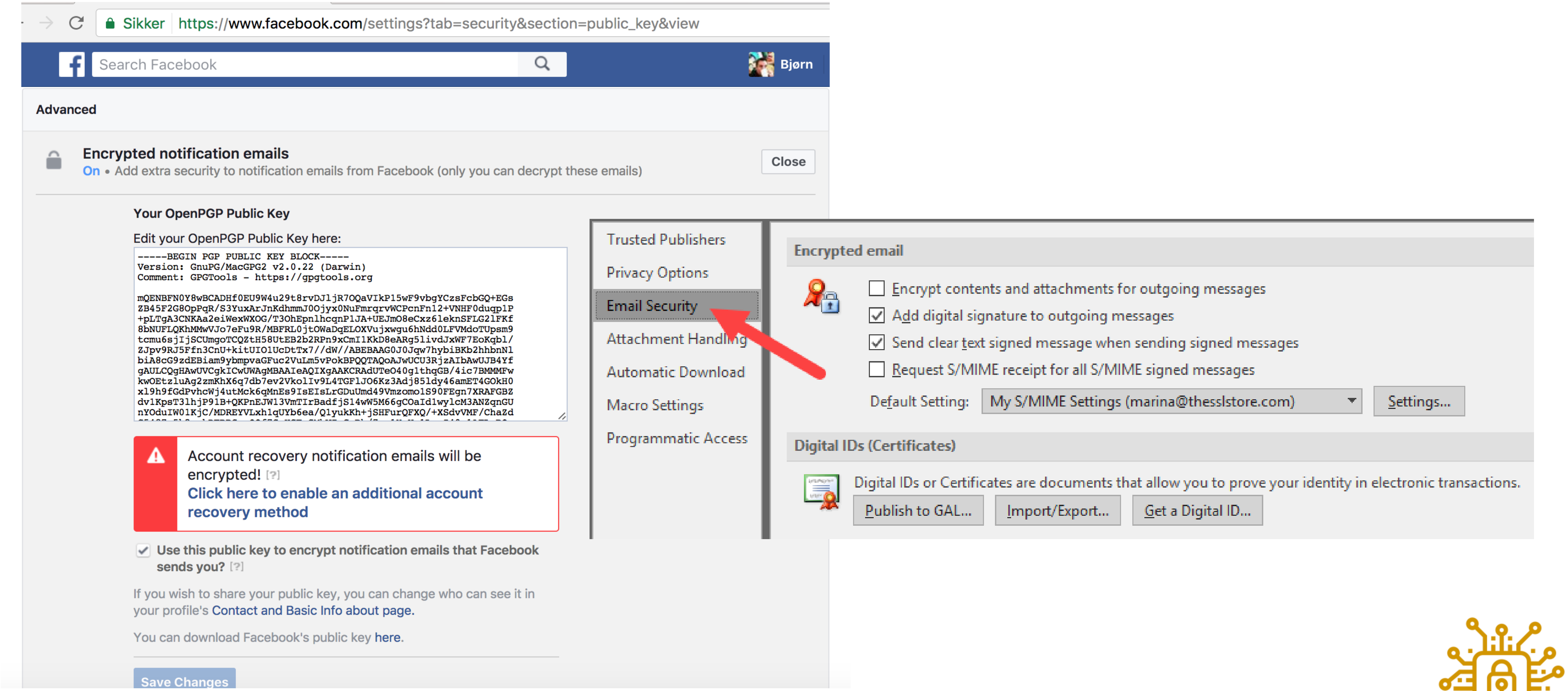
OpenSSL

- A general purpose cryptography library
- Open-source implementation of the SSL and TLS protocols
    - Performs encryption/decryption
- Includes tools for generating:
    - RSA private keys
    - Certificate Signing Requests (CSRs)
    - Checksums
- Can manage certificates
- Widely used by Internet servers and the majority of HTTPS websites

S/MIME

- S/MIME (Secure/Multipurpose internet Mail Extensions) is a widely accepted protocol for sending digitally signed and encrypted email messages
- You can use an online secure email provider or your local email client
- Obtain or create a certificate (public key)
- Select the certificate in the email client
  - Alternatively, upload the certificate to the email provider
- In an enterprise environment, user certificates are distributed and managed by the email server and/or directory service

Securing Email Example



Sikker https://www.facebook.com/settings?tab=security&section=public_key&view

Advanced

Encrypted notification emails

On • Add extra security to notification emails from Facebook (only you can decrypt these emails)

Your OpenPGP Public Key

Edit your OpenPGP Public Key here:

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: GnuPG/MacGPG2 v2.0.22 (Darwin)
Comment: GPGTools - https://gpgtools.org

mQENBFN0Y8wBCADHf0EU9W4u29t8rvDJljR7OQaVIkP15wF9vbgYCzsFcbGQ+EGs
ZB45F2G8OpPqR/S3YuxArJnKdhmmJ0Ojyx0NuFmrqrvWCPcnFn12+VNHF0duqp1P
+pLTgA3CNKAa2eiWexWXOG/T3OhEpnlhcqnP1JA+UEJmO8eCxz6leknSFLG2lFKf
8bNUFLQKhMMwVJo7eFu9R/MBFRL0jtOWaDqELOXVujxwgu6hNdd0LFVMdoTUpsm9
tcmu6sjIjSCUmgoTCQZtH58UtEB2b2RPn9xCmI1KkD8eARg5livdJxWF7EoKqbl/
ZJpv9RJ5Ffn3CnU+kitUIO1UcDtTx7//dW//ABEBAAG0J0Jqw7hybiBKb2hhbnNl
biA8cG9zdEBiam9ybmpvaGFuc2VuLm5vPokBPQQTAQoAJwUCU3RjzAIbAwUJB4Yf
gAULCQgHAwUVCgkICwUWAgMBAAIeAQIXgAAKCRAdUTeO40g1thqGB/4ic7BMMMFw
kwOEtzluAg2zmKhX6q7db7ev2VkolIv9L4TGFlJO6Kz3Adj85ldy46amET4GOkH0
xl9h9fGdPvhcWj4utMck6qMnEs9IsEIsLrGDuUmd49Vmzomo1S90FEgn7XRAFGBZ
dv1KpsT3lhjP91B+QKPnEJW13VmTIrBadfjS14wW5M66gCOaId1wylcM3ANZqnGU
nYOduIW01KjC/MDREYVLxh1qUYb6ea/Q1yukKh+jSHFurQFXQ/+XSdvVMF/ChaZd
```

Account recovery notification emails will be encrypted! [?]

Click here to enable an additional account recovery method

☑ Use this public key to encrypt notification emails that Facebook sends you? [?]

If you wish to share your public key, you can change who can see it in your profile's Contact and Basic Info about page.

You can download Facebook's public key here.

Encrypted email

- □ Encrypt contents and attachments for outgoing messages
- ☑ Add digital signature to outgoing messages
- ☑ Send clear text signed message when sending signed messages
- □ Request S/MIME receipt for all S/MIME signed messages

Default Setting: My S/MIME Settings (marina@thesslstore.com)

Digital IDs (Certificates)

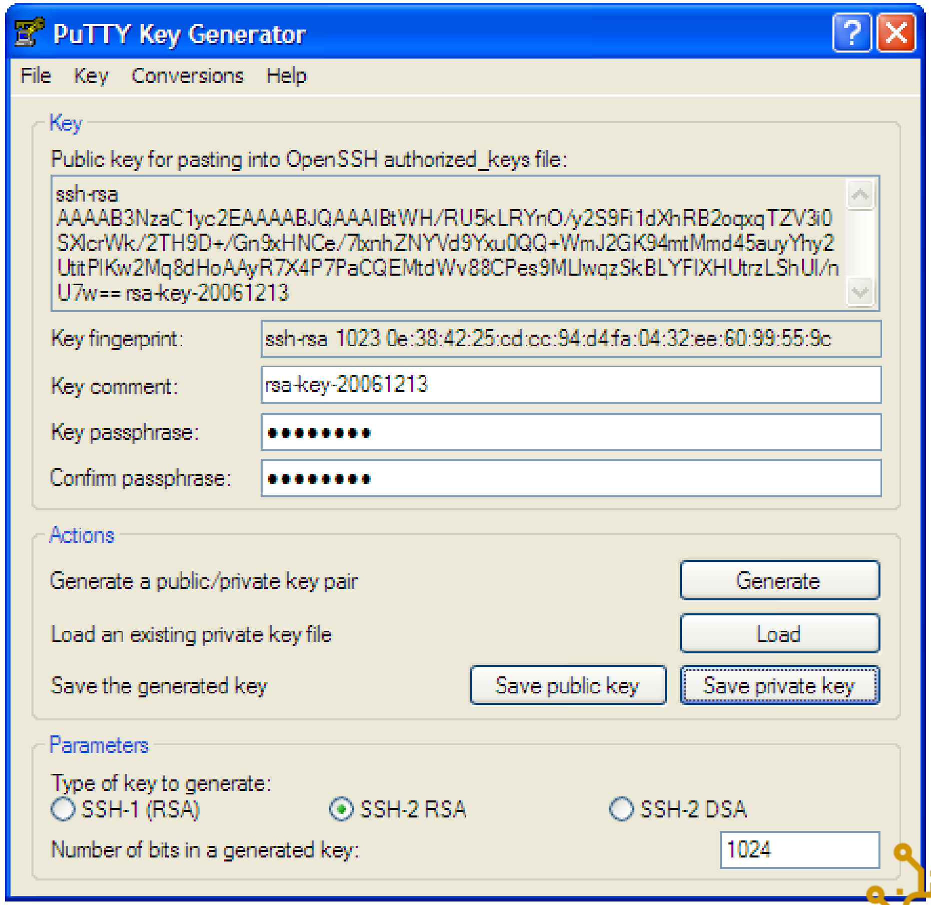
Digital IDs or Certificates are documents that allow you to prove your identity in electronic transactions.

Pretty Good Privacy (PGP)

- System for creating asymmetric key pairs and trading public keys
- Provides authentication and cryptographic privacy
- Used for digital signing, data compression, and to encrypt/decrypt emails, messages, files, and directories
- You can search MIT’s PGP Public Key Server
  - Use information about the person such as their email address
  - If someone’s public key is found, you can download it and put it on your key ring
- PGP was sold to Symantec in 2010
- Open source replacement is GPG

SSH Key Generation

- Tools such as PuTTY can create a key pair
- You can then use the generated public key to establish an SSH session



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAIBtWH/RU5kLRYnO/y2S9Fi1dXhRB2oqxqTZV3i0
SXlcrWk/2TH9D+/Gn9xHNCe/7bxnhZNYVd9Yxu0QQ+WmJ2GK94mtMmd45auyYhy2
UtitPIKw2Mq8dHoAAyR7X4P7PaCQEMtdWv88CPes9MLlwqzSkBLYFIXHUtrzLShUl/n
U7w== rsa-key-20061213
```

Key fingerprint: ssh-rsa 1023 0e:38:42:25:cd:cc:94:d4:fa:04:32:ee:60:99:55:9c

Key comment: rsa-key-20061213

Key passphrase: •••••••••

Confirm passphrase: •••••••••

Generate a public/private key pair

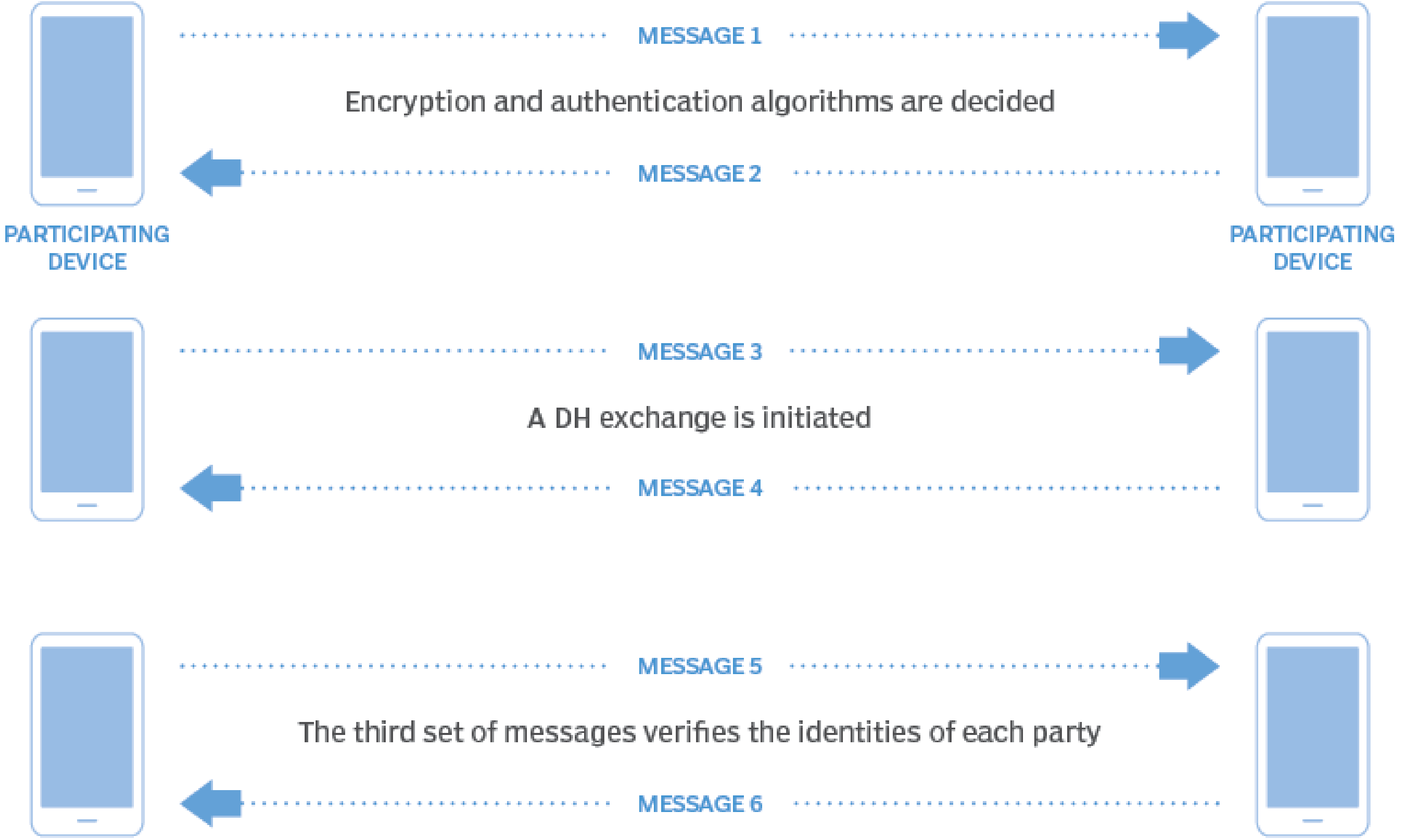
Load an existing private key file

Save the generated key

Type of key to generate:
○ SSH-1 (RSA) ● SSH-2 RSA ○ SSH-2 DSA

Number of bits in a generated key: 1024

IKE Example

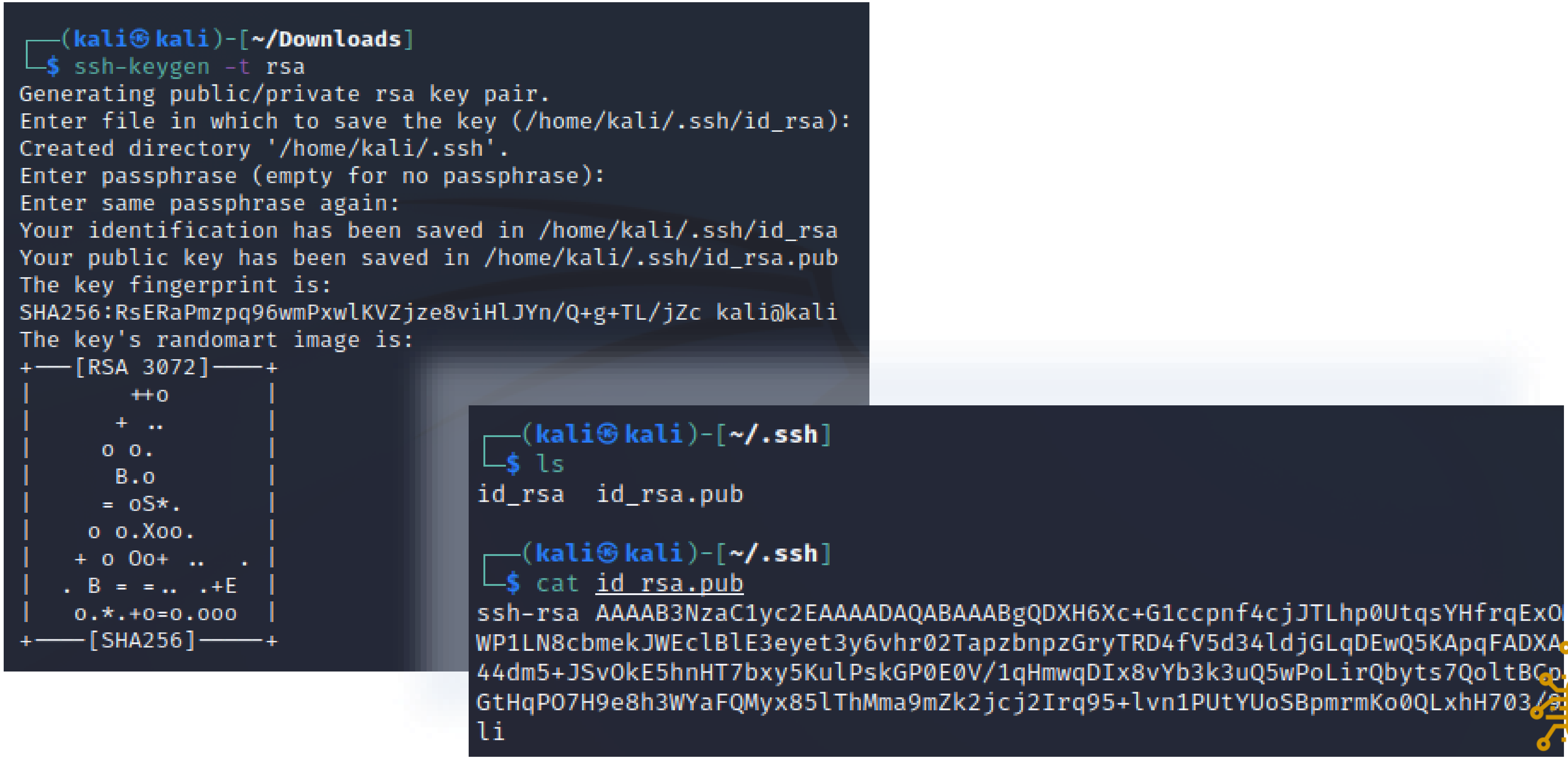


Encryption and authentication algorithms are decided

A DH exchange is initiated

The third set of messages verifies the identities of each party

Generating Your Own Key Pair Example



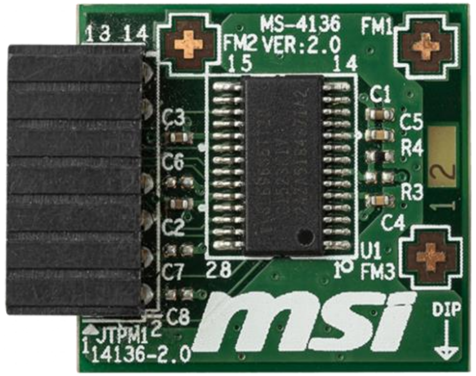
```
┌──(kali㉿kali)-[~/Downloads]
└─$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/kali/.ssh/id_rsa):
Created directory '/home/kali/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/kali/.ssh/id_rsa
Your public key has been saved in /home/kali/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:RsERaPmzpq96wmPxwlKVZjze8viHlJYn/Q+g+TL/jZc kali@kali
The key's randomart image is:
+---[RSA 3072]----+
|       ++o       |
|      + ..       |
|     o o.        |
|      B.o        |
|     = oS*.      |
|    o o.Xoo.     |
|   + o Oo+ ..   .|
|  . B = =.. .+E  |
|   o.*.+o=o.ooo  |
+----[SHA256]-----+
```

```
┌──(kali㉿kali)-[~/.ssh]
└─$ ls
id_rsa  id_rsa.pub

┌──(kali㉿kali)-[~/.ssh]
└─$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQDXH6Xc+G1ccpnf4cjJTLhp0UtqsYHfrqExO
WP1LN8cbmekJWEclBlE3eyet3y6vhr02TapzbnpzGryTRD4fV5d34ldjGLqDEwQ5KApqFADXA
44dm5+JSvOkE5hnHT7bxy5KulPskGP0E0V/1qHmwqDIx8vYb3k3uQ5wPoLirQbyts7QoltBCp
GtHqPO7H9e8h3WYaFQMyx85lThMma9mZk2jcj2Irq95+lvn1PUtYUoSBpmrmKo0QLxhH703/
li
```

Trusted Platform Module (TPM)

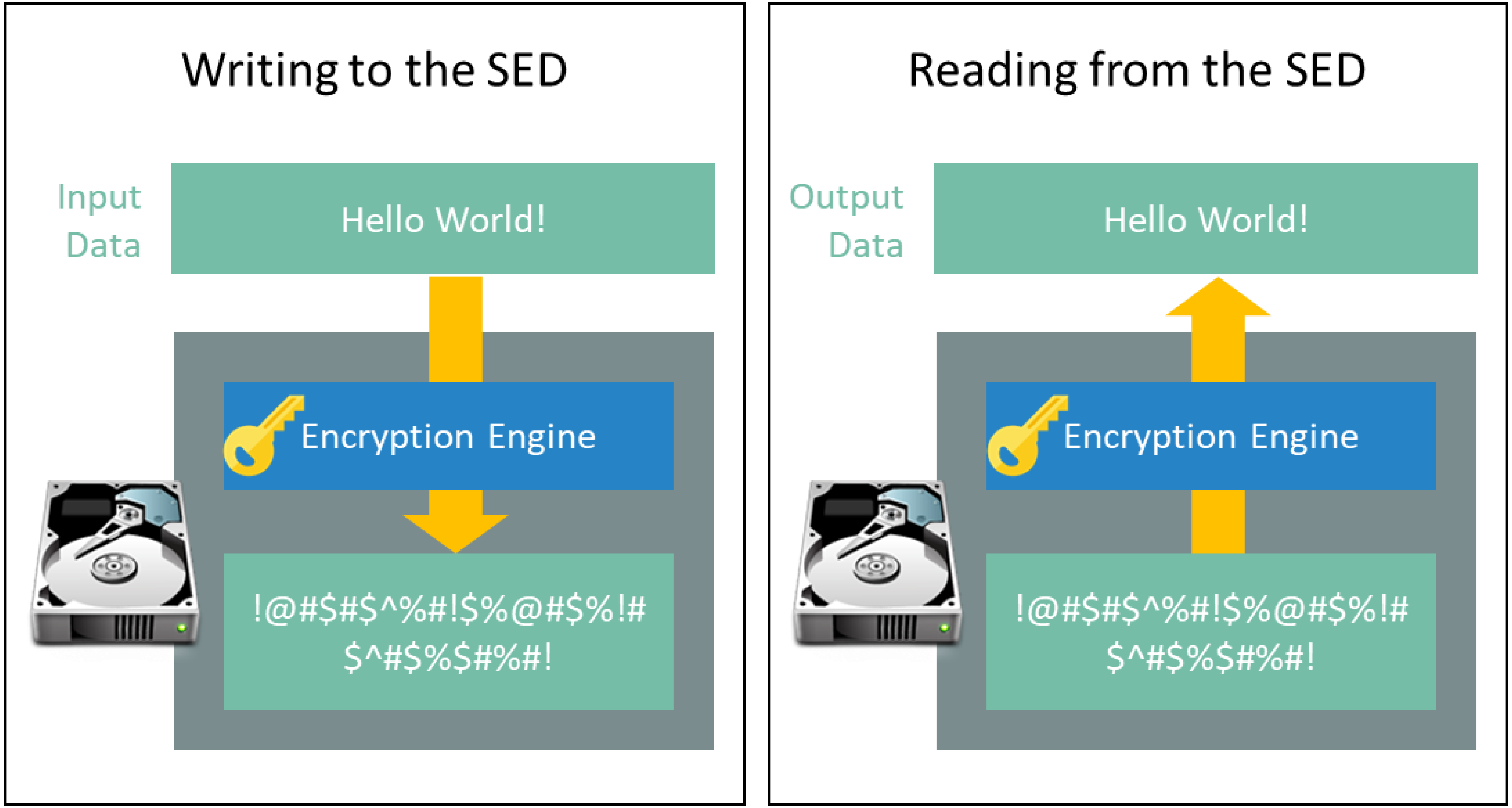
- A security chip embedded in the motherboard of a laptop or plugged into a desktop PC
- Acts as a “lockbox” for encryption keys
  - Microsoft Active Directory can provide key escrow to store TPM recovery keys
- Can be used to:
  - encrypt a drive (full disk encryption, or FDE)
  - verify OS integrity on bootup
- Operation is completely transparent to the user



Self-Encrypting Drive (SED)

- Alternative solution to TPM for full disk encryption
- The drive has an encryption key loaded into its firmware
  - The drive has its own on-board crypto processor to encrypt and decrypt data
- Requires a user to enter a key (password) at bootup
- Drive can be internal or external

SED Example



Writing to the SED

Input Data

Hello World!

Encryption Engine

!@#$#$^%#!$%@#$%!#
$^#$%$#%#!

Reading from the SED

Output Data

Hello World!

Encryption Engine

!@#$#$^%#!$%@#$%!#
$^#$%$#%#!

Hardware Security Module (HSM)

- Very secure dedicated hardware for securely storing cryptographic keys
- Can encrypt, decrypt, create, store and manage digital keys, and be used for signing and authentication
- Provides extra security for encryption keys by storing the keys on a separate, removeable device

HSM Examples

Caution: Risk of explosion if battery is replaced by an incorrect type. Dispose of used batteries according to the instructions.

Key Management System (KMS)

- A system that controls the entire lifecycle of cryptographic keys including:
  - Generation, Distribution, Use, Storage, Rotation, Backup/Recovery, Revocation, and Destruction
- Typically implemented as a server

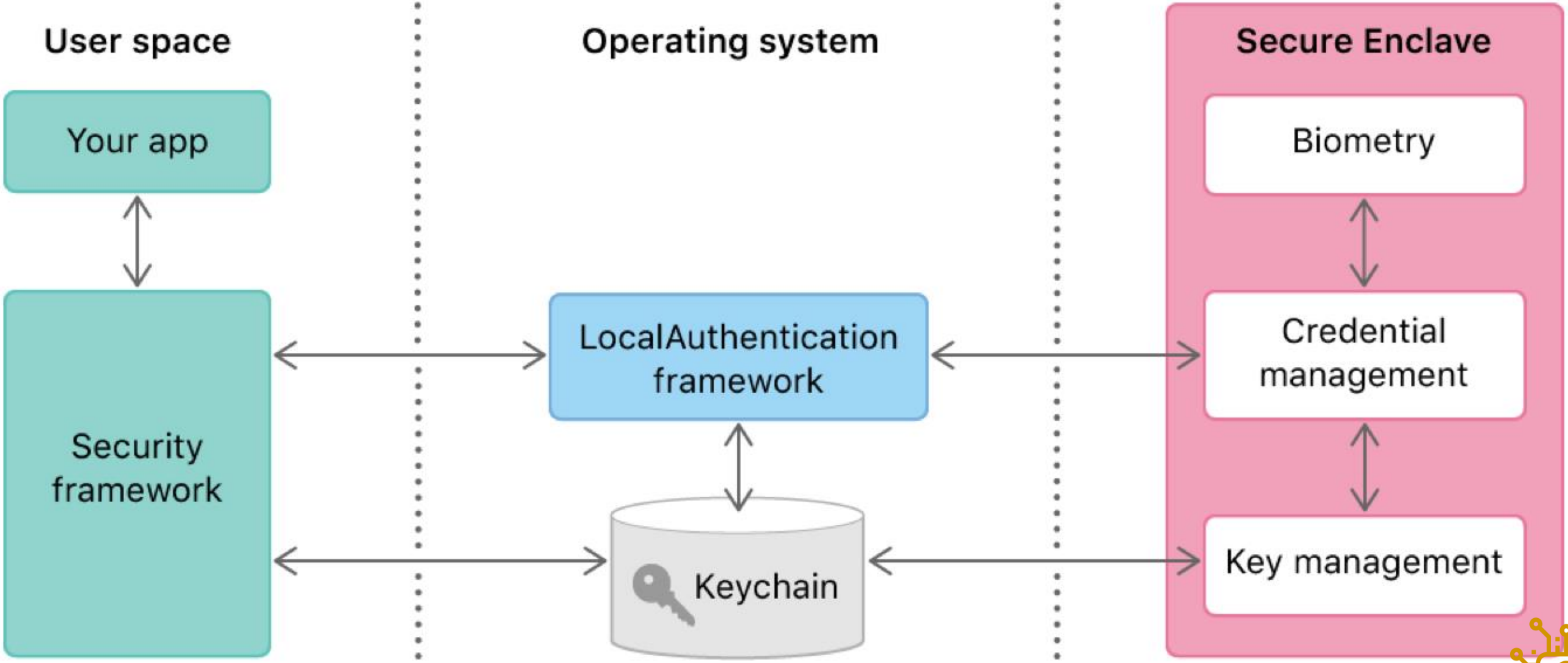
Secure Enclave

- A hardware-based trusted computing component of a microchip
- Found in modern macOS, iOS, and Android devices
- Designed to generate and store encryption keys as well as process critical information such as biometric data
- Responsible for handling sensitive operations relating to security and privacy
- Secure enclave components cannot be used outside of the device if removed
- Attackers cannot replace secure enclave components with malicious counterparts

Secure Enclave and Private Keys

- When you store a private key in a Secure Enclave, you never actually handle the key
- Instead, you instruct the Secure Enclave to create the key, securely store it, and perform operations with it
- You receive only the output of these operations, such as encrypted data or a cryptographic signature verification outcome.
- This makes it difficult for the key to become compromised

Apple Device Secure Enclave Example



Question

- Against your recommendation, the company set all user passwords on a server to “P@55w0rD”.
- Upon review of the /etc/passwd file, an auditor found the following:

```
alice:a8df3b6c4fd75f0617431fd248f35191df8d237f
bob:2d250c5b2976b03d757f324ebd59340df96aa05e
moo:ea981ec3285421d014108089f3f3f997ce0f4150
```

- Why do the encrypted passwords do not match?
- **Each password has a different salt, so their hashes are different.**

Question #2

- You want to store customer data in the cloud, but still allow the data to be accessed and manipulated while encrypted.
- You do not want the cloud service provider to be able to decipher the data due to its sensitivity.
- You don’t care about computational overheads and slow speeds.
- Which encryption type would BEST meet your requirement?
- **Asymmetric encryption**

Question #3

- An application developer accidentally uploaded a company's code-signing certificate private key to a public web server.
- You are concerned about malicious use of this certificate.
- What should you do FIRST?
- **Revoke the code-signing certificate.**

Question #4

- As part of the building process for a web application, the compliance team requires that all PKI certificates are rotated annually and can only contain wildcards at the secondary subdomain level.
- Which of the following certificate properties meets these requirements?

```
HTTPS://*.comptia.org, Valid from April 10 00:00:00 2021
- April 8 12:00:00 2022
```

- **The wildcard (*) in the subdomain part of the domain name**

Question #5

- You used an internal CA to issue a certificate to a public application
- When users try to connect, they receive the error message “Your connection is not private.“
- What can you do to fix this?
- **Send a CSR to a known CA and install the signed certificate on the application's server**

Question #6

- You want to buy some laptops that support built-in full-disk encryption (FDE).
- Before buying the laptops, what should you ensure is installed on them?
- **A Trusted Platform Module (TPM) chip**

Question #7

- You are implementing FDE for all laptops in your organization.
- You want to make sure that administrators can recover lost private keys as needed.
- What two things should you include in your planning?
- **Key escrow**
- **TPM presence in the laptops**

Question #8

- You want to protect data on employees’ laptops, even if the laptops are stolen.
- What encryption technique should you use?
- **Full disk**

Blockchain

- How it Works
- Use Cases

Blockchain

- A mechanism to store and secure digital data
- Blockchain is an open ledger that several parties can access at once
- Each new record becomes a block with a unique, identifying hash
- Linking the blocks into a chain of records forms a blockchain
- A primary benefit is that the recorded information is hard to change without an agreement from all parties involved
- Cryptocurrency uses blockchain technology

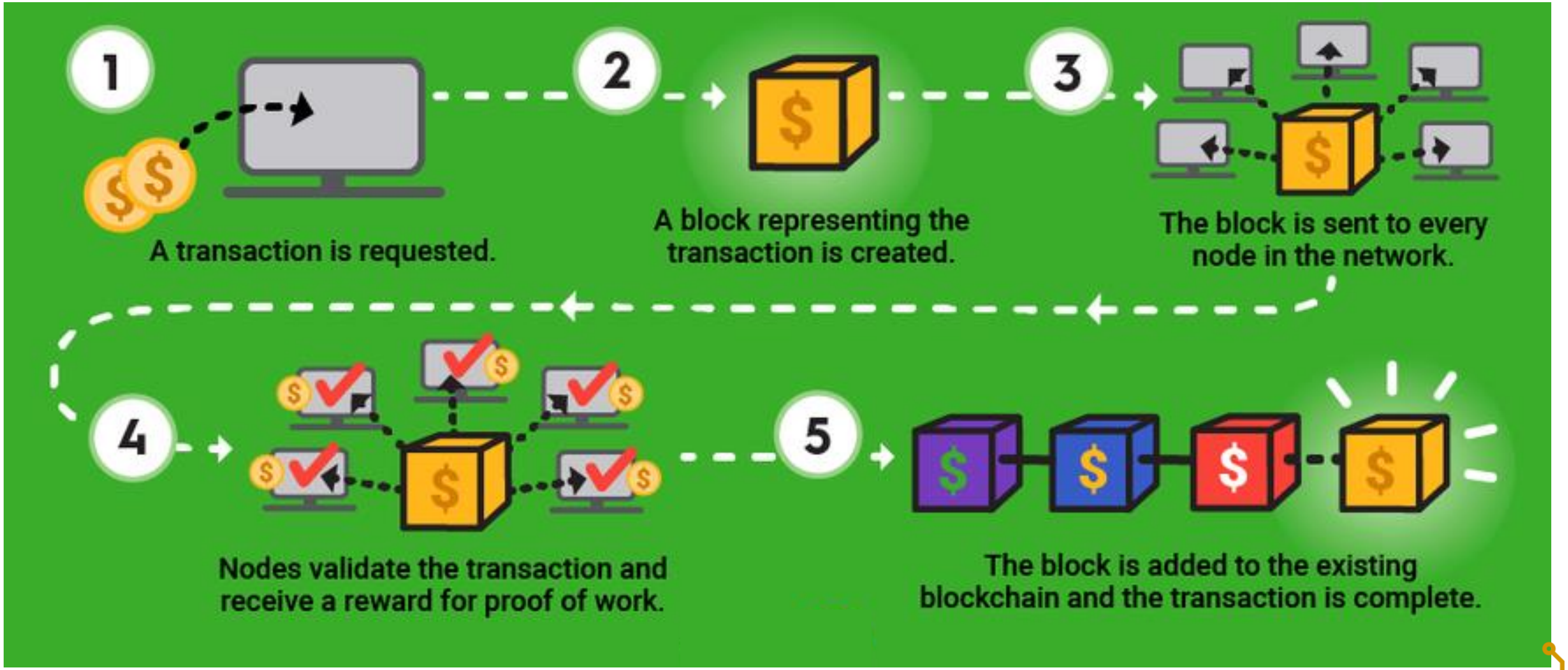
How Blockchain Works

- Blockchain systems rely on a peer-to-peer network of computers that analyze a shared digital ledger at regular intervals
- New transactions must be confirmed by a predetermined number of computer nodes
- When a new block of transactions gets the stamp of approval from enough nodes:
  - the new data is “written in stone”
  - the blockchain moves on to considering another list of new transactions
- Blockchain transactions can involve all sorts of data including:
  - Cryptocurrency transactions
  - Physical goods, car titles, land ownership, and more
  - Logging manufacturing and shipping events along a supply chain
  - Intangible assets like patents, branding, or intellectual property

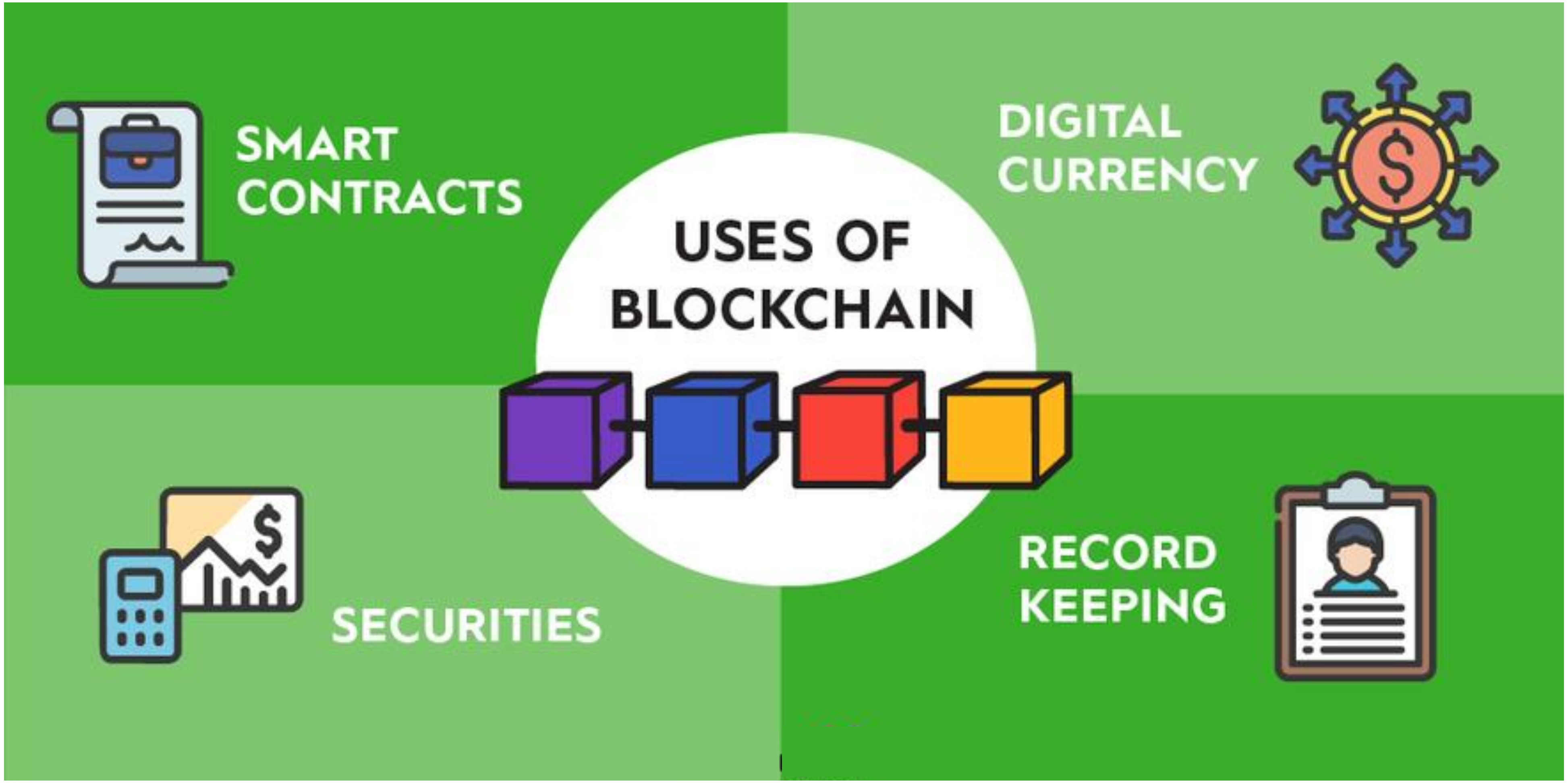
Accessing Data in the Blockchain

- Each block in a blockchain represents a transaction.
- All participating nodes have a copy of the blockchain, and can validate that none of the blocks (records) have been tampered with.
- It is cumbersome and impractical to try to read transaction data directly from the blockchain.
- Instead, you can implement a local database that is updated as each block is added.
- You save a copy of each transaction for the entire history of the blockchain
  - As you commit the updates to your database, you never clear the transaction log
  - You might archive the log periodically to keep its size manageable
- For speed and convenience, you query your database for the information you seek
  - You could review historical transactions in the log for verification or clarity
  - The other nodes in the blockchain can attest that the transactions happened as recorded

Blockchain Example


1
A transaction is requested.
2
A block representing the transaction is created.
3
The block is sent to every node in the network.
4
Nodes validate the transaction and receive a reward for proof of work.
5
The block is added to the existing blockchain and the transaction is complete.

Blockchain Use Case Examples



Open Public Ledger

- A ledger is a record-keeping system
- A digital or physical log that records transactions associated with a system (usually financial)
- It tracks a value as it moves around, so the viewer can always see exactly what value resides where at a given moment
- Traditional finance systems like banks use ledgers to track all transactions completed within a period
- A public ledger (open public ledger) is an open-access network
    - Anyone can join at any time
    - The ledger is fully decentralized
    - No single entity controls the blockchain network
    - Bitcoin and Ethereum blockchains are both considered open public ledgers

Question

- You decide to use blockchain to track product development and movement along a supply chain
- You want to be able to review any part of the history of a particular product as it is being developed by various partners and vendors
- What can you implement to facilitate ad-hoc queries of the product’s progress?
- **A local database that is updated by each successive transaction in the blockchain**

Non-Cryptographic Data Protection

- Steganography
- Tokenization
- Data Masking

Steganography

- The practice of concealing messages or information within other non-secret data (text, image, audio or video clip, network packet, etc.)
- The non-secret data could be publicly available
    - E.g., an image on a website
    - Concealed message is “hiding in plain sight”
- The concealed message can optionally be encrypted

Steganography Example

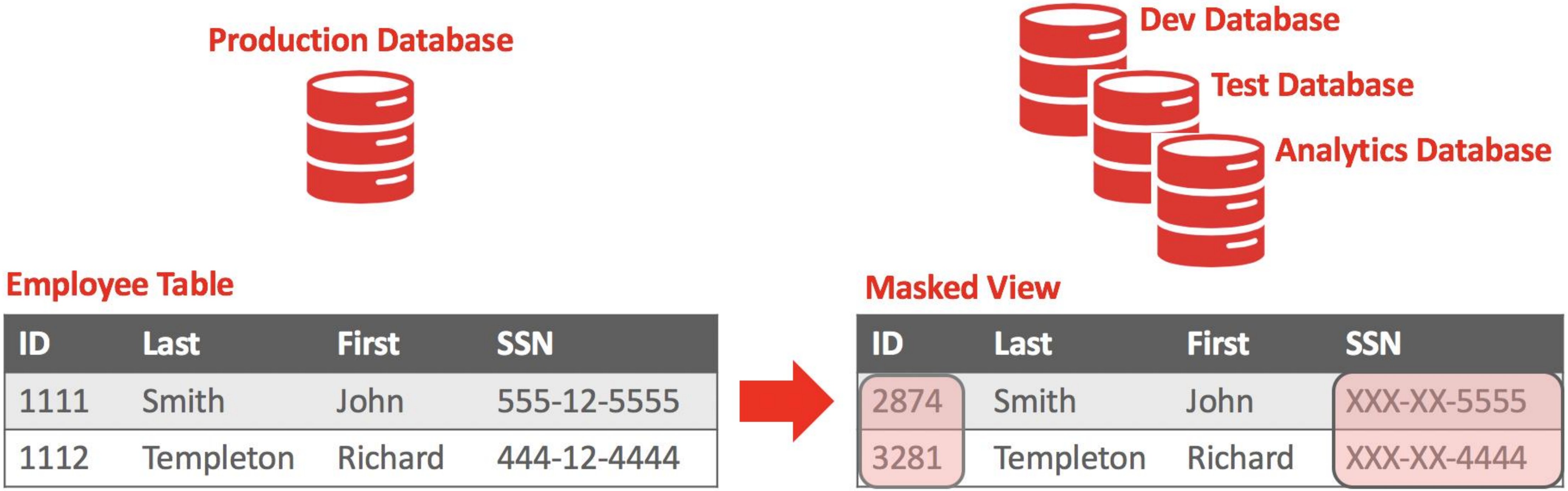
No Message

Attack at midnight

Data Masking

- AKA data obfuscation, anonymization, or tokenization
- A way to create a fake but realistic version of sensitive data
- The goal is to create a version that cannot be deciphered or reverse engineered
- Ways to alter data include:
  - character shuffling
  - word or character substitution
  - encryption
- Masked data can be used for software development, testing, or data analytics

Data Masking Example



Production Database

Dev Database

Test Database

Analytics Database

Employee Table

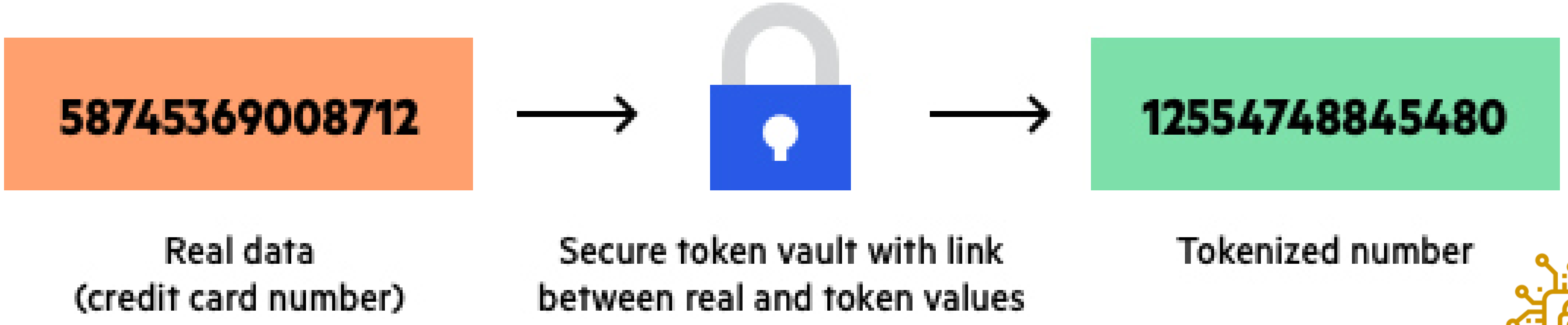
ID	Last	First	SSN
1111	Smith	John	555-12-5555
1112	Templeton	Richard	444-12-4444

Masked View

ID	Last	First	SSN
2874	Smith	John	XXX-XX-5555
3281	Templeton	Richard	XXX-XX-4444

Tokenization

- The process of substituting a sensitive data element with a non-sensitive equivalent (token)
- The token has no intrinsic or exploitable meaning or value
- The token is a reference (i.e. identifier) that maps back to the sensitive data through a tokenization system



Question

- The database administration team is requesting guidance for a secure solution that will ensure confidentiality of cardholder data at rest only in certain fields in the database schema
- The requirement is to substitute a sensitive data field with a non-sensitive field that is rendered useless if a data breach occurs
- What would best meet the requirement?
- **Tokenization**

Question #2

- What describes the process of concealing code or text inside a graphical image?
- **Steganography**

Question #3

- You want to protect credit card information that is stored in a database from being exposed and reused.
- However, your point-of-sale system does not support encryption.
- What else can you use to protect the data?
- **Tokenization**