VCT-GEN-IYPT-P Cryptographie

Basics of cryptography in corporate IT

Agenda

1. Fundamentals of Cryptography in Business Systems

A understanding of objectives, terms and usage of cryptographic systems

2. Fundamentals of Assessing Cryptographic Systems

Methods and tools to assess cryptographic systems

3. Fundamentals of Hardening Cryptographic Setups

Approaches to protect cryptographic systems against flaws of use and implementation

Training Goals - Knowledge

- You know the protection goals of the use of cryptographic measures.
- You know the basic challenges in the use of cryptography
- You know typical attack vectors against cryptographic systems
- You know the application purposes of representatives of cryptographic tools
- You know typical notations of cryptographic protocols (cipher suites).

Training Goals - Skill

- You will be able to evaluate the protective effect of cryptographic procedures in typical EDP application scenarios.
- You can determine the central cryptographic parameters in a simple cryptographic system.
- You can evaluate whether configurations of cryptographic systems are adequately selected for simple application scenarios.

Training Goals - Behavior

- You question the achievement of protection goals in the use and design of cryptographic systems

PT-P-GEN-46

Fundamentals of Cryptography in Business Systems

Course Goals - Knowledge

- You know the protection goals of the use of cryptographic measures.
- You know the basic challenges in the use of cryptography

Course Goals - Skill

- You can determine the central cryptographic parameters in a simple cryptographic system.

Fundamentals of Cryptography in Business Systems

1. Cryptographic Security

Security objectives and mathematical concepts

2. Ciphers

Different ways to implement ciphers

3. Algorithms

Implementation algorithms and their use cases

4. Public Key Infrastructure

PKI as a common approach of key distribution

5. Conclusion

Cryptography Goals

"Cryptography provides confidentiality, integrity, authentication, and nonrepudiation for sensitive information while it is stored (at rest), traveling across a network (in transit), and existing in memory (in use)"

Cryptography Goals

The different algorithms that enforce the four main goals of cryptography

Confidentiality

Symmetric and asymmetric cryptosystems

Integrity

Digital signatures, symmetric and asymmetric cryptosystems

Authentication

Symmetric and asymmetric cryptosystems

Nonrepudiation

Asymmetric cryptosystems

Traits of evil

Terminologies and Concepts

Plaintext Message (P)

A message that has not yet been encrypted with any cryptographic algorithm

Ciphertext Message (C)

Populated from an encrypted plaintext message

Key

Used to maintain the security for all cryptographic algorithms

Key Space (bit size)

The range of values that are applicable and valid for use as a key for a specific algorithm

Private Key (Secret Key)

All participants use a single shared key

Public Key Cryptosystems

Participants have their own pair of keys

Cryptographic Security

Objectives

- Applied cryptography
- Mathematical concepts in cryptography
- Achieving Security
- Cryptographic attacks

Applied Cryptography

Web Applications

SSL, TLS

Portable Devices

Disk encryption (MAC, Windows, Linux)

Email

PGP, S/MIME

Digital Rights Management

Copyright restrictions (movies, music, e-books)

Networks

IPsec, ISAKMP, Wireless Networks (WEP, WPA)

Watermarking

Intellectual property, embedded secret messages

Mathematical Concepts

- **Split Knowledge**

  Separation of duties and two-person Control (Example: Key Escrow -> *M of N Control*)

- **Zero-Knowledge Proof**

  Proving knowledge of a fact without revealing the fat. (Done with passwords and secret authenticators)

- **Nonce**

  Random number acting as a placeholder variable in mathematical functions; must be unique (Example: Initialization Vector)

- **Work Function**

  Measures the strength of cryptosystem by measuring the effect in terms of cost and/or time

- **One-way Functions**

  Mathematical operation that produces output values for each combination of inputs and makes it impossible to retrieve the input values (e.g. Public Cryptosystems)

Achieving Security

- **Key Length**

  The longer a key the stronger is the application of a cryptographic system with cost of performance

- **Key Generation**

  Generating a key can have process flaws which can compromise key confidentiality

- **Algorithm**

  The crypto-algorithm as such can be broken or have limitations towards attributes of the plaintext

- **Distributing Keys**

  Flaws during distributing keys can compromise key confidentiality and authenticity

- **Protecting Keys**

  Mistakes on protecting key material can compromise all objectives of the crypto-system

Cryptographic Attacks

- **Statistical Attack**

  Finds vulnerabilities in the hardware or OS hosting the cryptography application

- **Brute Force**

  Methodically guesses the key using high processing power (enhancement: Rainbow tables)

- **Analytic Attack**

  The logic of the algorithm is attacked to reduce its complexity

- **Meet-in-the-Middle**

  Affects block encryption ciphers (double encryption)

- **Implementation Attack**

  Exploits the software code that is used to program the encryption system

Cryptographic Attacks – Cont.

- **Known Plaintext**

  Both the encrypted message and the plaintext message are available to generate the ciphertext

- **Chosen Ciphertext**

  Decrypting portions of the ciphertext message and using it to discover the key

- **Chosen Plaintext**

  Encrypting the plaintext message and analysing the ciphertext output of the encryption algorithm

- **Frequency Analysis**

  Breaks most substitution ciphers by studying the frequency of letters or groups in a ciphertext to replace them with common letters

Ciphers

Objectives

- Substitution ciphers
- Block ciphers
- Stream ciphers

Substitution Cipher

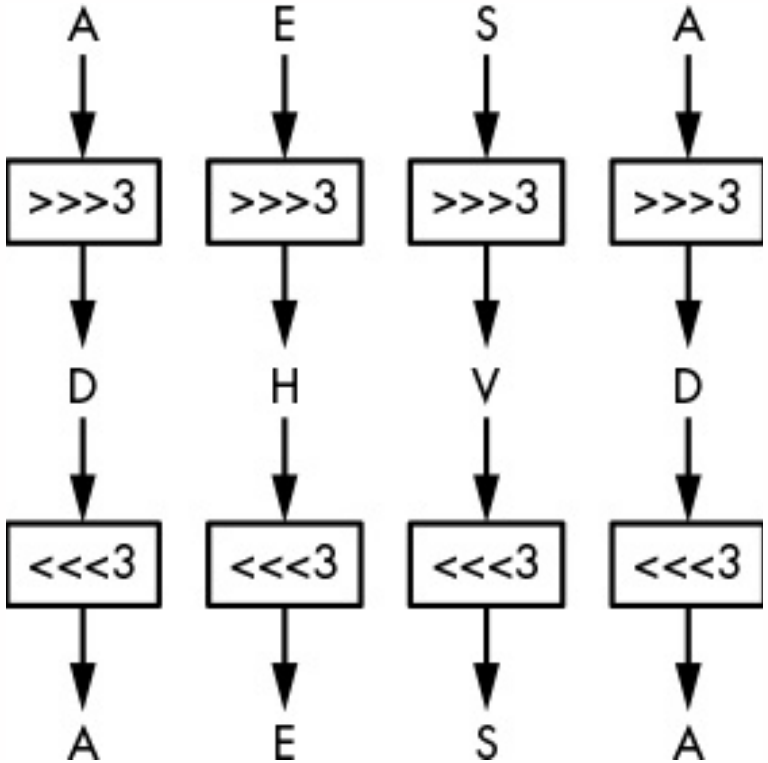
Replaces each character or bit of the plaintext messages with a different character

Examples:

- Caesar cipher
  - Breakable with frequency analysis
- Polyalphabetic substitution ciphers (e.g. Vigenère cipher)
  - Breakable with second-form of frequency analysis called period analysis
- One-Time pad (Vernam ciphers)
  - Powerful
  - Unbreakable if used properly

Example: Ceasar Cipher

https://mlhale.github.io/nebraska-gencyber-modules/
ciphers_and_cybersecurity_in_other_fields/README/ licensed under CC BY-SA-NC



- The key describes the substitution by the distance to the cipher text representation
- Easy to break when parts of plaintext are guessable
- Easy to break with chosen plaintext attack
- In ancient times the key was literally a ring
- (Most) Nowadays ciphers also do transposition

Block Ciphers

Operate on chunks, or blocks, of a message and apply the encryption to entire message block at the same time

- Symmetric encryption algorithm
- Slower processing
- Requires more resources
- Different modes of operation
- Commonly used

Examples:

- AES, DES, Blowfish, RC5, 3DES

Stream Ciphers

Operate on one character or bit of a message at a time

- Symmetric encryption algorithm
- Faster processing
- Requires fewer resources
- Can function as a block cipher

Examples:

- RC4 (wireless networks), A5 (GSM)
- SSL/TLS connections, Bluetooth connections, cellular and 4G connections

Algorithms

Objectives

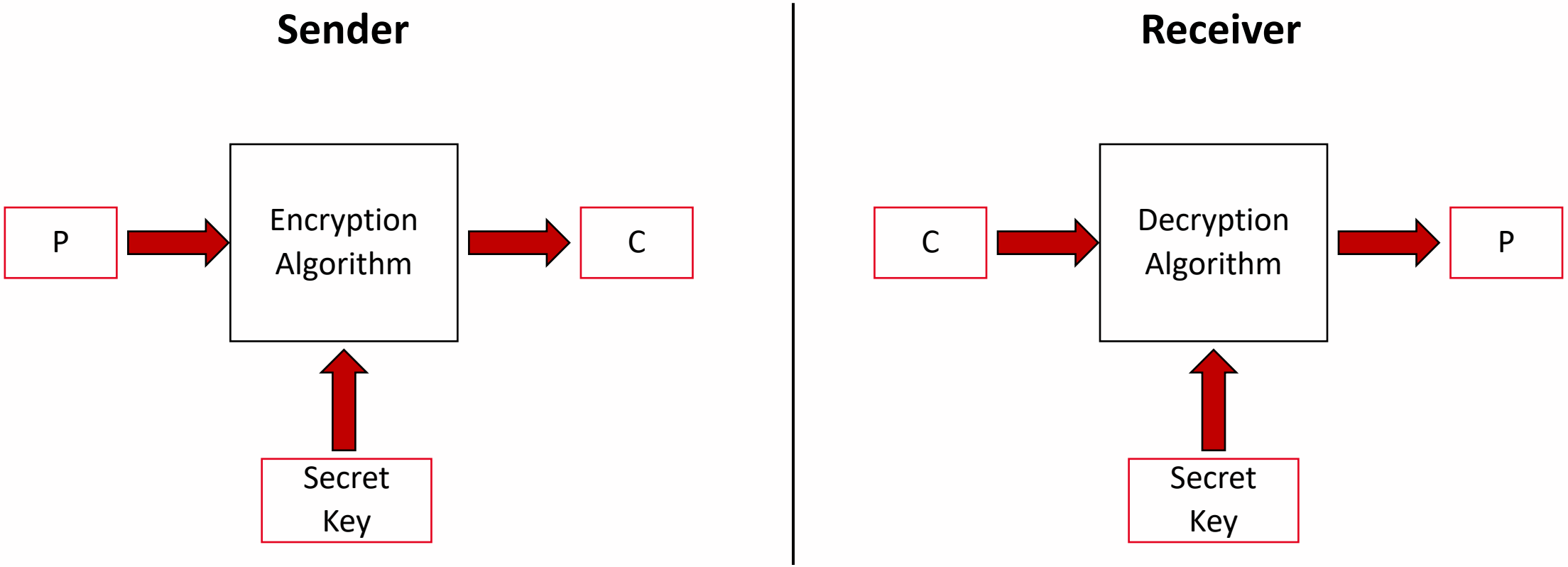
- Symmetric encryption algorithms
- Asymmetric encryption algorithms
- Hashing algorithms

Symmetric Encryption Algorithm

- Relies on a single shared key for encryption and decryption
- Also called secret or private key cryptography
- Fast
- Not scalable
- Key distribution is a major problem
- Achieves confidentiality
- Usage:
  - Data at rest
  - Encryption of large amounts of data (i.e. payment applications)

https://www.maxpixel.net/static/photo/2x/Ring-Hobbit-Magic-Lord-Of-The-Rings-Dragon-4612457.jpg licensed under CC BY-SA-NC

Symmetric Encryption Algorithm



$$Number\ of\ Keys = n(n-1)/2$$

n -> number of users to connect

Symmetric Encryption Algorithm

Name	Block size	Key Size
Advanced Encryption Standard (AES)	128	128, 192, 256
Rijndael	Variable	128, 192, 256
Blowfish	64	32 – 448
Data Encryption Standard (DES)	64	56
International Data Encryption Algorithm (IDEA)	64	128
Rivest Cipher 2 (RC2)	64	128
Rivest Cipher 5 (RC5)	32, 64, 128	0 – 2040
Skipjack	64	80
Triple DES (3DES)	64	112 pr 168
Twofish	128	1 - 256

Modes of Operation

- **Electronic Code Book (ECB) mode**

  Used for small amount of data, no initialization vector (IV)

- **Cipher Block Chaining (CBC) mode**

  Implements IV, if one block is corrupted it becomes harder to decrypt

- **Cipher Feedback (CFB) mode**

  Operates against data produced in real time, uses stream ciphers

- **Output Feedback (OFB) mode**

  Implements IV, no transmission errors, operates like CFB, uses stream cipher

- **Counter (CTR) mode**

  Uses stream ciphers, used in parallel computing

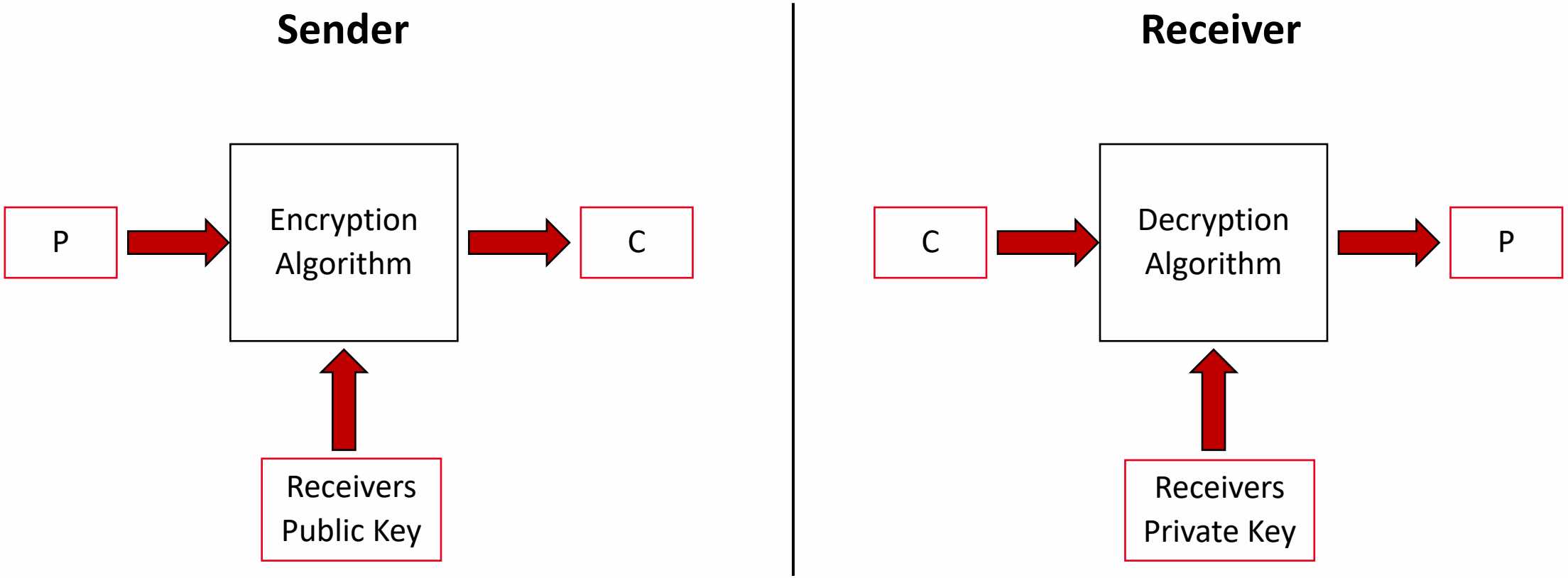
Asymmetric Encryption Algorithm

- Two keys are distributed to users:
  - Public key – shared with all users
  - Private key – kept secret, and only known to the user
- Known as public key algorithm
- Slow speed of operation
- Keys are easily removed
- Key distribution is a simple
- Provides integrity, authentication, and nonrepudiation
- No pre-existing communication link needs to exist
- Achieves confidentiality, integrity, authenticity, and nonrepudiation

Examples:

- RSA (Rivest-Shamir-Adleman)
- Merkle-Hellman Knapsack
- EL Gamal
- Diffie-Hellman
- Elliptic Curve

Asymmetric Encryption Algorithm



Hashing Algorithms

- Message digest (MD) is a summary of the messages content produced by a hashing algorithm
- Provides integrity
- Also known as: Hash, Hash value, Hash total, CRC, Fingerprint, Checksum, Digital ID)
- MD is 128 bits or longer (the longer the MD, the more reliable its verification of integrity)
- **Applicability:**
  - Makes sure that a received message was not altered
  - Implements a digital signature algorithm

Basic Requirements – Hash Functions

- Input can be of any length
- Output has a fixed length
- Easy to compute for any input
- One-way (hard to determine the input when you have the output)
- Collision free

Hashing Algorithms

Name	Hash Value Length	Comments
Secure Hash Algorithm (SHA1)	160 bit message digest (MD)	Weak
SHA2-256 / SHA3-256	256 bit MD using a 512 bit block size	SHA2 have weaknesses
SHA2-224 / SHA3-224	224 bit MD using a 512 bit block size	SHA3 uses a more secure algorithm
SHA2-512 / SHA3-512	512 bit MD using a 1024 bit block size	
SHA2-384 / SHA3-384	384 bit MD using a 1024 bit block size	
MD2	128 bit	• Attacks exist • Collisions can be found (not accepted) • not a one-way function
MD4	128 bit	• Increase level of security • Collisions can be found (not accepted)
MD5	128 bit	• Additional security • Reduces speed of MD • Collisions can be found (not accepted)
Hash Message Authenticating Code (HMAC)	Variable	• Doesn't provide nonrepudiation • Used best with symmetric key algorithms
Hash of Variable Length (HAVAL)	128, 160, 192, 224, and 256 bits	• Collisions can be found (128 bits)

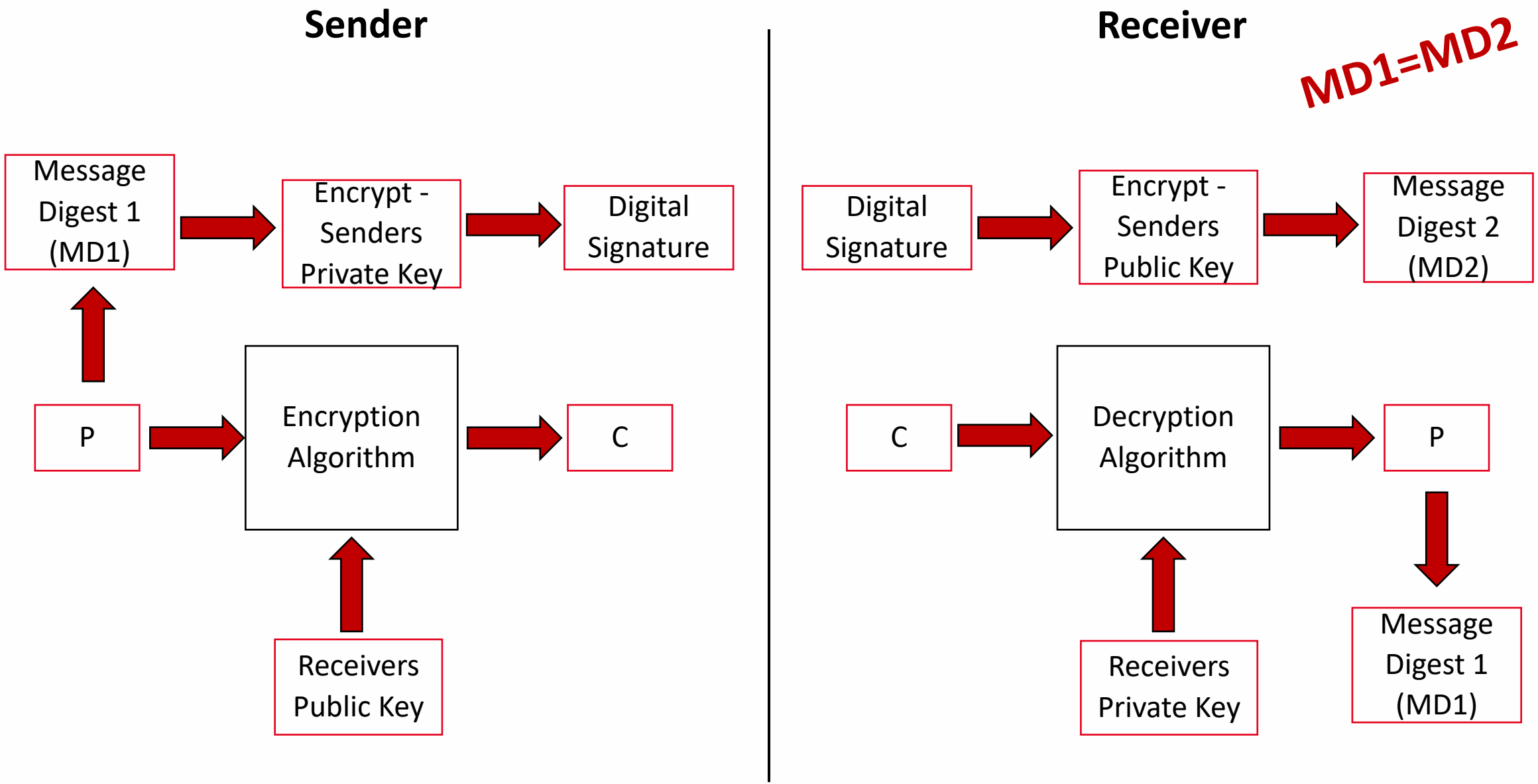
Digital Signatures

- Enforces nonrepudiation
- Protects against malicious modification or unintentional modification (validates integrity)
- Relies on public key cryptography and hashing functions

Digital Signature Standard (DSS):

- Approved Digital Signature Algorithm must use SHA-3
- Algorithms that support digital signature infrastructure:
  - DSA (Digital Signature Algorithm)
  - RSA
  - ECDSA (Elliptic Curve DSA)

Digital Signatures



Public Key Infrastructure

Objectives

- Digital Certificates
- Certificate authority
- Certificate life-cycle

Digital Certificates

*Gives assurance that the people your communicating with truly are who they claim to be**

Digital Certificate standards:

- Conformity X.509 version 3
- Serial number (certificate creator)
- Signature algorithm identifier
- Issuer name
- Validity period
- Subject’s name
- Subject’s public key

Demonstration

Certificate Authorities

- Issues digital certificates
- The glue that binds PKI together
- CAs validate user identities
- Registration Authorities (RAs) assists CAs in verifying users identities

Examples:

- DigiCert
- Sectigo
- GoDaddy
- IdenTrust
- Let’s Encrypt

Certificate Enrolment

Obtaining a Digital Certificate

Prove identity

Verify with CA your identity

Public Key

Provide CA with your public key

CA

Creates X.509 certificate with your information, digitally signs using CAs private Key and provides you with a copy

Distribute

You can start distributing your certificate

Certificate Verification

Obtaining a Digital Certificate

Check Digital Signature

Check the CAs digital signature using the CAs public key

Revocation List

Check if the certificate hasn’t been revoked using the Certificate Revocation List (CRL) or the Online Certificate Status Protocol (OCSP)

Information Check

Make sure information about someone is available in certificate

Certificate Revocation

- Certificate was compromised (obtaining the private key)
- Certificate was not properly verified by CA
- Details of certificate changes
- Security association changed

Conclusion

IY-P-GEN-56

Fundamentals of Assessing Cryptographic Systems

Course Goals - Knowledge

- You know typical attack vectors against cryptographic systems
- You know typical notations of cryptographic protocols (cipher suites).

Course Goals - Skill

- You will be able to evaluate the protective effect of cryptographic procedures in typical EDP application scenarios.

Fundamentals of Assessing Cryptographic Systems

1. Protocol exchange assessment

Assessing the exchange and negotiation of cryptographic protocols

2. Cipher analysis

Using CyberChef for statistical analysis

3. Steganography vs. Encryption

The differences between hiding and encrypting

4. Standards for applied cryptography

How to assess if a certain cryptographic system use appropriate components and tools

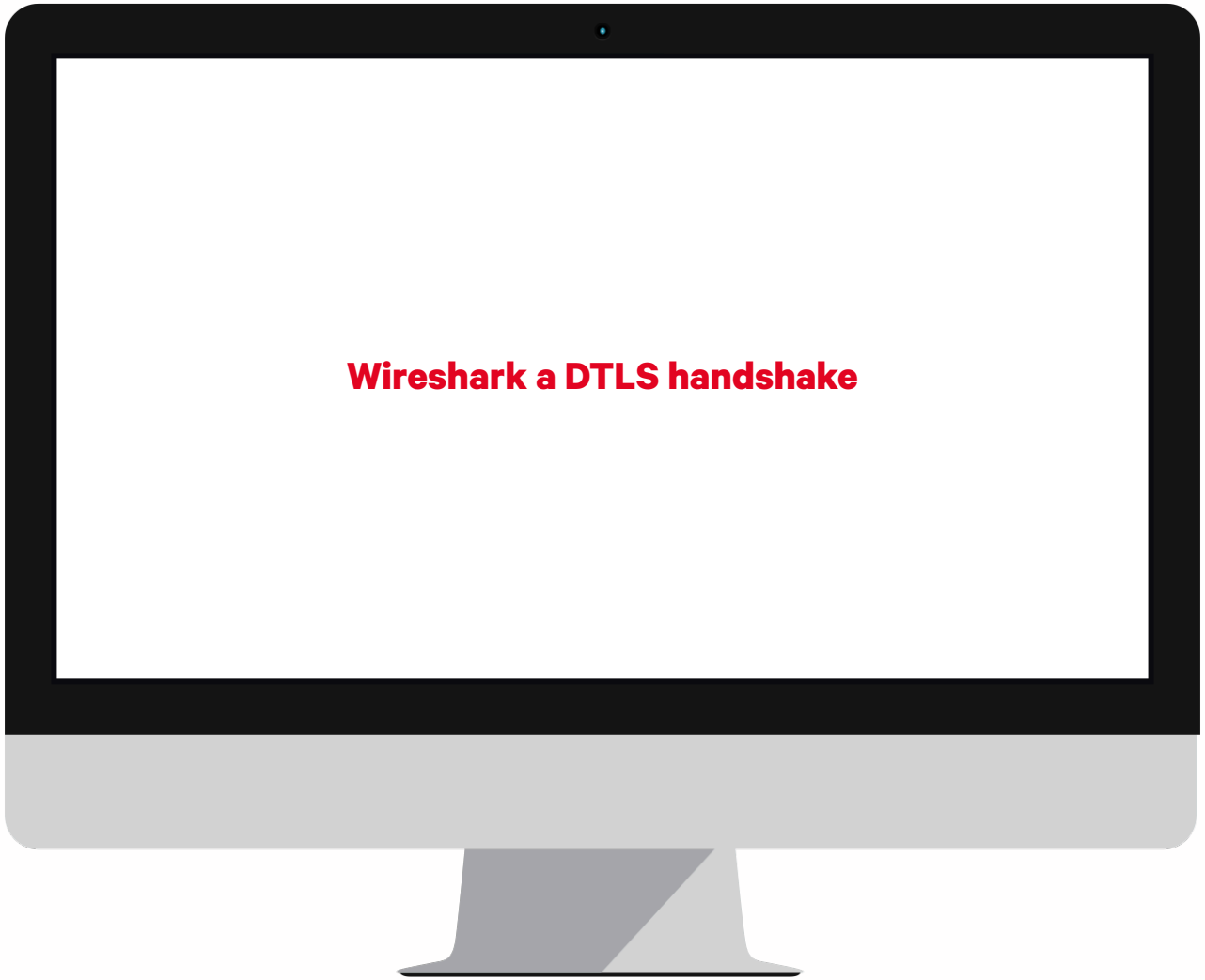
5. Implementation Review

Reviewing bad cryptographic implementation examples

6. Conclusion

Protocol exchange assessment

Assessing crypto network traffic



TLS 1.3 Handshake

To use cipher suites, the client and the server must agree on the specific cipher suite that is going to be used in exchanging messages. Both the client and the server must support the agreed upon cipher suite.

Client Hello

- List of prefered cipher suites
- key share

Server Hello

- List of prefered cipher suites
- key share
- certification

Finished

Confirmation of completion

Data Exchange

Encrypted communication

TLS Cipher Suites

Cipher suites decide how secure, compatible and fast your HTTPS website is. A cipher suite is a set of information that helps determine how your web server will communicate secure data over HTTPS.

ECDHE-RSA-AES256-GCM-SHA256

ECDHE

Protocol for key exchange

RSA

Certification Key to authenticate the service

AES256-GCM

Transport cipher for bulk encryption with shared key

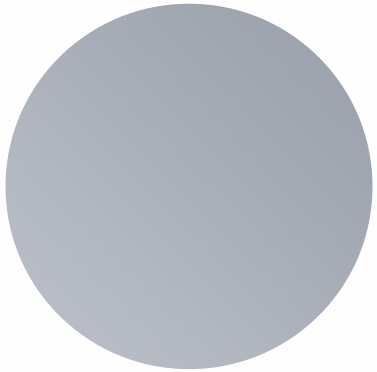
SHA256

Algorithm for the message authentication check

Assessing public TLS service

Testing internal TLS services

https://github.com/nabla-c0d3/sslyze



Alban Diquet

SSL/TLS scanning library.

Cipher analysis

Challenges decyperhing secrets

Is Encrypted?

Is it a cipher text or steagonography?

Find Cipher

The cipher needs to be determined to defeat it

Understand encoding

You need to have proper assumptions about the plaintext's encording

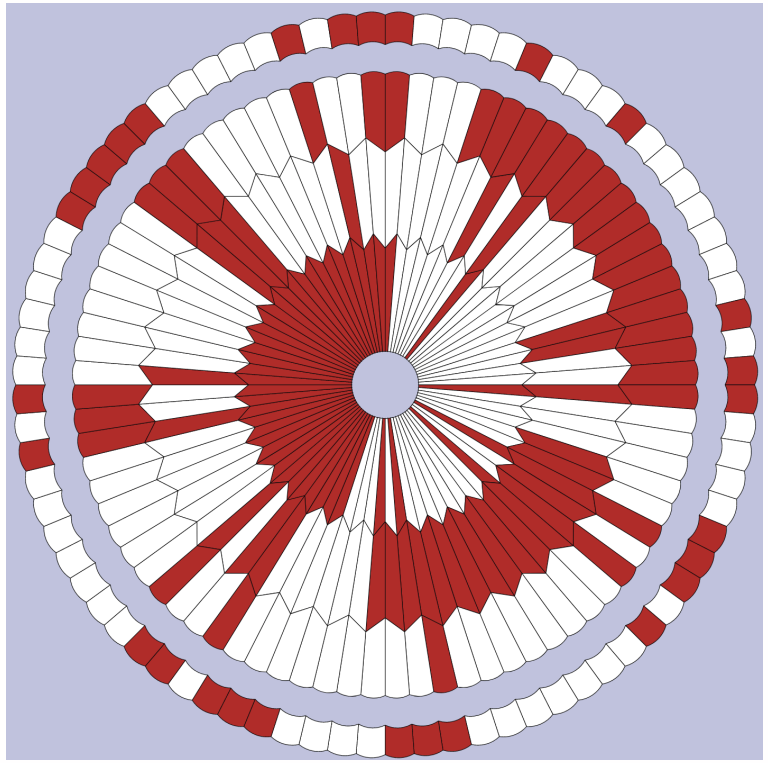
Unterstand language

For statistical analysis

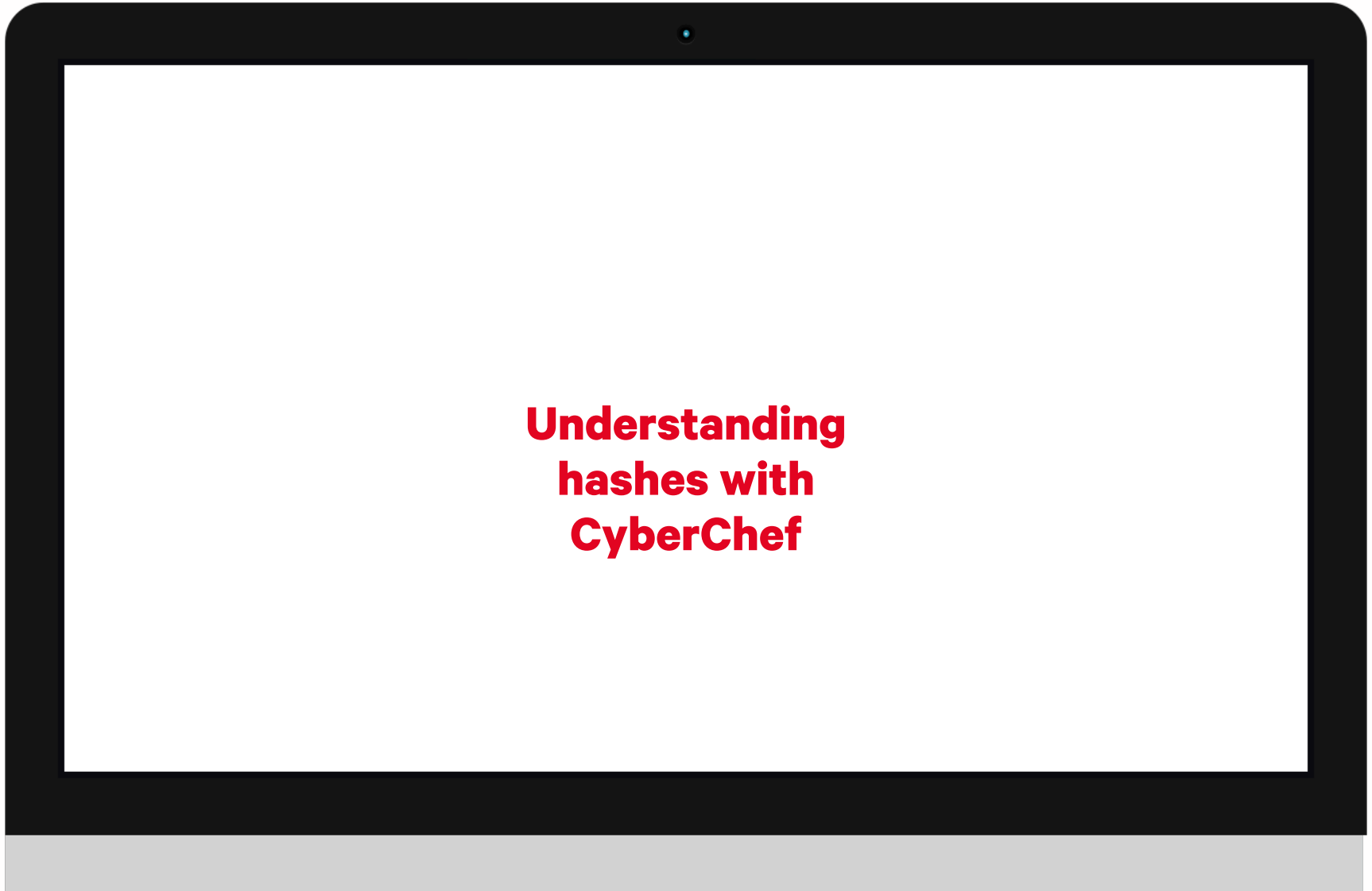
DARE MIGHTY THINGS

- Decoding of the mars rover parachute message

https://projects.noahliebman.net/encodemightythings/



Hash analysis



Steganography vs. Encryption

Arts of hiding secrets

Encryption

- Hinding secrets by creating cipher text from plain text
- Always uses a key
- Independent from encoding
- Secret are available to those who know key and cipher

Steagonography

- Hiding secret in another message
- Does not use a key
- Utilize layered encodings
- Secret are available to those who know how to decode them

Standards for applied cryptography

BSI Cryptographic Guidelines

Ciphers

BSI TR-02102-1

assessment of the security of selected cryptographic mechanisms. However, no claim to completeness

BSI TR-02102-2

recommendations for the use of the cryptographic protocol TLS.

TLS

Ipsec & IKE

BSI TR-02102-3

recommendations for the use of cryptographic mechanisms in the IPsec and IKE protocols

BSI TR-02102-4

recommendations for the use of the cryptographic protocol Secure Shell (SSH)

SSH

Common Crypto Implementation Flaws

- **Hard-coded keys**

  Hard-coding keys implies that whoever has access to the software knows the keys to decrypt the data

- **Choosing insufficient cryptographic parameters**

  Insufficient selection of parameter and use of weak defaults of cryptographic libraries

- **Wrong mode of operation**

  If block sizes and mode are not sufficient the cipher message leaks information of the plaintext

- **Passwords are not cryptographic keys**

  Passwords are things people remember and can be arbitrary length. Keys on the other hand are not limited to printable characters and have fixed length.

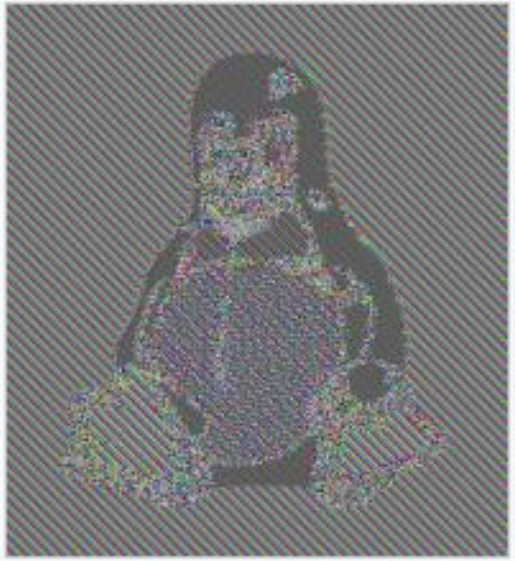
- **Assuming encryption provides message integrity**

  Encryption hides data, but an attacker might be able to modify the encrypted data which might be accepted by software

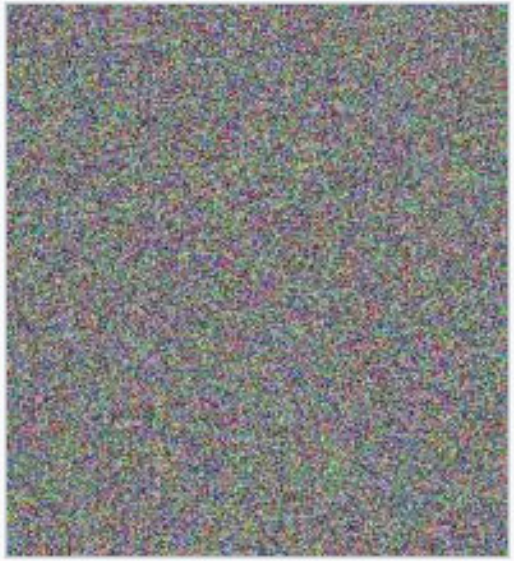
Implementation Review

Wrong mode of operation

Original image



Encrypted using ECB mode



Modes other than ECB result in pseudo-randomness

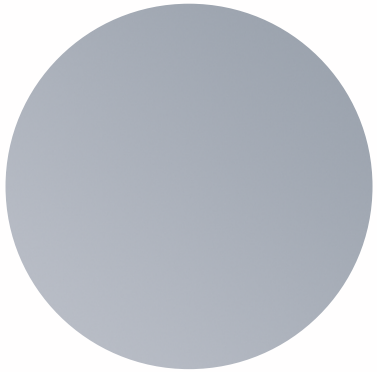
The image on the right is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as "random-looking".

Insufficient parameters

```
public class Encryptor {
        public static String encrypt(String key, String initVector, String value) {
        try {
                IvParameterSpec iv = new IvParameterSpec(initVector.getBytes("UTF-8"));
                SecretKeySpec skeySpec = new SecretKeySpec(key.getBytes("UTF-8"), "AES");

                Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
                cipher.init(Cipher.ENCRYPT_MODE, skeySpec, iv);

                byte[] encrypted = cipher.doFinal(value.getBytes());
                System.out.println("encrypted string: "+ Base64.encodeBase64String(encrypted));

                return Base64.encodeBase64String(encrypted);
        } catch (Exception ex) {
                ex.printStackTrace();
        }
        return null;
        }
public static void main(String[] args) {
        String key = "Bar12345Bar12345"; // 128 bit key
        String initVector = "RandomInitVector"; // 16 bytes IV
        System.out.println(decrypt(key, initVector,
        encrypt(key, initVector, "Hello World")));
}
```

Good crypto implementation examples

https://github.com/luke-park/SecureCompatibleEncryptionExamples



Luke Park

Secure Compatible Encryption Examples

Conclusion

To take away

- **Know the usage scenario**

  Except of violation against baselines – the use of crytopgraphic systems is only good when appropriate for the use case
- **Know the traits**

  Different cryptograpic modes have different traits. Typically message integrity is not given by encryption
- **Watch out for bad examples**

  There are plenty of bad examples out there. Stackoverflow votes don't make them reputable
- **Don't do security by obscurity**

  Never think you are the smartest person. There are many graves out there to put his on.
- **Assume change**

  In crypto what's fine today can be broken tomorrow

PT-P-GEN-57

Fundamentals of Hardening Cryptographic Setups

Course Goals - Knowledge

- You know the application purposes of representatives of cryptographic tools

Course Goals - Skill

- You can evaluate whether configurations of cryptographic systems are adequately selected for simple application scenarios.

Fundamentals of Hardening Cryptographic Setups

1. Compliance to standards

Resources to standard to comply to

2. Organizational protection

Organizational measures to protect and limit damage

3. Technical measures

Technical approaches to protect keys and integrity

4. Trusted platforms

Dependency to hardware and cryptographic software services

5. Conclusion

Compliance to standards

Transport Layer cheat sheet

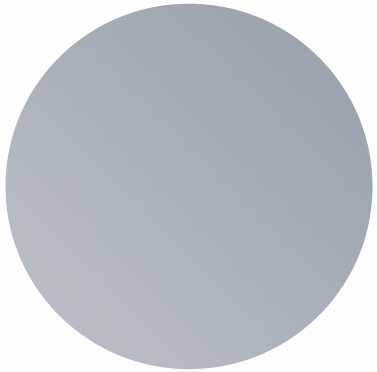
https://cheatsheetseries.owasp.org/cheatsheets/Transport_Layer_Protection_Cheat_Sheet.html

OWASP

Transport Layer Protection Cheat Sheet

Storage cheat sheet

https://cheatsheetseries.owasp.org/cheatsheets/Cryptographic_Storage_Cheat_Sheet.html



OWASP

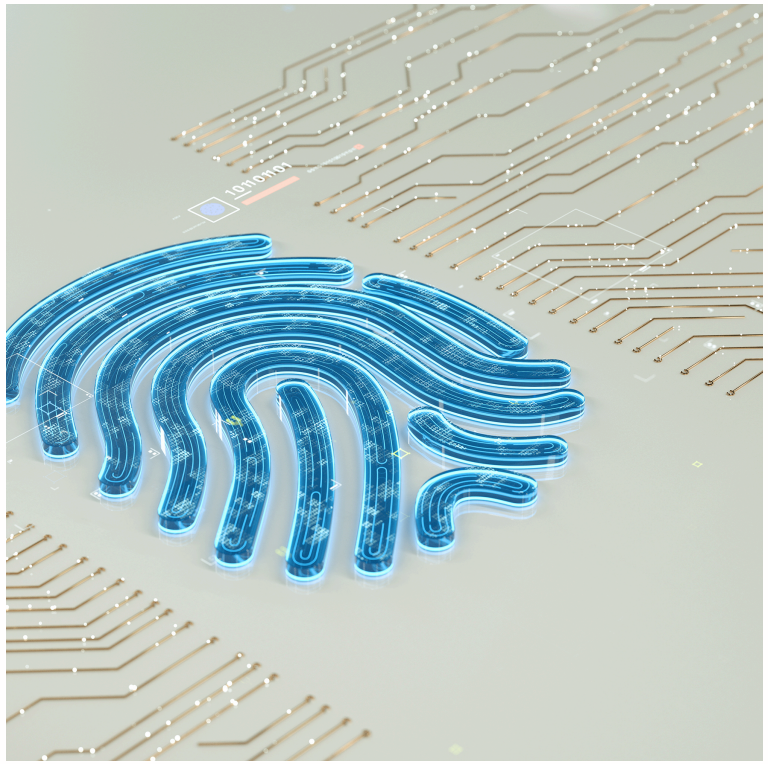
Cryptographic Storage Cheat Sheet

Organizational protection

Limit time of trust

- Certificates should expire as early as possible (TLS certificates are not trustworthy with validity for more then 12 months soon)
- Key's should be rotated and ciphertext reencrypted
- Time limit should depend on trust to the process and processors (e.g. domain vs. org. validation)

Use of trusted software



- Know who is the provider for the compontens of cryptographic systems
- Patchmanagement is crucial for all components of cryptographic systems
- Best is to trust but verify

Key Management Policy

- Have a key management policy and communicate it to all stakeholders
- Define organizational and technical measures over all areas of IT operations (including development)
- Define NoNo's (like hardcoded keys)

Technical measures

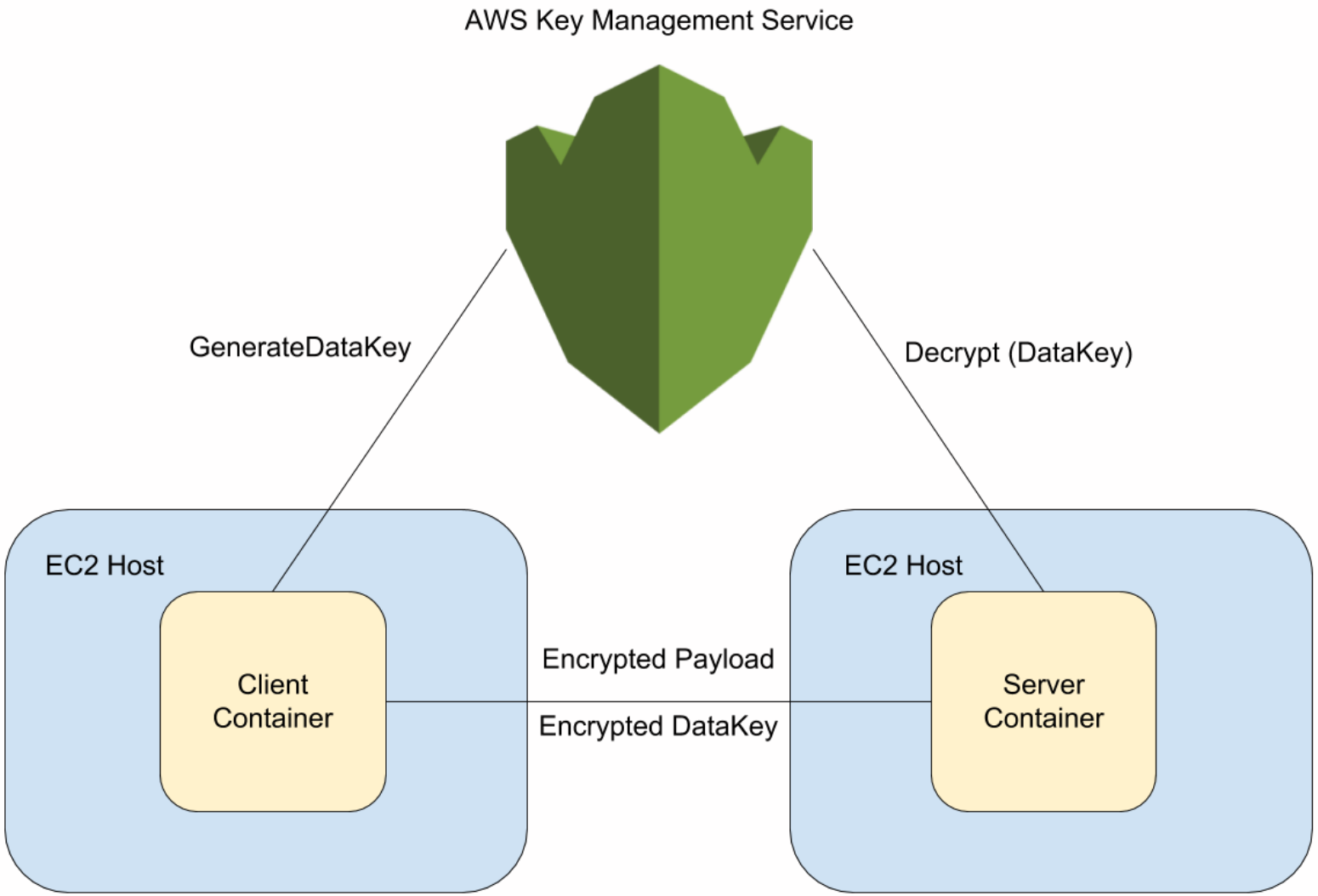
Key Vaults

This Photo by Unknown Author is licensed under CC BY-NC

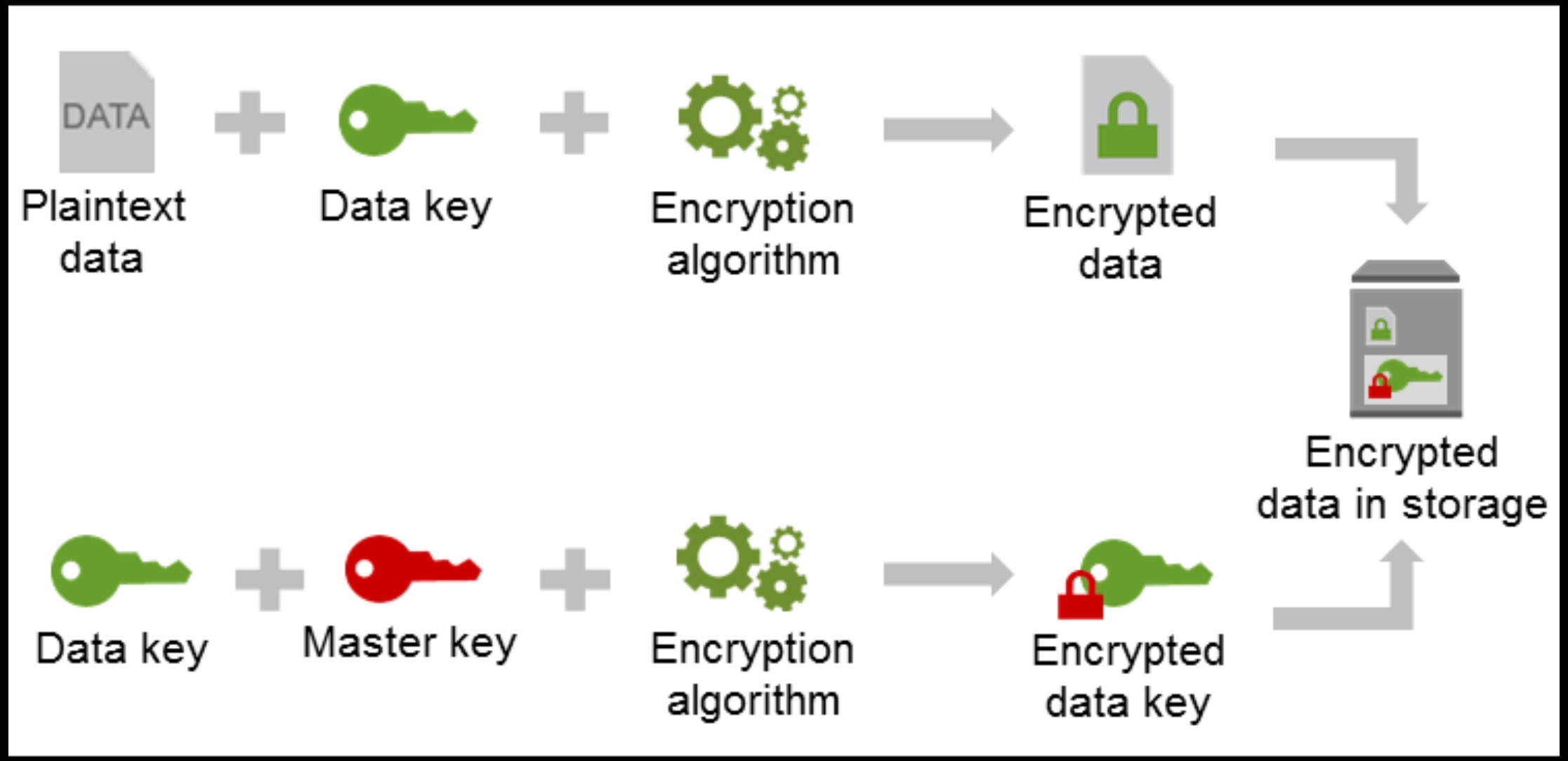
- Keys are typically stored and management in vaults.
- Vaults are nothing but formats that should be only accessed by a certain API and require the use of a secret (PIN/password) to open

PGP Key Vault

Key Management Services



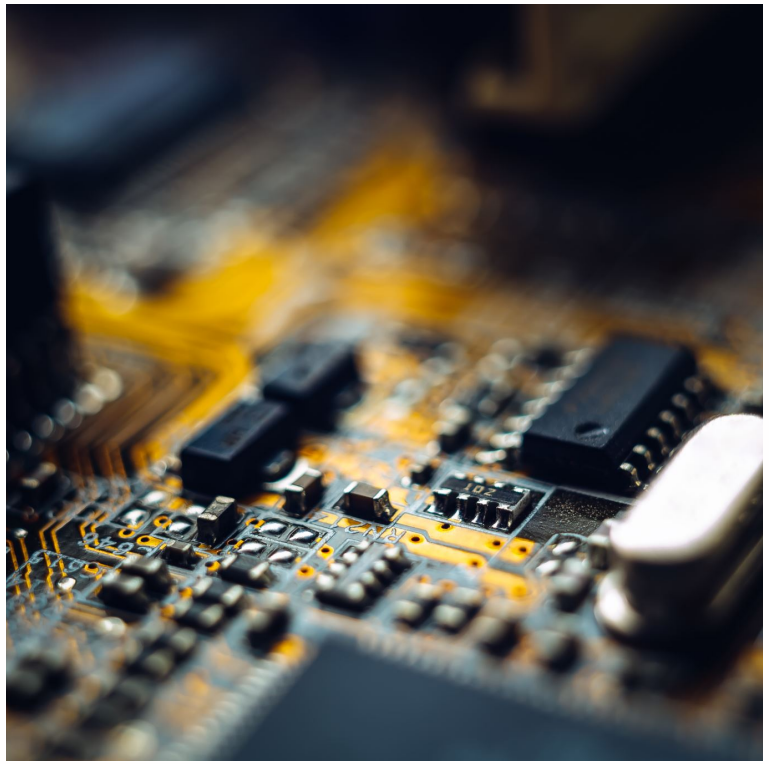
KMS Keys and use



Trusted platforms

Trusted Platform Module

- A hardware security module (processor and persitent memory storage)
- Limited input/output rountines
- Provides better randomness & key generation and hardware-based encryption
- Presists certain platform keys and provides API's to store other application keys in protected memory



Conclusion

To take away

- **Define your protection objectives**

  There are plenty of possibilities to protect cryptographic systems. Define your objectives before you validate what works for you.

- **Communicate key management policies**

  Employees and vendors and service providers should know your policies and being asked to comply to them.

- **Access management is core**

  Restricting access to private keys is essential for the security of cryptographic systems.

- **Cloud computing increases complexity**

  In a shared responsibility model key management can get utterly complex. Include this in the decision for or against a service model.

- **Use encryption when needed**

  Because encryption is a two-edged sword use it when needed and don’t do when not.

Homework

Create your own PKI with OpenSSL

This presentation/representation may contain confidential and/or privileged information and is eligible for demonstration purposes to the recipient only. SAP, ABAP, SAP NetWeaver, SAP S/4HANA are trademarks or registered trademarks of SAP SE or an SAP affiliate company.

NO MONKEY, NO MONKEY ACADEMY, NO MONKEY SECURITY and NO MONKEY ADVISORY are trademarks or registered trademarks of NO MONKEY GmbH or a NO MONKEY affiliate company or subsidiary.

© 2020