Module 20

Cryptography

Learning Objectives

01 Explain Cryptography Concepts and Different Encryption Algorithms

02 Explain Applications of Cryptography

03 Explain Different Cryptanalysis Methods and Cryptography Attacks

04 Explain Cryptography Attack Countermeasures

Objective 01

Explain Cryptography Concepts and Different Encryption Algorithms

Cryptography

Cryptography is the conversion of data into a scrambled code that is encrypted and sent across a private or public network

Cryptography is used to protect confidential data, such as email messages, chat sessions, web transactions, personal data, corporate data, and e-commerce applications

Objectives of Cryptography

- Confidentiality
- Authentication
- Integrity
- Nonrepudiation

Types of Cryptography

Symmetric Encryption

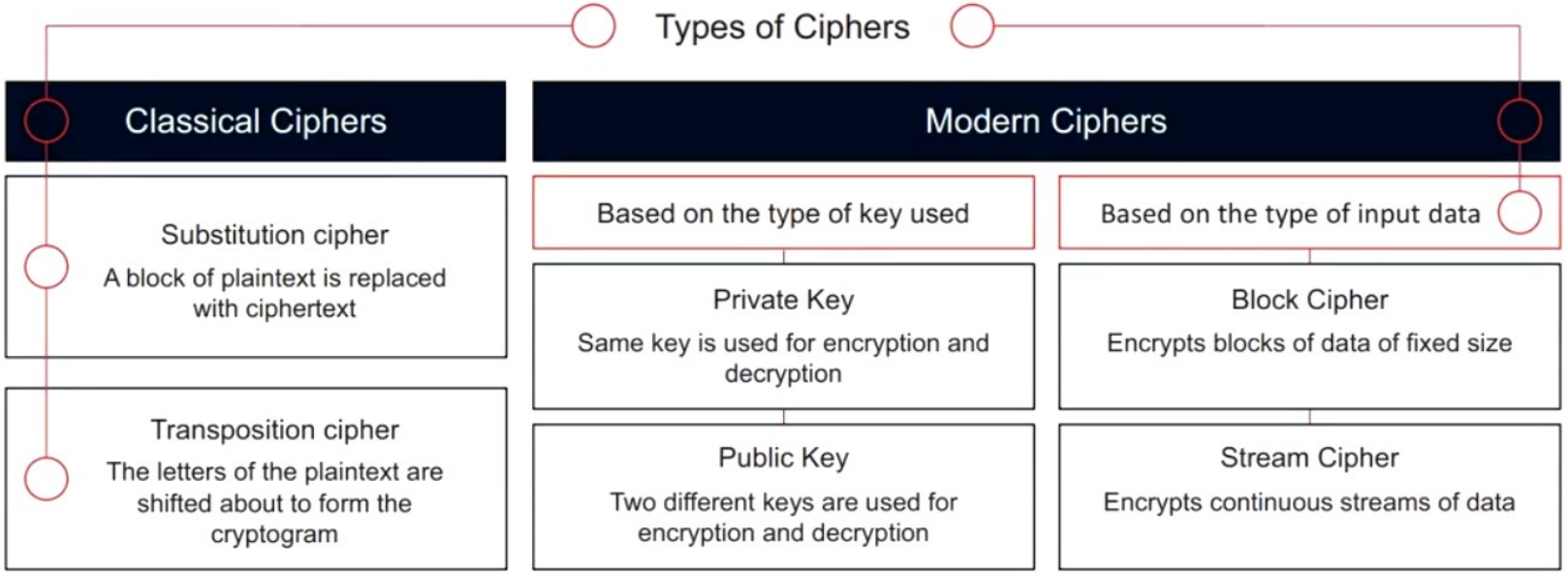
Symmetric encryption (secret-key, shared-key, and private-key) uses the same key for encryption as it does for decryption

Asymmetric Encryption

Asymmetric encryption (public-key) uses different encryption keys, which are called public and private keys for encryption and decryption, respectively

Ciphers

Ciphers are **algorithms** used to encrypt or decrypt the data



Symmetric Encryption Algorithms

Algorithm	Cipher Type	Key Size (bits)	Block Size (bits)	Application Areas
Data Encryption Standard (DES)	Block	56 bits	64 bits	Legacy systems, early encryption standards
Triple DES (3DES)	Block	112, 168 bits	64 bits	Financial services, payment systems
Advanced Encryption Standard (AES)	Block	128, 192, 256 bits	128 bits	Secure communications, storage encryption, government standards
RC4	Stream	40 to 2048 bits (variable)	-	Secure web traffic (HTTPS), Wi-Fi security (WEP/WPA), streaming encryption
RC5	Block	0 to 2040 bits (variable)	32, 64, 128 bits	Cryptographic libraries, secure communication
RC6	Block	128, 192, 256 bits	128 bits	Advanced encryption, AES competition finalist
Blowfish	Block	32 to 448 bits (variable)	64 bits	Replacement for DES, secure storage
Twofish	Block	128, 192, 256 bits	128 bits	File and disk encryption, open-source software
International Data Encryption Algorithm (IDEA)	Block	128 bits	64 bits	Secure email (PGP), data encryption
Threefish	Block	256, 512, 1024 bits	256, 512, 1024 bits	Disk encryption (Skein hash function)
Serpent	Block	128, 192, 256 bits	128 bits	High-security applications, AES competition finalist
Camellia	Block	128, 192, 256 bits	128 bits	Secure communications, Japanese encryption standard
Tiny Encryption Algorithm (TEA)	Block	128 bits	64 bits	Lightweight encryption, embedded systems
CAST-128	Block	40 to 128 bits	64 bits	Various software applications, secure communications
CAST-256	Block	128, 160, 192, 224, 256 bits	128 bits	Advanced encryption, cryptographic libraries
ChaCha20	Stream	256 bits	-	Secure communications, modern encryption protocols
Salsa20	Stream	256 bits	-	Secure communications, cryptographic protocols

Asymmetric Encryption Algorithms

Algorithm	Key Size (bits)	Application Areas
Rivest–Shamir–Adleman (RSA)	Variable	Encryption, digital signatures, key exchange
Digital Signature Algorithm (DSA)	Variable	Digital signatures
Diffie-Hellman	Variable	Key exchange, secure communication
Elliptic Curve Cryptography (ECC)	160-521 bits	Encryption, digital signatures, key exchange
ElGamal	Variable	Encryption, key exchange

Message Digest (One-Way Hash) Functions

Hash functions **calculate a unique fixed-size bit string** representation called a message digest of any arbitrary block of information

If any given bit of the function's input is changed, then every output bit has a **50 percent** chance of changing

It is computationally infeasible to have two files with the **same message digest value**

Note: Message digests are also called one-way hash functions because they cannot be reversed

Message Digest Functions

Algorithm	Output Size (bits)	Internal State Size (bits)	Block Size (bits)	Max Message Size (bits)	Rounds	Operations	Security (bits)	Application Areas
MD2	128	128	128	2^64	18	Permutation, Substitution	128	Legacy applications, checksum validation
MD4	128	128	512	2^64	48	Logical operations (AND, OR, XOR)	64	Obsolete, early cryptographic hash functions
MD5	128	128	512	2^64	64	Logical operations (AND, OR, XOR)	64	File verification, checksum, digital signatures
MD6	224, 256, 384, 512	1024	512	Unlimited	Variable	Logical operations (AND, OR, XOR)	128-256	Cryptographic applications, data integrity
SHA-0	160	160	512	2^64	80	Bitwise logical operations	0	Obsolete, replaced by SHA-1
SHA-1	160	160	512	2^64	80	Bitwise logical operations	80	Legacy systems, software updates, TLS
SHA-2	224, 256, 384, 512	256, 512	512, 1024	2^128	64, 80	Logical operations (AND, OR, XOR)	112-256	Secure applications, digital signatures, SSL
SHA-3	224, 256, 384, 512	1600	1088, 576	Unlimited	Variable	Sponge construction	112-256	Secure applications, next-generation cryptographic functions
RIPEMD-160	160	160	512	2^64	160	Logical operations (AND, OR, XOR)	80	Cryptographic applications, data integrity
WHIRLPOOL	512	512	512	2^256	10	Matrix operations, substitution	256	Secure hashing, cryptographic applications
Tiger	192	192	512	Unlimited	24	Logical operations (AND, OR, XOR)	192	High-speed applications, checksum validation
BLAKE2	256, 512	256, 512	512, 1024	Unlimited	10-14	Logical operations (AND, OR, XOR)	128-256	High-speed hashing, secure applications
BLAKE3	256	256	512	Unlimited	Variable	Logical operations (AND, OR, XOR)	128	High-performance cryptographic applications

Message Digest Functions Calculators

https://www.nirsoft.net

https://www.bullzip.com

MD6 Hash Generator
https://www.browserling.com

All Hash Generator
https://www.browserling.com

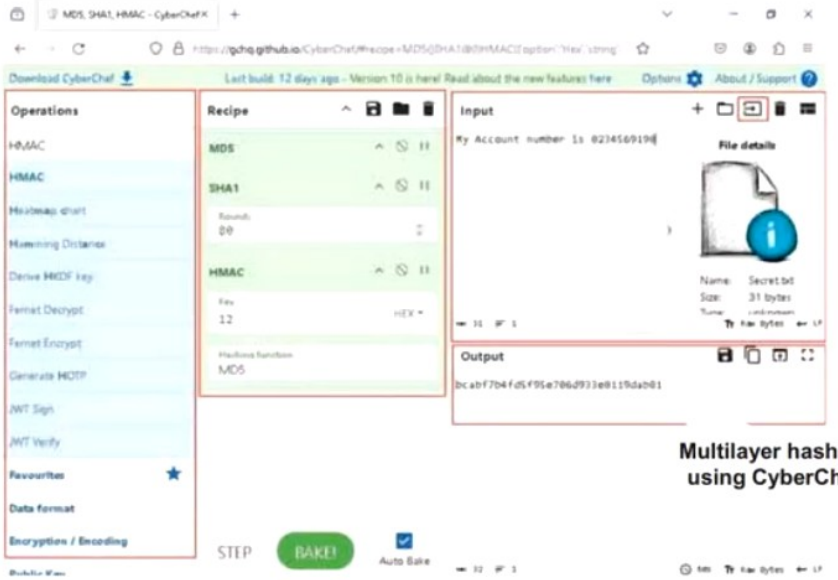
md5 hash calculator
https://onlinehashtools.com

Message Digester
https://www.freeformatter.com

MD6 Hash Generator
https://www.atatus.com

Multilayer Hashing Calculators

- Multilayer hashing, also known as nested hashing or recursive hashing, is a technique where a **hash function** is **applied multiple times** to an input or to the output of a previous hash operation
- You can use tools such as **CyberChef** to perform multilayer hashing



Multilayer hashing using CyberChef

https://gchq.github.io

Hardware-Based Encryption

- Hardware-based encryption **uses computer hardware** for assisting or replacing the software when the data encryption process is underway
- These devices are also capable of **storing encryption keys** and other **sensitive information** in secured areas of RAM or other nonvolatile storage devices

Types of hardware encryption devices

TPM	Trusted platform module (TPM) is a crypto-processor or chip that is present on the motherboard that can securely store the encryption keys, and it can perform many cryptographic operations
HSM	Hardware security module (HSM) is an additional external security device that is used in a system for crypto-processing and can be used for managing, generating, and securely storing cryptographic keys
USB Encryption	USB encryption is an additional feature for USB storage devices that offers onboard encryption services
Hard Drive Encryption	Hard drive encryption is a technology where the data stored in the hardware can be encrypted using a wide range of encryption options

Quantum Cryptography

Quantum cryptography is based on quantum mechanics, such as **quantum key distribution** (QKD)

Data is encrypted by a **sequence of photons** with a spinning trait while travelling from one end to another

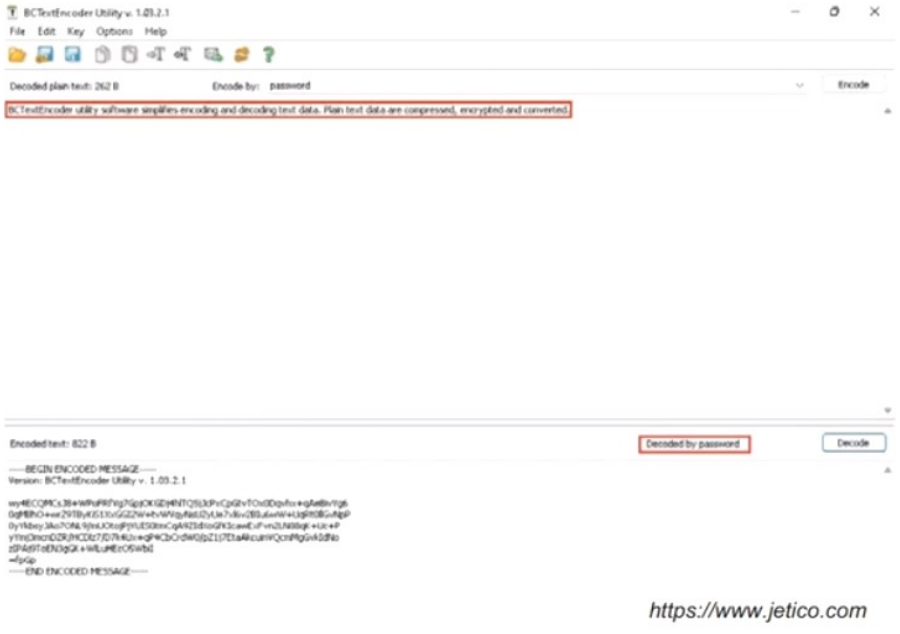
These photons keep changing their shapes during their course through filters: vertical, horizontal, forward slash, and backslash

Attackers can eavesdrop but cannot manipulate the data because the photons are transferred through arbitrary filters

Cryptography Tools

BCTextEncoder

- Encrypts **confidential text** in your **message**
- Uses strong symmetric and public-key algorithms for **data encryption**



https://www.jetico.com

CryptoForge
https://www.cryptoforge.com

AxCrypt
https://www.axcrypt.net

Microsoft Cryptography Tools
https://www.microsoft.com

Concealer
https://www.belightsoft.com

SensiGuard
https://www.sensiguard.com

Objective 02

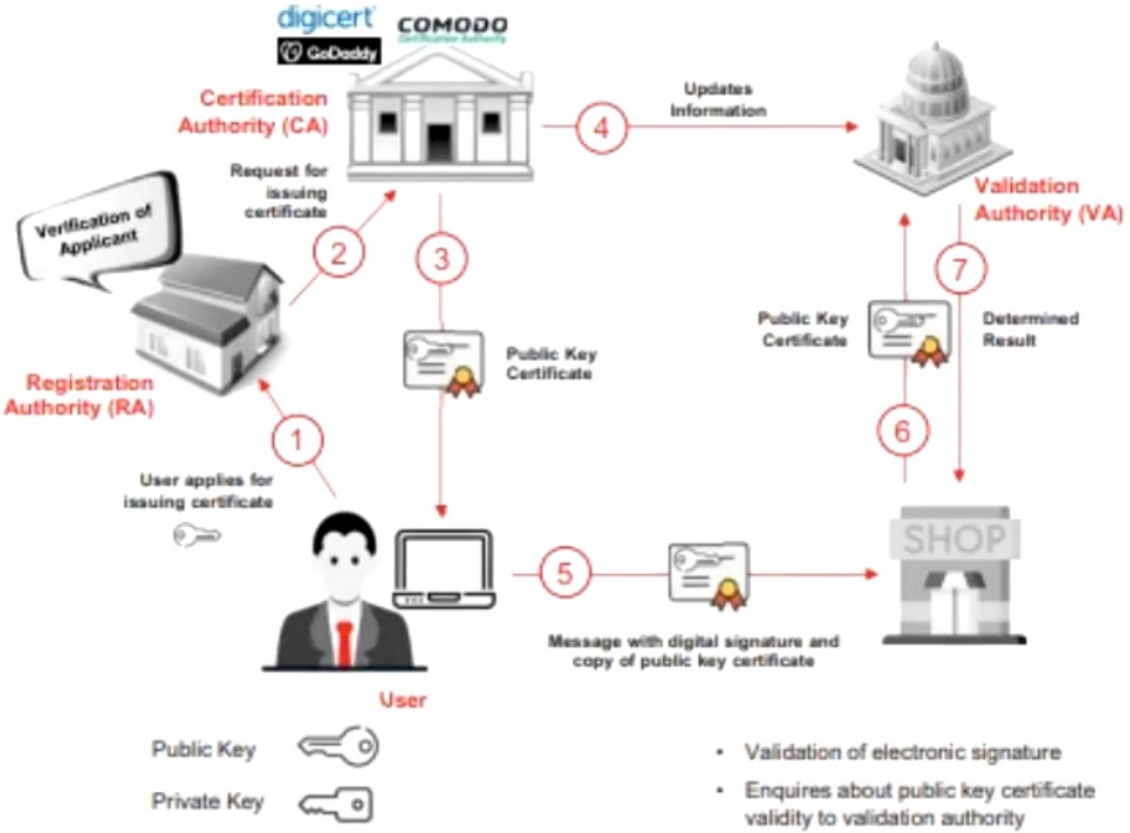
Explain Applications of Cryptography

Public Key Infrastructure (PKI)

PKI is a **set of hardware**, **software**, **people**, **policies**, and **procedures** required to create, manage, distribute, use, store, and revoke **digital certificates**

Components of PKI

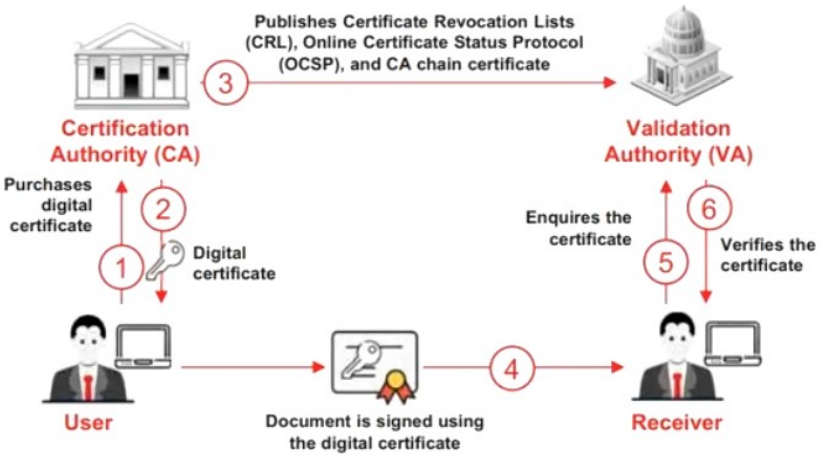
- **Certificate Management System**: Generates, distributes, stores, and verifies certificates
- **Digital Certificates**: Establish people's credentials in online transactions
- **Validation Authority (VA)**: Stores certificates (with their public keys)
- **Certificate Authority (CA)**: Issues and verifies digital certificates
- **End User**: Requests, manages, and uses certificates
- **Registration Authority (RA)**: Acts as the verifier for the certificate authority



Signed Certificate (CA) vs. Self-Signed Certificate

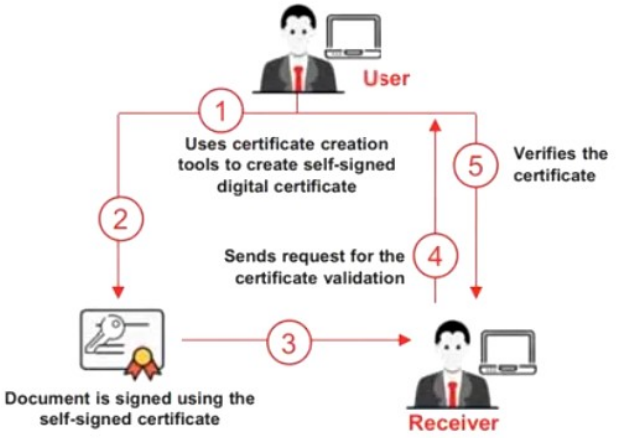
Signed Certificate

- User gets a **digital certificate** from a trustworthy CA
- The digital certificate contains name of the certificate holder, a serial number, expiration dates, **a copy of the certificate holder's public key** and the digital signature of the CA
- **User signs the document** using the **private key** and sends it to the receiver
- The receiver can verify the certificate by enquiring with the **validation authority** (VA)
- VA verifies the validity of the certificate



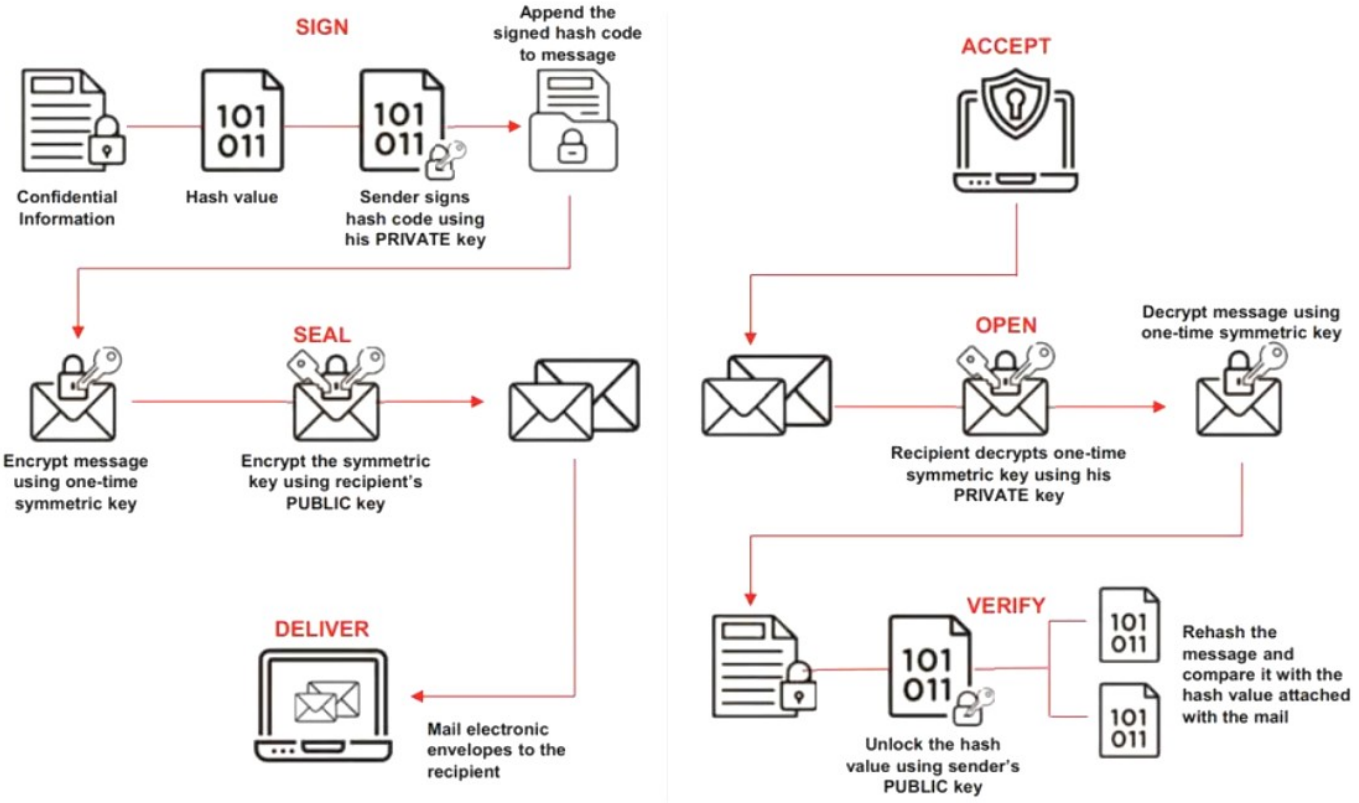
Self-Signed Certificate

- User creates self-signed digital certificate using a certification creation tools, such as **Adobe Acrobat Reader**, **Java keytool**, or **Apple's Keychain**
- The certificate contains name of the user, **user's public key** and his digital signature
- User **signs the document** using the **self-signed certificate** and sends to the receiver
- The receiver can verify the certificate by enquiring with the **user**
- User verifies the certificate to the receiver



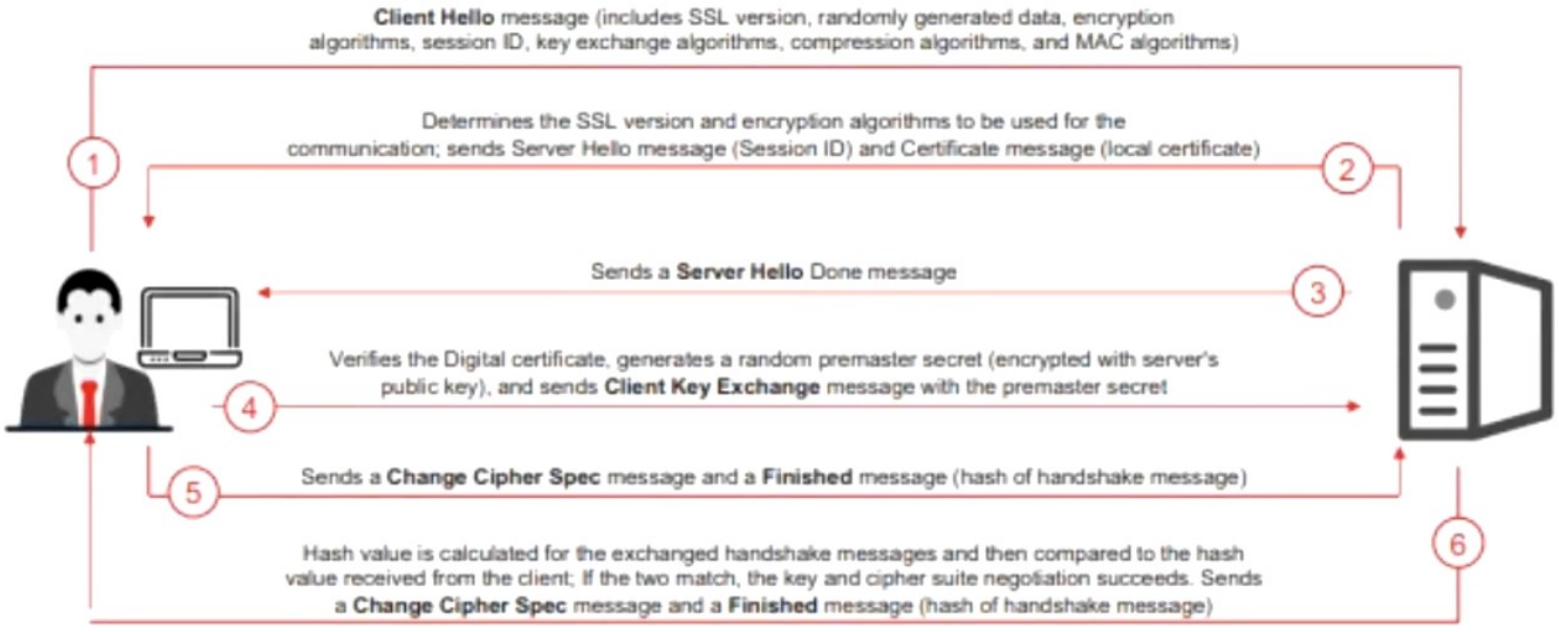
Digital Signature

- Digital signature uses asymmetric cryptography to simulate the security properties of a **signature in digital rather than written form**
- A digital signature may be further protected by **encrypting the signed email** for confidentiality



Secure Sockets Layer (SSL)

- SSL is an application layer protocol developed by Netscape for **managing the security** of message transmission on the Internet
- It uses **RSA asymmetric (public key) encryption** to encrypt data transferred over SSL connections



Transport Layer Security (TLS)

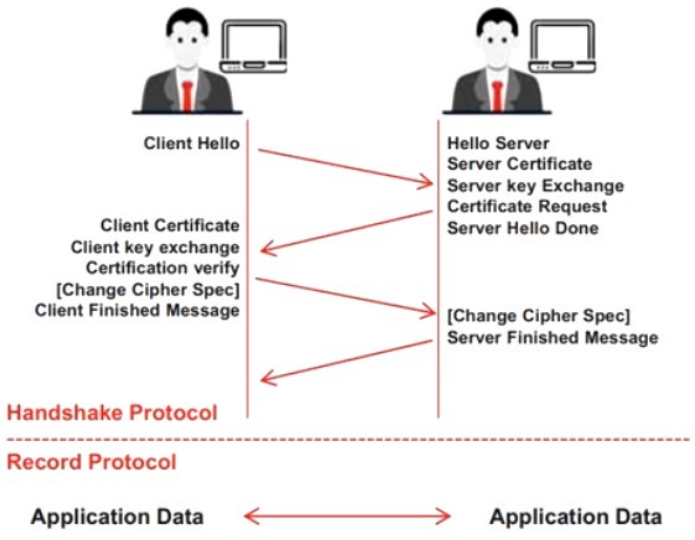
TLS is a protocol **to establish a secure connection** between a client and a server and ensure the privacy and integrity of information during transmission

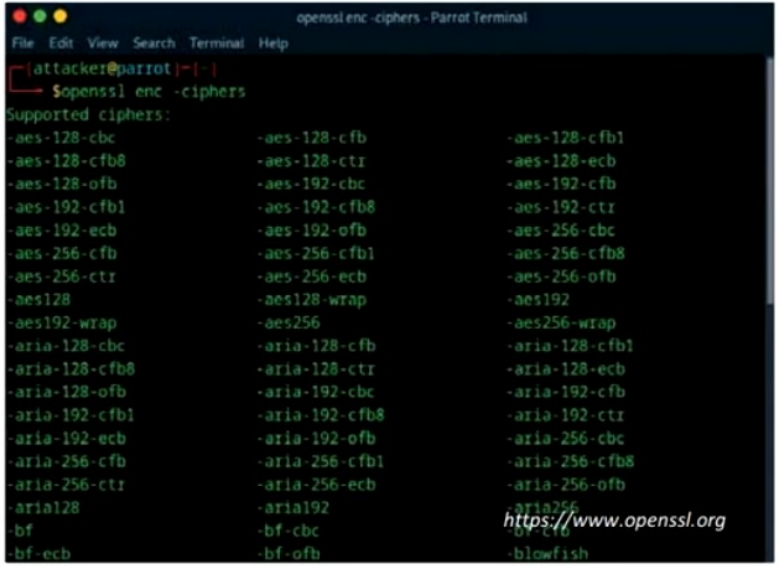
TLS Handshake Protocol

It allows the client and server to authenticate each other, select an encryption algorithm, and exchange a symmetric key prior to data exchange

TLS Record Protocol

It provides secured connections with an encryption method, such as DES





OpenSSL

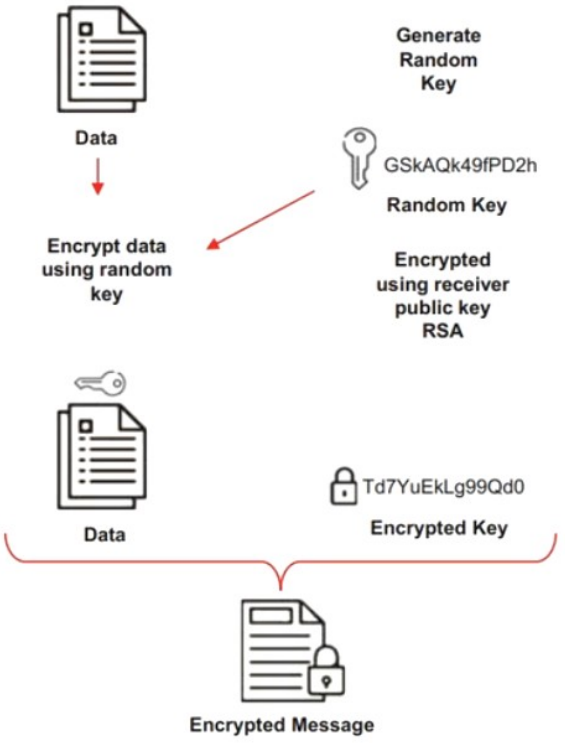
OpenSSL is an open-source cryptography toolkit implementing **SSL v2/v3** and **TLS v1** network protocols and the related cryptography standards required by them

Pretty Good Privacy (PGP)

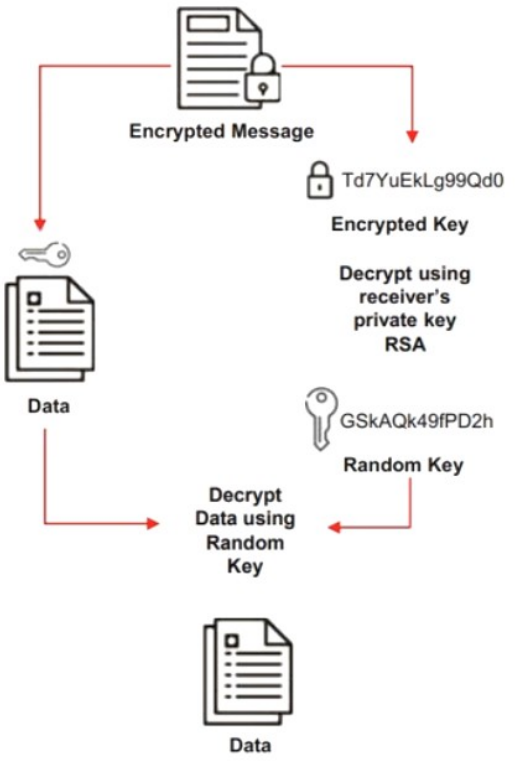
Pretty Good Privacy

- PGP is a protocol used to **encrypt** and **decrypt** data that provides **authentication** and **cryptographic privacy**
- It is often used for data **compression**, **digital signing**, encryption and decryption of **messages**, **emails**, **files**, **directories**, and to enhance the privacy of email communications
- It combines the best features of both **conventional** and **public key cryptography** and is therefore known as a **hybrid cryptosystem**

PGP Encryption



PGP Decryption

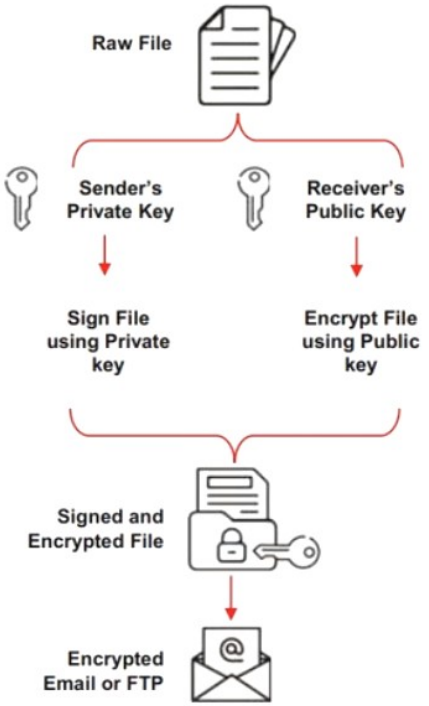


GNU Privacy Guard (GPG)

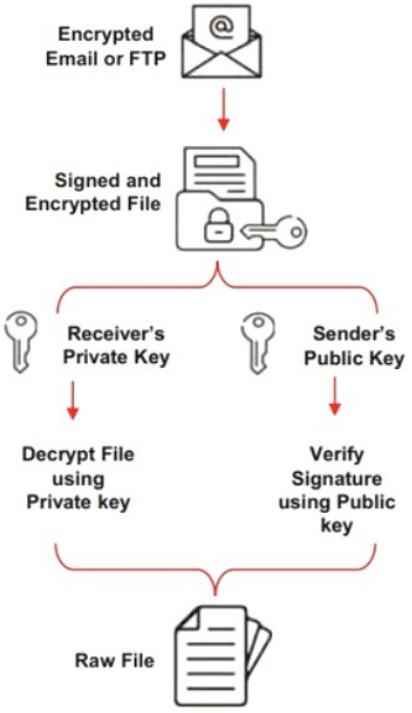
GNU Privacy Guard

- GPG is a **software replacement of PGP** and free implementation of the OpenPGP standard
- GPG is also called **hybrid encryption software** as it uses both symmetric key cryptography and asymmetric key cryptography
- It also supports S/MIME and Secure Shell (SSH)

GPG Signing and Encryption

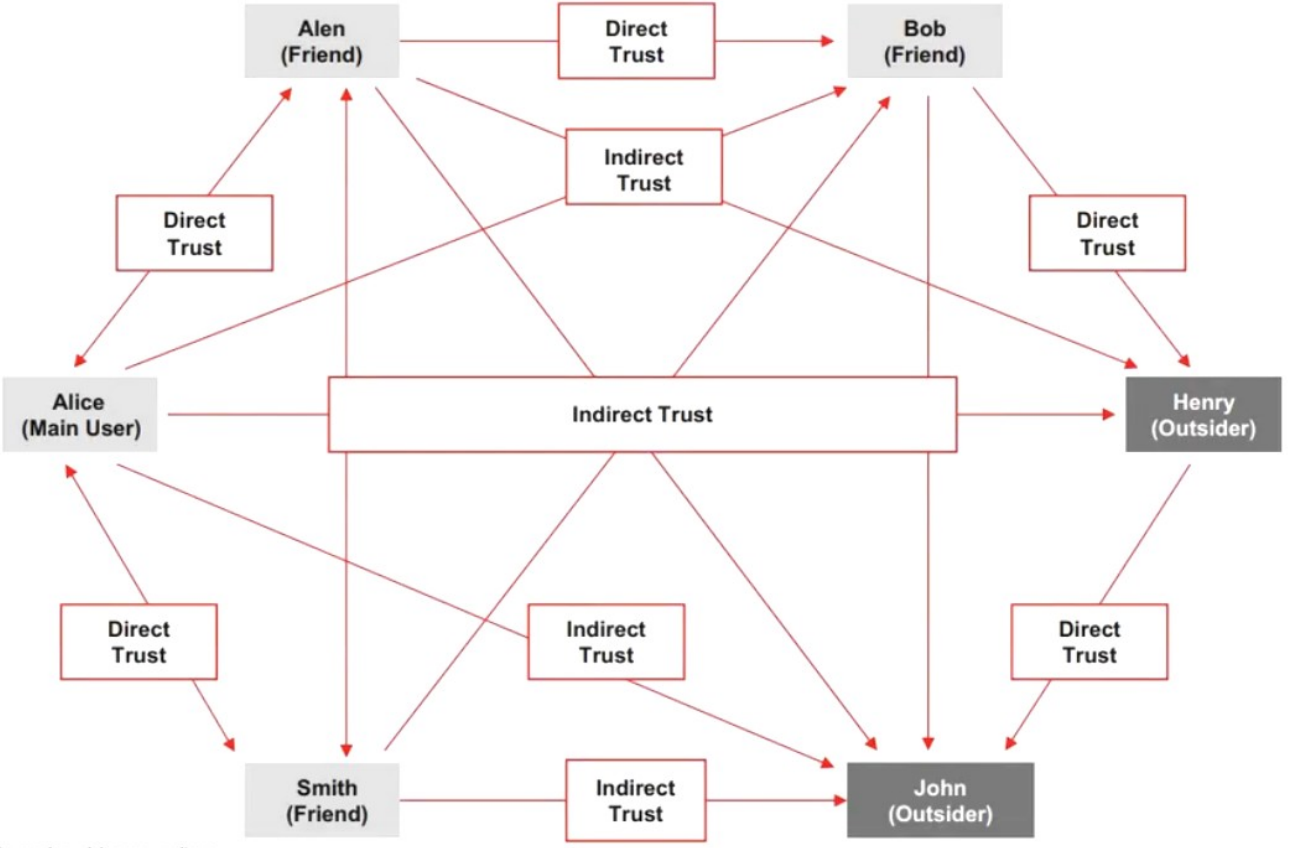


GPG Decryption and Verification



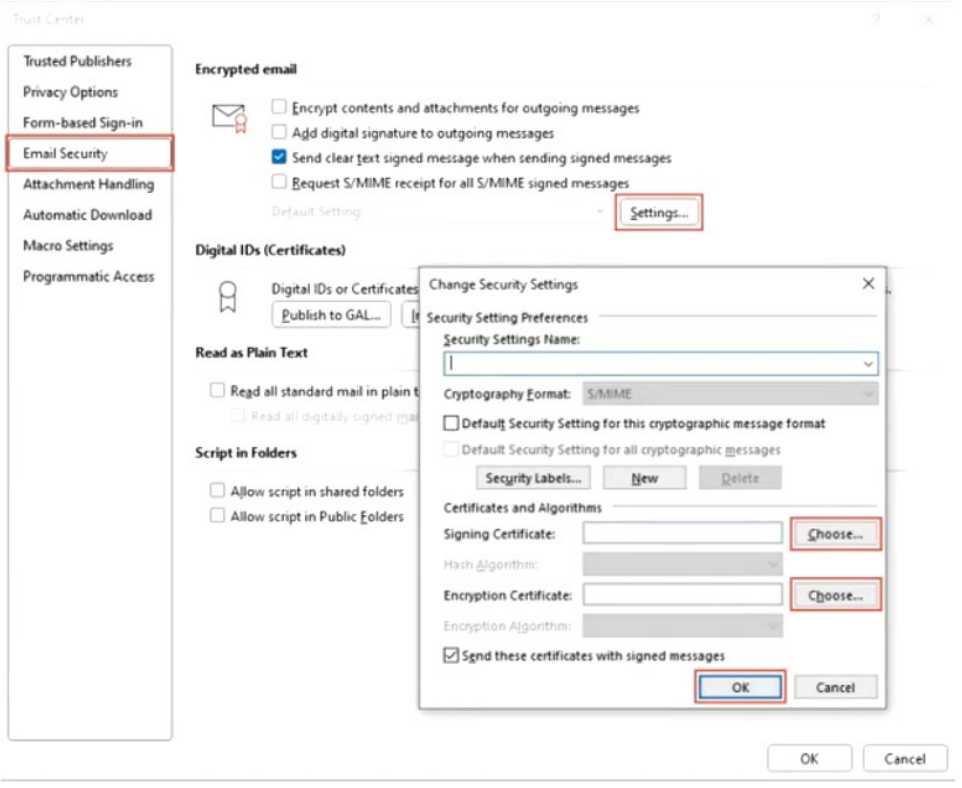
Web of Trust (WOT)

- Web of trust (WoT) is a **trust model of PGP**, OpenPGP, and GnuPG systems
- Everyone in the network is a Certificate Authority (CA) and signs for other trusted entities
- WoT is a **chain of a network** in which individuals intermediately validate each other's certificates using their signatures
- Every user in the network has a **ring of public keys** to encrypt the data, and they introduce many other users whom they trust



Encrypting Email Messages in Outlook: S/MIME Encryption

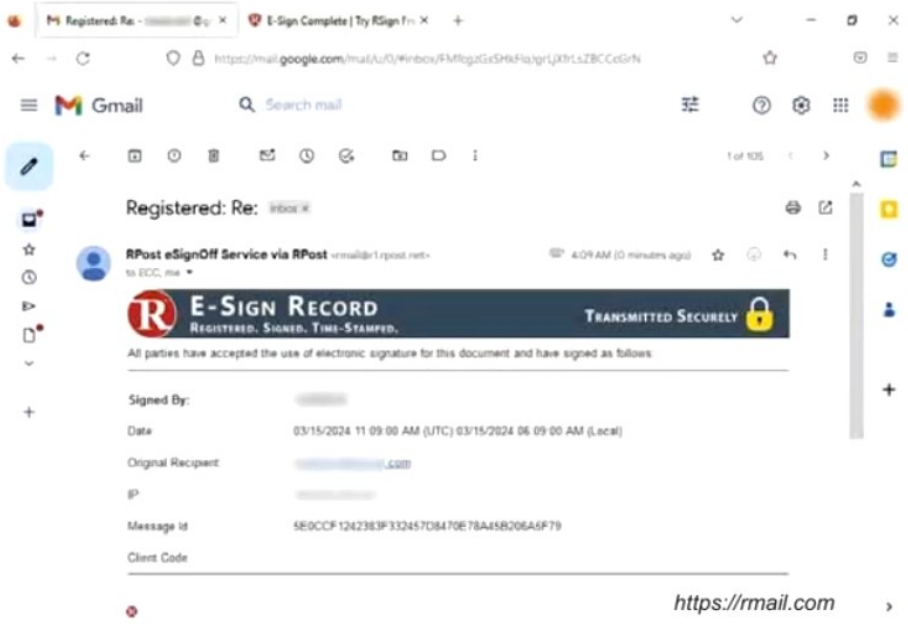
- Navigate to **File → Options → Trust Center → Trust Center Settings**
- Choose the **Email Security** option from the left pane
- In the **Encrypted email** section, click on the **Settings** option beside **Default Setting**
- In the **Change Security Settings** pop-up window, under the **Certificates and Algorithms** section, choose the **S/MIME certificate** for the **Signing certificate** and **Encryption certificate** options and click **OK**



Email Encryption Tools

RMail

RMail is an email security tool that provides open tracking, delivery proof, **email encryption**, **electronic signatures**, large file transfer functionality, etc.



https://rmail.com

Mailvelope
https://mailvelope.com

Virtru
https://www.virtru.com

Webroot™
https://www.webroot.com

Secure Email (S/MIME) Certificates
https://www.ssl.com

Proofpoint Email Protection
https://www.proofpoint.com

Disk Encryption

Confidentiality

Disk encryption protects the **confidentiality of the data** stored on disk by converting it into an unreadable code using disk encryption software or hardware

- **Privacy**
- **Passphrase**
- **Hidden Volumes**

Encryption

It works in a similar way as **text message encryption** and protects data even when the OS is not active

- **Volume Encryption**

Protection

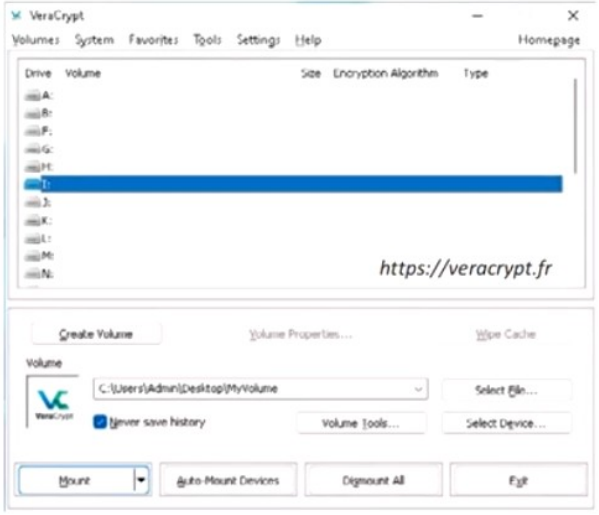
With the use of an encryption program for your disk, you can **safeguard any information** to burn onto the disk and keep it from falling into the wrong hands

- **USB Flash Drive**
- **External HDD**
- **Backup**

Disk Encryption Tools

VeraCrypt

VeraCrypt is software for establishing and maintaining an **on-the-fly-encrypted volume** (data storage device)

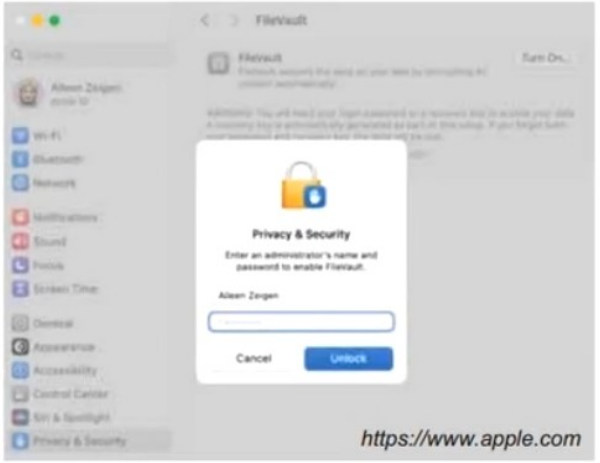


Rohos Disk Encryption

It allows users to **create hidden** and **encrypted partitions** on a computer, USB flash drive, or cloud storage service such as Google Drive, OneDrive, and Dropbox

FileVault

FileVault utilizes the **XTS-AES-128 encryption technology** along with a 256-bit key to prevent unauthorized access to the information on the startup disk



Other Disk Encryption Tools:

BitLocker Drive Encryption
https://www.microsoft.com

Symantec Encryption
https://www.broadcom.com

SafeGuard Enterprise Encryption
https://www.sophos.com

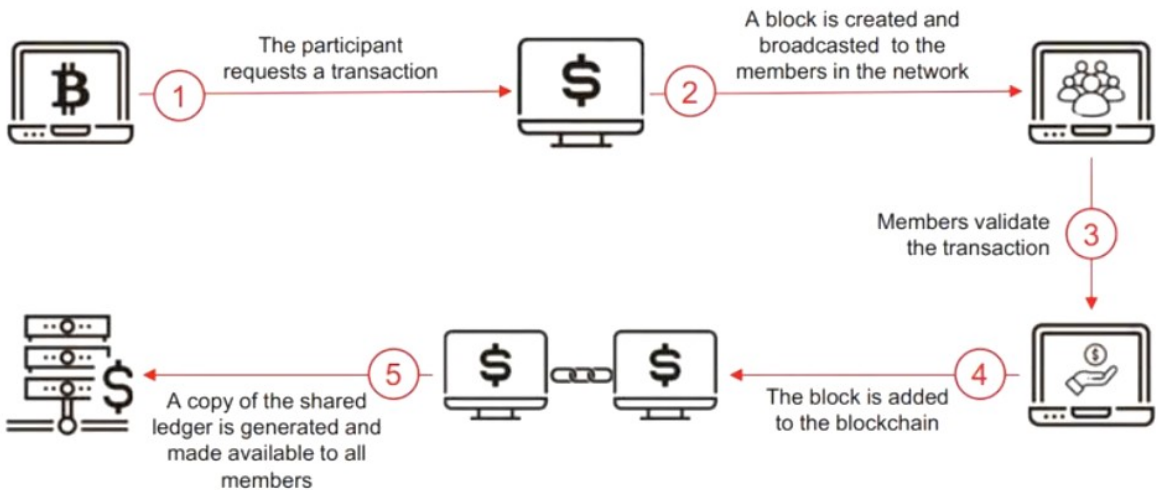
Cryptsetup
https://gitlab.com

Cryptmount
https://cryptmount.sourceforge.net

Blockchain

- A blockchain, also referred to as **distributed ledger technology** (DLT), is used to record and store the history of transactions in the form of blocks
- For multiple transactions, **multiple blocks are created**, which are linked together to form a "blockchain"

Process of Creating a Blockchain



Types of Blockchain

1. **Public Blockchain**
2. **Private Blockchain**
3. **Federated Blockchain**
4. **Hybrid Blockchain**

Objective 03

Explain Different Cryptanalysis Methods and Cryptography Attacks

Cryptanalysis Methods

Linear Cryptanalysis

- Commonly used on **block ciphers**
- It is a known plaintext attack and uses a **linear approximation** to describe the behavior of the block cipher
- Given sufficient pairs of **plaintext** and **corresponding ciphertext**, bits of information about the key can be obtained
- For example, with a **56-bit DES key, brute force** could take up to **2^{56} attempts**

Differential Cryptanalysis

- Differential cryptanalysis is a form of cryptanalysis applicable to **symmetric key algorithms**
- It is the **examination of differences** in an input and how that affects the resultant difference in the output
- It originally worked only **with chosen plaintext**
- It can now also work with **known plaintext** and **ciphertext only**

Integral Cryptanalysis

- This attack is useful against block ciphers based on **substitution-permutation networks**, an extension of differential cryptanalysis
- Integral analysis, for block size b, holds ***b-k* bits constant** and runs the other *k* through all ***2k* possibilities**
- For ***k*=1**, this is just differential cryptanalysis, but with ***k*>1** it is a new technique

Quantum Cryptanalysis

- Quantum cryptanalysis is the process of **cracking cryptographic algorithms** using a **quantum computer**
- To perform cryptanalysis, attackers must obtain the **encrypted content**, and the process requires significant time and **quantum resources** such as Circuit Width, Circuit Depth, Number of Gates, Number of T-Gates, T-Depth, and MAXDEPTH

Cryptography Attacks

Cryptography attacks are based on the assumption that the cryptanalyst has access to the **encrypted information**

Ciphertext-only Attack	Attacker has access to the cipher text; the goal of this attack is to **recover the encryption key** from the ciphertext
Adaptive Chosen-plaintext Attack	Attacker makes a **series of interactive queries**, choosing subsequent plaintexts based on the information from the previous encryptions
Chosen-plaintext Attack	Attacker **defines their own plaintext**, feeds it into the cipher, and analyzes the resulting ciphertext
Related-Key Attack	Attacker can obtain ciphertexts encrypted under **two different keys**; this attack is useful if the attacker can obtain the plaintext and matching cipher text
Dictionary Attack	Attacker constructs a **dictionary of plaintext** along with its corresponding ciphertext that they have learnt over a certain period of time

Cryptography Attacks (Cont'd)

Known-plaintext Attack	Attacker has **knowledge of some part of the plain text**; using this information, the key used to generate ciphertext is deduced to decipher other messages
Chosen-ciphertext Attack	Attacker obtains plaintexts corresponding to an **arbitrary set** of ciphertexts of their own choosing
Rubber Hose Attack	Extraction of cryptographic secrets (e.g., the password to an encrypted file) from a person by **coercion or torture**
Chosen-key Attack	Attacker usually breaks an **n bit** key cipher into **$2^{n/2}$** operations
Timing Attack	It is based on repeatedly measuring the **exact execution times** of modular exponentiation operations
Man-in-the-middle Attack	Attacker performs this attack on the **public key cryptosystems** where key exchange is required before communication takes place

Code Breaking Methodologies

One can measure the **strength of an encryption algorithm** using various code-breaking techniques

Brute Force	Cryptography keys are discovered by **trying every possible combination**
Frequency Analysis	• The study of the frequencies of letters or groups of letters in a **ciphertext** • It works based on the fact that in any given stretch of written language, certain letters and **combinations of letters** occur with varying frequencies
Trickery and Deceit	Involves the use of **social engineering techniques** to extract cryptography keys
One-Time Pad	A one-time pad contains many **non-repeating groups of letters** or number keys, which are chosen randomly

Brute-Force Attack

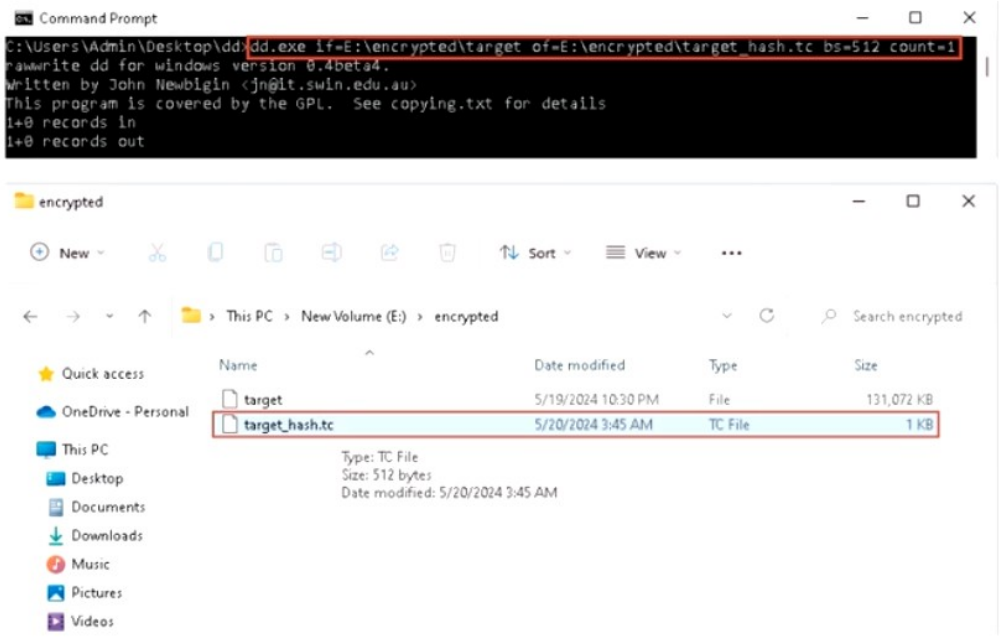
- Defeating a cryptographic scheme by **trying a large number of possible keys** until the correct encryption key is discovered
- Brute-force attack is a **high-resource and time intensive process**, but it is more guaranteed to achieve results
- Success of brute-force attack depends on the **length of the key**, **time constraint**, and **system security mechanisms**

Power/Cost	40 bits (5 char)	56 bits (7 char)	64 bits (8 char)	128 bits (16 char)
$ 2K (1 PC; can be achieved by an individual)	1.4 min	73 days	50 years	10^20 years
$ 100K (can be achieved by a company)	2 sec	35 hours	1 year	10^19 years
$ 1M (can be achieved by a huge organization or a state)	0.2 sec	3.5 hours	37 days	10^18 years

Estimated Time for Successful Brute-force Attack

Brute-Forcing VeraCrypt Encryption

- Brute-forcing VeraCrypt encryption is an attack technique in which attackers attempt to decrypt the encrypted data
- Attackers use dd command to extract the hash value from the encrypted container and hashcat or John the Ripper tool to brute-force the password



Steps to Brute-force VeraCrypt encryption Using hashcat

- Run the command to extract first 512 bytes of hash value from encrypted container

  dd.exe if=<path_to_container> of=<path_to_hashfile.tc> bs=512 count=1

- Run the command for brute-forcing numeric password:

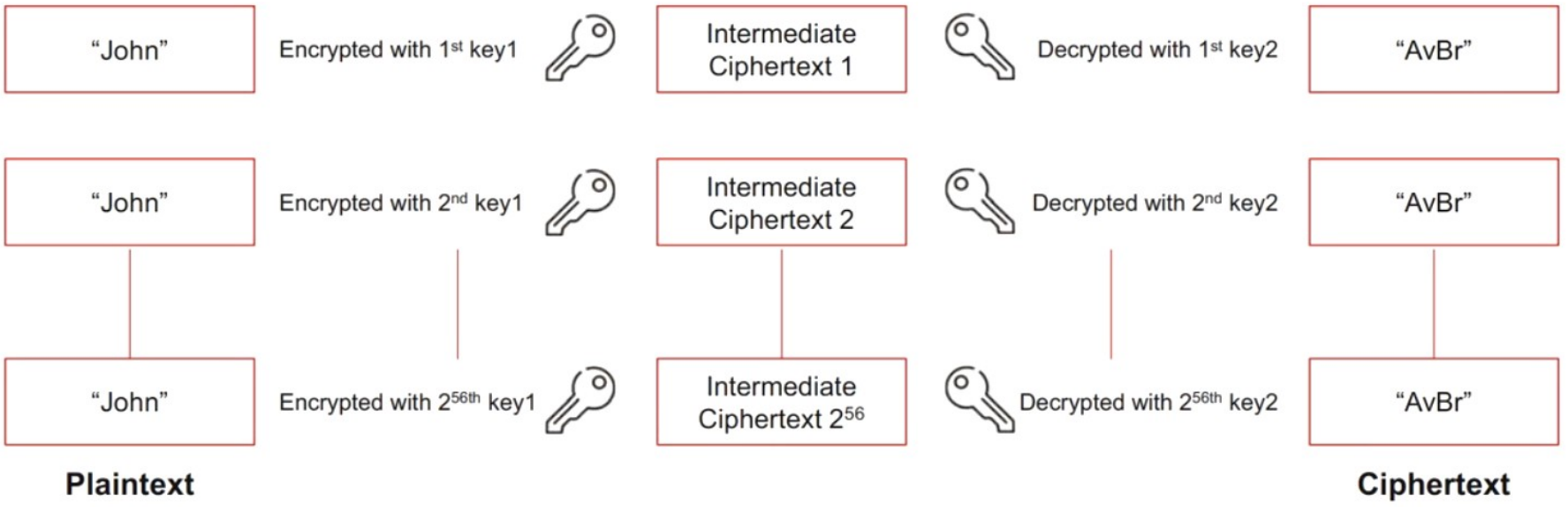
  hashcat.exe -a 3 -w 1 -m 13721 <path_to_hashfile.tc> ?d?d?d?d

- Run the command for brute-forcing with a wordlist.txt file that contain default passwords

  hashcat.exe -w 1 -m 13721 hash.tc wordlist.txt

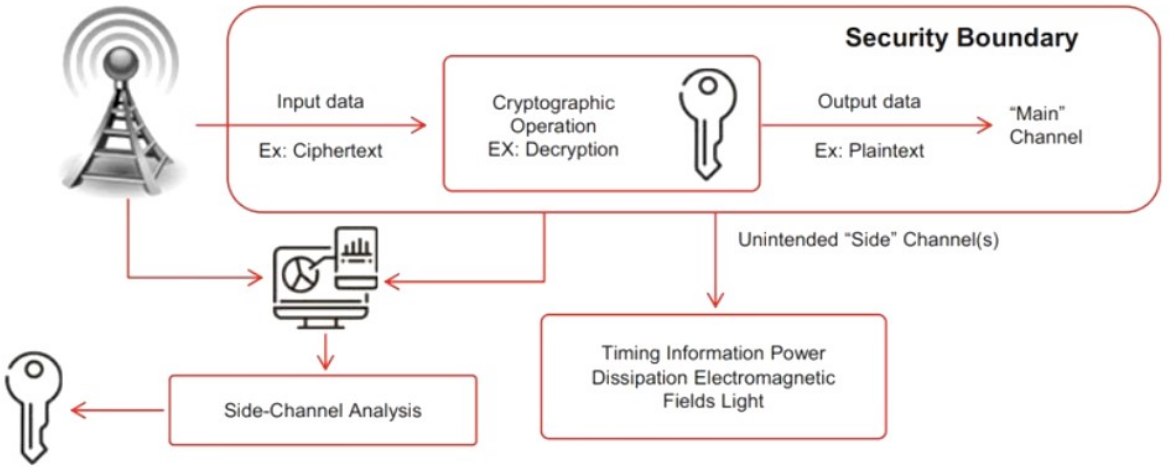
Meet-in-the-Middle Attack on Digital Signature Schemes

- The attack works by **encrypting from one end** and **decrypting from the other end**, thus meeting in the middle
- It can be used for **forging messages** that use multiple encryption schemes



Side-Channel Attack

- A side-channel attack is a **physical attack** performed on a cryptographic device/cryptosystem to gain sensitive information
- Cryptography is generally part of the hardware or software that runs on physical devices, such as semi-conductors (including resistors, transistors, etc.)
- These physical devices are affected by various **environmental factors**, including power consumption, electro-magnetic field, light emission, timing and delay, and sound
- In a side-channel attack, an attacker **monitors these channels (environmental factors)** and tries to acquire the information useful for cryptanalysis



- Assume that encrypted data is to be decrypted and displayed as plain text inside a **trusted zone**
- At the time of decryption in a cryptosystem, **physical environmental factors**, such as timing and power dissipation, acting on the components of a computer are recorded by an attacker
- The attacker analyzes this information in an attempt to **gain useful information** for cryptanalysis

Hash Collision Attack

A hash collision attack is performed by finding **two different input messages** that result in the same hash output

This allows the attacker to perform cryptanalysis by **exploiting the digital signature** used to generate a different message with same hash value

The SHA-1 algorithm converts input messages into **constant-length unstructured strings** of numbers and alphabets, which act as a fingerprint for the sent file

Attacker is able to forge the victim's **digital signature** of message a1 on the incorrect message a2

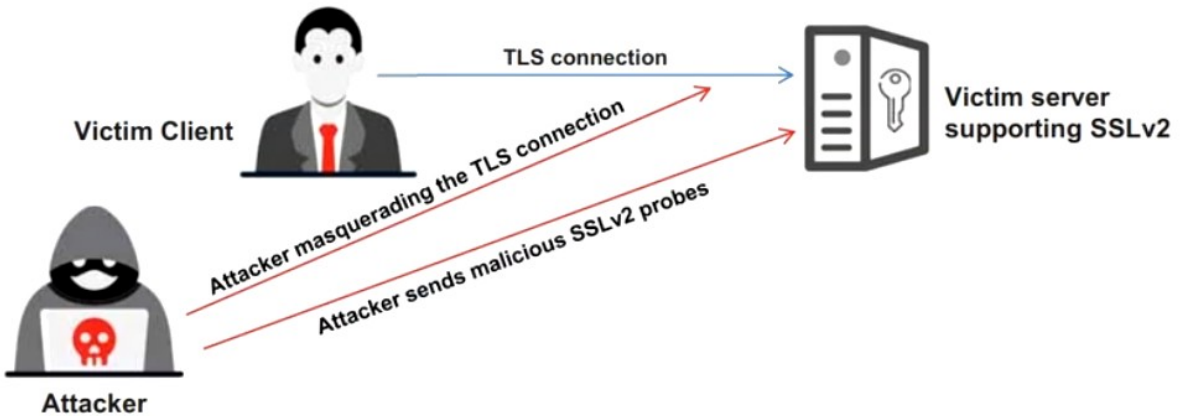
Once the attacker is able to detect any collisions in the hash, they try to identify more collisions by **concatenating data** to the matching messages

DUHK Attack

1. DUHK (Don't Use Hard-Coded Keys) is a **cryptographic vulnerability** that allows an attacker to **obtain encryption keys** used to secure VPNs and web sessions
2. This attack mainly affects any hardware/software using the ANSI X9.31 **random number generator** (RNG)
3. **Pseudorandom number generators** (PRNGs) generate random sequences of bits based on the initial secret value, called a seed, and the current state
4. Both these factors are the key issues of a DUHK attack as any attacker could combine ANSI X9.31 with the hard-coded seed key **to decrypt the encrypted data** sent or received by that device
5. Using this attack, attackers identify encryption keys and **steal confidential information**, such as critical business data, user credentials, and credit card details

DROWN Attack

- A DROWN attack is a **cross-protocol weakness** that can communicate and initiate an attack on servers that support recent SSLv3/TLS protocol suites
- It affects cryptographic protocols like HTTPS and cryptographic services that depend on SSL and TLS
- A DROWN attack makes the attacker **decrypt the latest TLS connection** between the victim client and server by launching malicious SSLv2 probes using the same private key
- Attackers perform a DROWN attack as part of an **online MitM attack**, breaking the encrypted keys and sniffing sensitive information, such as passwords and bank account details

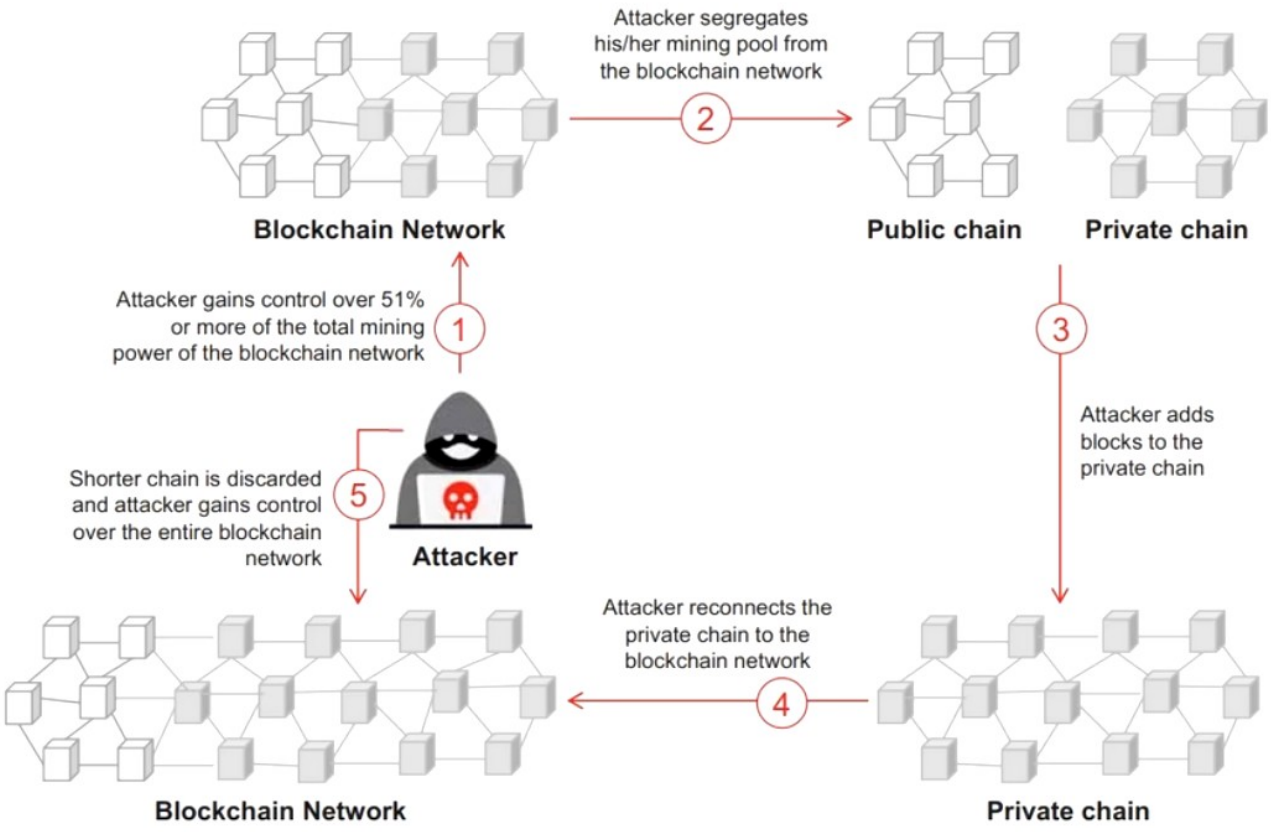


Rainbow Table Attack

1. A rainbow table attack is a type of cryptography attack where an **attacker uses a rainbow table to reverse cryptographic hash functions**
2. A rainbow table is a **precomputed table** that **contains word lists** like dictionary files and brute force lists and their hash values
3. It uses the **cryptanalytic time-memory trade-off technique** to crack the cryptography, which requires less time than some other techniques
4. An attacker computes the hash for a list of possible passwords and compares it to the precomputed hash table (rainbow table). If the attacker find a match, **they can crack the password**

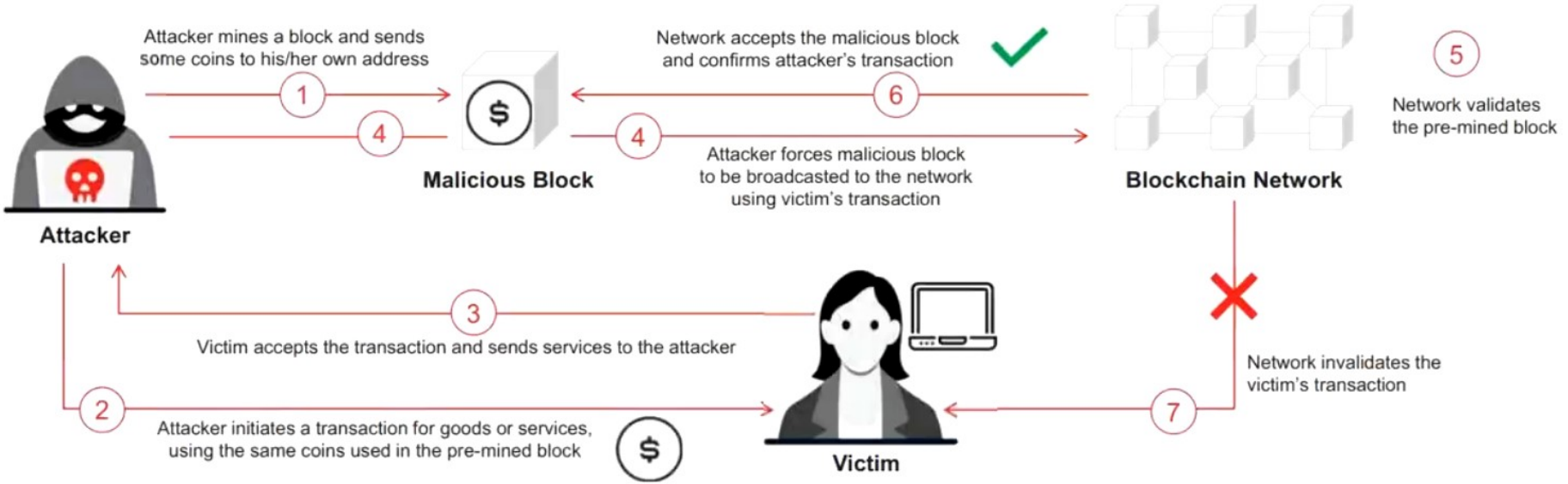
Attacks on Blockchain: 51% Attack

- 51% attack, also known as majority attack, occurs when an attacker or group of attackers gains control of **more than 50% of the computational power** (hash rate) or staking power in a blockchain network
- This level of control allows them to manipulate the blockchain by conducting **double-spending attacks**, **denial-of-service** (DoS) attacks, **transaction reversals**, and other malicious activities



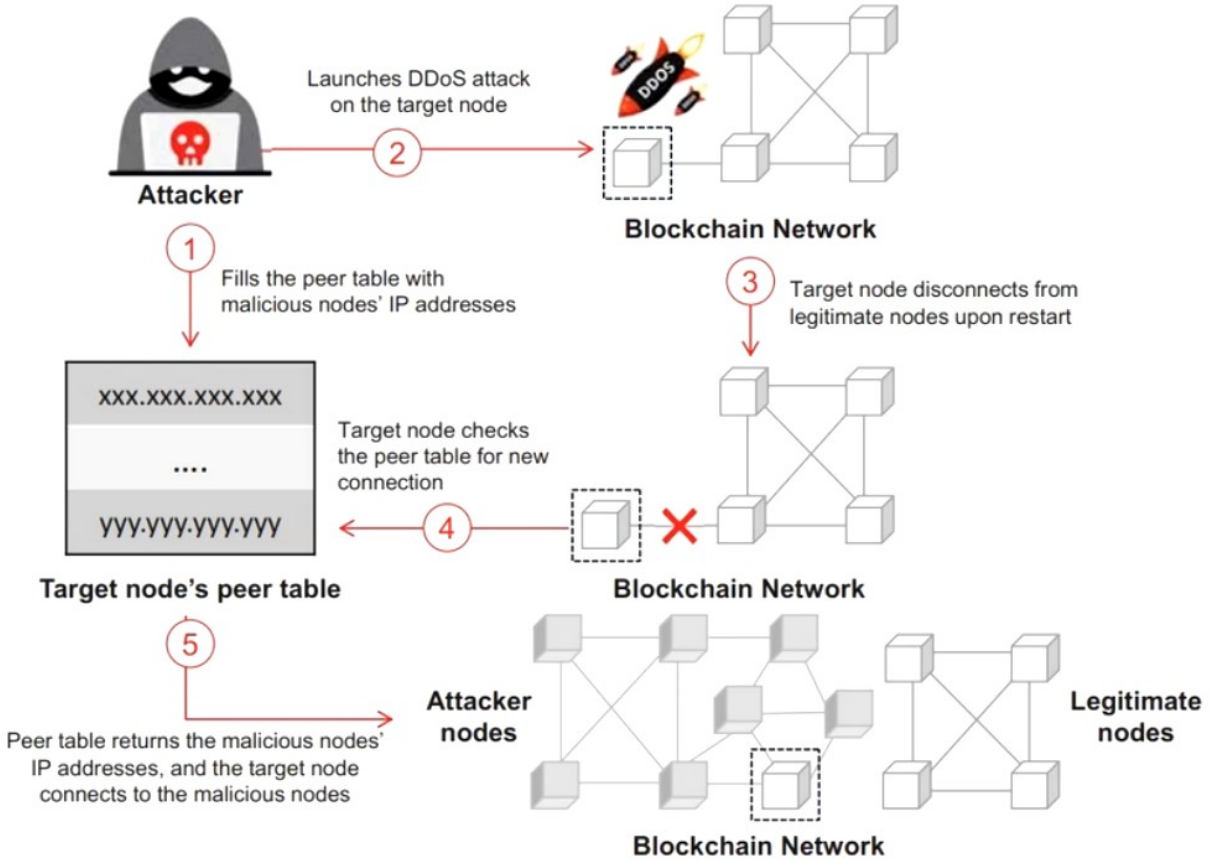
Attacks on Blockchain: Finney Attack

- Finney attack is a type of blockchain attack that involves an attacker leveraging the time **delays between the broadcasting and the confirmation of transactions** in cryptocurrency networks to **reverse the transactions** before they are confirmed
- Attackers perform this attack to **double-spend the cryptocurrency**, effectively getting goods or services for free while retaining their coins



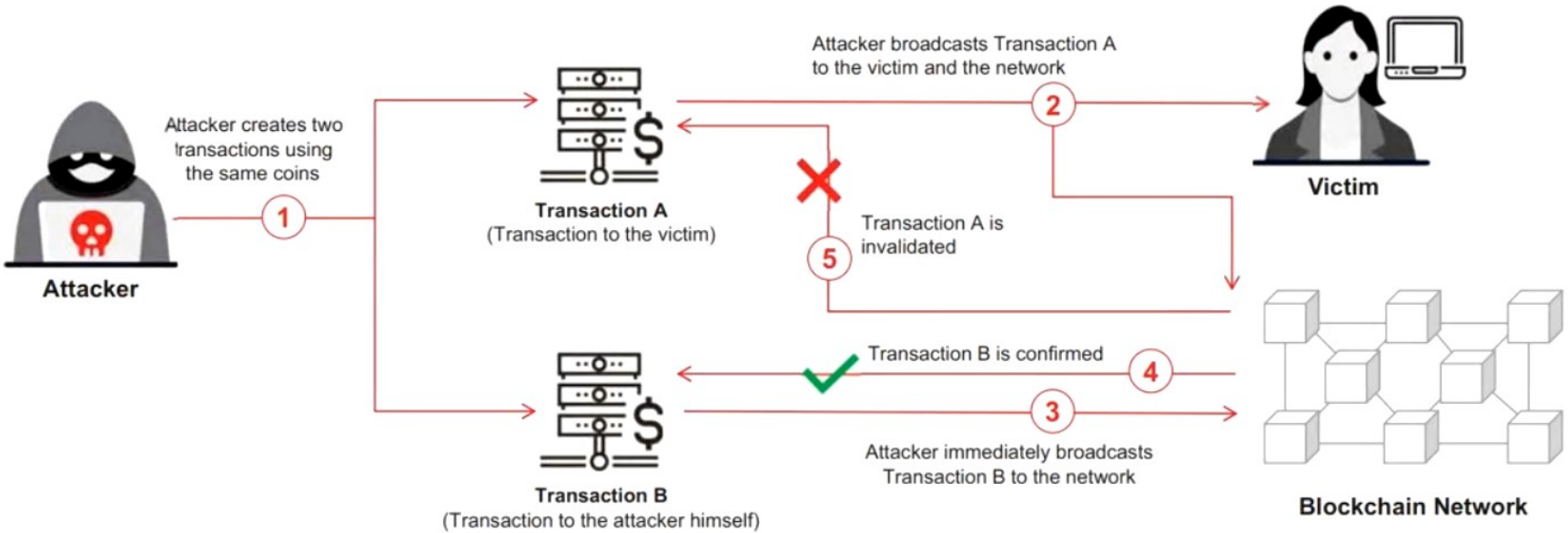
Attacks on Blockchain: Eclipse Attack

- Eclipse attack is a type of blockchain attack where the attacker **isolates a target node** from the rest of the network **by surrounding it with malicious nodes**, effectively controlling the node's view of the blockchain
- This attack allows the attacker to exploit the target node for various malicious purposes such as **disrupting transaction processing**, split mining power, facilitating **double-spending attacks** and so on



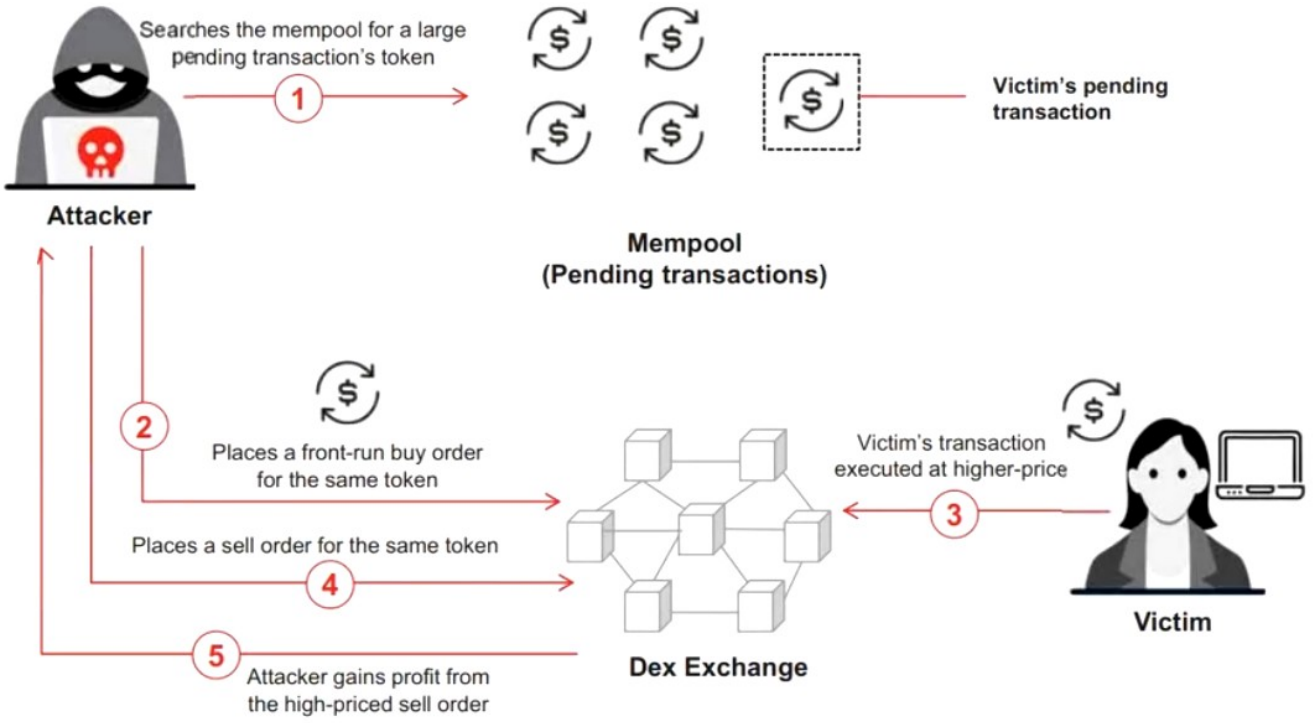
Attacks on Blockchain: Race Attack

- Race attack is a double-spending attack that exploits the **delay in transaction confirmation** in blockchain networks to obtain goods or services without actually paying for them
- Unlike the Finney attack, a race attack **does not require the attacker to pre-mine blocks** to reverse the victim's transaction but relies on the attacker's **ability to broadcast transactions quickly** and exploit the network's latency



Attacks on Blockchain: DeFi Sandwich Attack

- DeFi sandwich attack exploits the time delay and order execution mechanisms in decentralized exchanges (DEXs) to manipulate the price of a token
- This attack targets tokens with significant larger transactions
- Sandwich attacks can cause victims to buy tokens at an inflated price or sell them at a deflated price, leading to financial losses



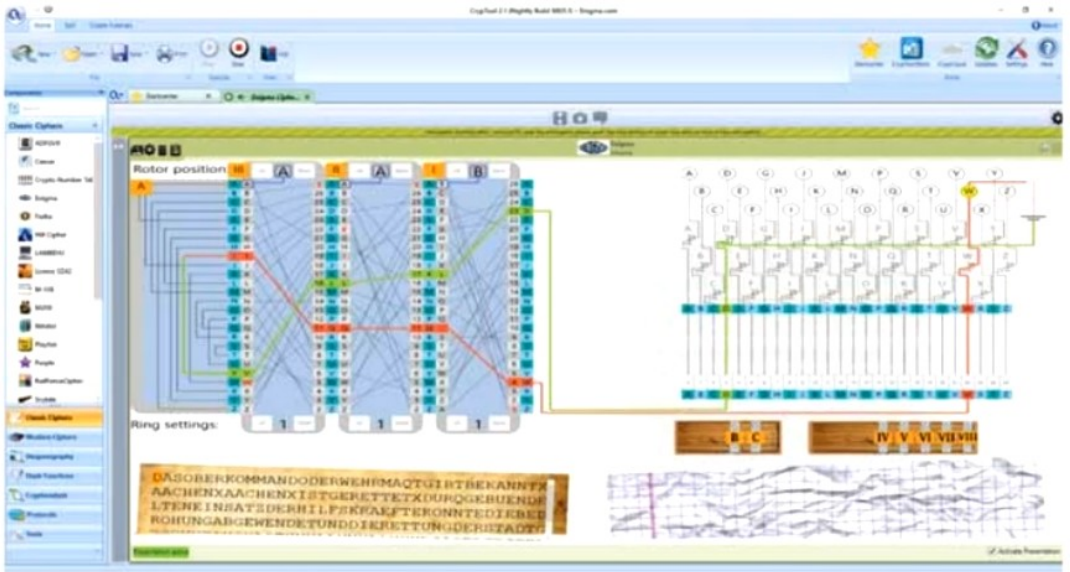
Quantum Computing Attacks

1. Quantum Cryptanalysis Attack
2. Quantum Side-Channel Attack
3. Classical-to-Quantum Transition Attack
4. Harvest-Now, Decrypt-Later Attack
5. Quantum Trojan Horse Attack
6. Quantum Supply Chain Attack
7. Quantum Computer Sabotage Attack
8. Fault Injection Attack on Quantum Hardware
9. Quantum Denial-of-Service (DoS) Attack
10. Quantum Data Eavesdropping
11. Quantum Bit Flipping Attack
12. Quantum Error Correction Mechanism Exploitation
13. Quantum Replay Attack

Cryptanalysis Tools

CrypTool

- CrypTool is a e-learning program in the area of **cryptography** and **cryptanalysis**
- It consists of e-learning software (CT1, CT2, JCT, and CTO)



https://www.cryptool.org

RsaCtfTool
https://github.com

Msieve
https://sourceforge.net

Cryptol
http://github.com

CryptoSMT
https://github.com

MTP
https://github.com

Objective 04

Explain Cryptography Attack Countermeasures

How to Defend Against Cryptographic Attacks

1. Access to **cryptographic keys** should be given to the application or user directly
2. **Intrusion detection system** should be deployed to monitor exchange and access of keys
3. Passphrases and passwords must be used to **encrypt the key** if it is stored on the disk
4. Keys should not be present inside the **source code** or **binaries**
5. For certificate signing, **transfer of private keys** should not be allowed
6. For symmetric algorithms, key sizes of **256 bits** should be preferred for a secure system, especially in large transactions
7. **Message authentication** must be implemented for encryption of symmetric-key protocols
8. For asymmetric algorithms, key sizes of at least **2048 bits** should be considered for secure and highly protected applications
9. In the case of hash algorithms, a hash length of **256 bits** or **higher** should be considered for secure applications
10. Recommended **tools** and **products** should be preferred over creating **self-engineered** crypto **algorithms** and functions
11. **Avoid encryption key relationships** being simple, i.e., each encrypted key should be created from KDF
12. The output of the hash function should have a **higher bit length**, making it difficult to decrypt

How to Defend Against Blockchain Attacks

1. Implement **Decentralized Identifiers** (DIDs) to enhance identity verification security and privacy
2. Use **zero-knowledge proofs** to verify transactions and identities without revealing sensitive information
3. Store cryptographic keys in **HSMs** to protect against unauthorized access and tampering
4. Use **multi-signature wallets** that require multiple keys to authorize a transaction
5. Use **atomic swaps** for cross-chain trading to reduce the risk of incomplete transactions
6. Use **out-of-band verification** methods to check the validity of blockchain data from trusted sources
7. Wait for **multiple confirmations** to accept transactions
8. Use a set of trusted **bootstrapping nodes** to help new nodes connect to the network securely
9. Increase the **speed at which transactions propagate** across the network to minimize attack windows
10. Use **batch processing** and **fair sequencing** to prevent transaction reordering and manipulation

How to Defend Against Quantum Computing Attacks

1. Use **larger keys** for symmetric cryptography to counteract the reduction in security from quantum attacks
2. Integrate **quantum-resistant digital signatures** into blockchain protocols
3. **Encrypt** stored data with quantum-resistant algorithms
4. Break data into **fragments** and distribute it across multiple locations to avoid reconstruction of the original data
5. Develop **quantum-specific firewalls** to filter and protect quantum communication channels
6. Use **quantum-resistant zero-knowledge proofs** to authenticate users
7. Implement quantum-resistant **distributed ledger technology** for secure decentralized transaction records
8. Use **Trusted Platform Modules** that support quantum-resistant cryptographic algorithms for secure boot process
9. Include quantum-resistance checks in **SDLC** and **code review** processes.
10. Use **HSMs** for secure storage of quantum-resistant cryptographic keys and ensure they are regularly updated

Key Stretching

Key stretching refers to the process of strengthening a key that might be slightly too weak, usually by making it longer

PBKDF2

PBKDF2 (**Password-Based Key Derivation Function 2**) is a part of **PKCS #5 v. 2.01**. It applies some function (such as hash or HMAC) to the password or passphrase along with Salt to produce a derived key

Bcrypt

bcrypt is used with passwords; it essentially uses a derivation of the **Blowfish algorithm**, converted to a hashing algorithm to hash a password and add Salt to it

Module Summary

In this module, we discussed the following:

- Basic cryptography concepts used to protect confidential data along with different types of cryptography
- Ciphers and different encryption algorithms used to encrypt or decrypt the data
- Various cryptography tools
- Importance of public key infrastructure (PKI) for encryption in detail
- Email encryption protocols and tools in detail
- Disk encryption and various disk encryption tools in detail
- Types of cryptanalysis methods and code breaking methodologies currently in use
- Various cryptanalysis attacks along with cryptanalysis tools
- Countermeasures used to defend against various cryptography attacks

EC-Council

www.eccouncil.org