19z701
Cryptography Presentation

Role of Cryptography in Security Protocols

Introduction & Need for Cryptography in Protocols

Prateekshaa T (22z246)

Every time you send a message, shop online, or log in — how is your data kept safe?

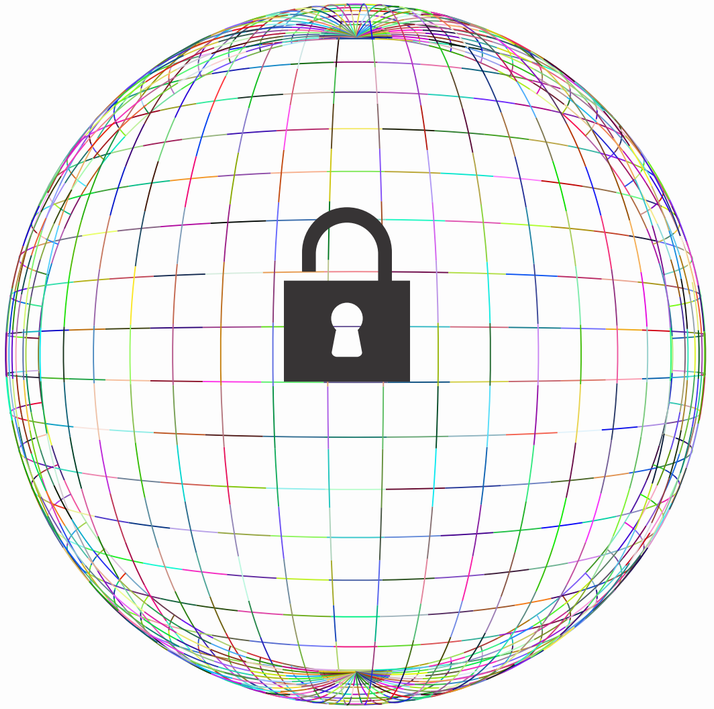
Security Protocols: Rules that define how data is securely transmitted over networks.

Cryptography: The science of protecting data through encoding and decoding.

Why We Need Security Protocols

- **Rising Frequency & Cost of Cyberattacks**

More than 4100 publicly disclosed breaches happened last year — that's roughly 11 breaches per day.



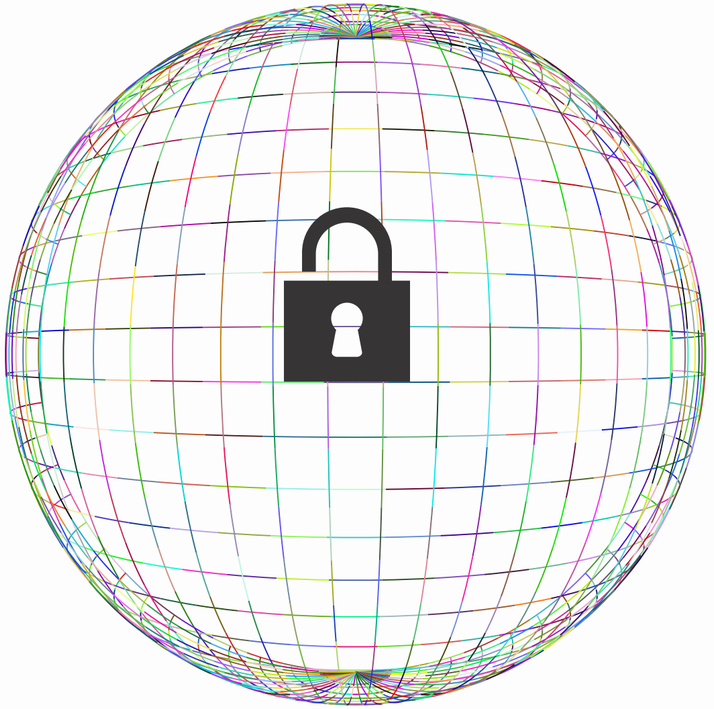
- **Protection of Sensitive Information (Credentials, Personal Data)**

Without encryption and secure protocols, such data can be intercepted or modified.
Security protocols, using cryptographic techniques, ensure that even if data is captured, it cannot be read or misused by unauthorized parties.

Continued...

Erosion of Trust & Reputational Damage

For example, Qantas Airways in 2025 experienced a breach where personal data of over 5 million customers was exposed.
Such breaches lead to customer distrust, regulatory fines, and long-term brand damage.



References

- The Guardian
- Times of India

Cryptography in Security Protocols

1. Confidentiality

2. Integrity

3. Authentication

4. Non-repudiation

Cryptography serves as the core engine behind several widely used security protocols that protect our everyday online interactions:

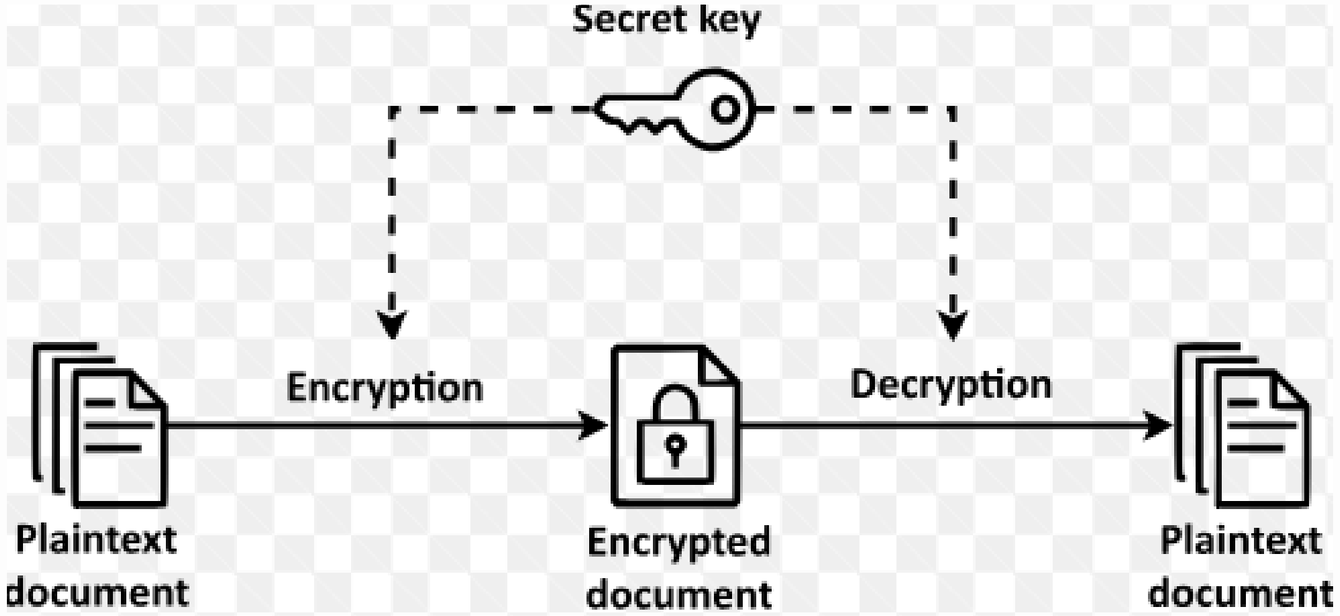
- SSL/TLS (for HTTPS websites)
- SSH (Secure Shell)
- VPN (Virtual Private Network)

Symmetric Cryptography In Protocols

Adish kumar S (22z206)

Symmetric Cryptography

- Uses a common key
- Confidentiality
- Integrity
- Authenticity
- Secured Communication



Types of Symmetric cryptography

Block cipher

- Encrypts fixed-size blocks of data (e.g., 64-bit or 128-bit).
- Same key is applied to each block using rounds of transformation.
  Example: AES, DES

Stream cipher

- Encrypts data bit by bit or byte by byte.
- Uses a keystream generator that mixes key bits with plaintext bits..
  Example: RC4

Advanced Encryption Standard

- AES is the most widely used symmetric block cipher standard.
- Developed to replace DES.
- Uses block size: 128 bits and key sizes of 128, 192, or 256 bits.
- Performs multiple rounds of substitution, permutation, and mixing operations.
- Provides strong security, speed, and efficiency.

Advanced Encryption Standard

Example

Plaintext: HELLO Key: MYKEY

Step 1 : Characters->ASCII ->Binary

Character	ASCII	Binary
H	72	1001000
E	69	1000101
L	76	1001100
L	76	1001100
O	79	1001111

Key Char	ASCII	Binary
M	77	1001101
Y	89	1011001
K	75	1001011
E	69	1000101
Y	89	1011001

Continues...

Step 2: XOR Each Letter

Plaintext	Binary	Key	Binary	XOR	Result (Decimal)
H	1001000	M	1001101	101	5
E	1000101	Y	1011001	11100	28
L	1001100	K	1001011	111	7
L	1001100	E	1000101	1001	9
O	1001111	Y	1011001	10110	22

Continues...

Step 3: Ciphertext Output

[5, 28, 7, 9, 22]

Cipher text (in binary)

00000101 00011100 00000111 00001001 00010110

Step 4: Decryption

Plaintext=Ciphertext ⊕ Key

Plaintext :: **“HELLO”**

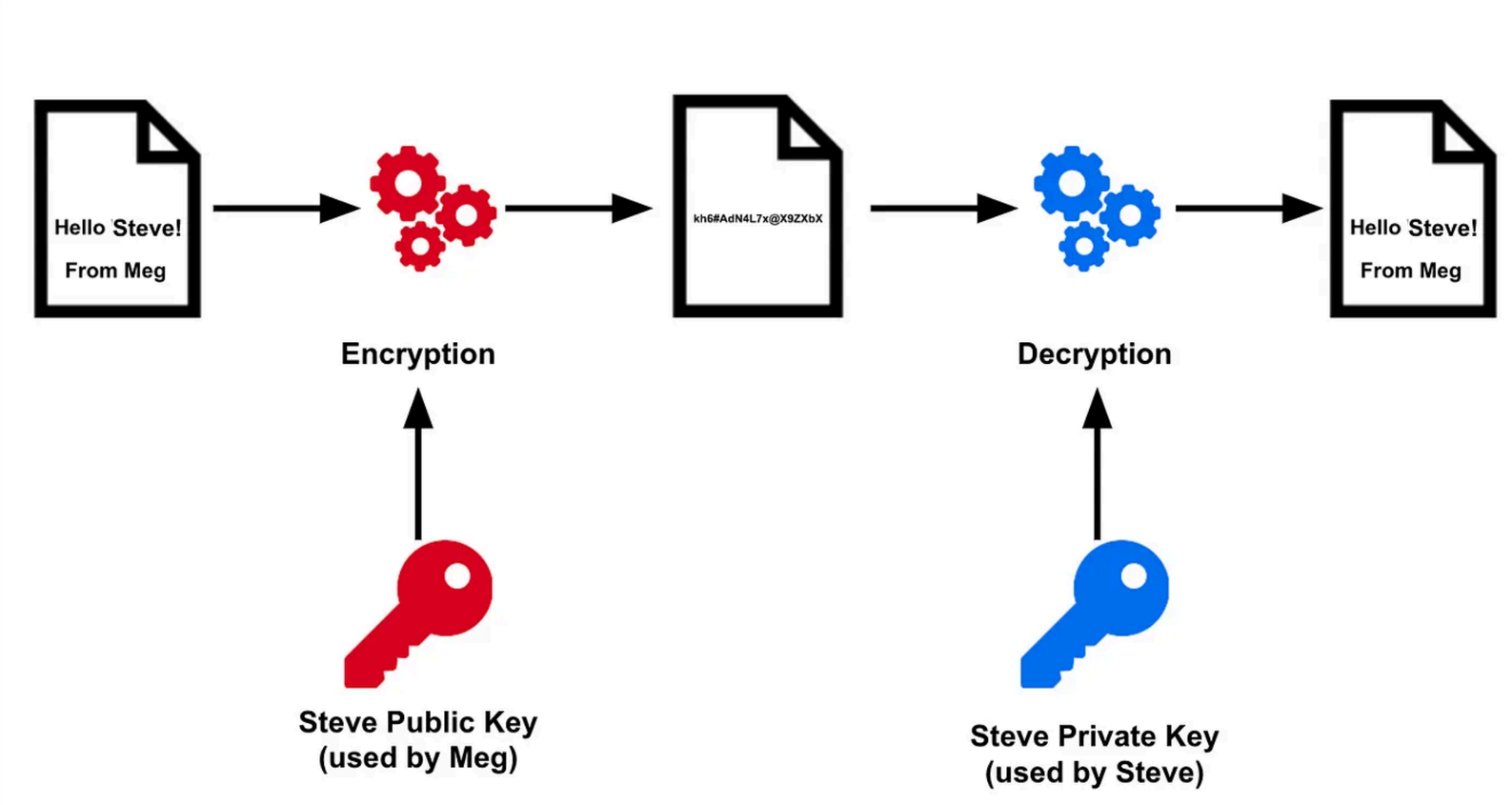
Asymmetric Cryptography In Protocols

Jayavarshini S S (22z227)

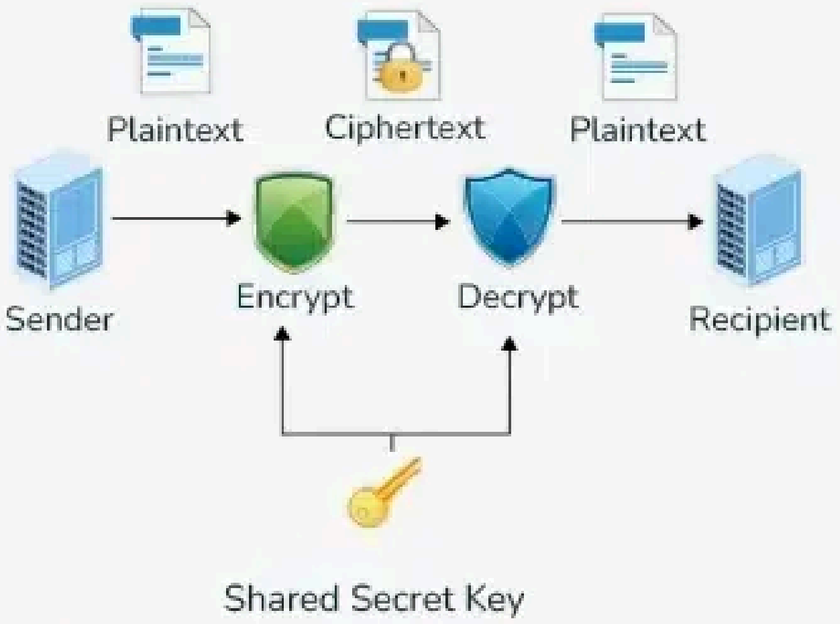
Asymmetric Key Cryptography

- Asymmetric cryptography (also called **public-key cryptography**) is a class of cryptographic systems that uses a pair of keys for its operations:
  - **a public key** (which can be distributed openly)
  - **a private key** (which must be kept secret by its owner).
- The public key is used for encryption or verification; the private key is used for decryption or signing.

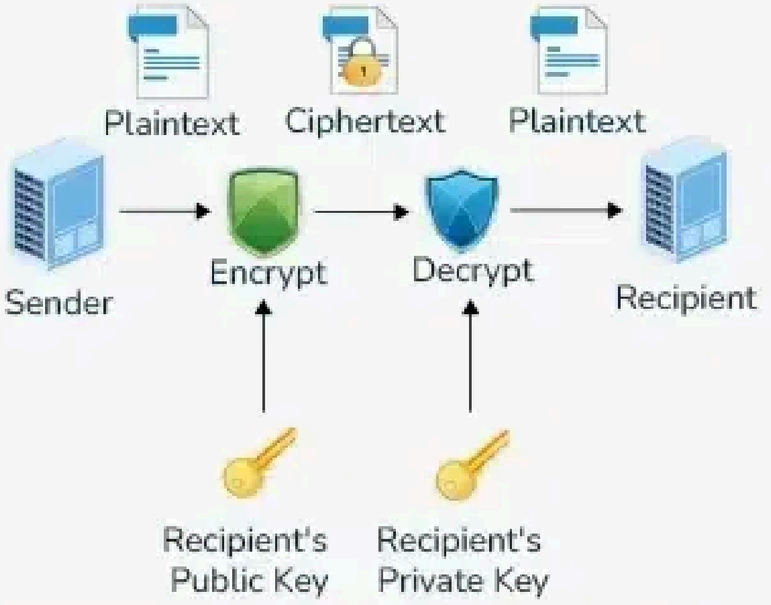
Working


Hello Steve!
From Meg
kh6#AdN4L7x@X9ZXbX
Hello Steve!
From Meg
Encryption
Decryption
Steve Public Key
(used by Meg)
Steve Private Key
(used by Steve)

Difference Between Symmetric and Asymmetric Key Encryption



Same key is used to encrypt & decrypt message

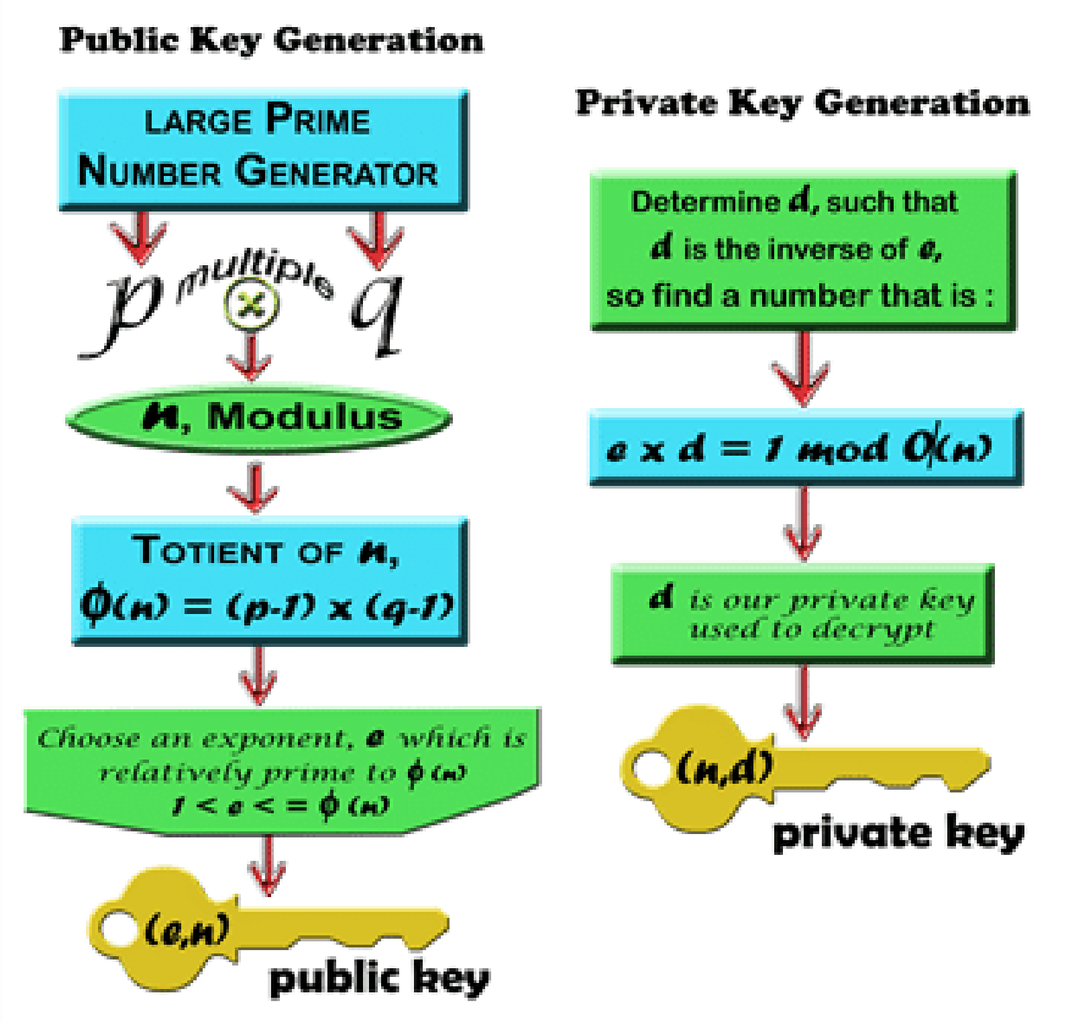


Different keys are used to encrypt & decrypt message

RSA (Rivest–Shamir–Adleman)

It is commonly utilized to ensure secure communication and for creating digital signatures. It uses large integer prime numbers for key generation. It Encrypts data with the public key and decrypts with the private key. It is Slower than some other algorithms but offers strong security.

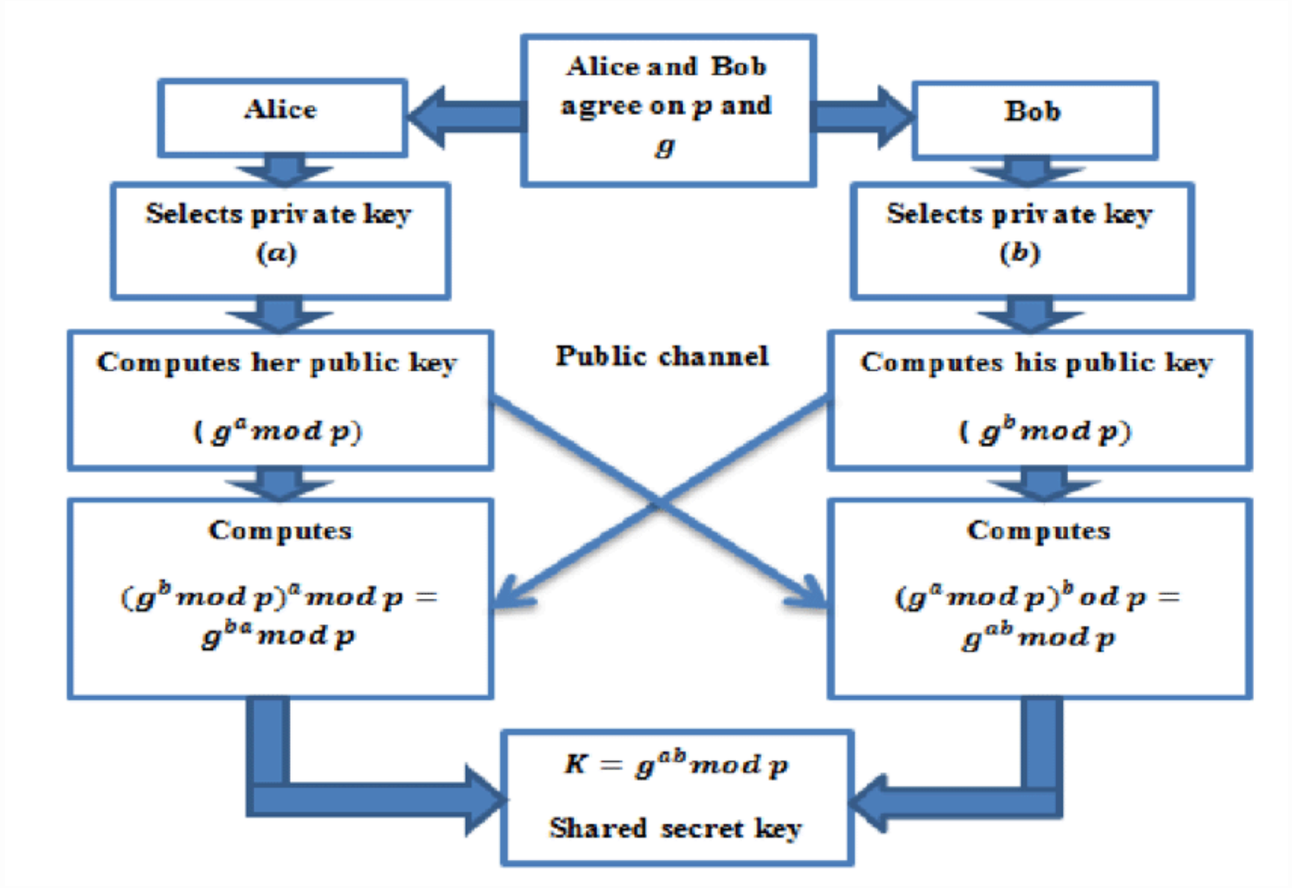
Key Generation



Diffie-Hellman Key Exchange

It doesn't directly encrypt data but establishes a shared secret key for secure communication. Two parties can generate a common secret key without ever exchanging it directly. It often used in conjunction with other algorithms like RSA for key exchange.

Diffie-Hellman Key Exchange



Advantages

- Enhanced Security
- Secure Key Distribution
- Digital Signatures and Authentication
- Non-Repudiation
- Scalability
- Used in Hybrid Systems

Authentication & Integrity

Hemanthkumar V (22z225)

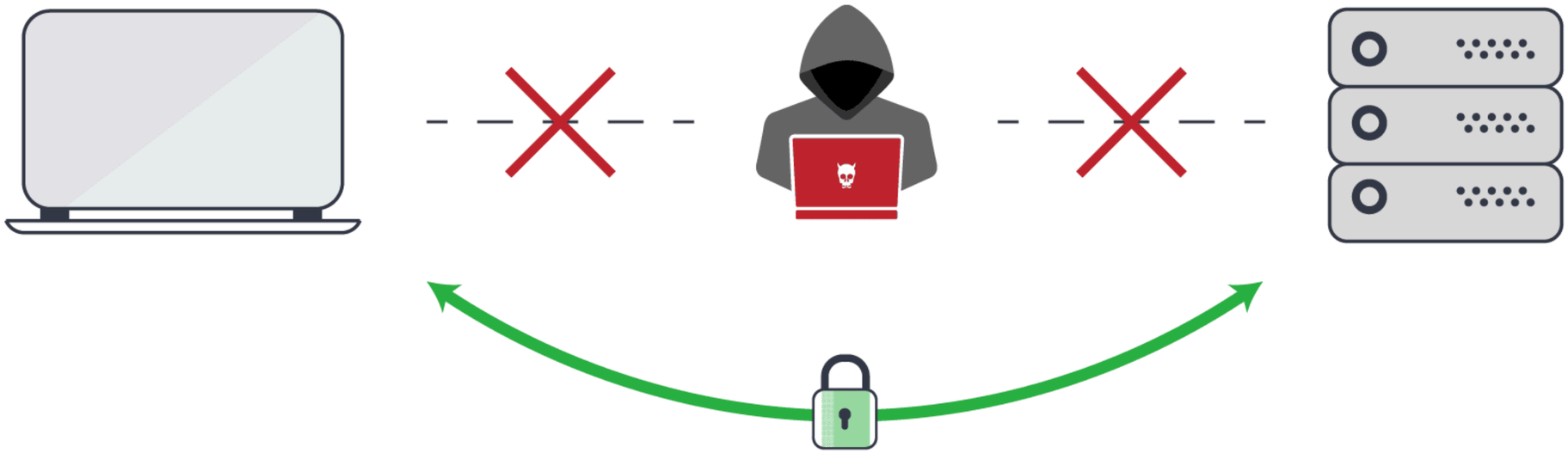
Authenticity

- Authentication ensures that the message is truly from the sender it claims to be.

Integrity

- Integrity ensures that the message hasn’t been altered during transmission.

Encryption** alone doesn't prove **who sent** the data or if it **was changed



MITM violates...

- Authentication → because the receiver cannot confirm the true identity of the sender (attacker can impersonate them)
- Integrity → because the message could be modified in transit (attacker changes its contents).

Digital Signatures

- A digital signature is a cryptographic method that verifies the sender's identity and ensures a document's integrity using public key cryptography.

- It provides authenticity, security, and is legally binding in many countries.

How does digital signatures work?

Key Generation

- A private key (kept secret) and a public key (shared) are created.

Hashing

- The document's hash is encrypted with the private key to form the digital signature.

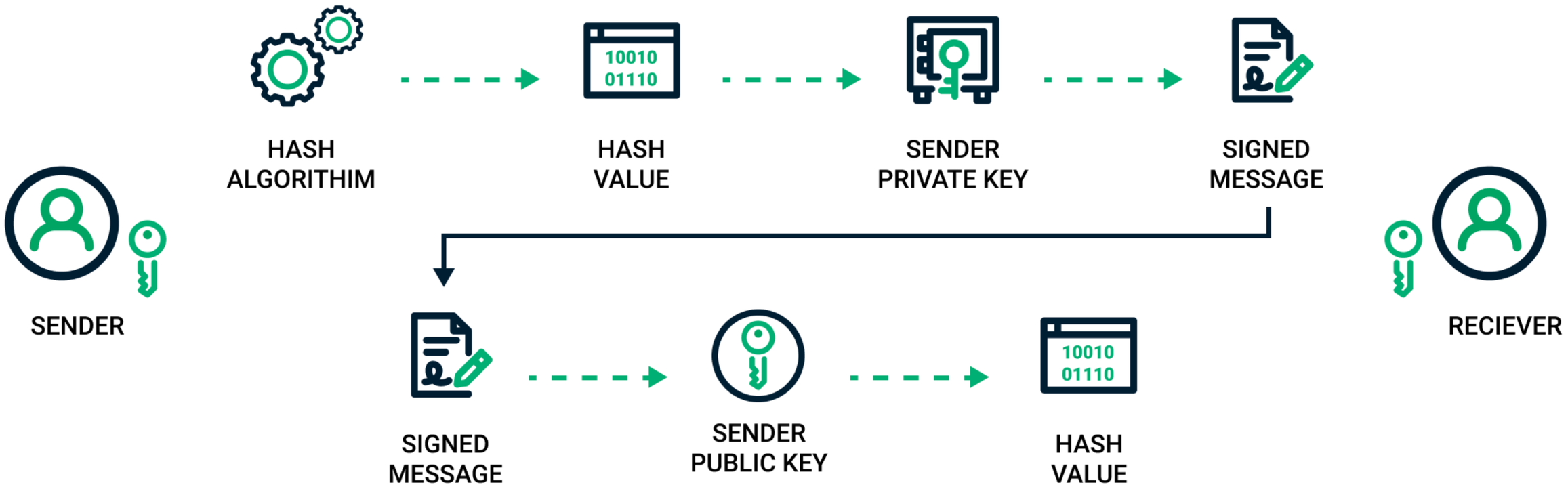
Verification

- The recipient uses the public key to decrypt the signature and generate a new hash.

Validation

- If both hashes match, authenticity and integrity are confirmed.

How does digital signatures work?



Digital Signature Demo

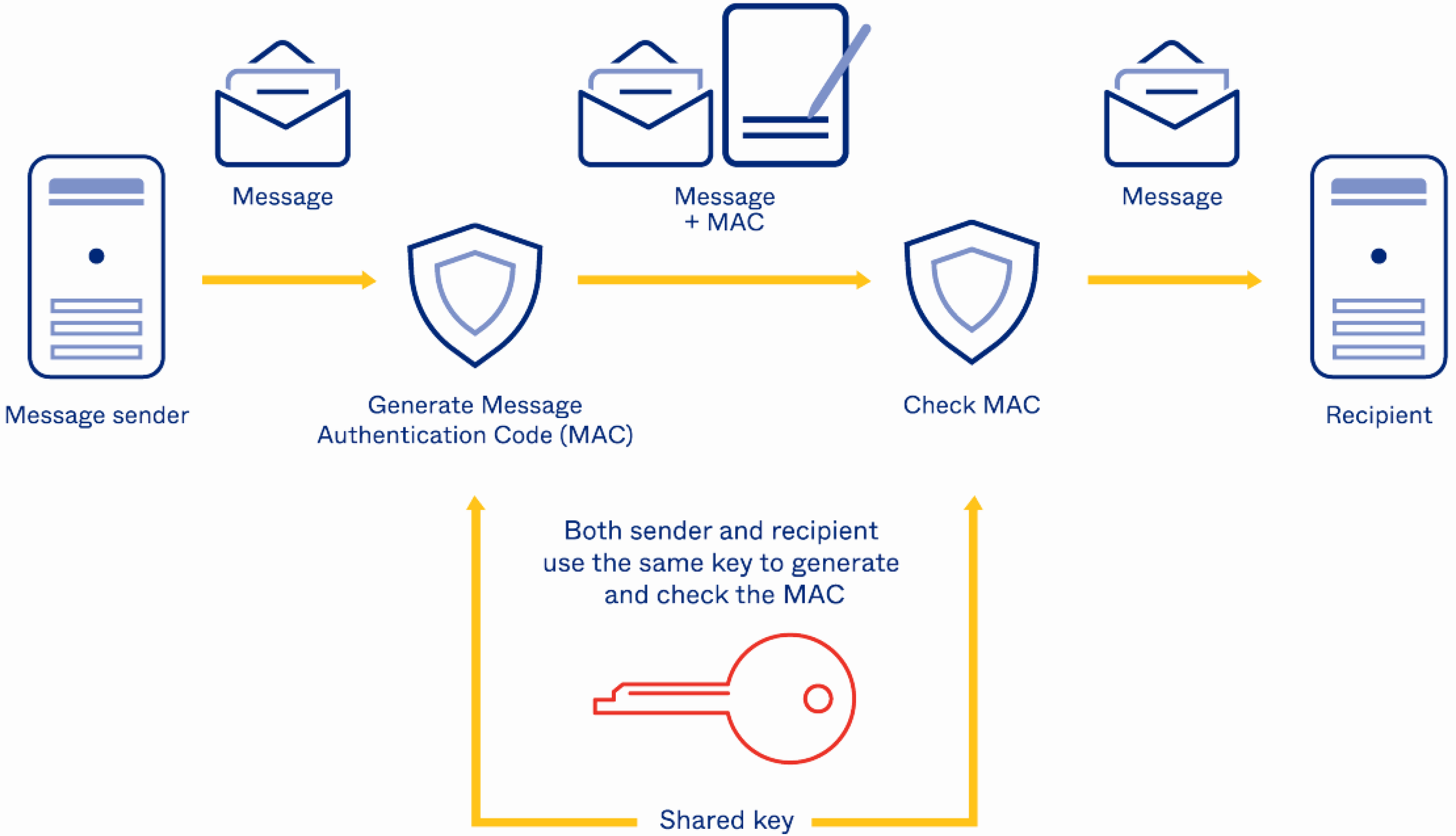
https://crypto-digital-sign-demo.streamlit.app/

HMAC (Hash-based Message Authentication Code)

- A cryptographic method that combines a secret key with a hash function to ensure message integrity and authenticity.
- The sender hashes the message with the key and sends the HMAC; the receiver recomputes it—if both match, the message is verified and untampered.

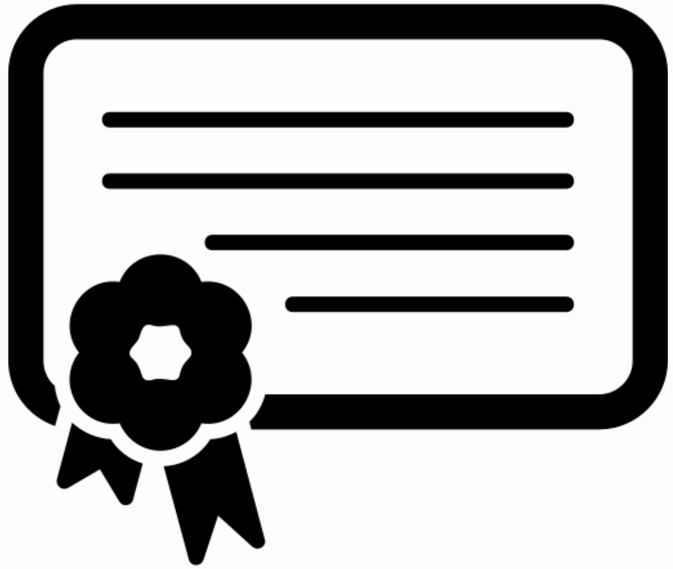
How does HMAC work?

- **Shared Key:** A secret key is shared between sender and receiver.
- **Hashing:** The sender hashes the message with the secret key to create the HMAC.
- **Transmission:** The message and HMAC are sent to the receiver.
- **Verification:** The receiver recalculates the HMAC using the same key.
- **Validation:** If both HMACs match, the message is authentic and unaltered.

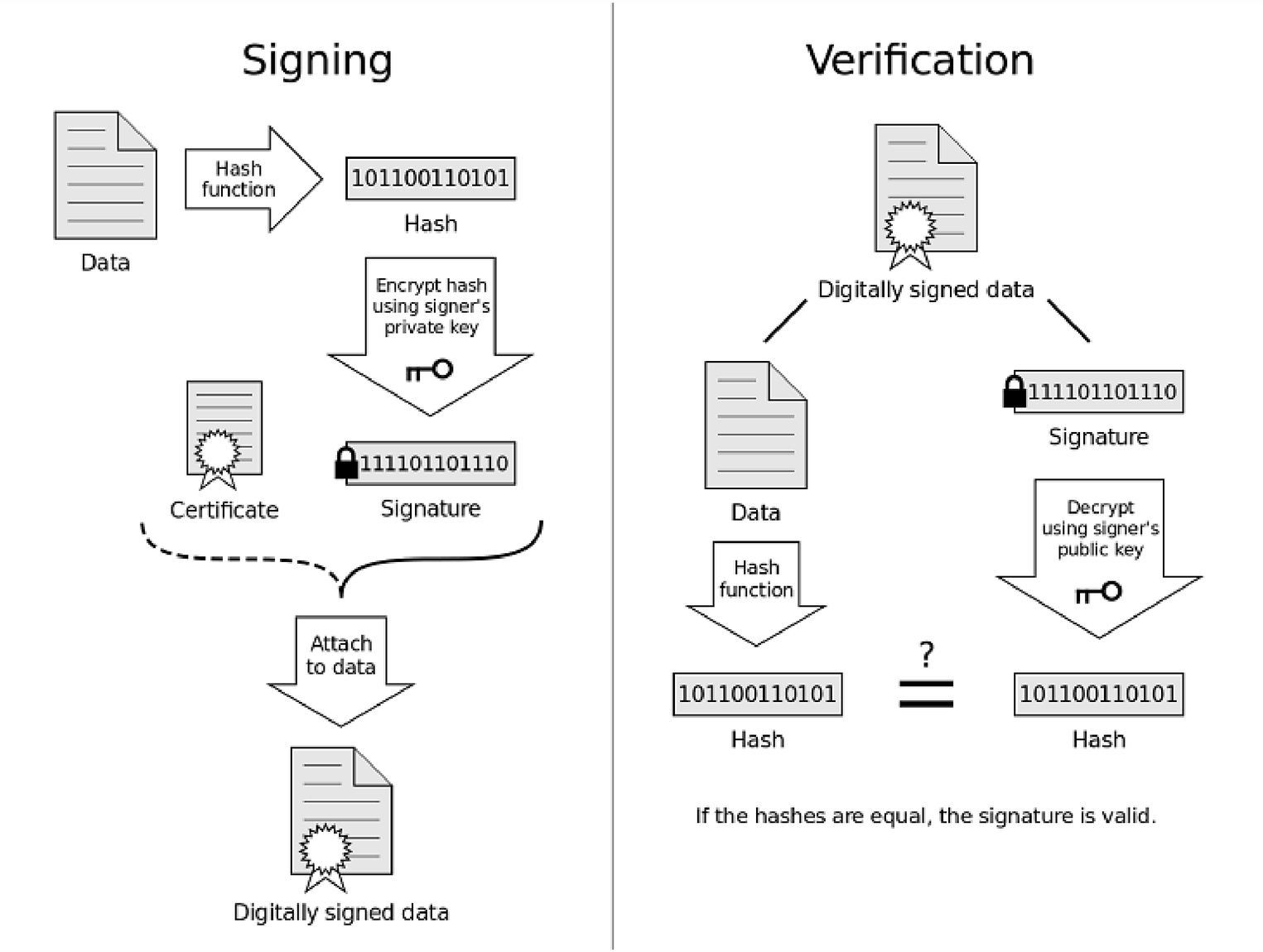

Message
Message
+ MAC
Message
Message sender
Generate Message
Authentication Code (MAC)
Check MAC
Recipient
Both sender and recipient
use the same key to generate
and check the MAC
Shared key

Digital Certificates

- An electronic credential that authenticates the identity of a user, device, or server using public key cryptography.
- Issued by trusted ***Certificate Authorities (CAs)*** and contain information like a public key, ownership details, and a digital signature from the CA.



How does digital certificates work?



Secure Communication Protocols – TLS/SSL, HTTPS, VPN

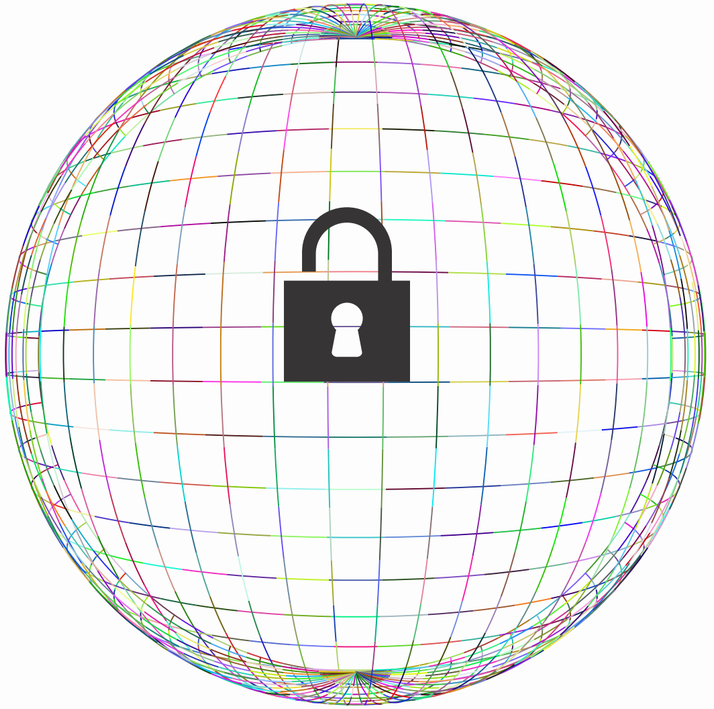
Manojkumar K (22z236)

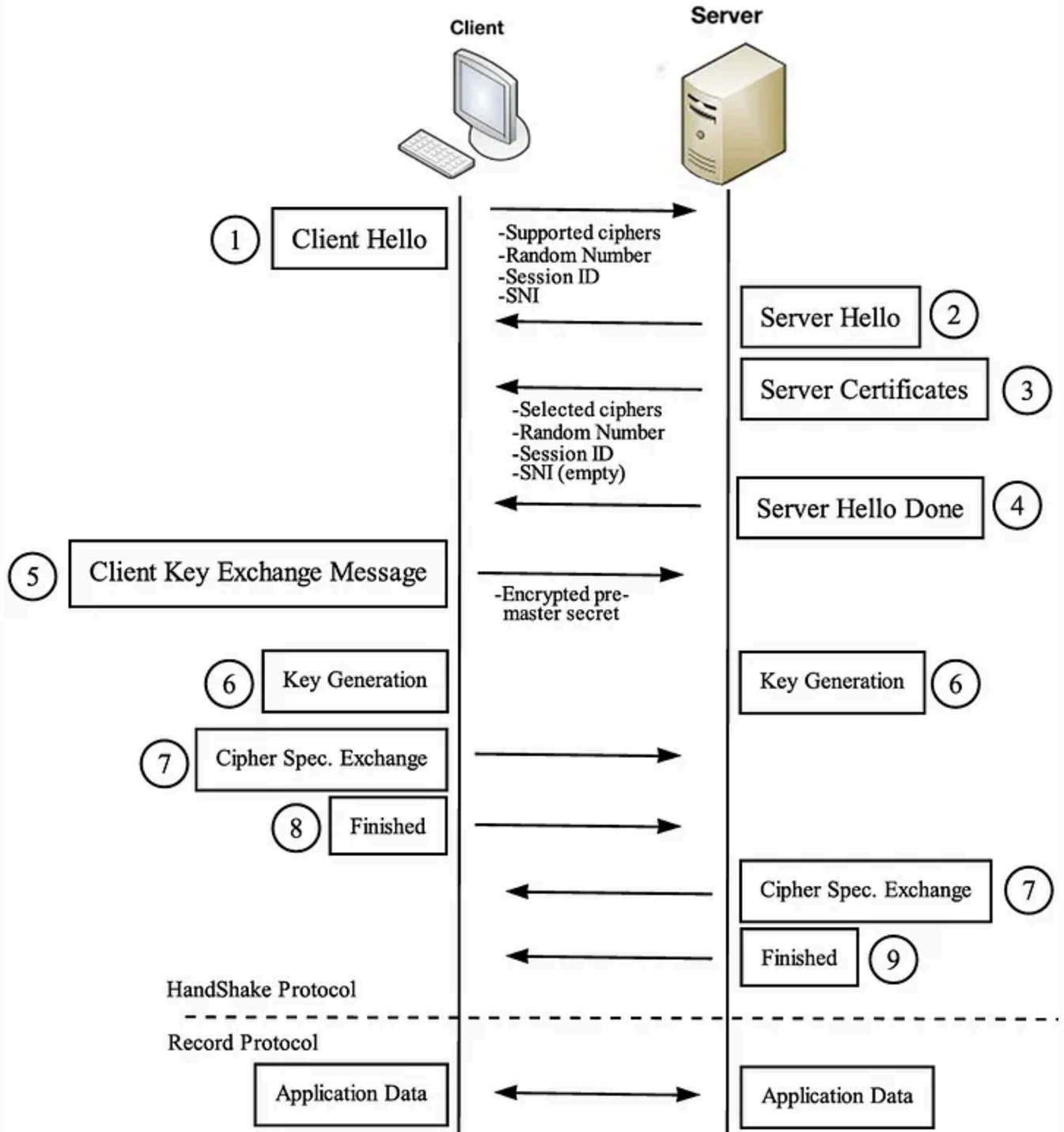
What are Secure Communication Protocols?

Ensure safe data transfer between client and server

Protect against:

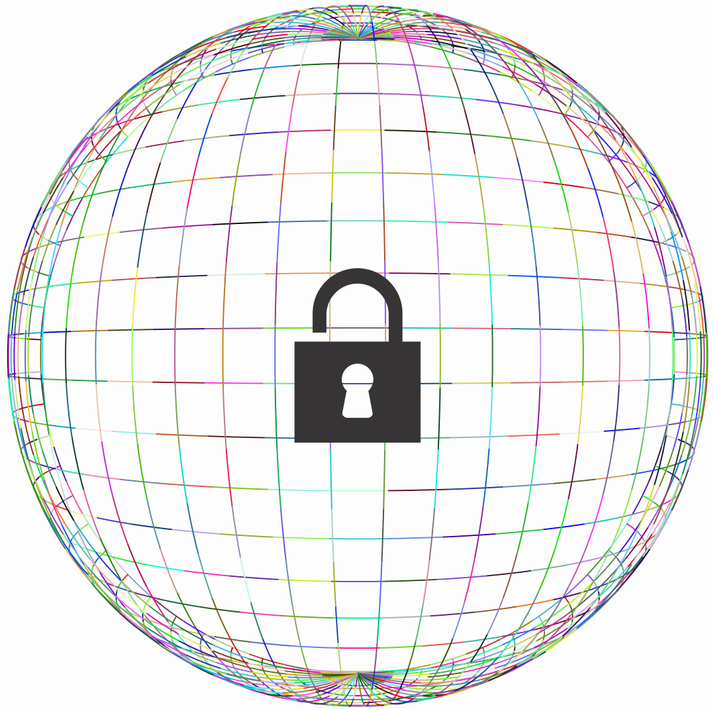
- Eavesdropping
- Data Tampering
- Identity Spoofing

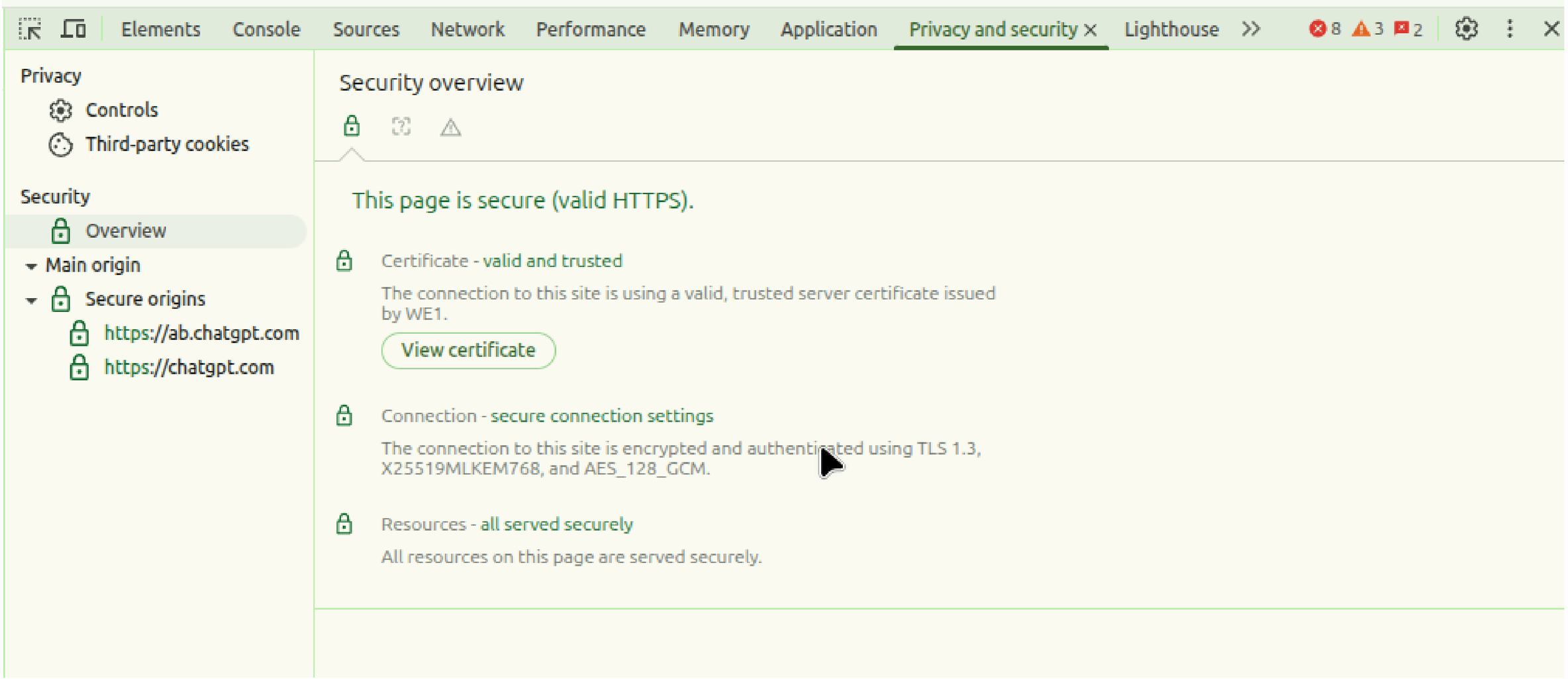


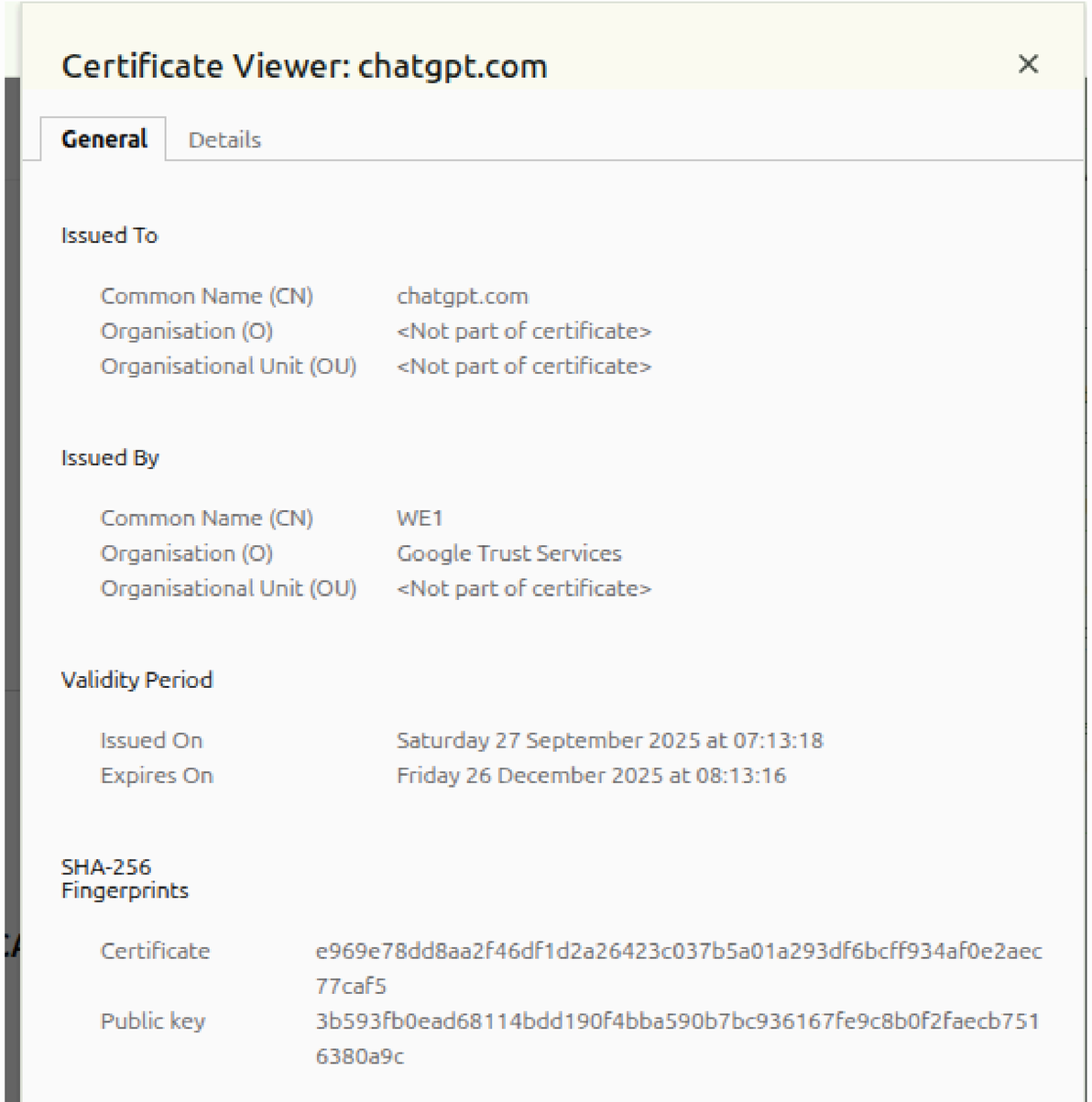

Client
Server
1
Client Hello
-Supported ciphers
-Random Number
-Session ID
-SNI
Server Hello
2
Server Certificates
3
-Selected ciphers
-Random Number
-Session ID
-SNI (empty)
Server Hello Done
4
5
Client Key Exchange Message
-Encrypted pre-master secret
6
Key Generation
Key Generation
6
7
Cipher Spec. Exchange
8
Finished
Cipher Spec. Exchange
7
Finished
9
HandShake Protocol
Record Protocol
Application Data
Application Data

HTTPS (HTTP + TLS)

- HTTPS = HTTP + Security Layer (TLS)
- Ensures all web requests and responses are encrypted
- Uses digital certificates issued by trusted Certificate Authorities (CAs)
- Modern browsers show lock symbol for HTTPS websites



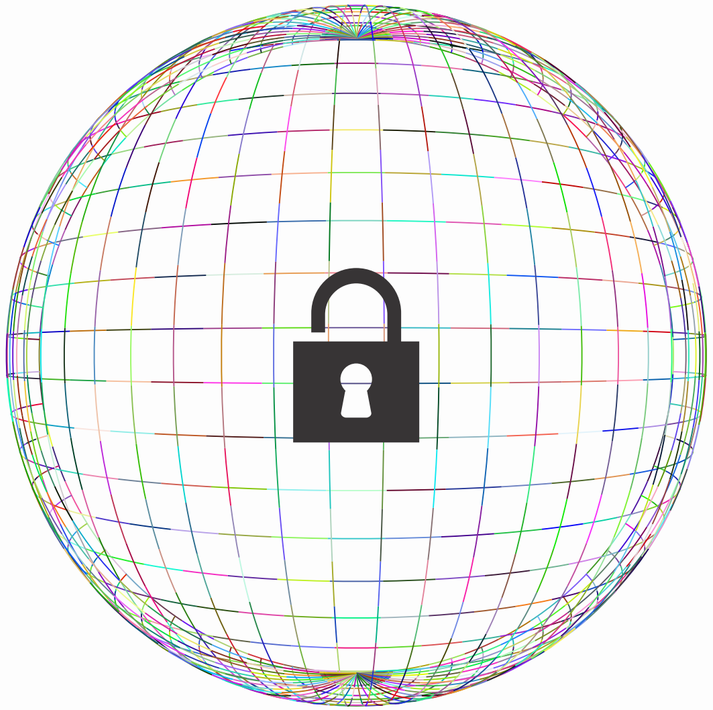

Elements
Console
Sources
Network
Performance
Memory
Application
Privacy and security
Lighthouse
8
3
2
Privacy
Controls
Third-party cookies
Security
Overview
Main origin
Secure origins
https://ab.chatgpt.com
https://chatgpt.com
Security overview
This page is secure (valid HTTPS).
Certificate - valid and trusted
The connection to this site is using a valid, trusted server certificate issued by WE1.
View certificate
Connection - secure connection settings
The connection to this site is encrypted and authenticated using TLS 1.3, X25519MLKEM768, and AES_128_GCM.
Resources - all served securely
All resources on this page are served securely.

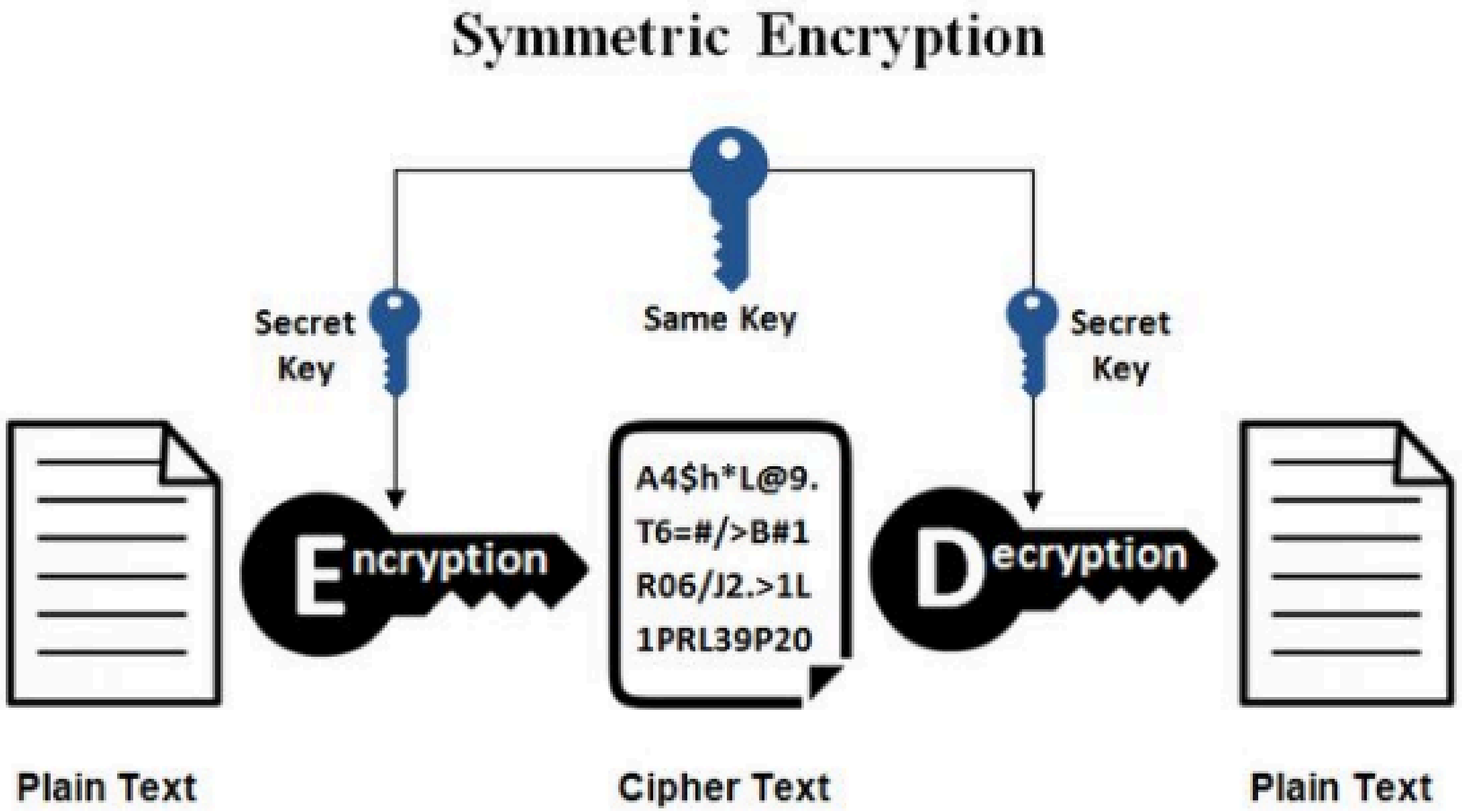

Certificate Viewer: chatgpt.com
General
Details
Issued To
Common Name (CN) chatgpt.com
Organisation (O) <Not part of certificate>
Organisational Unit (OU) <Not part of certificate>
Issued By
Common Name (CN) WE1
Organisation (O) Google Trust Services
Organisational Unit (OU) <Not part of certificate>
Validity Period
Issued On Saturday 27 September 2025 at 07:13:18
Expires On Friday 26 December 2025 at 08:13:16
SHA-256 Fingerprints
Certificate e969e78dd8aa2f46df1d2a26423c037b5a01a293df6bcff934af0e2aec77caf5
Public key 3b593fb0ead68114bdd190f4bba590b7bc936167fe9c8b0f2faecb7516380a9c

VPN (Virtual Private Network)

- Creates an encrypted tunnel between your device and VPN server
- Protects all internet traffic, not just websites
- Common uses:
  - Secure access for remote employees
  - Protecting data on public Wi-Fi
  - Bypassing geo-blocks or restricted content
- Example:

  Company VPN → secure internal communication





Demonstrating and Analysing the TLS Handshake Using Wireshark

Introduction & Background

Medium / Aug 13, 2024

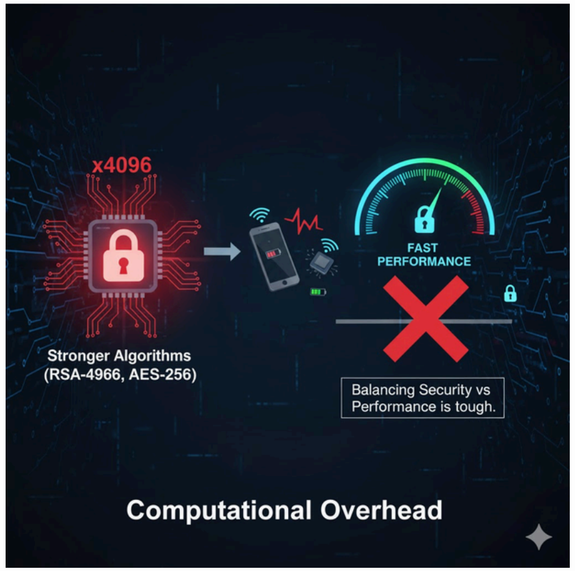
Challenges and Emerging Trends

Monish Rajan L (22z240)

Current Challenges in Security Protocols

a. Key Management Complexity

- Generating, distributing, storing, and revoking keys securely is a major challenge.
- Especially in large-scale systems, IoT, and cloud environments.



b. Computational Overhead

- Stronger algorithms (like RSA-4096 or AES-256) consume more processing power, impacting low-power or mobile devices.
- Balancing security vs performance is tough.

Current Challenges in Security Protocols

c. Quantum Threat

- Quantum computers can break RSA and ECC using Shor's algorithm.
- Forces migration to Post-Quantum Cryptography (PQC).

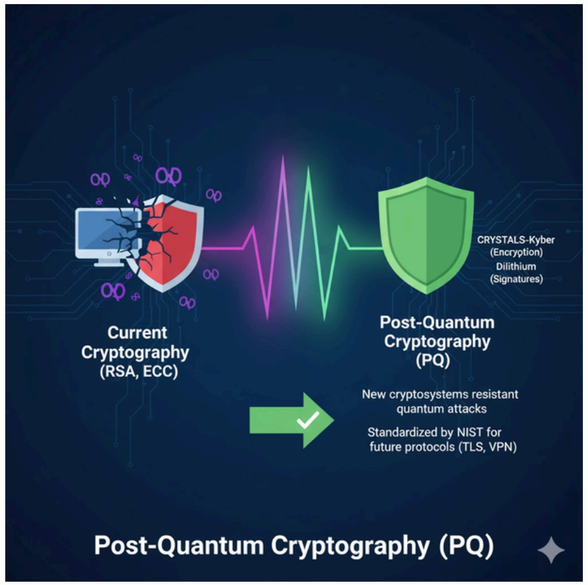
d. Human & Implementation Errors

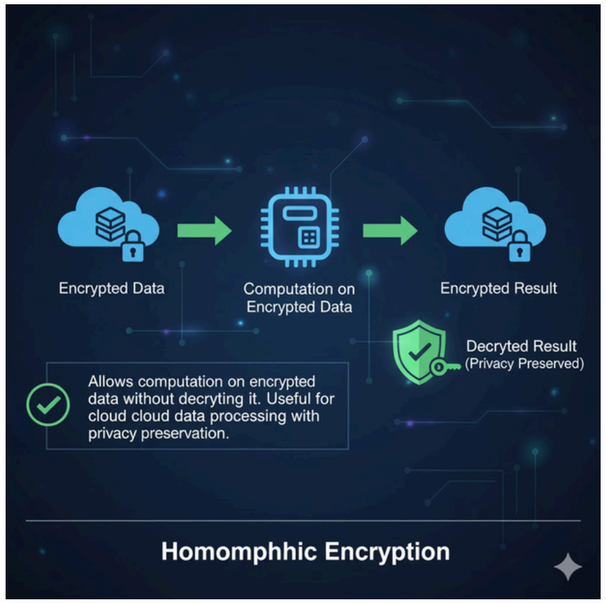
- Misconfigured certificates, weak passwords, and bad randomness lead to vulnerabilities even with strong cryptography.

Emerging Trends & Future Directions

a. Post-Quantum Cryptography (PQC)

- New cryptosystems resistant to quantum attacks.
- Examples: CRYSTALS-Kyber (encryption), Dilithium (signatures).
- Being standardized by NIST for future protocols (TLS, VPN, etc.).





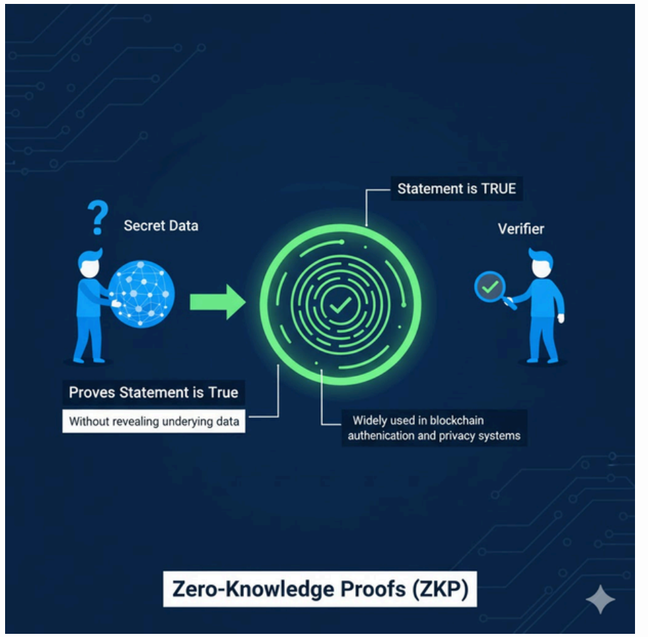
b. Homomorphic Encryption

- Allows computation on encrypted data without decrypting it.
- Useful for cloud data processing with privacy preservation.

Emerging Trends & Future Directions

c. Zero-Knowledge Proofs (ZKP)

- Proves a statement is true without revealing underlying data.
- Widely used in blockchain authentication and privacy systems.



d. Lightweight Cryptography

- Designed for IoT and embedded devices with limited memory and power.
- Example: Ascon, TinyAES, PRESENT.

About Shor’s Algorithm

- **Classical bottleneck**: Finding the period r is hard classically for large N.
- **Quantum trick**: Use a quantum register in superposition of many inputs, compute the modular function in superposition, then apply the Quantum Fourier Transform (QFT).
- **Interference**: QFT causes constructive interference at frequencies corresponding to the period r. Measuring the quantum state gives information that, with high probability, lets you deduce r (using continued fractions to turn measured frequency into r).
- After r is known, the rest is classical (gcd via Euclid).