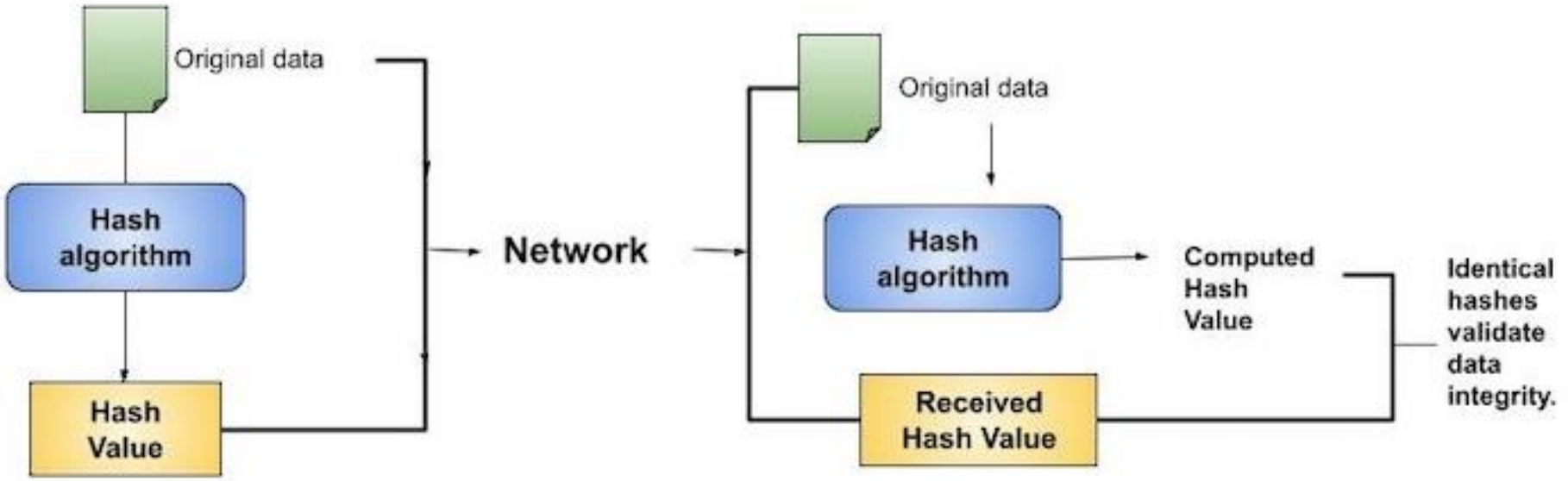
Cryptograph

Hashing and Encryption

Hashing Algorithm

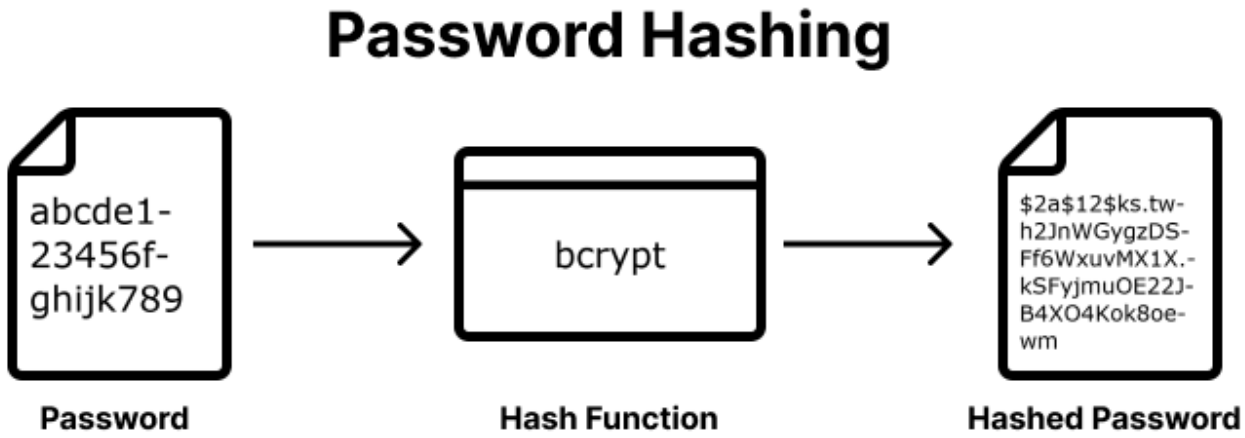
- A hashing algorithm is a mathematical function that takes input data of any size and generates a fixed-size output, known as a hash value or digest.
- The key characteristic of a hashing function is that it is a one-way function, meaning it is:
  - Deterministic: Given the same input, it always produces the same output.
  - Non-invertible: It is computationally infeasible to reverse-engineer the original input from the output hash value.
- Demo: https://guggero.github.io/blockchain-demo/#!/hash



Hashing For Data Integrity

- Data Integrity: Hashing ensures that data is not tampered with or altered during transmission or storage.
- Software Download: Apache Web server: https://httpd.apache.org/download.cgi

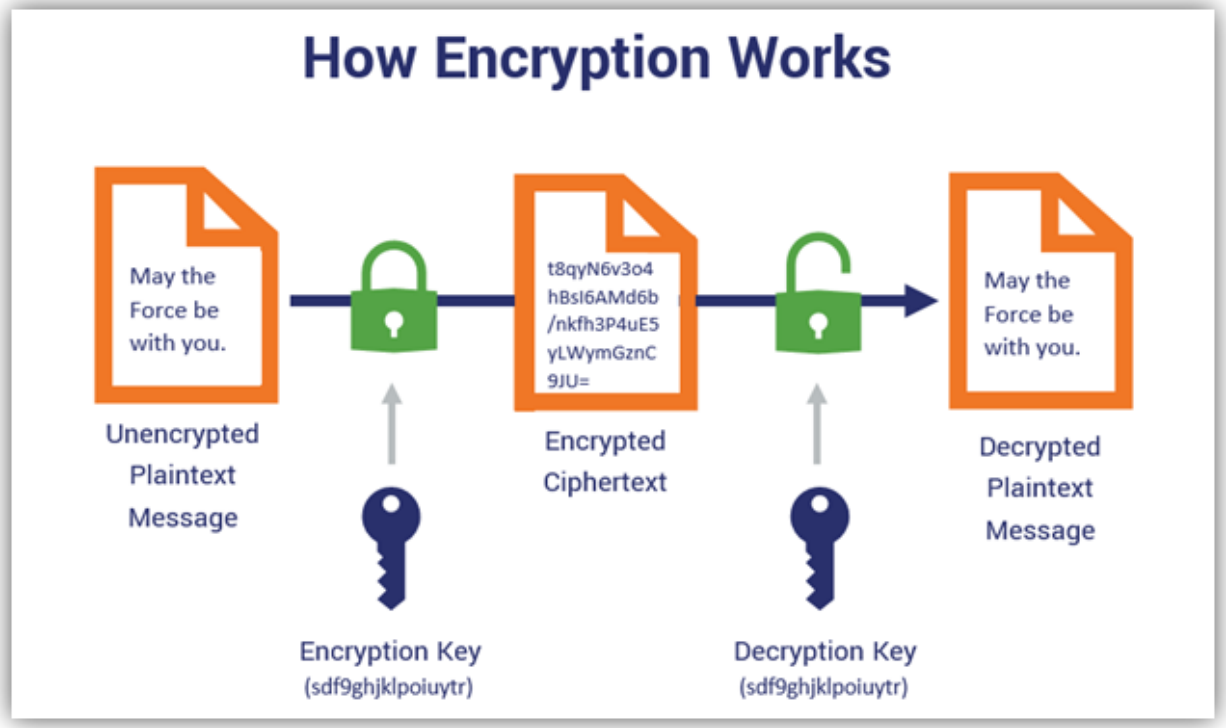
Hashing for Password Storage



- Password Storage: Hashing is used to store passwords securely, making it difficult for attackers to obtain the original password.
- When a user creates a password, a hash value is generated and stored instead of the actual password. When the user logs in, the input password is hashed and compared with the stored hash value to authenticate the user.
- This way, even if an attacker gains access to the stored hash values, they cannot obtain the original passwords.
- bcrypt in the above picture is a popular password hashing function because it allows the computational cost to be increased as computational power grows, enhancing security.

Symmetric Encryption

- Symmetric encryption is a type of cryptography that uses the same secret key for both encryption and decryption.
- This means that the key used to lock the information is the same key used to unlock it. Symmetric encryption algorithms are fast, efficient, and widely used in various applications, including secure web browsing, email encryption, and digital signatures.



Symmetric Encryption Algorithms

- Data Encryption Standard (DES): introduced in 1976, DES was the first widely adopted symmetric encryption algorithm. DES uses a 56-bit key and operates on 64-bit blocks. Its short key length and vulnerability to brute-force attacks led to its eventual deprecation.
- Triple Data Encryption Algorithm (Triple DES): introduced in the late 1990s, aimed to address DES's security concerns. It uses three separate 56-bit keys, effectively increasing the key length to 168 bits.
- Advanced Encryption Standard (AES): introduced in 2001, revolutionized symmetric encryption. AES uses a variable key length (128, 192, or 256 bits) and operates on 128-bit blocks.

Symmetric Encryption Drawbacks

- Key distribution: A major challenge with symmetric encryption is securely distributing the secret key to all parties involved. If the key is intercepted during transmission, the security of the encrypted data is compromised.
- Key Management: Managing the keys becomes increasingly complex as the number of users grows. Each pair of users needs a unique key, leading to many keys that must be securely stored and managed.
- Lack of Non-repudiation: Symmetric encryption does not provide non-repudiation. Since the same key is used for both encryption and decryption, it is impossible to prove which party encrypted the message, making it unsuitable for scenarios where proof of origin is required.

Key Sharing Among 10 People

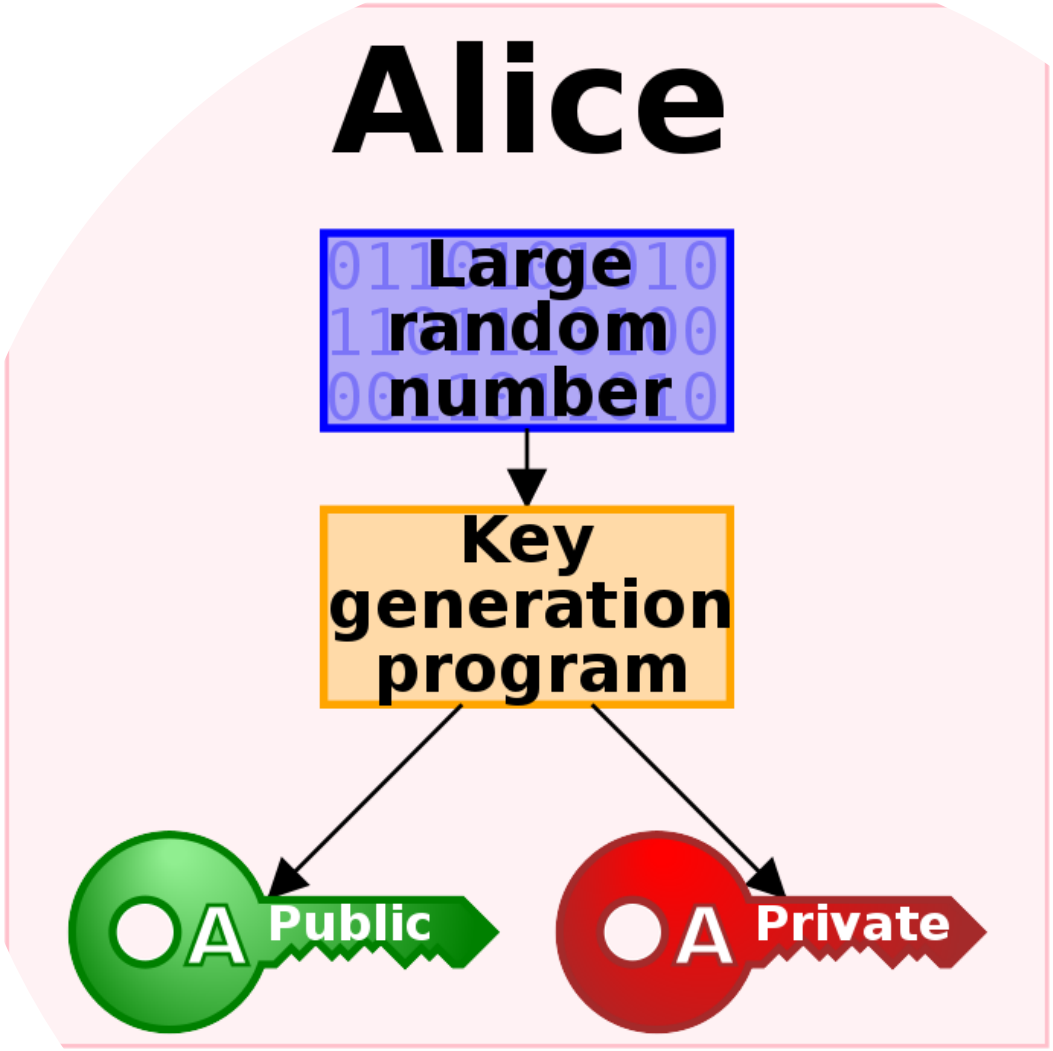
- Meet in person: This would require each person to meet with every other person individually, resulting in 45 separate meetings (10 people x 9 others = 90, but since each meeting involves two people, we divide by 2).
- Share via email or messaging: This would require each person to send the key to every other person, resulting in 45 separate key transmissions However, this method is insecure since the keys could be intercepted or compromised during transmission.
- Use a central key server: This would require each person to trust the key server and connect to it to retrieve the shared key. However, this creates a single point of failure and a potential target for attacks.

Mission Impossible

- As the number of users increases, the complexity of sharing symmetric keys grows exponentially. In a larger group, managing and securing the key exchange process becomes impractical.
- For example, with 100 users, each person would need to share the key with 99 others, resulting in 4,950 separate key exchanges (100 people x 99 others = 9,900, but since each exchange involves two people, we divide by 2).
- In Internet that has millions of businesses and billions of users, sharing symmetric key is an impossible mission.

Asymmetric Encryption Concepts

- A pair of keys: a public key and a private key.
- It is generated from a large random number.
- You can encrypt data with one key and decrypt the data with another key – but encryption/decryption is very slow compared to symmetric encryption/decryption,



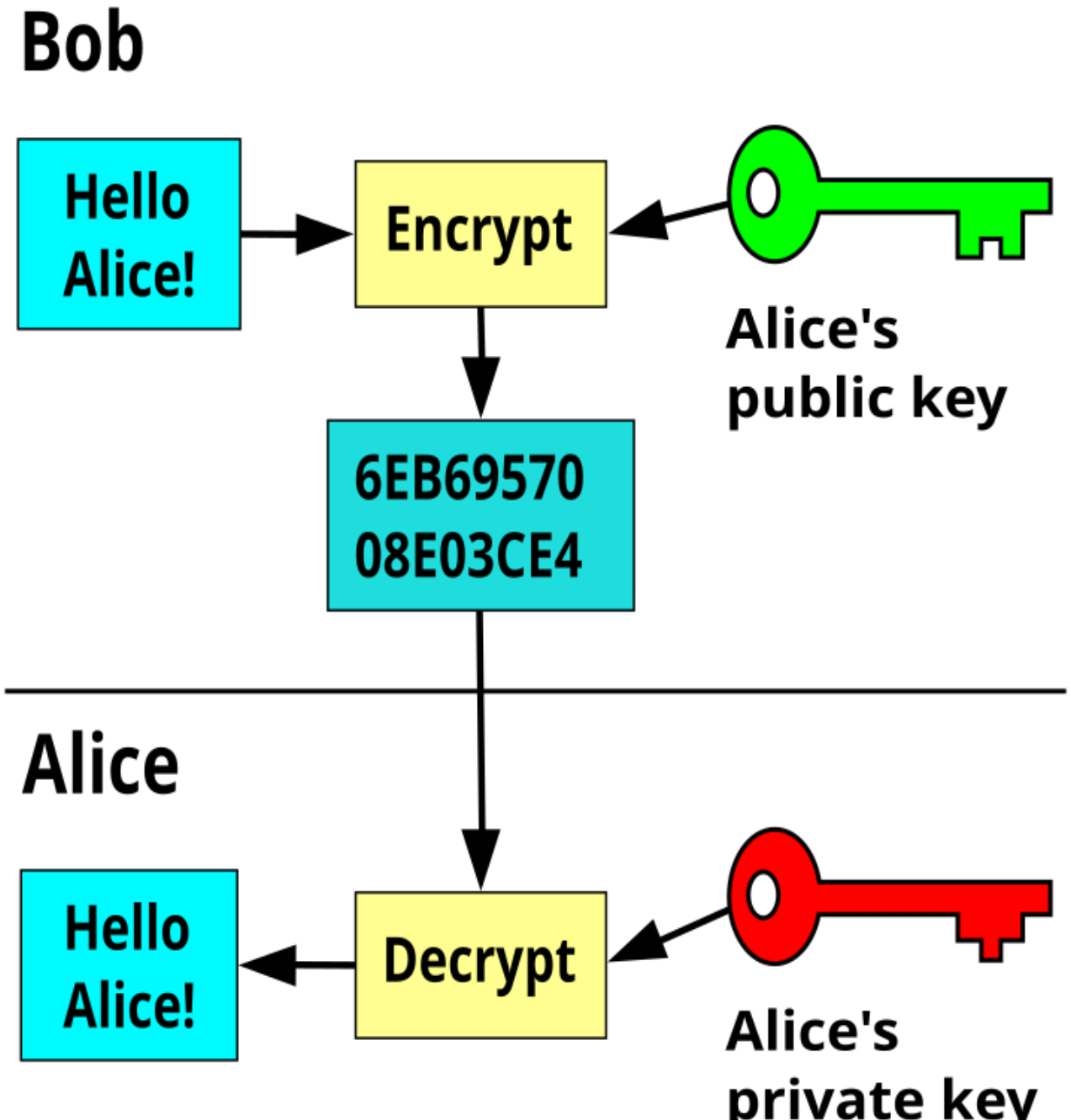
Public Key

- Publicly available.
- Used for encryption/decryption, digital signature, and key exchange.
- Not sensitive, shared openly.
- It is computationally impossible to find out private key from the public key.

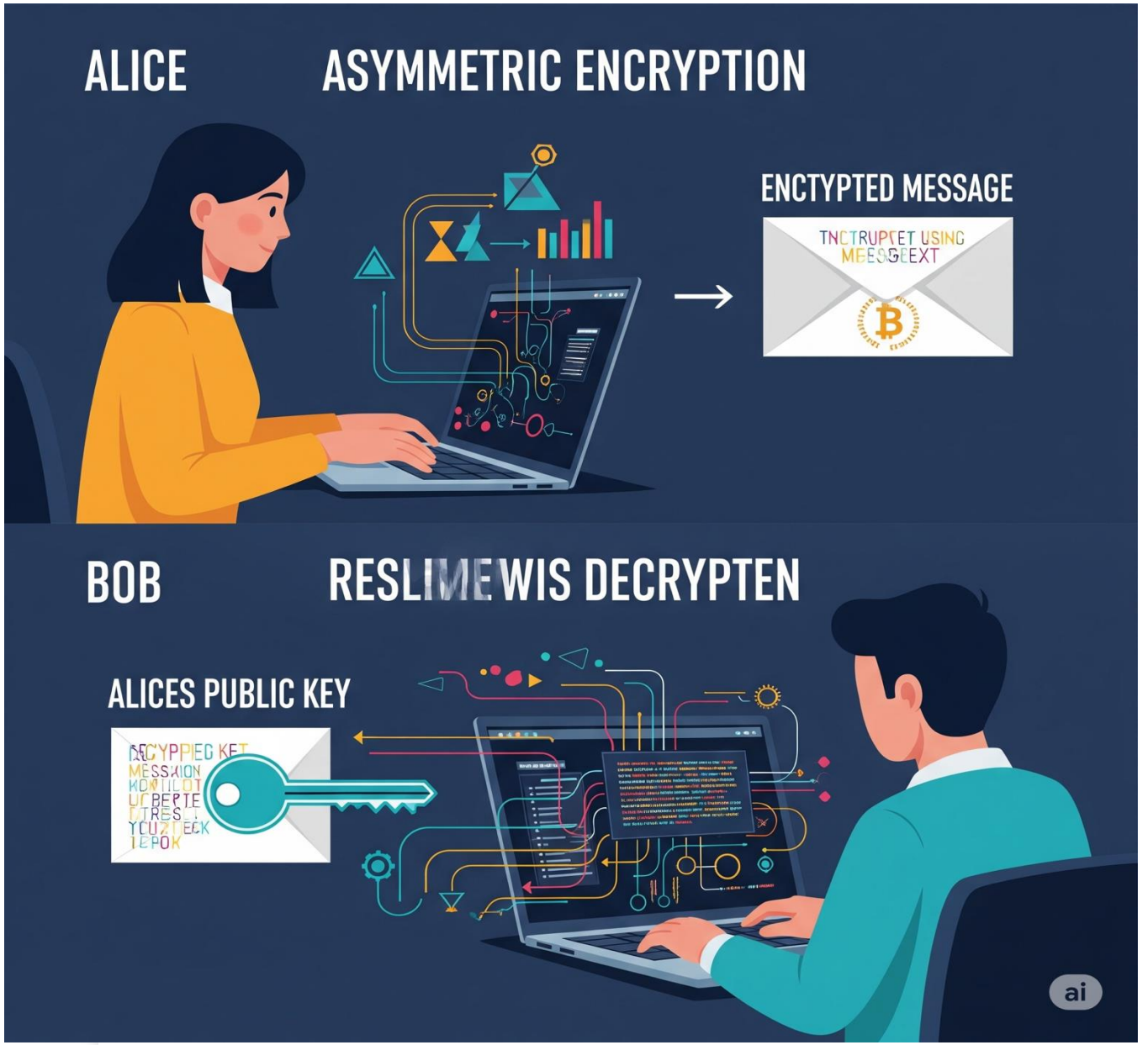
Private Key

- Kept confidential: it is computationally impossible to find out the private key from its public key
- Used for encryption/decryption, digital signature, and key exchange
- Sensitive, not shared publicly

1) Bob encrypt data with Alice’s public key
2) Bob pays bitcoin to Alice’s public key (address)

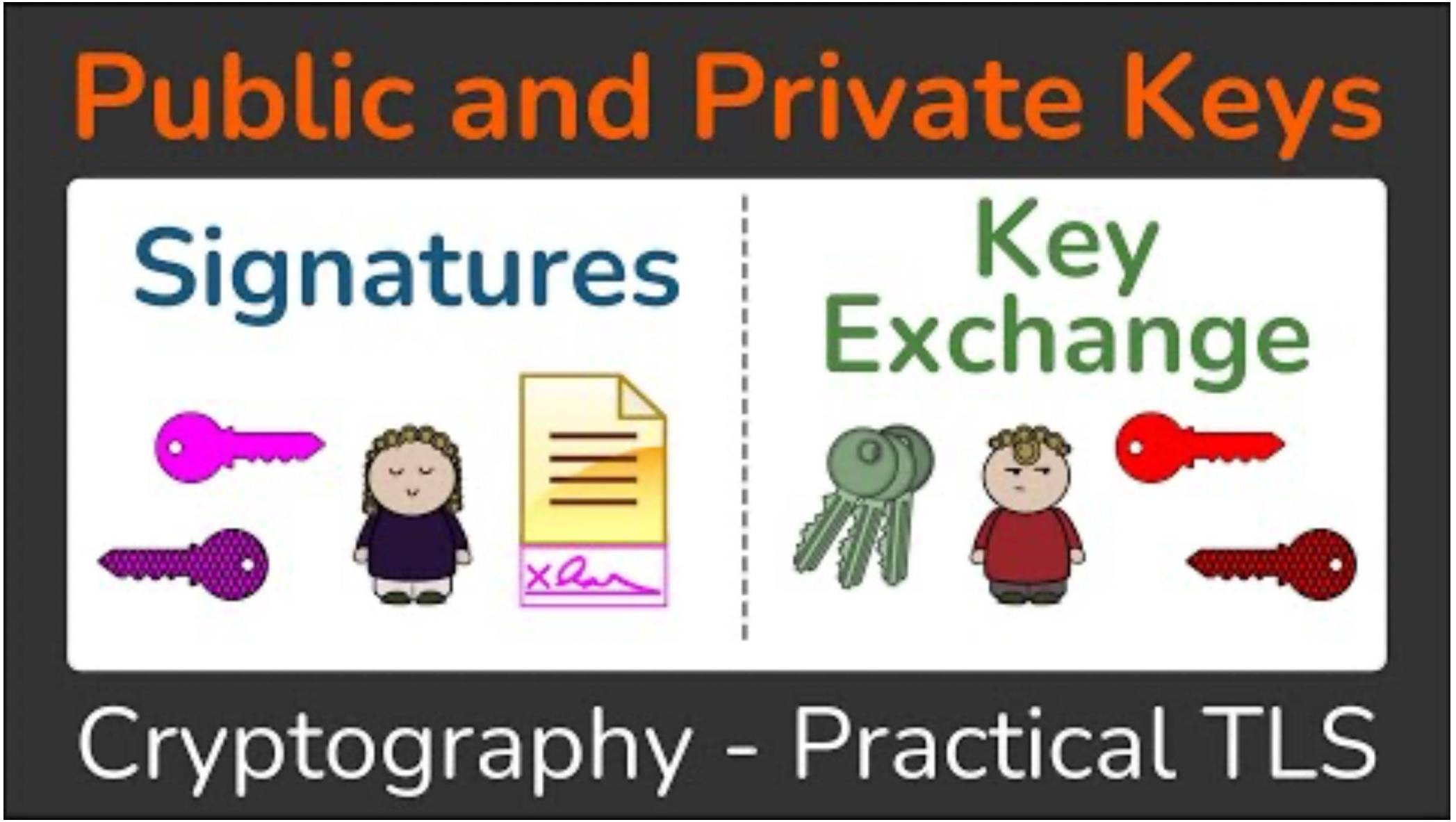


Alice encrypts data with her private Key: everyone can decrypt it. Why do this?



Public Key and Private Key:

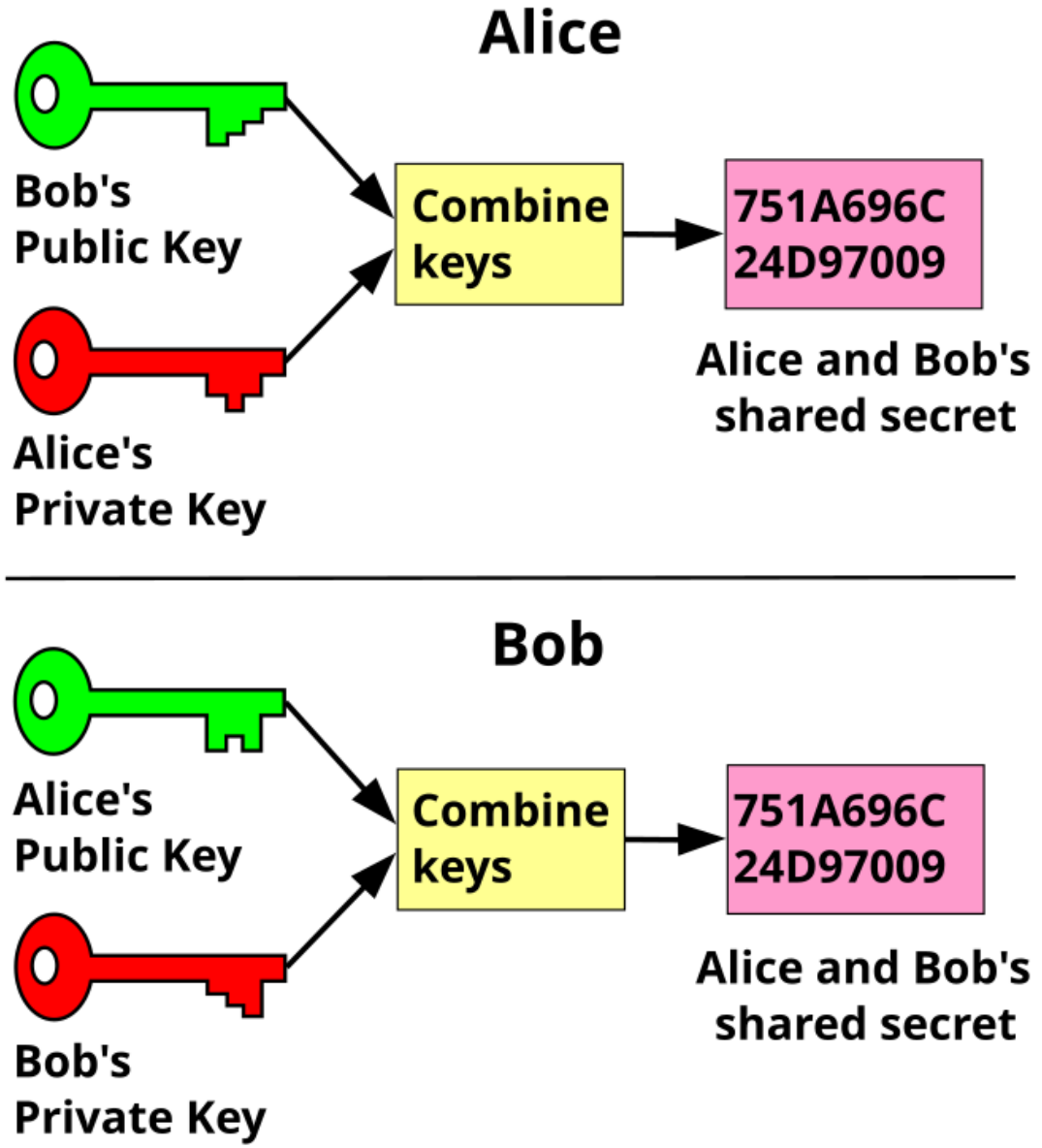
https://www.youtube.com/watch?v=_zyKvPvh80



Public Key for Authentication

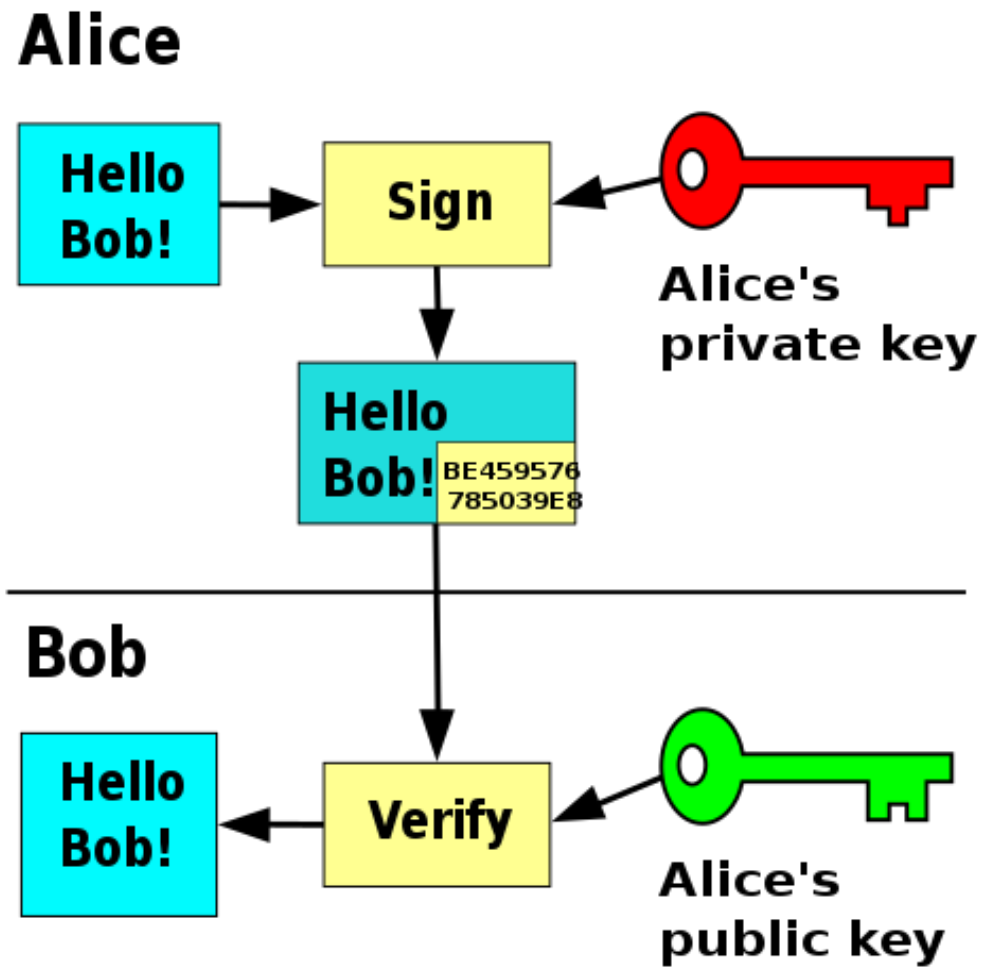
1. **Generate Key Pair:** Create a linked public and private key pair for the user.
2. **Bind Identity:** Associate the public key with the user's identity (e.g., username, email) and store this information in the service provider's database.
3. **Send Challenge:** When the user requests access, the server sends a random challenge (nonce) to the user.
4. **Sign and Verify:** The user signs the challenge with their private key and sends the signature to the server. The server verifies the signature with the public key. If valid, the user is authenticated and granted access.

Key Exchange



Key Exchange is Everywhere

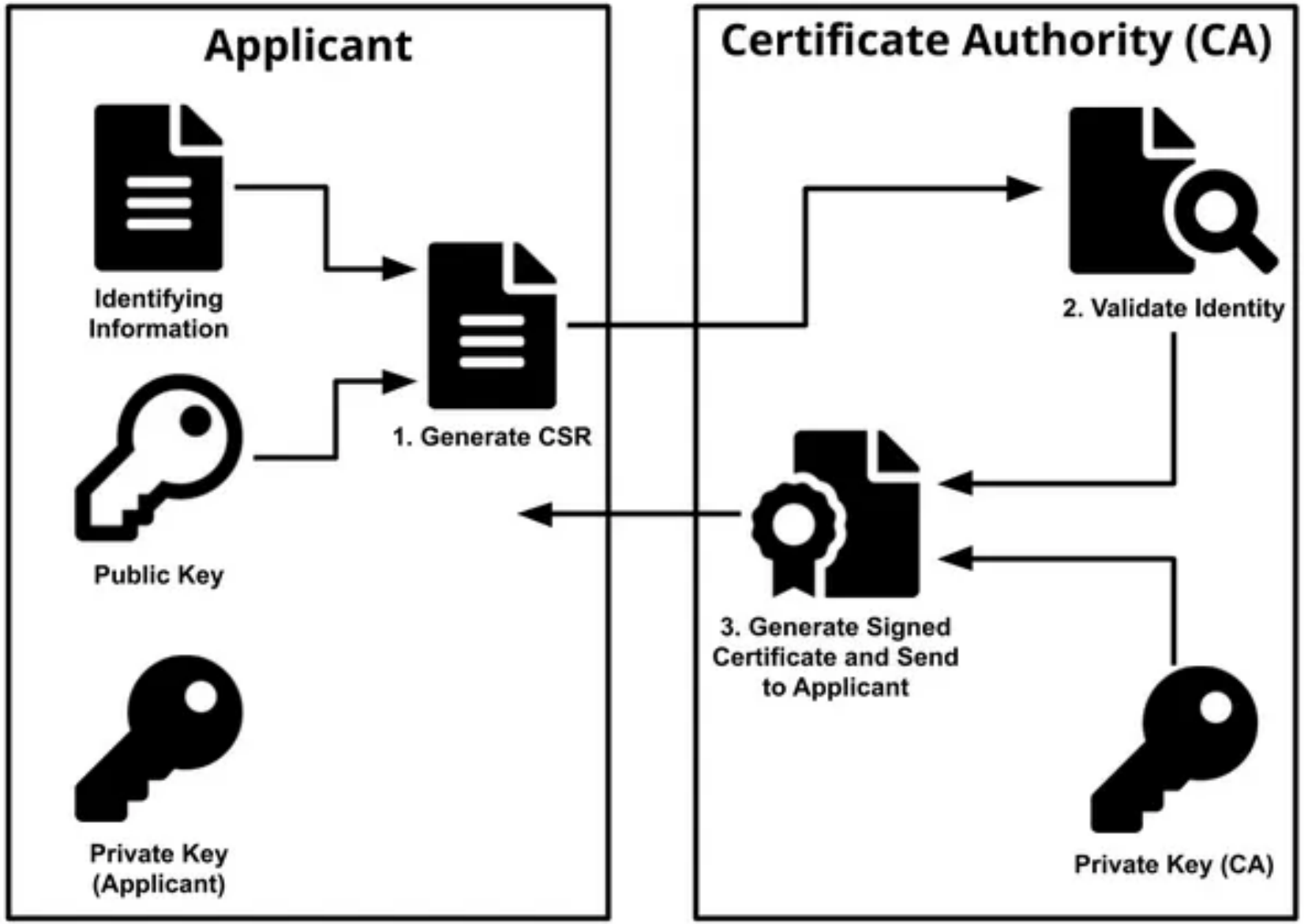
- TLS/SSL used by HTTPS
- VPN
- WIFI
- SSH for remote secure shell



Digital Signature

- Process
  - Alice encrypts the hash of the message to create a digital signature.
  - The digital signature is then sent along with the original message.
  - On the receiving end, Bob uses Alice's public key to decrypt the digital signature.
  - Bob also hashes the received message using the same algorithm as Alice.
  - He then compares the decrypted digital signature with the newly generated hash.
- Result: integrity and authentication

PKI


Applicant
Certificate Authority (CA)
Identifying Information
1. Generate CSR
2. Validate Identity
Public Key
3. Generate Signed Certificate and Send to Applicant
Private Key (Applicant)
Private Key (CA)

WEB/HTTPS Authentication

- Every browser has Amazon (or its CA Certificate Authority)’s public key.
- When you access amazon.com, your browser sends a random text (nonce – one time) to amazon, ask Aamzon.com to encrypt it and send it back
- When receiving the encrypted message, the browser decrypt it with Amazon’s public key, if you get the same message, it must be from Amazon.

What is PKI? https://www.youtube.com/watch?v=0ctat6RBrFo