Applied Cryptography

Cryptography fundamentals for a developer

Boney Johns
19/03/2019

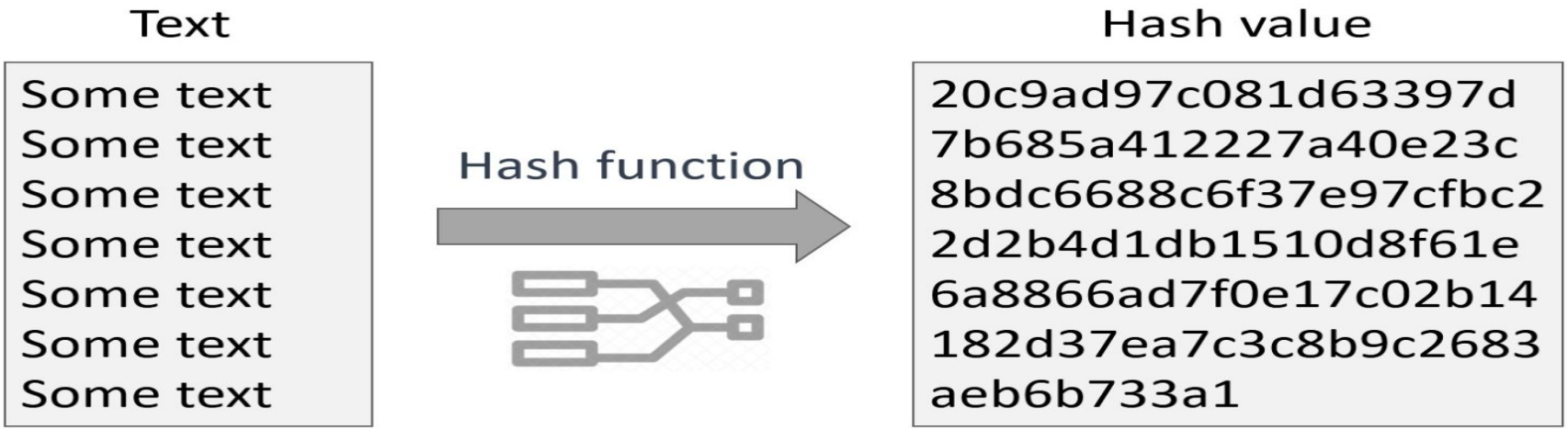
Topics

- Hashing
- Symmetric Encryption
- Asymmetric Encryption
- SSL
- SSH
- Digital Signatures

Disclaimer: All images taken from internet

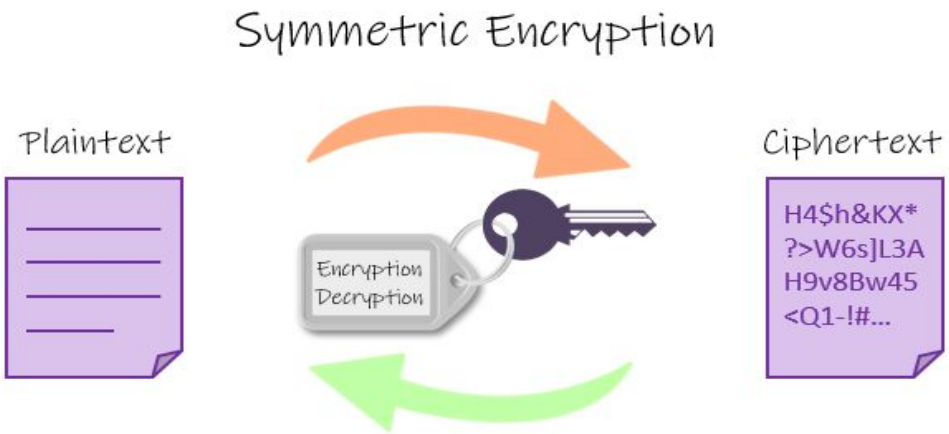
Hashing

- Converts data into a unique string
- Uses hash function (One way and non reversible)
- Use case: Storage of passwords
- Common algorithms: SHA256



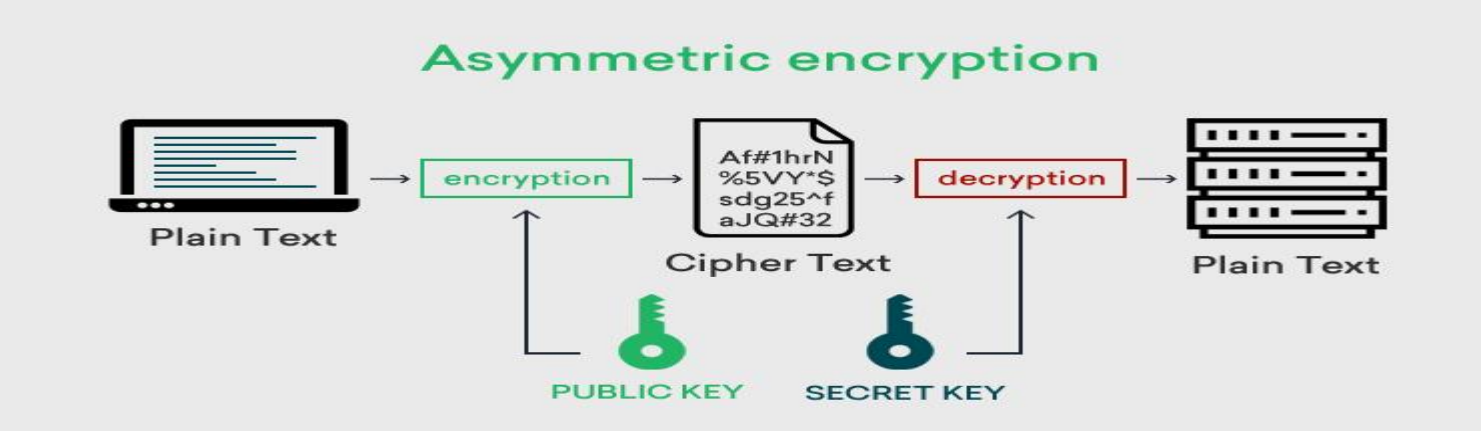
Symmetric Encryption

- Single key used for encryption and decryption
- 1000 times faster than Asymmetric encryption
- Main block of SSL, SSH



Asymmetric Encryption

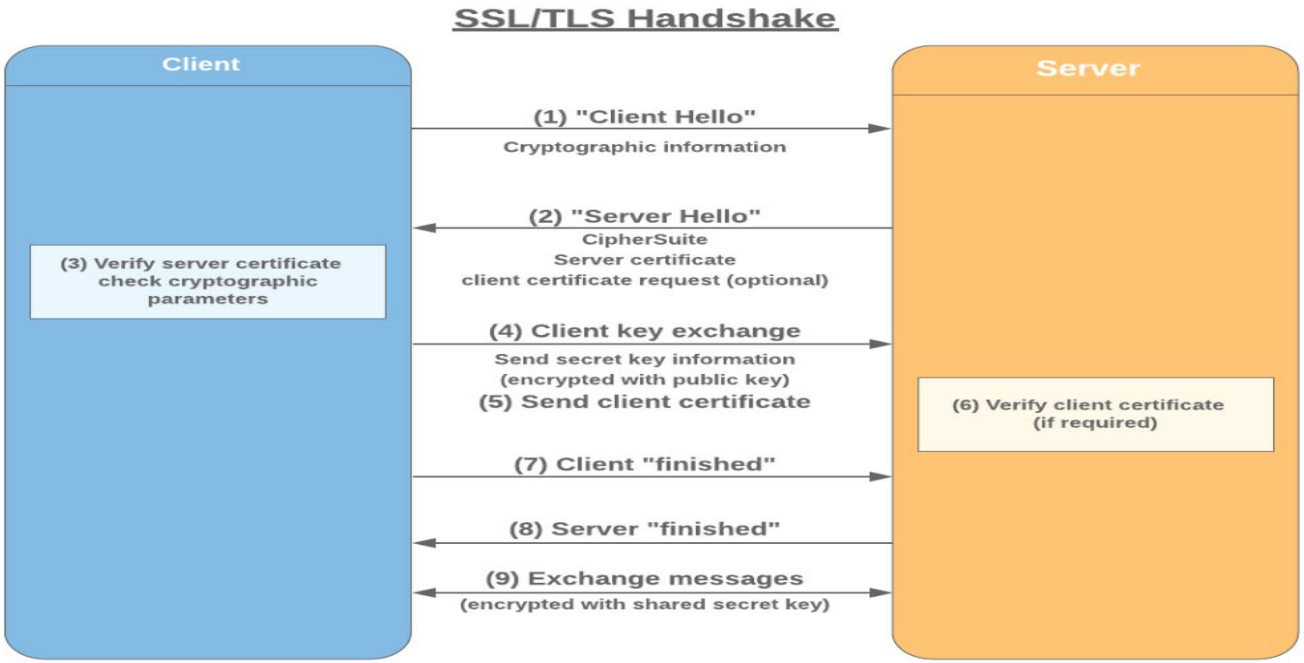
- Key Pair - Public key & Private key
- Usually public key will be used for encryption and private key for decryption (except for digital signatures)
- Boot loader for SSL, SSH



TLS - 1

- CA (Certificate Authority) will sign the server certificate
- Certificate contains the public key & passed to client
- Private key with server
- Symmetric session key will be generated by browser, encrypted using public key and passed to server
- Symmetric session key used to encrypt traffic in subsequent calls
- Tools to create certificate: openssl, makecert

TLS - 2


SSL/TLS Handshake
Client
Server
(1) "Client Hello"
Cryptographic information
(2) "Server Hello"
CipherSuite
Server certificate
client certificate request (optional)
(3) Verify server certificate
check cryptographic
parameters
(4) Client key exchange
Send secret key information
(encrypted with public key)
(5) Send client certificate
(6) Verify client certificate
(if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

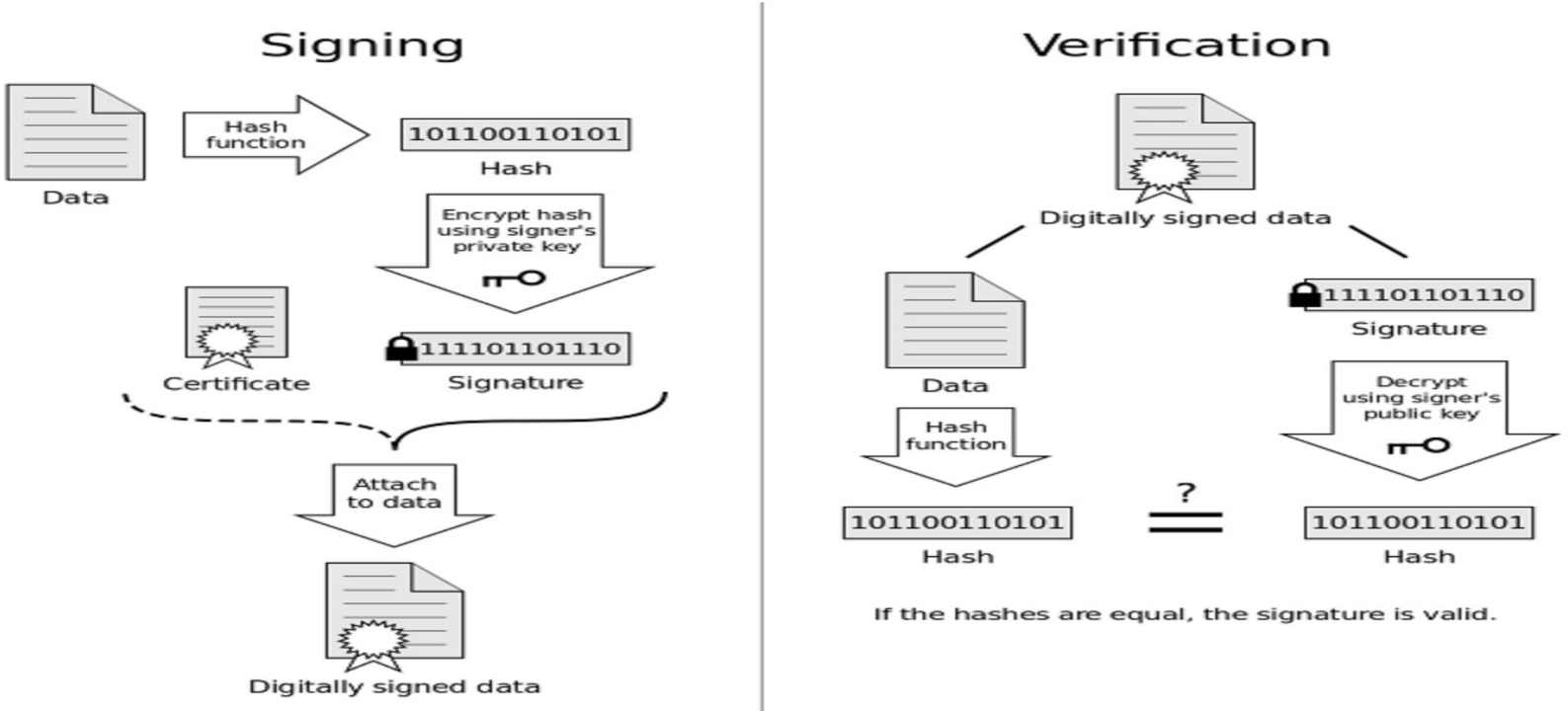
SSH

- ssh-keygen - will generate key pair
- Known hosts (ssh-keyscan)
- First create the encryption channel (using seperate key pairs at client side & server side)
- Then authenticate the user (password/SSH key) via the encryption channel
- Use cases: Remote login to the server (SSH enabled), get code from Git repository
- In depth: https://www.digitalocean.com/community/tutorials/understanding-the-ssh-encryption-and-connection-process

Digital Signature - 1

- To verify someone is whom they claim to be
- Private key and public key is with the sender
- Create signature using hash of commonly agreed data, signed with private key
- Pass signature and public key to the receiver
- Receiver verifies the identity of the sender
- Use cases: Hub provisioning, AWS SNS message verification, .NET Dll verification

Digital Signature - 2



Food for thought

- If encrypted web traffic can’t be decrypted by sniffers, how does fiddler decrypt the traffic ?
- In asymmetric cryptography, why data encrypted by public key cannot be decrypted by any other key, except for the corresponding private key ?

Code snippets

- https://github.com/bnyjohns/cryptography

Thank You