Cryptography

a lecture by ya boy, Bill Nye (Jerry)

Key Terms

- Key
  - Data used for encryption
- Cyphertext
  - a secret or disguised way of writing; a code.
- Plaintext
  - text that is not computationally tagged, specially formatted, or written in code.
- Encryption
  - the process of encoding a message or information in such a way that only authorized parties can access it and those who are not authorized cannot.
- Decryption
  - The conversion of encrypted data into its original form

Types

Types

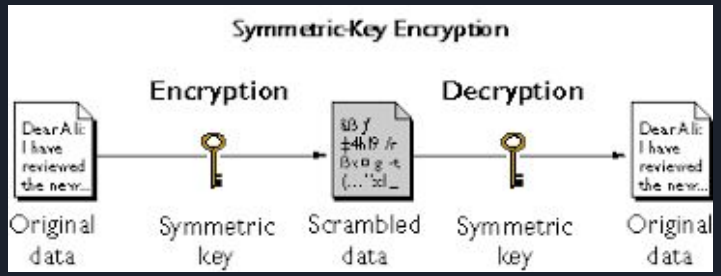
Symmetric

Asymmetric

Hashed

Types

Symmetric

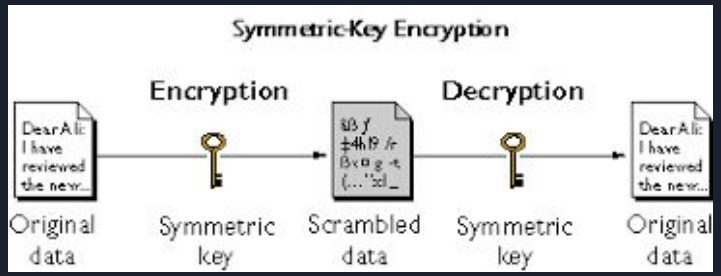


Asymmetric

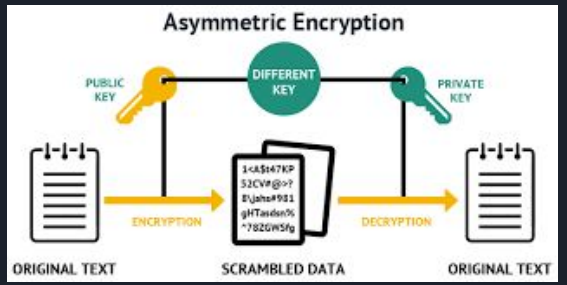
Hashed

Types

Symmetric



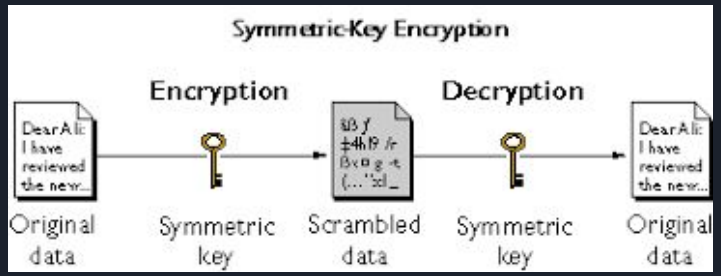
Asymmetric



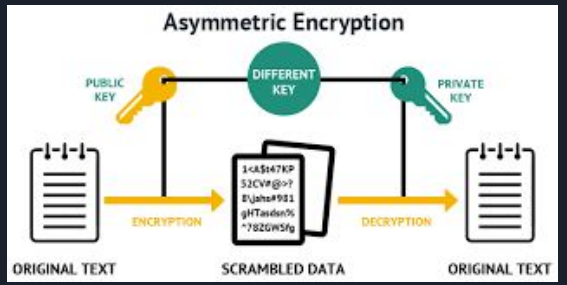
Hashed

Types

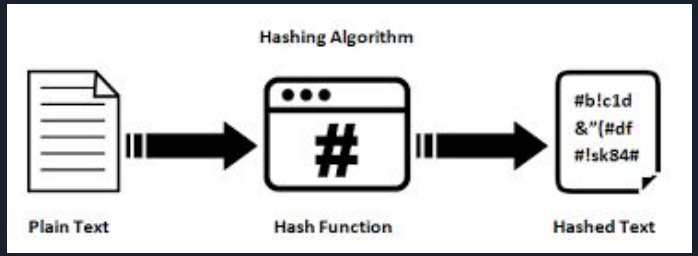
Symmetric



Asymmetric

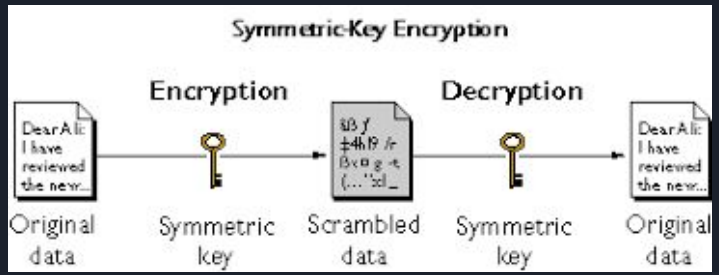


Hashed



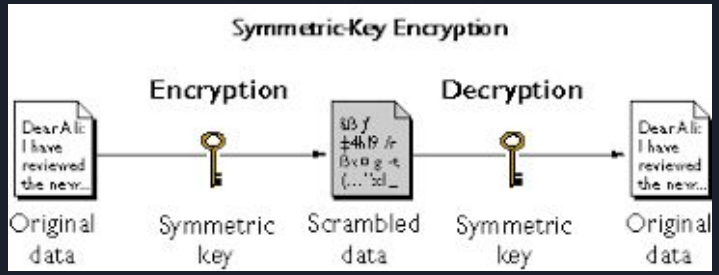
Symmetric - Definition

- AKA Private Key Encryption
- Both sender and receiver own identical private keys
- Sender encrypts data using an algorithm and the key
- Receiver decrypts data using the key
- Key is considered “shared secret”
- The key can never be misplaced or revealed to outside parties.
    - This becomes on of the big hindrances of symmetric encryption



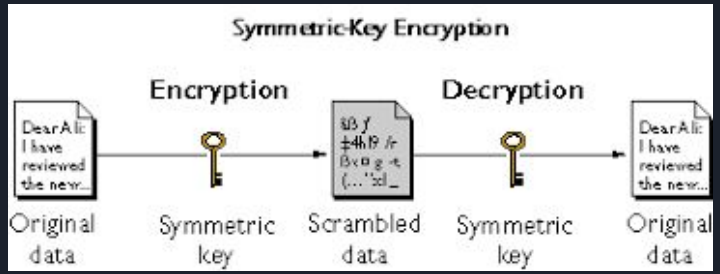
Symmetric - Use Cases

- Key Management is not an issue
- Encryption/ decryption time is a factor
- Less compute power available
- Implementations
  - HTTPS
  - SSL
  - DES
  - 3DES



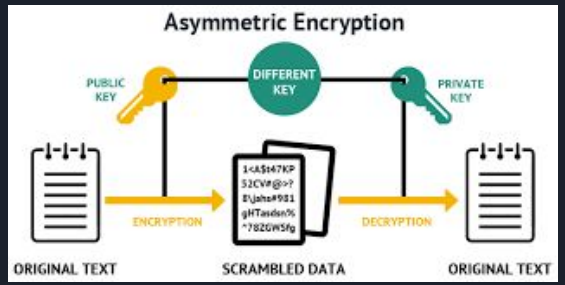
Symmetric - Examples

- Substitution ciphers
- Polygraphic substitution ciphers
- One time Pads
- Hill Cipher
- Steam and Block Ciphers
  - Steam Ciphers
  - Block Ciphers
    - DES, 3DES
    - AES
- Electronic Code Book (ECB)
  - Cipher Block Chaining (CBC)
- Output Feedback (OFB)
- Counter Mode (CTR)



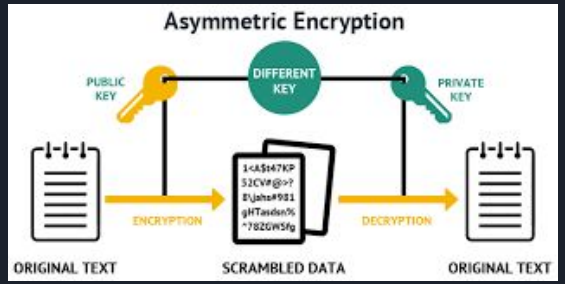
Asymmetric - Definition

- AKA Public Key Encryption
- Uses two keys, public and private
  - Anyone can access the public key
  - Private key is kept secret
  - The two keys are mathematically related, however it is impossible to derive the private key from the public key
- Sender encrypts with public key
- Receiver decrypts with private key



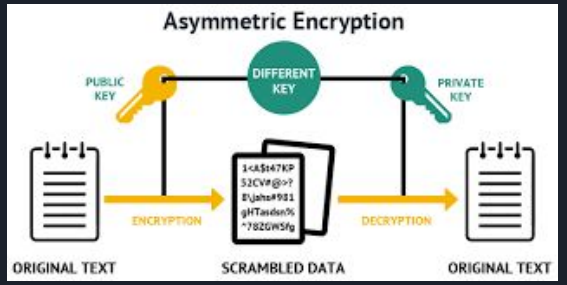
Asymmetric - Use Case

- Key Management is an issue
- Encryption/ decryption time is not a factor
- Heavy compute power available
- A common use for public key encryption is to use it to exchange secret keys
    - This allows for the secure use of the more efficient symmetric cryptosystem without worrying about exposure during key exchange
- Implementations
    - HTTPS
    - SSL
    - SSH
    - Bitcoin
    - PGP



Asymmetric - In Detail

- Trap door functions
  - Basis of asymmetric cryptography
  - Like hash functions, these functions should be easy to compute, and hard to invert
  - Unlike hash functions, the inverse is easy to compute if you know the secret key
- Factoring Primes
  - The type of trap door function that drives modern asymmetric encryption
  - Given two prime numbers, p and q, the product n of those primes will have 4 factors: n, 1, p, q

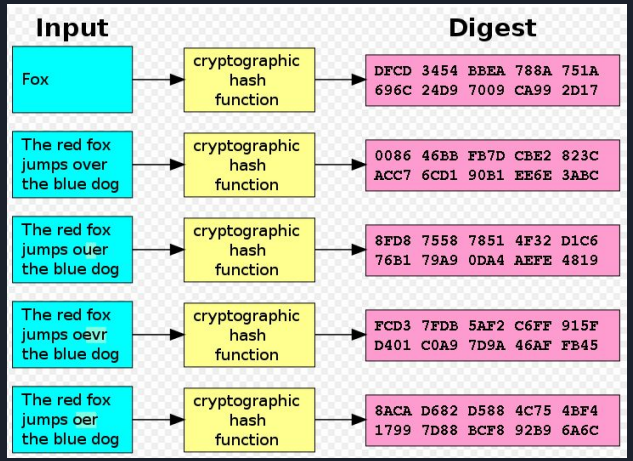


Symmetric V Asymmetric

https://www.youtube.com/watch?v=AQDCe585Lnc&vl=en

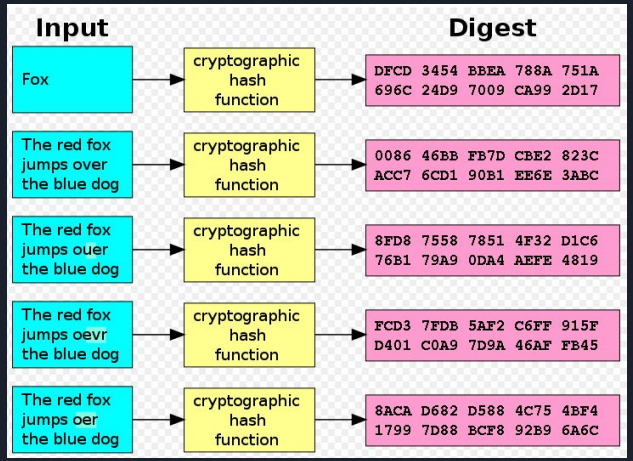
Hashed - Definition

- A function that takes an input and produces a fixed-size alphanumeric string called a hash value
- Easy to compute a hash value for any input
- Hard to invert (take the output and discover the input)
- Small changes to the input should produce large changes to the output



Hashed - Use Case

- Passwords
  - Store passwords as hash values
  - Hash passwords every time you auth
- Digital Signature
  - Verifying a download is legitimate
  - Verifying integrity of a filesystem

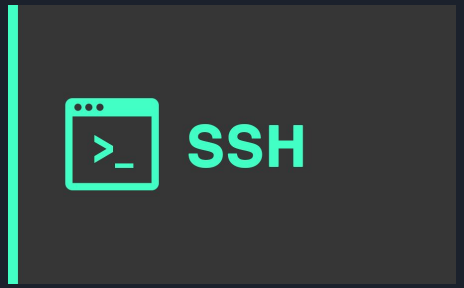


Attack Types

- Ciphertext-Only Attack
  - The cryptanalyst has access to the ciphertext of one or more messages, all encrypted using the same key, K. The goal is to determine the plaintext of these messages, or better yet, the key K.
- Known Plaintext Attack
  - The cryptanalyst has access to one or more plaintext-ciphertext pairs, each encrypted with K. The goal is to determine the key K.
- Chosen Plaintext Attack
  - The cryptanalyst can chose one or more plaintext messages and get the ciphertext that is associated with each one, based on the use of key K.
- Chosen Ciphertext Attack
  - The cryptanalyst can choose one or more ciphertext messages and get the plaintext that is associated with each one, based on the use of the same key K.

Secure Communication Tools

- PGP
- SSH



PGP

- Pretty Good Privacy (PGP)
- What does it do:
  - generates a public key (to encrypt messages) and a private key (to decrypt messages)
  - Used as a method to securely send data
- Some History
  - Created by a software engineer named Phil Zimmermann in 1991.
  - Is now owned by Symantec
  - OpenPGP was created by Zimmermann in 1997
  - GPG is based off of OpenPGP, but developed further to combat Symantec's software

SSH

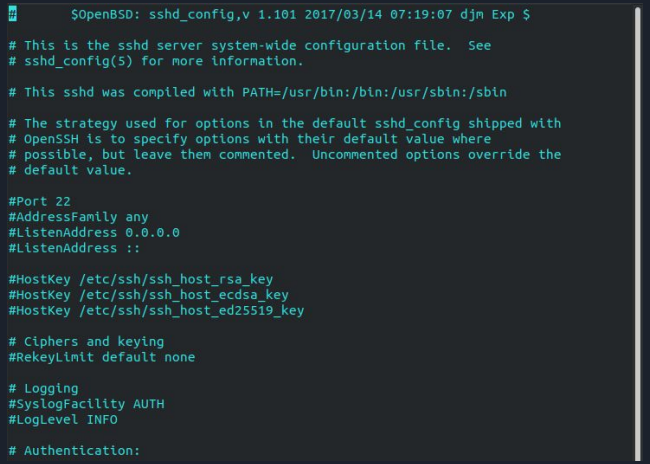
- Both parties agree on a large prime number, which will serve as a seed value.
- Both parties agree on an encryption generator (typically AES), which will be used to manipulate the values in a predefined way.
- Independently, each party comes up with another prime number which is kept secret from the other party. This number is used as the private key for this interaction (different than the private SSH key used for authentication).
- The generated private key, the encryption generator, and the shared prime number are used to generate a public key that is derived from the private key, but which can be shared with the other party.
- Both participants then exchange their generated public keys.
- The receiving entity uses their own private key, the other party's public key, and the original shared prime number to compute a shared secret key. Although this is independently computed by each party, using opposite private and public keys, it will result in the same shared secret key.
- The shared secret is then used to encrypt all communication that follows

SSH

- In short:
  - Authentication is encrypted through asymmetric encryption
  - Actual connection is encrypted through symmetric encryption
    - AES, Blowfish, 3DES, CAST128, and Arcfour
- Because of the encryption, listening to the traffic might not yield much information, so attackers sometimes get creative
  - That's why we harden our SSH configuration

SSH Hardening

- /etc/ssh/sshd_config
- Make edits such as
  - Change default port
  - Whitelist specific users
  - Disable root login
  - Disconnect idle sessions
  - Generate a ssh key so you can disable password auth
- Test the config
  - sshd -t
- Reload the new config
  - sudo systemctl reload sshd



```
#       $OpenBSD: sshd_config,v 1.101 2017/03/14 07:19:07 djm Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options override the
# default value.

#Port 22
#AddressFamily any
#ListenAddress 0.0.0.0
#ListenAddress ::

#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_ecdsa_key
#HostKey /etc/ssh/ssh_host_ed25519_key

# Ciphers and keying
#RekeyLimit default none

# Logging
#SyslogFacility AUTH
#LogLevel INFO

# Authentication:
```

Resources

- https://resources.infosecinstitute.com/basics-of-cryptography-the-practical-application-and-use-of-cryptography/
- https://www.marksei.com/encryption-and-hashing-differences-and-use-cases/
- https://medium.freecodecamp.org/how-does-pretty-good-privacy-work-3f5f75ecea97
- https://medium.com/@jasonrigden/hardening-ssh-1bcb99cd4cef
- https://www.digitalocean.com/community/tutorials/understanding-the-ssh-encryption-and-connection-process/
-