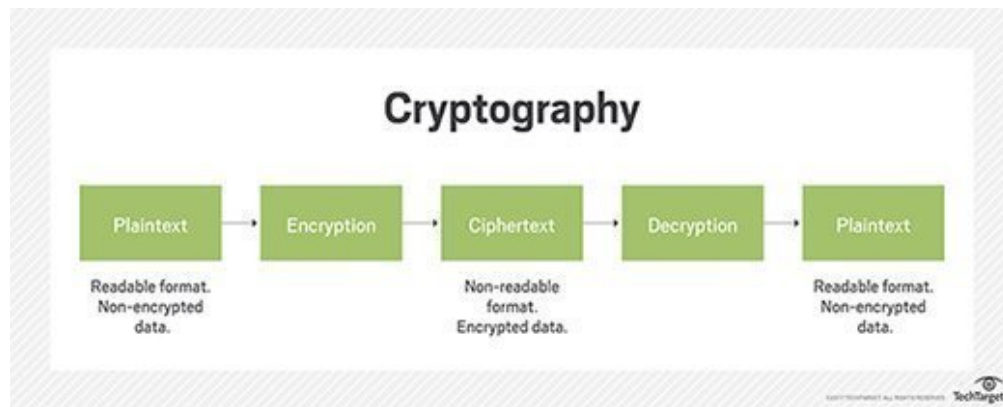


# Cryptography

---



## The goals of Cryptography:

- C.I.A. + Nonrepudiation
  - **Nonrepudiation** - Means by which a recipient can ensure the identity of the sender and neither party can deny sending.

## Basic Terms & Concepts

---

- **Cryptography**
  - Science or study of protecting information whether in transit or at rest
  - Renders the information unusable to anyone who can't decrypt it
  - Takes plain text, applies cryptographic method, turn it into cipher text
- **Cryptanalysis**
  - Study and methods used to crack cipher text
- **Linear Cryptanalysis**
  - Works best on block ciphers
- **Differential Cryptanalysis**
  - Applies to symmetric key algorithms
  - Compares differences in the inputs to how each one affects the outcome
- **Integral cryptanalysis**

- input vs output comparison same as differential; however, runs multiple computations of the same block size input
- Plain text doesn't necessarily mean ASCII format - it simply means unencrypted data
- **Key clustering** - Different encryption keys generate the same ciphertext from the same plaintext message

## Where to Encrypt & Decrypt?

---

- **Data-in-Transit / Data-in motion:** Transport / Network
  - Not much protection as it travels
    - Many different switches, routers, devices
  - Network-based protection:
    - Firewall, IPS
  - Provide transport encryption:
    - TLS, IPsec
- **Data-at-Rest:** Resides in storage
  - Hard drive, SSD, flash drive, etc
  - Encrypt the data
    - Whole disk encryption
    - Database encryption
    - File or/ folder-level encryption
  - Apply permissions
    - Access control lists
    - Only authorized users can access the data
- **Data-in-use / Data-in-process:** RAM & CPU
  - The data is in memory or CPU registers and cache
  - The data is almost always decrypted

## Encryption Algorithms

---

- **Algorithm** - step-by-step method of solving a problem
- **Two General Forms of Cryptography**
  - **Substitution** - bits are replaced by other bits
  - **Transposition** - doesn't replace; simply changes order
- **Encryption Algorithms** - mathematical formulas used to encrypt and decrypt data
- **Stream Cipher** - readable bits are encrypted one at a time in a continuous stream
  - Usually done by an XOR operation
  - Work at a high rate of speed

- **Block Cipher** - data bits are split up into blocks and fed into the cipher
  - Each block of data (usually 64 bits) encrypted with key and algorithm
  - Are simpler and slower than stream ciphers
- **XOR** - exclusive or; if inputs are the same (0,0 or 1,1), function returns 0; if inputs are not the same (0,1 or 1,0), function returns 1
- Key chosen for cipher must have a length larger than the data; if not, it is vulnerable to frequency attacks

## Symmetric Encryption

---

- **Symmetric Encryption** - One Single Key / Session Key to encryption and decryption.
  - **Known as:**
    - Single key cryptography
    - Secret key cryptography
    - Shared key cryptography
    - Session key cryptography



**One key is used to encrypt and decrypt the data.**

- Suitable for large amounts of data
- 128-bit or larger symmetric keys are common
- Harder for groups of people because more keys are needed as group increases
- Can be very fast to use
  - Less overhead than asymmetric encryption
  - Often combined with asymmetric encryption
- **Problems/Weaknesses of Symmetric Encryption:**
  - Problems include key distribution and management / not scalable
  - Non-repudiation possible because everyone has a copy of the key
  - Key must be regenerated whenever anyone leaves the group of keyholders

## Cryptosystem

Defines key properties, communication requirements for the key exchange; actions through encryption and decryption process.

*e.g.: Using asymmetric encryption to exchange Session keys after that communicate using Symmetric encryption.*

- **Key escrow** (also known as a “fair” cryptosystem) is an arrangement in which the keys needed to decrypt encrypted data are held in escrow so that, under certain circumstances, an authorized third party may gain access to those keys.

## Symmetric Cryptosystems:

Algorithm	Block or Streaming	Block Size	Rounds	Key Size	Notes
DES	Block	64-bit	16	56 bits	Uses five modes of operation: ECB, CBC, CFB, OFB and CTR.
Blowfish	Block	64-bit	16	32-448 bits	Public domain algorithm.
Twofish	Block	128-bit	16	128, 192 and 256 bits	Public domain algorithm.
3DES	Block	64-bit	16	168 bits (56 x 3)	Repeats DES process 3 times.
AES	Block	128-bit	10, 12, or 14	128, 192 or 256 bits	Encryption standard for the US Gov.; Used in WPA2
RC4	Streaming	N/A	1	40-2048 bits	Used in WEP, SSL and TLS; largely deprecated in current technologies.
IDEA	Block	64-bit	8	128 bits	Made for replacement for the DES

- Larger keys than symmetric encryption; Common to see key lengths of 3,072 bits or larger

## Asymmetric Encryption

Uses a Key pair:

- **Public Key** - Anyone can see this key; give it away
- **Private Key** - Keep this private; used for decryption; The private key is used to digitally sign a message.



- **Algorithms:**
  - **Diffie-Hellman** - Developed as a key exchange protocol; used in SSL and IPSec; if digital signatures are waived, vulnerable to MITM attacks
  - **Elliptic Curve Cryptosystem (ECC)** - Uses points on elliptical curve along with logarithmic problems; uses less processing power; good for mobile devices
  - **RSA** - Achieves strong encryption through the use of two large prime numbers; factoring these create key sizes up to 4096 bits; modern de facto standard
  - **El Gamal** - Not based on prime number factoring; uses solving of discrete logarithm problems
- Only downside is it's slower than symmetric especially on bulk encryption and processing power

## Hashes

- One-way encryption
- Verify the Integrity of the message.
- Verify the authenticity of the message (proof of origin & non-repudiation)
- Impossible to recover the original message from the digest
- Used to store passwords providing confidentiality.

Hash	Algo.
MD5	128 bit hash
SHA-1	160 bit hash
SHA256	256 bit hash

*Examples:*

String: hello world!

MD5 Hash: FC3FF98E8C6A0D3087D515C0473F8677

SHA-1 Hash: 430CE34D020724ED75A196DFC2AD67C77772D169

SHA256 Hash: 7509E5BDA0C762D2BAC7F90D758B5B2263FA01CCBC542AB5E3DF163BE08E6CA9



If you change a single character, the entire Hash value changes. See the example below, changing the last character '!' to '.'

- String: hello world!
  - MD5 Hash: FC3FF98E8C6A0D3087D515C0473F8677

- String: **hello world.**
  - MD5 Hash: 3C4292AE95BE58E0C58E4E5511F09647

## Message digest

A message digest or hash, can be used to verify the integrity of a message by comparing the original hash to one generated after receipt of the message. If the two match, then integrity is assured. If they do not match, then the message was altered between transmission and receipt.

 **Message digests are also called:**

- hashes
- hash values
- hash total
- CRC
- fingerprint
- checksum
- digital ID

## Hashing Algorithms


---


### MD5 - Message Digest Algorithm

- First published in April 1992
- Replaced MD4
- 128-bit hash value
- 1996: Vulnerabilities found
  - Not collision resistant

 **Collision** - occurs when two or more files create the same output

- Can happen and can be used as an attack; rare, though

 **Key space** - Represents the total number of possible values of keys in a cryptographic algorithm or other security measure, such as password

 **DUHK Attack** (Don't Use Hard-Coded Keys) - allows attackers to access keys in certain VPN implementations; affects devices using ANSI X9.31 with a hard-coded seed key

 **Rainbow Tables** - contain precomputed hashes to try and find out passwords

### SHA - Secure Hash Algorithm

- Developed by NSA

## SHA-1

- Widely used
- 160-bit digest
- Weak; 2005: *Collision attacks published*

## SHA-2 Family

- SHA-256 | minor version: SHA-224
- SHA-512 | minor version: SHA-384

## SHA-3

- Uses a hash function called Keccak and has the same length of SHA-2.
- SHA-1 and SHA-2 have been replaced by the latest iteration of SHA known as SHA-3.

## HMAC

Hash Message Authentication Code - Used in conjunction with symmetric key both to authenticate and verify integrity of the message.

- Verify data **integrity** and **authenticity**
  - No fancy asymmetric encryption is required
- Used in network encryption protocols
  - IPsec, TLS
- Requires each side of the conversation to have the same key

## RIPEMD

RACE Integrity Primitives Evaluation Message Digest.

- Not very common
- Open Standard
- 128, 168, 256, 320 bit digests (*RIPEMD-128, RIPEMD-256, RIPEMD-320*)
- *Original RIPEMD was found to have collision issues (2004)*
  - Effectively replaced with RIPEMD-160 (no known collision issues)
  - Based upon MD4 design but performs similar to SHA-1

# Keystretching

Combine a very long salt and a huge number of hashing iterations to make cracking even more harder. (e.g Hashing the hashed password **N** times)

Two most popular Key stretching libraries/ functions:

- **PBKDF2** (Password-Based Key Derivation Function 2) algorithm
  - Part of RSA public key cryptography standards (PKCS #5, RFC 2898)
- **bcrypt**
  - Generates hashes from passwords
  - An extension to the UNIX crypt library
  - Uses Blowfish cipher to perform multiple rounds of hashing

*Example:*

- **PBKDF2**

Password: 123456

Hash:

rYoSDg62evyzhE1+lWBa9A==:YaeMu71c8KU3H0RYFP1e0Q==

- **bcrypt**

Password: 123456

Hash:

\$2b\$10\$vES9mCPsE10//v0c1u01XeUVmJrZyHGMPaRfo390IUoJ2g7iPtDnu



**Key streaming** - involves sending individual characters of the key through an algorithm and using a mathematical XOR function to change the output.

## Cryptographic nonce

*Cryptographic randomization schemes*

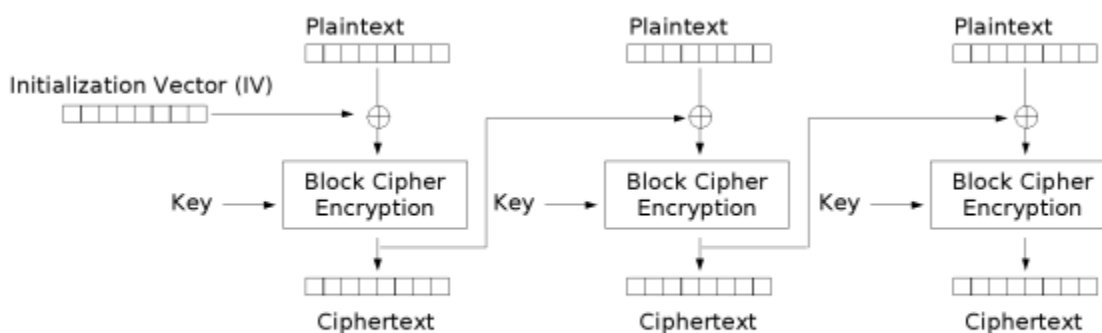
- Used once - 'for the nonce'/ for the time being
- A random or pseudo-random number
  - Something that can't be reasonably guessed



- Can also be a counter
- Use a nonce during the login process
  - Server gives you a nonce
  - Calculate your password hash using the nonce
  - **Each password hash sent to the host will be different**, so a replay attack won't work

## Initialization vectors (IV)

- Is a type of nonce
  - Used for randomizing an encryption scheme
  - The more random the better
- Use in encryption ciphers, WEP, and older SSL implementations



Cipher Block Chaining (CBC) mode encryption

## Digital Signatures

- When signing a message, you sign it with your **private** key and the recipient decrypts the hash with your **public** key
- **Digital Signature Algorithm (DSA)** - used in generation and verification of digital signatures per FIPS 186-2



### Digital Signature Standard (DSS):

- Document that NIST puts out to specify the digital signature algorithms & the encryption algorithms approved for use by the US gov.


## PKI System

*Public Key Infrastructure (PKI) - structure designed to verify and authenticate the identity of individuals*



- Also refers to the binding of public keys to people or devices
  - The certificate authority (CA)
  - It's all about trust
- X.509 v3 is current format most widely used. Part of the X.500 family of standards

## Digital Certificates

- **Certificate** - electronic file that is used to verify a user's identity; provides nonrepudiation
- X.509 - standard used for digital certificates
- **Contents of a Digital Certificate:**
  -  **digi-cert**
    - **Version** - identifies certificate format
    - **Serial Number** - used to uniquely identify certificate
    - **Subject** - who or what is being identified
    - **Algorithm ID** (Signature Algorithm) - shows the algorithm that was used to create the certificate
    - **Issuer** - shows the entity that verifies authenticity
    - **Valid From and Valid To** - dates certificate is good for
    - **Key Usage** - what purpose the certificate serves
    - **Subject's Public Key** - copy of the subject's public key
    - **Optional Fields** - Issuer Unique Identifier, Subject Alternative Name, and Extensions
- Some root CAs are automatically added to OSes that they already trust; normally are reputable companies
- **Self-Signed Certificates** - certificates that are not signed by a CA; generally not used for public; used for development purposes
  - Signed by the same entity it certifies

## Registration Authority

- Verifies user identity

## Certificate Authority

- Third party to the organization; creates and issues digital certificates

## Certificate Revocation List (CRL)

- Used to track which certificates have problems and which have been revoked

### Validation Authority

- Used to validate certificates via Online Certificate Status Protocol (OCSP)

### Trust Model

- How entities within an enterprise deal with keys, signatures and certificates

### Cross-Certification

- Allows a CA to trust another CS in a completely different PKI; allows both CAs to validate certificates from either side

### Single-authority system

- CA at the top

### Hierarchical trust system

- CA at the top (root CA); makes use of one or more RAs (subordinate CAs) underneath it to issue and manage certificates

## Key Wrapping and Key Encryption Keys (KEK)

---

- KEKs are used as part of key distribution or key exchange.
- key Wrapping - Protect session keys
- If the cipher is a symmetric KEK, both the sender and the receiver will need a copy of the same key
- If using an asymmetric cipher, with public/private key properties, to encapsulate a session key, both the sender and the receiver will need the other's public key



Protocols such as SSL, PGP, and S/MIME use the services of KEKs to provide session key confidentiality, integrity, and sometimes to authenticate the binding of the session key originator and the session key itself.

## Full Disk Encryption - FDE

---

- **Data at Rest (DAR)** - data that is in a stored state and not currently accessible
  - Usually protected by **full disk encryption (FDE)** with pre-boot authentication
  - Example of FDE is Microsoft BitLocker and McAfee Endpoint Encryption
  - FDE also gives protection against boot-n-root

# Encrypted Communication

---

- **Often-Used Encrypted Communication Methods:**
  - **Secure Shell (SSH)** - secured version of telnet; uses port 22; relies on public key cryptography; SSH2 is successor and includes SFTP
  - **Secure Sockets Layer (SSL)** - encrypts data at transport layer and above; uses RSA encryption and digital certificates; has a six-step process; largely has been replaced by TLS
  - **Transport Layer Security (TLS)** - uses RSA 1024 and 2048 bits; successor to SSL; allows both client and server to authenticate to each other; TLS Record Protocol provides secured communication channel
  - **Internet Protocol Security (IPSEC)** - network layer tunneling protocol; used in tunnel and transport modes; ESP encrypts each packet
  - **PGP** - Pretty Good Privacy; used for signing, compress and encryption of emails, files and directories; known as hybrid cryptosystem - features conventional and public key cryptography
  - **S/MIME** - standard for public key encryption and signing of MIME data; only difference between this and PGP is PGP can encrypt files and drives unlike S/MIME
- **Heartbleed** - attack on OpenSSL heartbeat which verifies data was received correctly
  - Vulnerability is that a single byte of data gets 64kb from the server
  - This data is random; could include usernames, passwords, private keys, cookies; very easy to pull off
  - `nmap -d --script ssl-heartbleed --script-args vulns.showall -sV [host]`
  - Vulnerable versions include Open SSL 1.0.1 and 1.0.1f
  - CVE-2014-0160
- **FREAK (Factoring Attack on RSA-EXPORT Keys)** - man-in-the-middle attack that forces a downgrade of RSA key to a weaker length
- **POODLE (Paddling Oracle On Downgraded Legacy Encryption)** - downgrade attack that used the vulnerability that TLS downgrades to SSL if a connection cannot be made
  - SSL 3 uses RC4, which is easy to crack
  - CVE-2014-3566
  - Also called PoodleBleed
- **DROWN (Decrypting RSA with Obsolete and Weakened Encryption)** - affects SSL and TLS services

- Allows attackers to break the encryption and steal sensitive data
- Uses flaws in SSL v2
- Not only web servers; can be IMAP and POP servers as well

## Cryptography Attacks

---

*Cryptographic attacks approaches that seek to exploit one or more vulnerabilities in a cryptosystem to break it; **Note: Patterns Kill! and it's all about the key!***

- **Frequency Analysis & the Ciphertext Only Attack**
  - Examine frequency of letters appearing in the ciphertext
  - Attempt to figure out what letters they correspond to plaintext
- **Known Plain-text attack**
  - Has both plain text and cipher-text; plain-text scanned for repeatable sequences which is compared to cipher text
- **Chosen Cipher-text Attack**
  - Chooses a particular cipher-text message
  - Attempts to discern the key through comparative analysis
  - RSA is particularly vulnerable to this
- **Chosen Plain-text attack**
  - Attacker encrypts multiple plain-text copies in order to gain the key
- **Adaptive chosen plain-text attack**
  - Attacker makes a series of interactive queries choosing subsequent plaintexts based on the information from the previous encryptions; idea is to glean more and more information about the full target cipher text and key
- **Cipher-text-only attack**
  - Gains copies of several encrypted messages with the same algorithm; statistical analysis is then used to reveal eventually repeating code
- **Replay attack**
  - Usually performed within context of MITM attack
  - Hacker repeats a portion of cryptographic exchange in hopes of fooling the system to setup a communications channel

- Doesn't know the actual data - just has to get timing right
- **Side-Channel Attack**
  - Monitors environmental factors such as power consumption, timing and delay
- **Meet-in-the-Middle**
  - Used against algorithms that use 2 rounds of encryption. (reason that 2-DES was defeated).
- **Man-in-the-Middle**
- **Birthday Attack / Collision Attack / Reverse Hash matching**
  - Find flaws in the one-to-one association of the hash function
- **Timing Attack**
  - Based on examining exact execution times of the components in the cryptosystems
- **Rubber-Hose Attack**
  - Based on the use of threats or torture to extract need information
- **Don't Use Hard-Coded Keys (DUHK) Attack**
  - Used against hardware/software that implements ANSI X9.31 Random Number Generation.
- **Social Engineering Attack**
  - Social eng. can be very efficient to grab passwords etc

## Tools

- Carnivore and Magic Lantern - used by law enforcement for cracking codes
- L0phtcrack - used mainly against Windows SAM files
- John the Ripper - UNIX/Linux tool for the same purpose
- PGPcrack - designed to go after PGP-encrypted systems
- CrypTool
- Cryptobench
- Jipher
- Keys should still change on a regular basis even though they may be "unhackable"
- Per U.S. government, an algorithm using at least a 256-bit key cannot be cracked

## How to defeat attack:

---

- **Salt the passwords** - A nonce most commonly associated with password randomization, making the password hash unpredictable.
  - *If the password database is breached, you can't correlate any passwords because even users with the same password have different hashes stored.*
- **Pepper** - A large constant number stored separately from the hashed password.
- **Key stretching** - Combine a very long salt and a huge number of hashing iterations to make cracking even more harder. (e.g Hashing the hashed password N times).