Flávio's Crypto 101

Notes from Flávio Costa's CompTIA Security+ Lesson #5.

The goals of cryptography are:

✓ **Confidentiality**: Unauthorized parties cannot access information (primary goal).

✓ **Integrity**: Assurance that the message was not modified during transmission, accidentally or intentionally.

✓ **Authenticity**: Validating the source of the message to ensure the sender is properly identified. Note: NOT **availability.**

✓ **Non-repudiation:** A sender cannot deny sending the message later.

Key Terms

✓ **Plaintext** is the unencoded message.

✓ **Ciphertext** is the coded message.

✓ **Cipher** cryptographic algorithm used to encrypt/decrypt data.

✓ **Cipher Suite** multiple pieces put together, with one or more modes of operation (signature algorithm, key exchange, key agreement, bulk encryption cipher).

✓ **Cryptanalysis** is the art of cracking cryptographic systems (not brute-forcing).

✓ **Key Length** range of key values in the keyspace: For example, a password with 4 digits. I have ten options, from 0-9, for each one of the 4 positions. So mathematically, the keyspace, or the maximum number of combinations for that password, is 10^4).

Encoding (e.g., Base64) **≠** encryption (e.g., AES, DES, RSA, etc.) **≠** hashing (e.g., SHA1, MD5)

When we discussed cryptography, we learned about these concepts:

✓ **Symmetric** cryptography.

✓ **Asymmetric** cryptography.

Symmetric cryptography

Symmetric encryption is used to ensure the confidentiality of some data. Symmetric ciphers use **one key** to encrypt data and **the same key** to decrypt data. That **private key** must be a secret. Many symmetric ciphers can process high data volumes without causing a significant impact on network, power, or memory. Common symmetric encryption algorithms include AES (Advanced Encryption Algorithm), DES (Data Encryption Standard), RC5 (Rivest Cipher 5). Both DES and RC5 are outdated and should only be used if AES is not available.

Modern symmetric ciphers have been developed to process data in one of two forms: **block and stream**.

Block ciphers are often used with asymmetric encryption on files and messages processing blocks of data. The files or messages are separated into fixed sized blocks and each block is encrypted or decrypted. By processing blocks of data, we add the ability to recover from possible errors experienced in data transmission. We do not have to start the process again from the start; we just must figure out which block was not processed properly, get a copy of that block, and process it again. AES, DES and RC5 & RC6 are block ciphers.

Stream ciphers are used when the data that is generated continually by some source. Audio (music) and video (movies) are examples of streaming data. Stream ciphers perform encryption and decryption on a bit-by-bit basis. RC4 is a stream cipher.

Flávio’s Crypto 101

Notes from Flávio Costa’s CompTIA Security+ Lesson #5.

The most commonly used symmetric operation modes for block ciphers are:

- **Electronic Codebook (ECB):** In this mode, each block of plaintext is encrypted separately using the same key. This mode is simple, but it is **not secure** because identical plaintext blocks produce identical ciphertext blocks.

- **Cipher Block Chaining (CBC):** In this mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This adds randomness to the encryption process, making it more secure than ECB mode.

- **Counter Mode (CTR):** In this mode, a counter is encrypted to generate a stream of key values that are XORed with the plaintext. This mode allows for parallel encryption and decryption and is used in many high-speed applications.

- **Galois/Counter Mode (GCM):** In this mode, **CTR mode is used for encryption** and a universal hash function called **Galois/Counter Mode Authentication (GMAC) is used for authentication**. This mode is often used in secure network communication protocols such as **TLS 1.2 and the preferred option for 1.3**.

- **Output Feedback (OFB):** In this mode, a keystream is generated by encrypting an initialization vector (IV) using the key. The keystream is then XORed with the plaintext to produce the ciphertext. This mode is similar to CTR mode but has the disadvantage of being vulnerable to errors in the keystream.

Asymmetric Cryptography

Asymmetric Encryption uses **two keys** to ensure the confidentiality of some data. If the **public key** is used for encryption, then the related **private key** is used for decryption. If the private key is used for encryption, then the related public key is used for decryption. As implied in the name, the Private Key is intended to be private so that only the authenticated recipient can decrypt the message.

Due to the ‘cost’ of encrypting and decrypting data (many asymmetric cipher algorithms are slower than symmetric) the message size is often limited so asymmetric encryption is not suitable for encrypting large amounts of data, but mainly for authentication and non-repudiation. Asymmetric cryptography is sometimes referred to as **Public Key** cryptography as it defines a key system uses a pair of keys.

Symmetric encryption is **unauthenticated** encryption.
Asymmetric encryption is **authenticated** encryption.
Common asymmetric encryption algorithms include RSA (Rivest Shamir Adleman), ECC (Elliptic-curve cryptography), DSA (Digital Signature Algorithm), Diffie-Hellman Key Exchange and ElGamal.

Forward Secrecy (also referred to as **Perfect Forward Secrecy (PFS)**) is feature of some cryptographic systems that changes the keys at intervals automatically. If a past key is disclosed or discovered that key can only be used to decrypt a small amount of old data -> *If the confidentiality of the private key is compromised, any previously encrypted data can be decrypted. PFS mitigate that vulnerability, hence, it gives better protection over time in the event of key disclosure.*

The terms "key exchange" and "key agreement" are often taken to mean the same thing, but there are different mechanisms. With key agreement, the client does not transmit an encrypted session key to the server. The client and server use Diffie-Hellman (D-H) to derive the same secret key value.

Flávio’s Crypto 101

Notes from Flávio Costa’s CompTIA Security+ Lesson #5.

Message / Data Integrity

A **hashing** (or **hash**) algorithm maps data of an arbitrary size (called the "message") to a bit array of a fixed size (the "hash value", "hash", or "**message digest**"). It is a one-way function which is practically infeasible to invert or reverse the computation. Two examples of hashing algorithms are: SHA (Secure Hash Algorithm) and MD5 (Message Digest 5).

Passwords are often stored as hashes instead of storing the actual password. Applications often **salt** passwords with extra characters before hashing them to defend against dictionary and brute force password cracking.

One way to find a message that produces a given hash is to attempt a brute-force search of possible messages to see if they produce a matching digest. A **rainbow table** is a list of messages and their digests that can be created in advance and then searched using the digest to try and discover the message. A rainbow table is a tool used in a brute force attack often used for breaking password hashes. Think of the rainbow as all the different combination of characters (a-z and 0-9) that could be used to create an n-character password (where n is the length of the password).

A Message Authentication Code (**MAC**) is a short piece of information used for authenticating a message. In other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed (integrity). The MAC value protects a message's data integrity, as well as its authenticity, by allowing verifiers (who know the algorithm used and secret key) to detect any changes to the message content. MACs differ from digital signatures as MAC values are both generated and verified using the same secret key. Digital signatures use public and private keys.

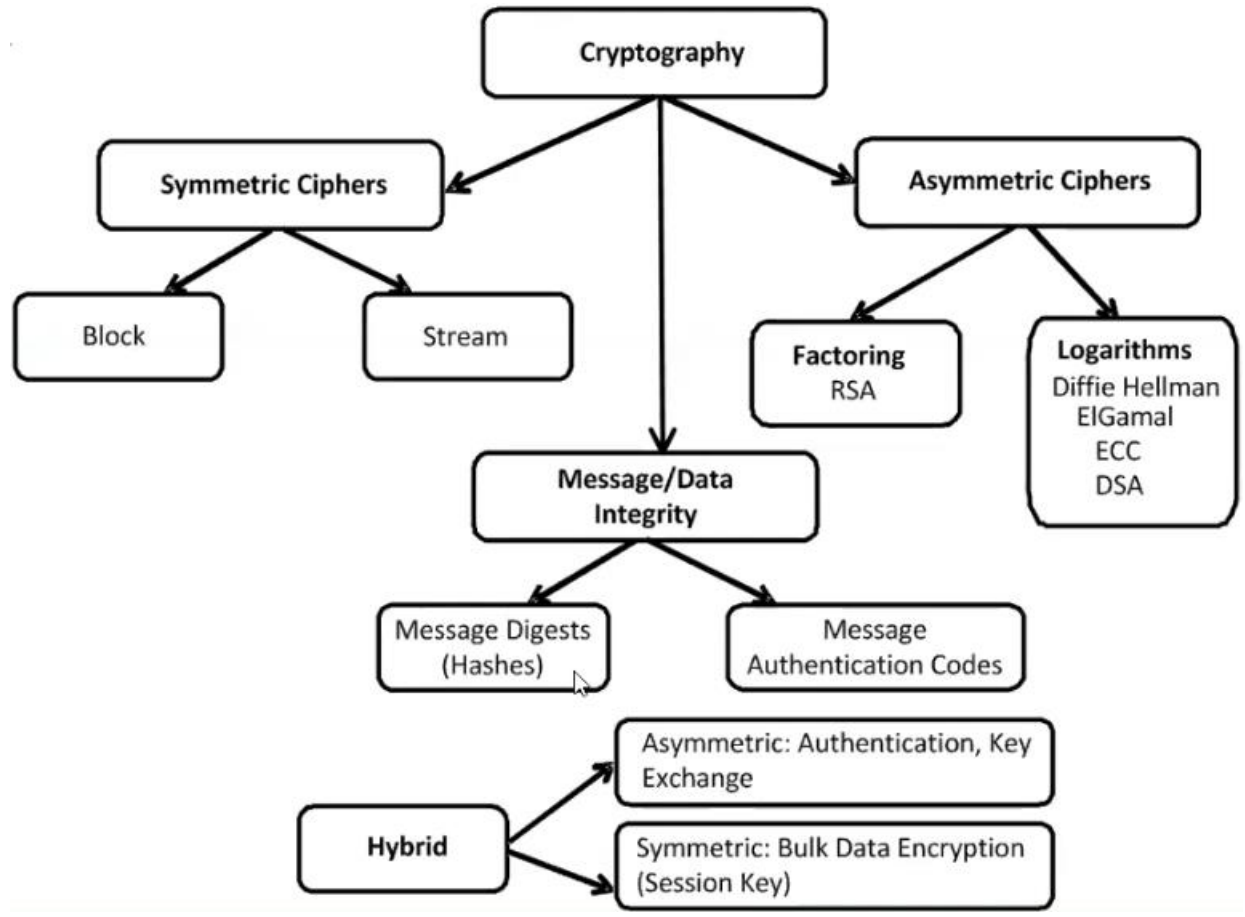
An **HMAC** (Hash-based Message Authentication Code) is a specific type of message authentication code (MAC) involving a cryptographic hash function and a secret cryptographic key. As with any MAC, it may be used to simultaneously verify both the data integrity and authenticity of a message.

Flávio's Crypto 101

Notes from Flávio Costa's CompTIA Security+ Lesson #5.

Crypto Summary

- Symmetric encryption uses one key.
- Asymmetric encryption uses a pair of keys (public and private).
- A hash algorithm creates a digest that cannot be reversed.
- Hashes are typically used for verifying the integrity of data, not for encryption.
- A MAC (Message Authentication Code) requires a secret key shared between the sender and receiver to generate and verify the authenticity and integrity of a message.
- HMAC (Hash-based Message Authentication Code) is a specific type of MAC that uses a cryptographic hash function and a secret key to provide message authentication and integrity.


Cryptography
Symmetric Ciphers
Asymmetric Ciphers
Block
Stream
Factoring
RSA
Logarithms
Diffie Hellman
ElGamal
ECC
DSA
Message/Data Integrity
Message Digests (Hashes)
Message Authentication Codes
Hybrid
Asymmetric: Authentication, Key Exchange
Symmetric: Bulk Data Encryption (Session Key)