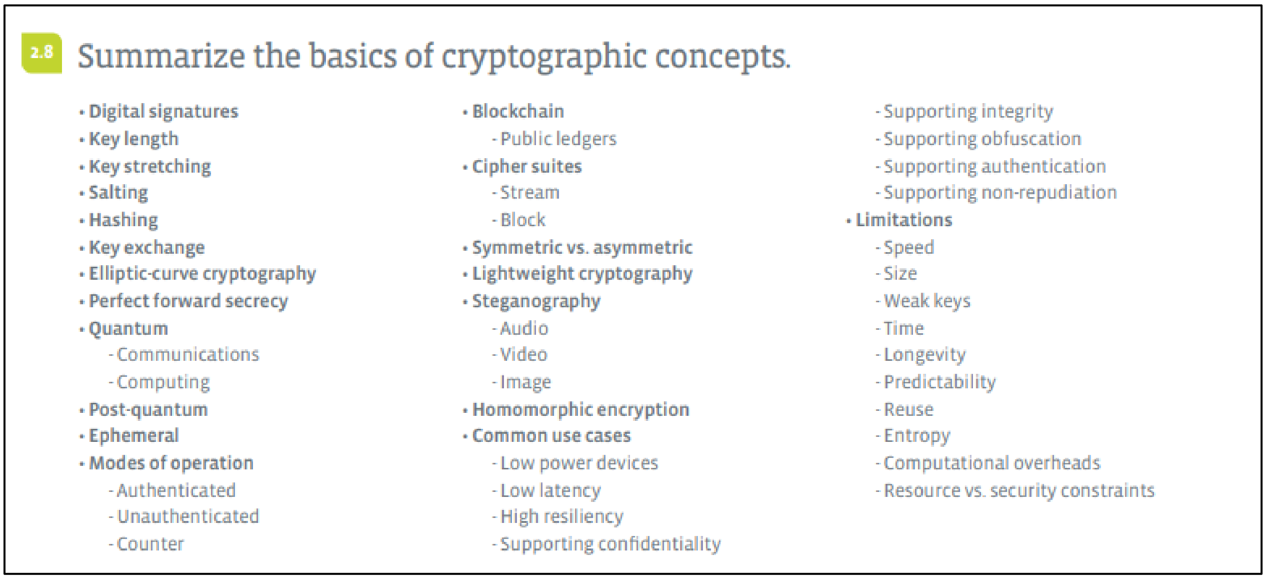
Ford’s Notes: Cryptographic Concepts

From the CompTIA Security+ exam objectives.



2.8 Summarize the basics of cryptographic concepts.

- Digital signatures
- Key length
- Key stretching
- Salting
- Hashing
- Key exchange
- Elliptic-curve cryptography
- Perfect forward secrecy
- Quantum
  - Communications
  - Computing
- Post-quantum
- Ephemeral
- Modes of operation
  - Authenticated
  - Unauthenticated
  - Counter
- Blockchain
  - Public ledgers
- Cipher suites
  - Stream
  - Block
- Symmetric vs. asymmetric
- Lightweight cryptography
- Steganography
  - Audio
  - Video
  - Image
- Homomorphic encryption
- Common use cases
  - Low power devices
  - Low latency
  - High resiliency
  - Supporting confidentiality
  - Supporting integrity
  - Supporting obfuscation
  - Supporting authentication
  - Supporting non-repudiation
- Limitations
  - Speed
  - Size
  - Weak keys
  - Time
  - Longevity
  - Predictability
  - Reuse
  - Entropy
  - Computational overheads
  - Resource vs. security constraints

Goals of Cryptography

Confidentiality – protecting data from disclosure.

Integrity – protecting data from alteration or changes.

Authenticity – an assessment of whether data has been changed.

Non-repudiation – linking data to a subject (a user).

Cryptographic Terms

Cleartext or Plaintext - information that is stored or sent in an unencrypted form. It is already in its expected form, consumable and readable.

Ciphertext - encrypted text transformed from plaintext using an encryption algorithm.

Cipher - are systems for encrypting and decrypting data.

Encrypt (or encode) – applies cryptography techniques to some data.

Decrypt (or decode) – uses cryptography on some ciphertext.

Keys or Secrets - is the piece of information or parameter that is used to encrypt and decrypt messages in a symmetric, or secret-key, encryption.

Cryptographic Primitive - A low-level mathematical or cryptographic algorithm used as a basic building block for higher-level cryptographic algorithms.

Cryptographic System (Cryptosystem) - Most practical cryptographic systems combine two elements:

- A process or algorithm which is a set of rules that specify the mathematical steps needed to encrypt or decrypt data.
- A cryptographic key (a string of numbers or characters), or keys.

Cryptanalysis - the art or process of deciphering coded messages without being told the key.

Nonce - random or pseudo-random number that authentication protocols attach to message to increase entropy. Examples of nonces include:

- **Initialization vector (IV)** - a nonce used along with a secret key for data encryption, and
- **Salt** - a nonce added to the hashing function to protect passwords.

Cryptographic Concepts

Hashing - 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. 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 in a human readable format.

Symmetric Encryption - 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.

Block and Stream Ciphers - Modern 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 don’t 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.

Asymmetric Encryption - 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. 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.

Common asymmetric encryption algorithms include RSA (Rivest Shamir Adleman), ECC (Elliptic-curve cryptography), and TLS/SSL (Transport Layer Security / Secure Sockets Layer).

Asymmetric cryptography is sometimes referred to as Public Key cryptography as it defines a key system uses a pair of keys.

Keys and Key Space - A cryptographic **key** is a string of bits used by a cryptographic algorithm to transform plain text into cipher text or vice versa. This key remains a secret and ensures secure communication.

The **key space** is the set of all valid, possible, distinct keys of a given cryptosystem. Short keys produce a smaller key space but have potentially lower processing (compute) and storage requirements. Larger or longer keys produce a larger key space but potentially increase compute and storage requirements.

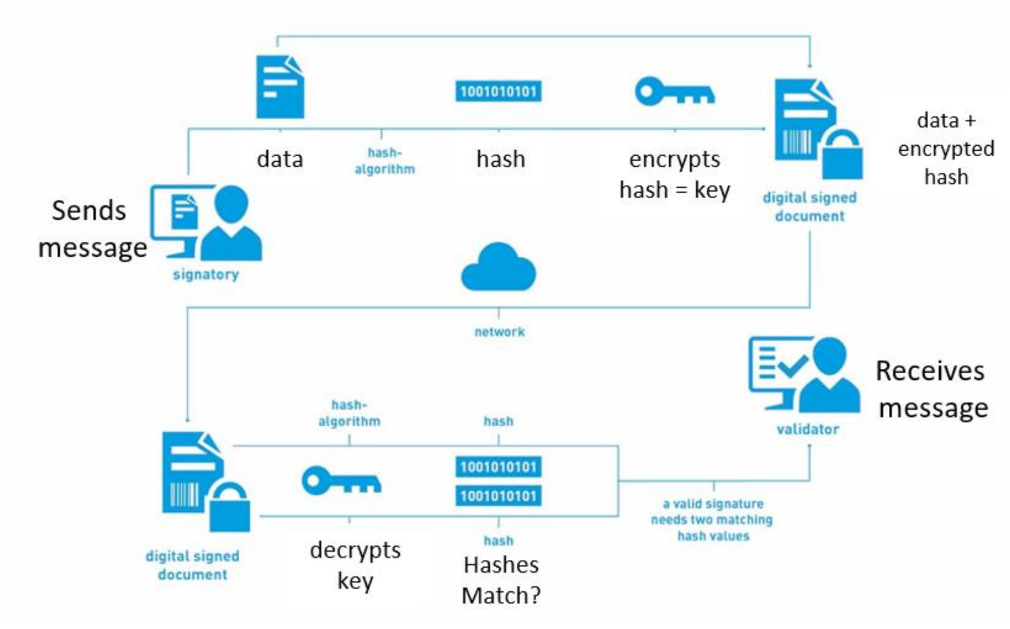
Hybrid Encryption - Hybrid Encryption combines the efficiency of symmetric encryption with the convenience of public-key (asymmetric) encryption. Only users with the private key can decrypt the data. To encrypt a message, a fresh symmetric key is generated and used to encrypt the plaintext data. The recipient’s public key is used to encrypt the symmetric key only. The final ciphertext consists of the symmetric ciphertext and the encrypted symmetric key.

RSA – The RSA algorithm is based on the math that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private key is also derived from the same two prime numbers. If a threat actor can factorize the large number, the private key is compromised. Given this RSA encryption strength totally lies on the key size; doubling or tripling the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long.

Elliptic Curve Cryptography (ECC) - is a key-based technique for encrypting data. ECC is an asymmetric cryptographic algorithm that uses pairs of public and private keys for decryption and encryption of web traffic. ECC, an alternative technique to RSA, is a powerful cryptography approach. It generates security between key pairs for public key encryption by using the mathematics of elliptic curves. ECC relies upon the difficulty of the Elliptic Curve Discrete Logarithm Problem (ECDLP).

Applied Cryptography

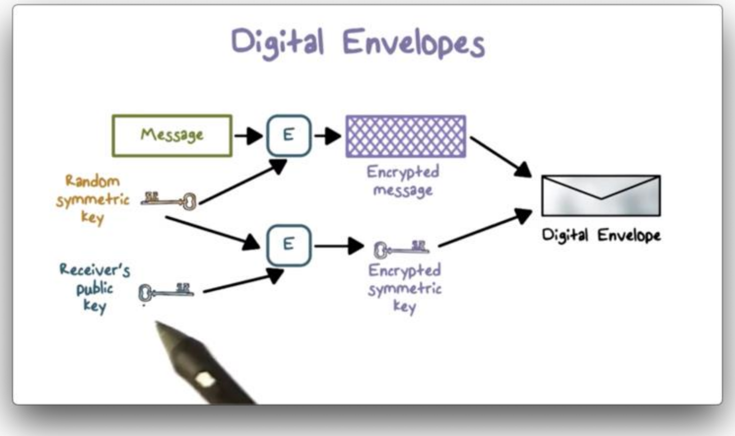
Digital Signature



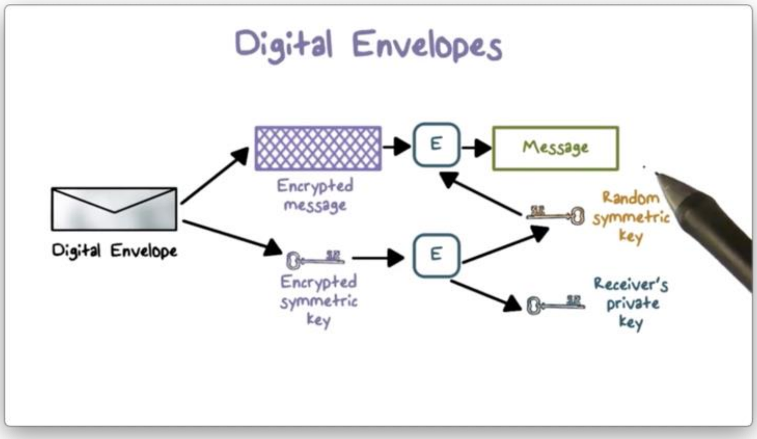
Digital signatures validate the authenticity and integrity of a message but do not provide confidentiality.

Digital Envelope

The process of inserting a message into a digital envelope uses the recipient’s public key and a random symmetric key. The message is encrypted using the random key. The random key is encrypted using the recipients’ public key. The encrypted key and the encrypted message are wrapped together in the digital envelope.



The recipient receives a wrapped envelope containing an encrypted message and the encrypted key. Retrieving a message from a digital envelope the recipient uses their private key to decrypt the random symmetric key. That random key is then used to decrypt the message.



Digital envelopes provide confidentiality and validate the authenticity and integrity of a message.

Public Key Infrastructure - The Public key infrastructure (PKI) is the set of hardware, software, policies, processes, and procedures required to create, manage, distribute, use, store, and revoke digital certificates and public-keys. PKIs are the foundation that enables the use of technologies, such as digital signatures and encryption, across large user populations.

Digital Certificates - A digital certificate is a file or electronic password that proves the authenticity of a device, server, or user using cryptography and the public key infrastructure (PKI). Digital certificate authentication helps organizations ensure that only trusted devices and users can connect to their networks. Another common use of digital certificates is to confirm the authenticity of a website to a web browser, which is also known as a secure sockets layer or SSL certificate. A digital certificate contains identifiable information, such as a user’s name, company, or department and a device’s Internet Protocol (IP) address or serial number. Digital certificates contain a copy of a public key from the certificate holder, which needs to be matched to a corresponding private key to verify it is real. A public key certificate is issued by certificate authorities (CAs), which sign certificates to verify the identity of the requesting device or user.

Digital certificates are often stored and distributed as files using x.509 base64 (ASCII) encoding. You get one of those in a zip file downloaded from your computer or receive such file from the certificate authority.

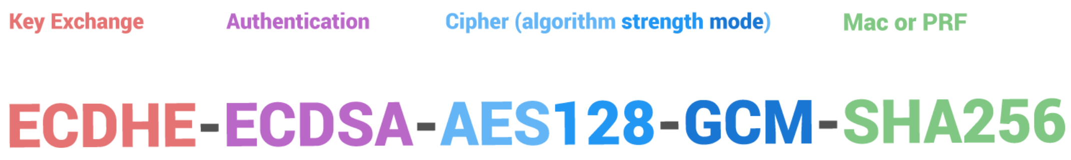
Forward Secrecy - Perfect Forward Secrecy (PFS), also known as forward secrecy, is a style of encryption that enables short-term, private key exchanges between clients and servers. PFS can be found within transport layer security (SSL/TLS) and prevents hackers from decrypting data from other sessions, past or future, even if the private keys used in an individual session are stolen at some point.

Diffie Hellman - The Diffie-Hellman key exchange allows two parties who have not previously met to securely establish a key which they can use to secure their communications.

With some key exchange methods that don’t support PFS, the same key will be generated if the same parameters are used on either side. This can cause problems as an intruder could guess the key, or even where the key was static and never changed. Using **ephemeral keys** a different key is used for each connection, so any leakage of any long-term key would not cause all the associated session keys to be breached. The problem with the Diffie-Hellman method is that the keys are not ephemeral, so it should be avoided for generating keys.

A **trapdoor function** is a function that is easy to compute in one direction yet believed to be difficult to compute in the opposite direction (finding its inverse) without special information, called the "trapdoor". An example of a trapdoor is a padlock. You can close a padlock without having the key. You must have the key and be able to turn the key to open the padlock. Trapdoor functions are a means of comparing different cryptosystems.

Cipher Suite - Cipher suites are sets of instructions that enable secure network connections through Transport Layer Security (TLS), often still referred to as Secure Sockets Layer (SSL). Behind the scenes, these cipher suites provide a set of algorithms and protocols required to secure communications between clients and servers.


Key Exchange
Authentication
Cipher (algorithm strength mode)
Mac or PRF
ECDHE-ECDSA-AES128-GCM-SHA256

To initiate an HTTPS connection, the two parties; often a web server and a client browser, perform an SSL handshake. During the handshake process the two parties agree on a mutual cipher suite. The cipher suite is then used to negotiate a secure HTTPS connection. During the handshake, the client and the web server will agree to use:

- A key exchange algorithm, to determine how symmetric keys will be exchanged
- An authentication or digital signature algorithm, which dictates how server authentication and client authentication (if required) will be implemented
- A bulk encryption cipher, which is used to encrypt the data
- A hash/MAC function, which determines how data integrity checks will be carried out

These ciphers are required at various points of the connection to perform authentication, key generation and exchange, and a checksum to ensure integrity. To determine what specific algorithms to use, the client and the web server start by mutually deciding on the cipher suite to be used.

Cryptographic Modes of Operation

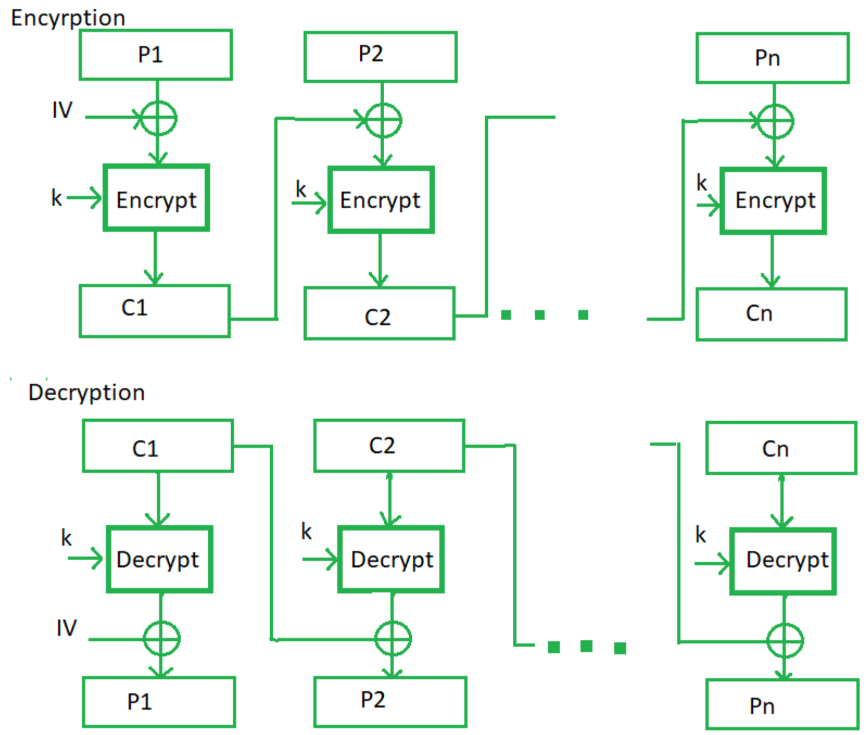
CBC (Cipher Block Chaining) - Encryption algorithms are divided into two categories based on the input type, as a block cipher and stream cipher. Block cipher is an encryption algorithm that takes a fixed size of input say b bits (b bits equals a block) and produces a ciphertext of b bits again. If the input is larger than b bits, it can be divided further. For different applications and uses, there are several modes of operations for a block cipher.

Electronic code book (ECB) is the easiest block cipher. It is easier because of direct encryption of each block of input plaintext and output is in form of blocks of encrypted ciphertext. Generally, if a message is larger than b bits in size, it can be broken down into a bunch of blocks and the procedure is repeated.

Cipher block chaining or CBC is an advancement made on ECB. In CBC, the previous cipher block is given as input to the next encryption algorithm after XOR with the original plaintext block.

An XOR is a Boolean logic operation that is widely used in cryptography. XOR compares two input bits and generates one output bit. The logic is simple. If the bits are the same, the result is 0. If the bits are different, the result is 1.

Using CBC, a cipher block is produced by encrypting an XOR output of the previous cipher block and present plaintext block.

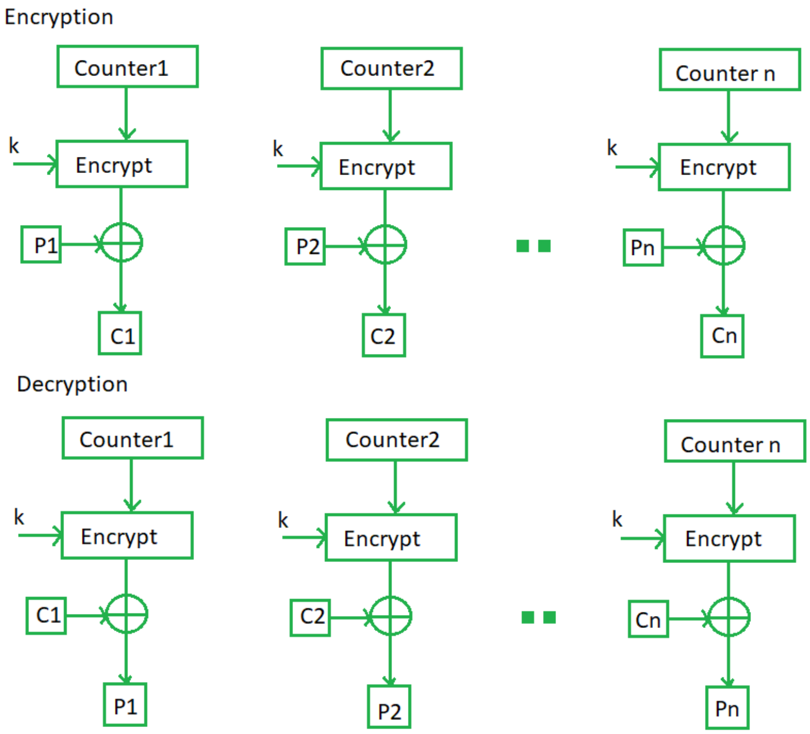


In the diagram above the plaintext is divided into blocks P1, P2, P3. K is the key. IV is an initialization vector (random value) used in the first block since there is no previous block to XOR. C1, C2, C3 represents ciphertext. Note that C1 is XOR’ed with P2 and the result is encrypted.

Note: That while encrypting each block (P2) is dependent on the block that came before it (C1).

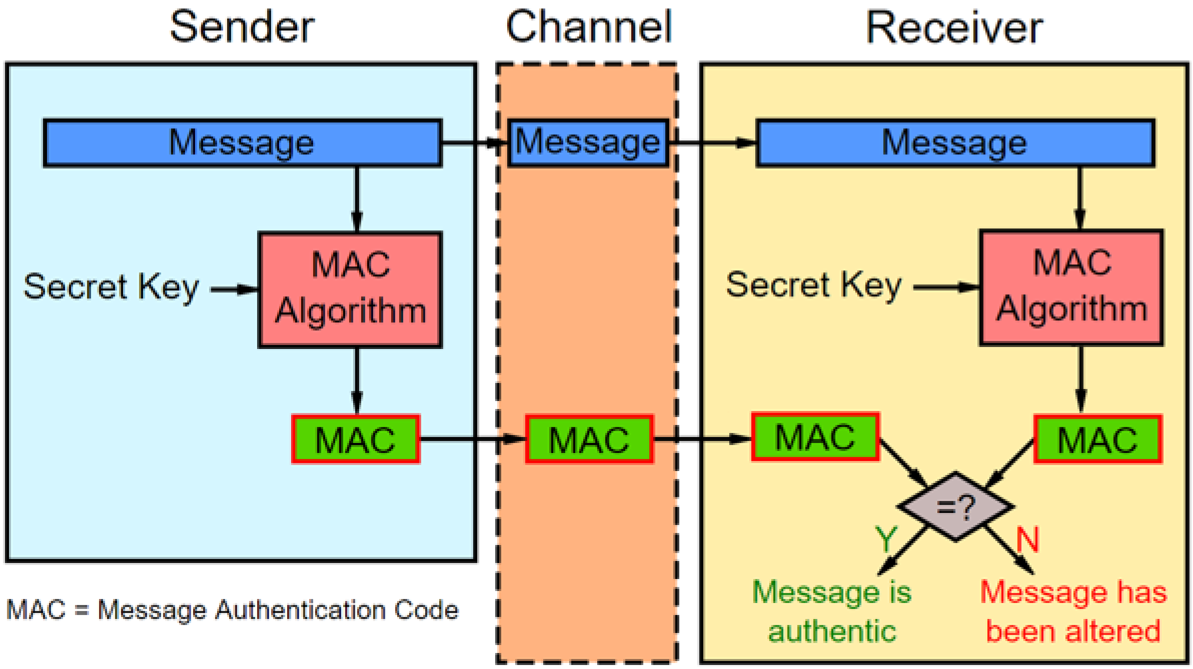
Counter Mode (CTR) - The Counter Mode or CTR is a simple counter-based block cipher implementation. Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block. Note: the counter value will be incremented by 1. Unlike

CBC, each block in CTR mode doesn't rely on previously processed data and the algorithm can be implemented in parallel over multiple processors.



In the diagram above the system starts with a Counter value (Counter1) which is encrypted with the key (k). Each block of cleartext (P1, P2, P3,...) is then XOR'ed with the now encrypted counter to produce ciphertext (C1, C2, C3,...).

Message Authentication Code (MAC) – also referred to as a tag, is used to authenticate the origin and nature of a message. In order to produce a MAC tag some cleartext message and a key are arguments to a MAC algorithm.



The first step in the MAC process is the establishment of a secure channel between the receiver and the sender (creating a pre-shared key).

To encrypt a message, the MAC system uses a MAC algorithm, which uses a symmetric key and the plain text message being sent. The MAC algorithm then generates authentication tags of a fixed length by processing the message. The resulting computation is the message's MAC. The MAC algorithm behaves like a hash function: a minor change in the message or in the key results to totally different MAC value.

This MAC is then appended to the message and transmitted to the receiver. The receiver computes the MAC using the same algorithm. If the resulting MAC the receiver computes equals the one sent by the sender, the message is verified as authentic and not tampered with.

The MAC uses a secure key only known to the sender and the recipient. Without this information, the recipient will not be able to open, use, read, or even receive the data being sent. If the data is to be altered between the time the sender initiates the transfer and when the recipient receives it, the MAC information will also be affected.

When the recipient attempts to verify the authenticity of the data using the MAC, the key will not work, and the result will not match that of the sender. When this kind of discrepancy is detected, the data packet can be discarded, protecting the recipient’s system.

There are many algorithms for calculating message authentication codes (MAC); however, the most popular are based on hashing functions (HMAC) or the use of symmetric encryption (Cipher-based MAC - CMAC and Galois MAC - GMAC).

Keyed-Hash Message Authentication Code (HMAC) – is a MAC algorithm that produces a one-way hash used to create a unique MAC value for every message sent. The input parameters can have various values assigned and making them very different from each other may produce a higher level of security.

Unauthenticated Encryption – Think about using symmetric (single key) algorithms. What if the secret key leaked and becomes known by others? The use of a single secret key cannot prove integrity. The leakage of keys makes an unauthenticated cryptographic system vulnerable to insertion and modification attacks.

Authenticated Encryption (AE) – An authenticated mode of operation simultaneously assures the confidentiality and authenticity of data. This can be accomplished using MAC (described earlier).

Implementations of AE may be vulnerable to padding oracle attacks; where the "oracle" (a server) leaks data about the padding of an encrypted message. This gives an attacker information for their cryptanalysis about the true size of the message contained in N blocks.

Authenticated Encryption with Additional Data (AEAD) - The AEAD primitive is the most common primitive for data encryption. The Additional Data in AEAD is taken from the data being encrypted (the content of the message). For example, if we are transmitting patient information the additional data might be the patient ID. Where a MAC is a tag generated by a hashing algorithm; AEAD uses some feature of data that is transferred. AEAD has the following properties:

- Secrecy: Nobody will be able to get any information about the encrypted plaintext, except the length.
- Authenticity: Without the key it is impossible to change the plaintext underlying the ciphertext undetected.
- Symmetric: Encrypting the message and decrypting the ciphertext is done with the same key.
- Randomization: The encryption is randomized. Two messages with the same plaintext will not yield the same ciphertext. This prevents attackers from knowing which ciphertext corresponds to a given plaintext.

Limitations of Cryptography

Entropy – is a measure of randomness. If a system exhibits low entropy, then it is not generating truly random numbers and that could be used in cryptanalysis to break the cipher. To create random numbers for cryptographic primitives like nonces, the crypto system needs a source of seemingly random input from outside the machine. Typically, operating systems are primarily responsible for supplying sources of entropy to programs. A lack of good entropy can leave a crypto system vulnerable and unable to encrypt data securely.

Frequency Analysis – is a cryptanalysis technique based on the study of the frequency of letters or groups of letters in a ciphertext. The method is used as an aid to breaking substitution ciphers (e.g. mono-alphabetic substitution cipher, Caesar shift cipher, Vatsyayana cipher). Frequency analysis consists of counting the occurrence of each letter in a text. Frequency analysis is based on the knowledge that in most languages for any given text; certain letters and combinations of letters occur with varying frequencies.

Lightweight cryptography - also known as lightweight encryption, is a form of encryption designed for resource-constrained devices. Lightweight encryption technology uses less memory, fewer computing resources, and a smaller amount of power to provide secure solutions for limited resources in a network.

Steganography

Steganography is a method in which secret message is hidden in a cover media. Steganography means covered writing. Steganography is less popular than Cryptography.

In steganography, structure of data is not usually altered. While in cryptography, structure of data is altered.

In steganography, the fact that a secret communication is taking place is hidden. While in cryptography only secret message is hidden.

In steganography, not much mathematical transformations are involved. Cryptography involves the use of number theory or mathematics (factoring, logarithms) to modify data.

A classic steganography primitive is the Least Significant Bit (LSB) Algorithm. LSB is used to conceal the existence of secret data inside a “public” cover. The LSB algorithm hides messages inside a cover image by replacing the least significant bits of image (the eight bit of each byte) with the bits of message to be hidden. The least significant bit in each byte can be used to store a bit of the secret message without significantly altering the original file.

Image - Hiding data by taking the cover object as the image is known as image steganography. In digital steganography, images are a widely used cover source because there are a huge number of bits present in the digital representation of an image. For hiding data within the images, the LSB (Least Significant Byte) approach is generally used. An image file is a file that shows multiple colors and intensities of light on different location of an image. The best type of image files to hide data inside is a 24 Bit BMP (Bitmap) image.

Audio – In audio steganography, the secret message is embedded into an audio signal which alters the binary sequence of the corresponding audio file. Hiding secret messages in digital sound is a much more difficult process when compared to others, such as Image Steganography. Audio steganography is accomplished using LSB and Enhanced Audio Steganography (EAS).

Video - In Video Steganography you can hide data into the digital video format. The advantage of using video is a large amount of data can be hidden inside and the fact that it is a moving stream of images and sounds. You can think of this as the combination of Image Steganography and Audio Steganography.

Homomorphic Encryption

Homomorphic encryption is the conversion of data into ciphertext that can be analyzed and worked with as if it were still in its original form. This protects data at rest and in use. Homomorphic encryption implementations allow complex mathematical operations to be performed on encrypted data without compromising the encryption. Data can remain confidential while it is processed, enabling useful tasks to be accomplished with data residing in untrusted environments.

The problems with implementing fully homomorphic encryption in that it is very specialized and very slow. There are some partial homomorphic encryption algorithms that will work for N number of queries but after N that results are no longer reliable.

Quantum Computing - is an area of study focused on the development of computer-based technologies centered around the principles of quantum theory (Qubits). Quantum theory explains the nature and behavior of energy and matter on the quantum (atomic and subatomic) level. Qubits can express states (0 or 1) but also the relationship between states (as in the current bit is 1 but the last bit setting was 0).

Quantum communications - The use case for qubits from a cybersecurity perspective is that if a hacker tries to observe them in transit, their super-fragile quantum state “collapses” to either 1 or 0. This means a hacker can’t tamper with the qubits without leaving behind a telltale sign of the activity.

Post quantum cryptography - The goal of post-quantum cryptography (also called quantum-resistant cryptography) is to develop cryptographic systems that are secure against both quantum and classical computers and can interoperate with existing communications protocols and networks.

Blockchain - A blockchain is a shared database that is distributed among the nodes of a computer network. It stores the database in a digital, ledger like format. The innovation of blockchain is that it guarantees the security of records of data and generates trust without the involvement of a third party. Every transaction recorded to the block is signed by the sender by digital signature and ensures that the data is not corrupted. The blockchain is an incorruptible digital ledger of economic transactions. As the new block enters the network it is chained to the previous block and makes the data chained together.

Cryptographic Limitations

Speed - Speed (often very platform-dependent): Cycles/byte averages for large messages or multiple uses with the same key;

Size - The strength of an encryption algorithm is measured in key sizes, not bits of the algorithm. The design of an algorithm determines its strength. In general, a bigger key size makes an algorithm more difficult to brute-force, making it stronger.

Weak keys – A limitation based on history that vendors of embedded products, such as network devices, IoT devices, and modems too often left hardcoded SSH keys and HTTPS server certificates in their devices to enable web access to the devices and for use by other protocols such as EAP/802.1X or FTPS. These keys have been embedded, essentially “baked in” to the firmware image (operating system) of devices and are mostly used for providing HTTPS and SSH access to the device. This is a problem because all devices that use that firmware use the exact same keys. Since these keys and certificates are the same across multiple products, they are relatively easy to exploit.

Time - Time to set up a new key. The time needed to perform cryptographic functions is often related to the Size limitation (described above).

Longevity – How long has a specific cryptographic algorithm or primitive been publicly available? How well known and implemented is the algorithm or primitive?

Reuse - Despite recommendations and the inherent security risks, many vendors are motivated to reuse cryptographic keys, because key reuse can reduce:

- storage requirements for certificates and keys,
- the costs of key certification,
- the certificate verification time, and
- the footprint of cryptographic code and development effort.

Entropy – High entropy means that there is much randomness in the subject. Low entropy means there are discernable patterns.

Computational overheads - The more secure the encryption used and the higher the key length, the more processing power and memory that the server will need.

Resource vs. security constraint - If there are not enough resources on the server, it could be vulnerable to a resource exhaustion attack, which causes the systems to hang or even crash—it is like a denial-of-service attack. We must strike a balance between the hardware resources that the server has and the amount of processing power.