1 Key exchange

- TLS_RSA
- TLS_DH_anon (anonymous)
- TLS_PSK (pre shared key)
- TLS_SRP (secure remote password)

Only these methods provide **forward secrecy** since the public key changes for every instance of the protocol. The last 'e' stands for *ephemeral*.

- TLS_DHE
- TLS_ECDHE

The following methods provide no authentication of server and user, hence suffers from MITM attacks, rarely used

- TLS_DH
- TLS_ECDH

Remark 1. *These methods can be combined together for instance we can have DHE_RSA, ECDHE_PSK, PSK_RSA etc.*

1.1 Secure Remote Password

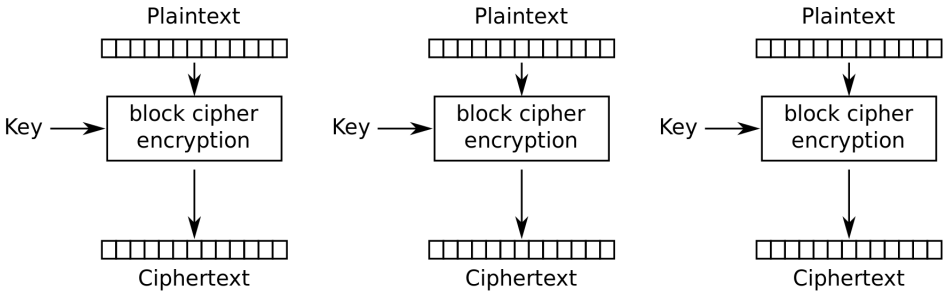
TLS_SRP provides **mutual authentication** while TLS with server certificates only authenticates the server to the client.
Moreover the user does not need to check the URL being certified since if the server does not know the password the connection can not be established. This prevent phising.
Note that using password based authentication does not require reliance of certification authorities.

2 Block cipher modes

A block cipher by itself is only suitable for the secure cryptographic transformation (encryption or decryption) of one fixed-length group of bits called a block. A mode of operation describes how to repeatedly apply a cipher's single-block operation to securely transform amounts of data larger than a block.

Most modes require a unique binary sequence, often called an initialization vector (IV), for each encryption operation. The IV has to be non-repeating and, for some modes, random as well. The initialization vector is used to ensure distinct ciphertexts are produced even when the same plaintext is encrypted multiple times independently with the same key.

2.1 Electronic Codebook (ECB)



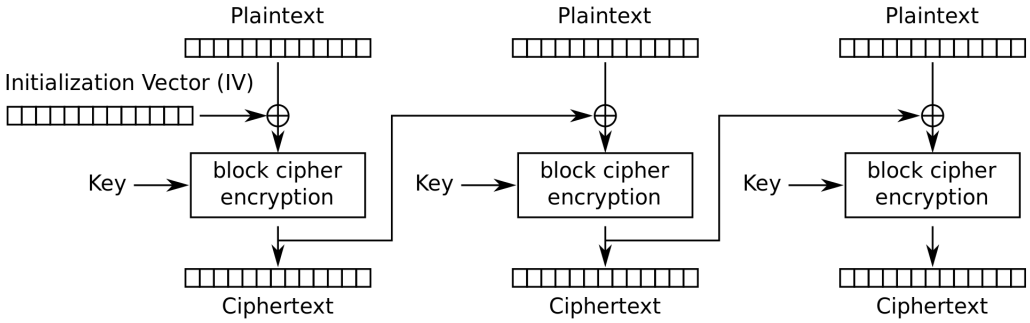
Electronic Codebook (ECB) mode encryption

The message is divided into blocks, and each block is encrypted separately. The disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks; thus, it does not hide data patterns well. In some senses, it doesn't provide serious message confidentiality, and it is not recommended for use in cryptographic protocols at all

Example 1: you see patterns in an image
Example 2: headers are standard hence the attacker has pairs (PT,CT) for free!

Remark 2. *it is good to store short information such as password*

2.2 Cipher Block Chaining (CBC)

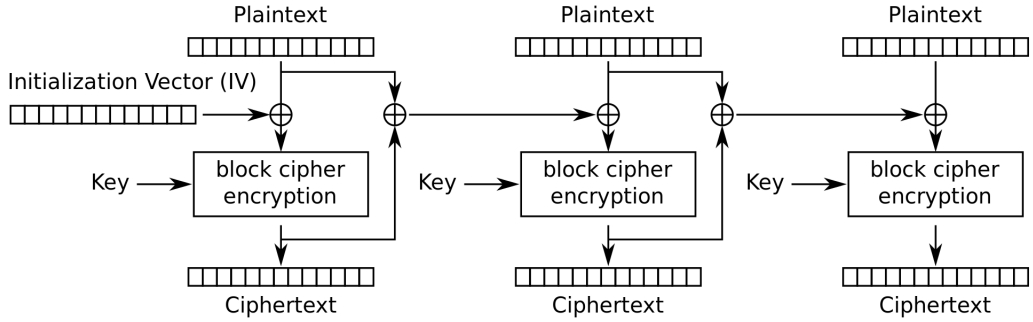


Cipher Block Chaining (CBC) mode encryption

In CBC mode, each block of plaintext is XORed with the previous ciphertext block before being encrypted. This way, each ciphertext block depends on all plaintext blocks processed up to that point. To make each message unique, an initialization vector must be used in the first block.

Remark 3. *usually the IV is encrypted with ECB.*

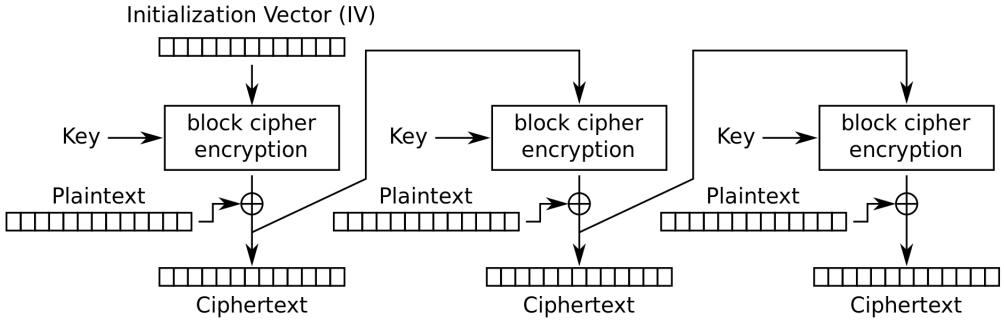
2.3 Propagating Cipher Block Chaining (PCBC)



Propagating Cipher Block Chaining (PCBC) mode encryption

In PCBC mode, each block of plaintext is XORed with both the previous plaintext block and the previous ciphertext block before being encrypted. As with CBC mode, an initialization vector is used in the first block.

2.4 Cipher Feedback (CFB)



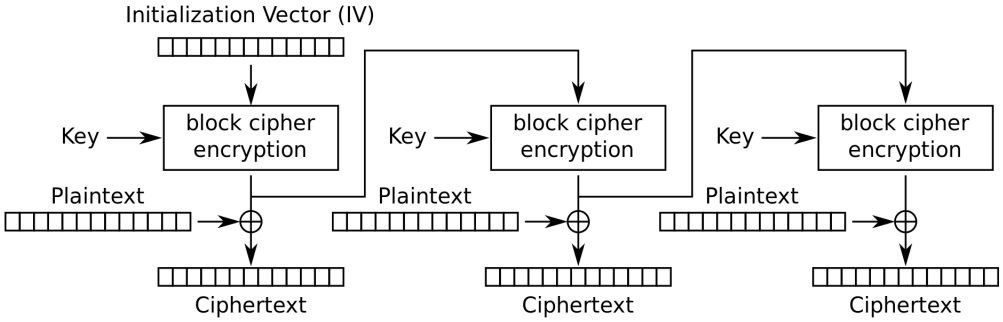
Cipher Feedback (CFB) mode encryption

The Cipher Feedback (CFB) mode, a close relative of CBC, makes a block cipher into a self-synchronizing stream cipher. Operation is very similar; in particular, CFB decryption is almost identical to CBC encryption performed in reverse. By definition of self-synchronising cipher, if part of the ciphertext is lost (e.g. due to transmission errors), then receiver will lose only some part of the original message (garbled content), and should be able to continue correct decryption after processing some amount of input data.

Remarks:

- 8 bits encrypted at a time. Then there is a shift in the IV. The image above is too generic.
- Another important problem is the error propagation: if I receive for instance $\bar{c}_1, c_2.., c_n$ then I can not decrypt anything! This problem is solved with OFB because the 8-bits are used in the next IV before being XORed with the PT
- It is used for video.

2.5 Output Feedback (OFB)

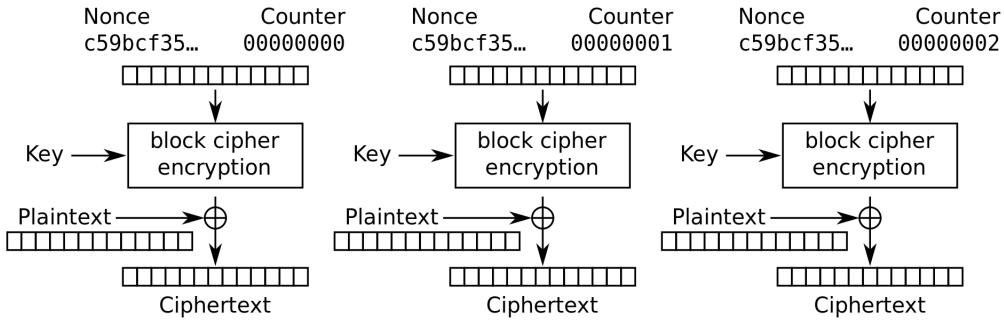


Output Feedback (OFB) mode encryption

The Output Feedback (OFB) mode makes a block cipher into a synchronous stream cipher. It generates keystream blocks, which are then XORed with the plaintext blocks to get the ciphertext. Just as with other stream ciphers, flipping a bit in the ciphertext produces a flipped bit in the plaintext at the same location. This property allows many error correcting codes to function normally even when applied before encryption.

Remark 4. *this modes corrects the error propagation of CFB however it is still not parallelizable. Here it comes CTR.*

2.6 Counter (CTR)



Counter (CTR) mode encryption

Counter mode turns a block cipher into a stream cipher. It generates the next keystream block by encrypting successive values of a "counter". The counter can be any function which produces a sequence which is guaranteed not to repeat for a long time, although an actual increment-by-one counter is the simplest and most popular.

Remark 5. *it is used in ATM, IPSEC.*

Remark 6. *in this mode decryption is performed by swapping around PT and CT hence the block cipher is used only in ecryption mode!*

2.7 XTS

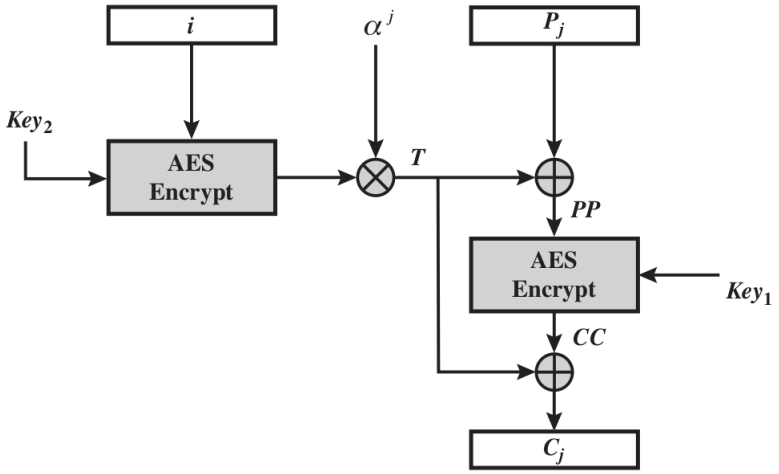
The scenario with hard disk is completely different from a TLS connection. With an hard disk there is a huge quantity of information. In an hard diks a sector is made of blocks.

$$Sector_1 : [block_1, ..., block_n]$$
$$Sector_2 : [block_1, ..., block_n]$$

Problem 1: the same info can be saved in different blocks or in different sectors in the same way. It is the same issue of ECB. To overcome this problem the encryption depends on the sector number i and block number j.

Definitions:

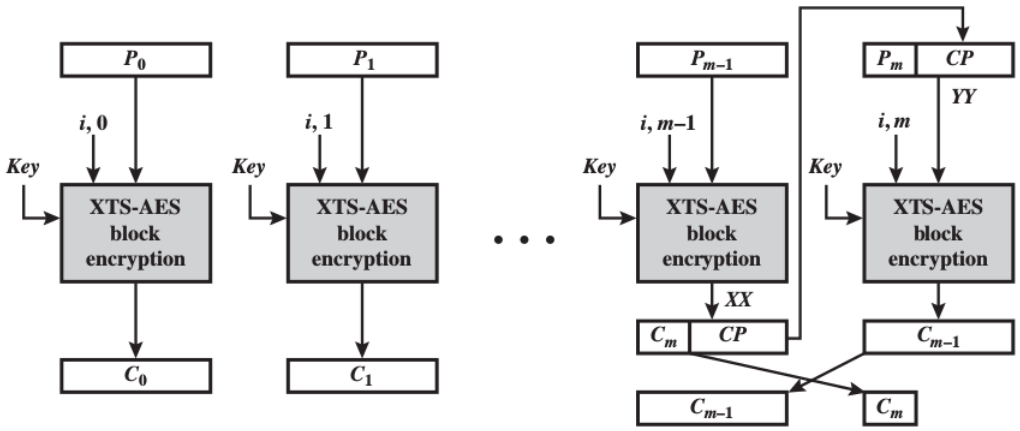
- $Key = Key_1 || Key_2$
- P_j: j-th block of PT. All blocks except possibly the last one have length 128 bits.
- C_j: j-th block of CT. All blocks except possibly the last one have length 128 bits.
- i: sector number
- j: block number
- α primitive element in $GF(2^1 28)$
- $\otimes$: modular multiplication of two polynomials modulo $x^{128} + x^7 + x^2 + x + 1$



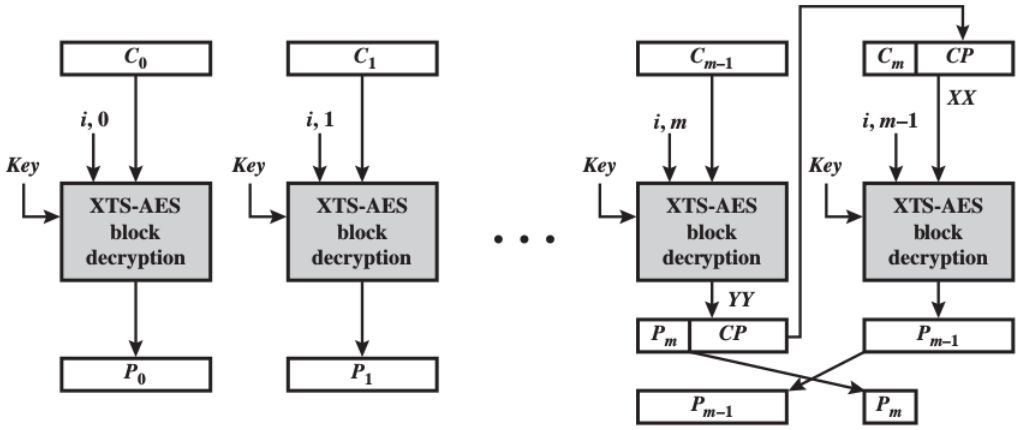
XTS_AES block operations:

- $T = E(K_2, i) \otimes \alpha^j$
- $PP = P \oplus T$
- $CC = E(K_1, PP)$
- $C = CC \oplus T$

Problem 2: consider encrypting a file of 129 bits with a block size of 128 bits. How is it possible to keep safe even the last bit which is saved in a new block?



(a) Encryption



(b) Decryption

3 Cryptographic hash functions

- SHA-2, SHA-3
- Scrypt
- Argon
- BLAKE

3.1 MAC - Message Authentication Code

A MAC algorithm, sometimes called a keyed (cryptographic) hash function accepts as input a secret key and an arbitrary-length message to be authenticated, and outputs a MAC. The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content.

Remark 7. *MACs differ from digital signatures as MAC values are both generated and verified using the same secret key. This implies that the sender and receiver of a message must agree on the same key before initiating communications, as is the case with symmetric encryption. For the same reason, MACs do not provide the property of non-repudiation offered by signatures specifically in the case of a network-wide shared secret key*

4 Digital Signature

A digital signature is a mathematical scheme for demonstrating the authenticity of a digital message or documents. Digital signature provides

- **Authentication:** the recipient has reason to believe that the message was created by a known sender
- **Non repudiation:** the sender cannot deny having sent the message
- **Integrity:** the message was not altered

Some algorithms:

- RSA
- ECDSA

5 Key derivation function

6 Certificates

6.1 Certificate pinning