Galois Counter Mode(GCM)

- Born with the idea of increasing the flaws of Counter Mode which is considered the best for performance speed encryption. The CM provides no protetion against bit-flipping attacks.
- It's a block cipher mode which use Galois Field to provide authenticated encryption

- Implementations:

1) hardware → high performance in speed(low cost-low latency)
2) Software → high performance only if we use table-driven field operation

- GCM admits pipelined and parallelized implementations
- GCM is capable of acting as MAC which let to protect both a message's data integrity as well as authenticity
- GCM supports the usage of IV of arbitrary length.

Implementation

It give in input 4 elements in Encryption :

1. Secret key k with a size appropriete for the underlying block cipher
2. IV-Initialization Vector
3. Plaintext P
4. Additional authentiated data(AAD) → not encrypted

Output are: Ciphertext with same size of P, an authentication tag T with a variabel lenght

In decryption function we add also the authentication tag in input. The output could be the right Plaintext or a fail symbol which indicates that the input are not authentic.

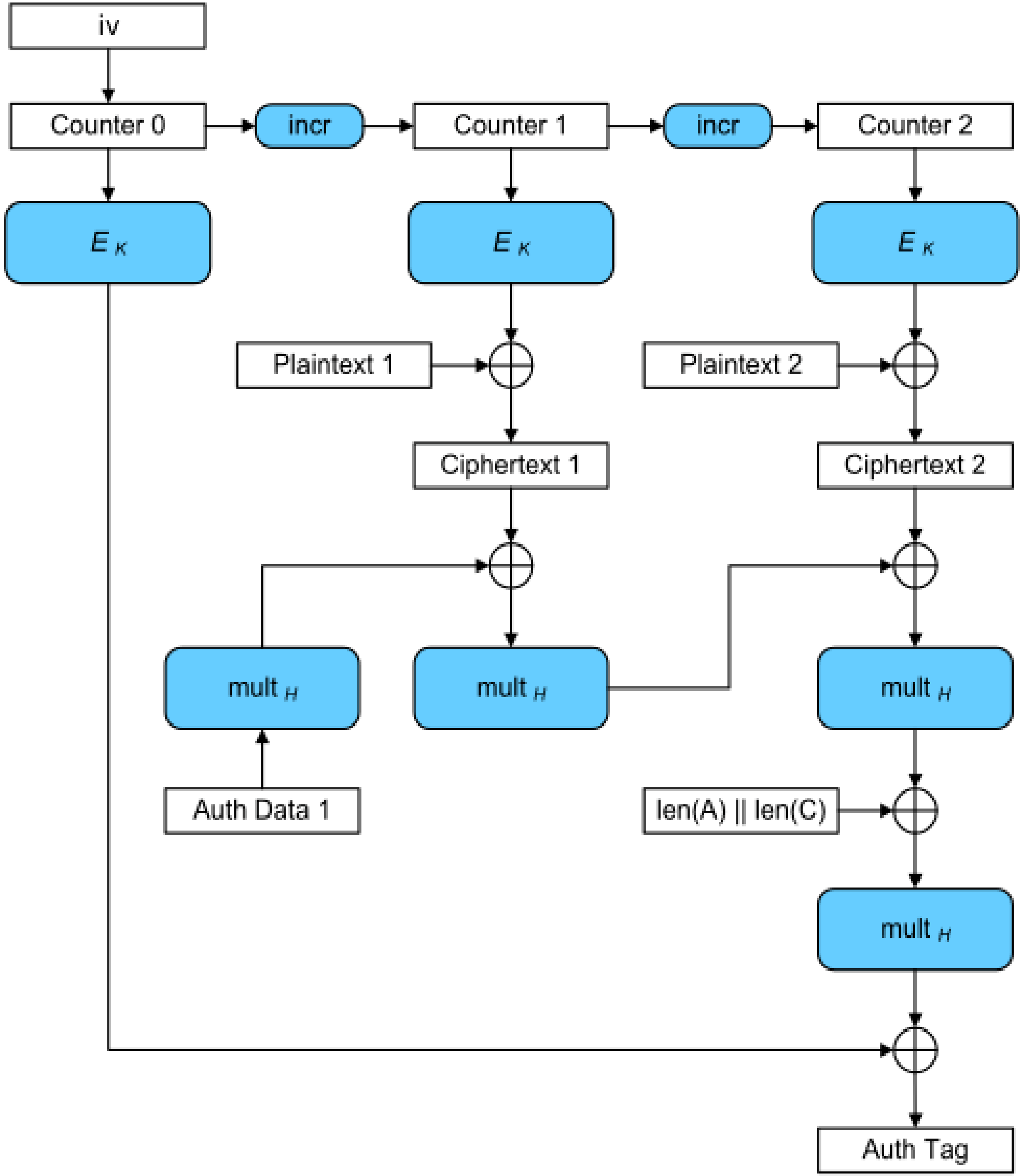
GCM functions are block cipher encryption E(k,P) and multipliation X·Y over the field GF(2^128), means X,Y€GF(2^128) and the addition is the xor operation..

The result is then encrypted, producing an authentication tag that can be used to verify the integrity of the data. The encrypted text then contains the IV, ciphertext, and authentication tag.

- ENCRYPTION SCHEMA→

- GCM combines CM of encryption with the new Galois mode of authentication. The key feature is that the Galois field multiplication used for authentication.

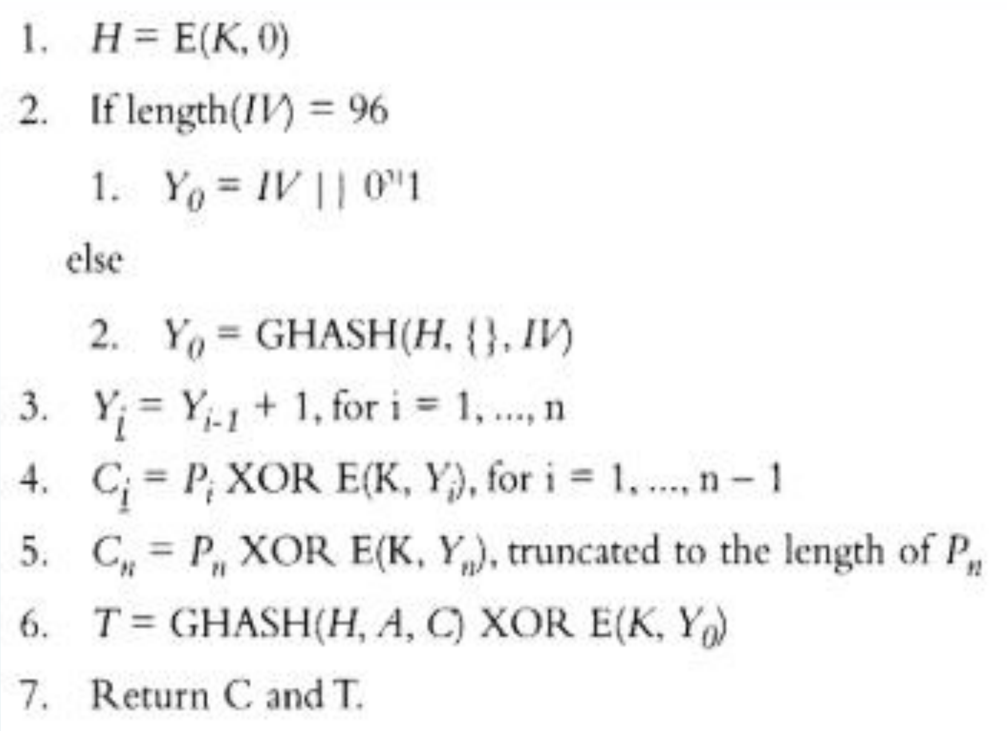
- Each blok is identify as a sequence number, then the first is combined with IV and enrypted with a blok cipher E usually Aes. Result is xored with plaintext to produe ciphertext



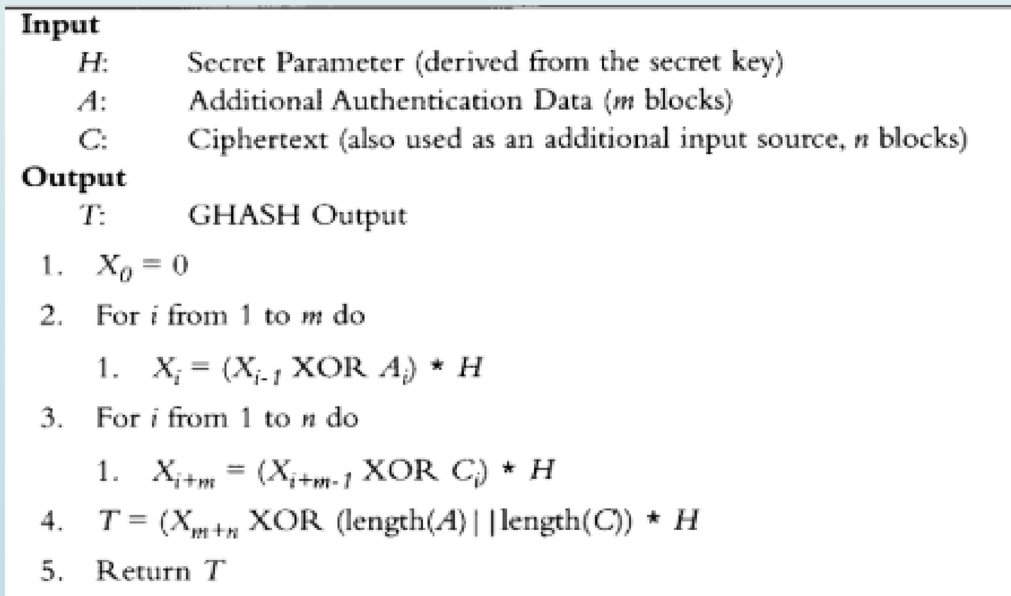
- PSEUDO-CODE ENRYPTION→

- Its security relies on the fact that the underlying block cipher cannot be distinguished from a random permutation, an assumption which is common in cryptographic designs and which appears to be valid for the AES.

- PSEUDO-CODE GHASH ROUTINE→



```
1.  $H = E(K, 0)$
2.  If length($IV$) = 96
        1.  $Y_0 = IV \,||\, 0^{31}1$
    else
        2.  $Y_0 = GHASH(H, \{\}, IV)$
3.  $Y_i = Y_{i-1} + 1$, for i = 1, ..., n
4.  $C_i = P_i$ XOR $E(K, Y_i)$, for i = 1, ..., n – 1
5.  $C_n = P_n$ XOR $E(K, Y_n)$, truncated to the length of $P_n$
6.  $T = GHASH(H, A, C)$ XOR $E(K, Y_0)$
7.  Return C and T.
```



Input

H: Secret Parameter (derived from the secret key)

A: Additional Authentication Data (*m* blocks)

C: Ciphertext (also used as an additional input source, *n* blocks)

Output

T: GHASH Output

```
1.  $X_0 = 0$
2.  For $i$ from 1 to $m$ do
        1.  $X_i = (X_{i-1}$ XOR $A_i) * H$
3.  For $i$ from 1 to $n$ do
        1.  $X_{i+m} = (X_{i+m-1}$ XOR $C_i) * H$
4.  $T = (X_{m+n}$ XOR (length($A$)||length($C$)) $* H$
5.  Return $T$
```