Authenticated Encryption

2019. 4 .9

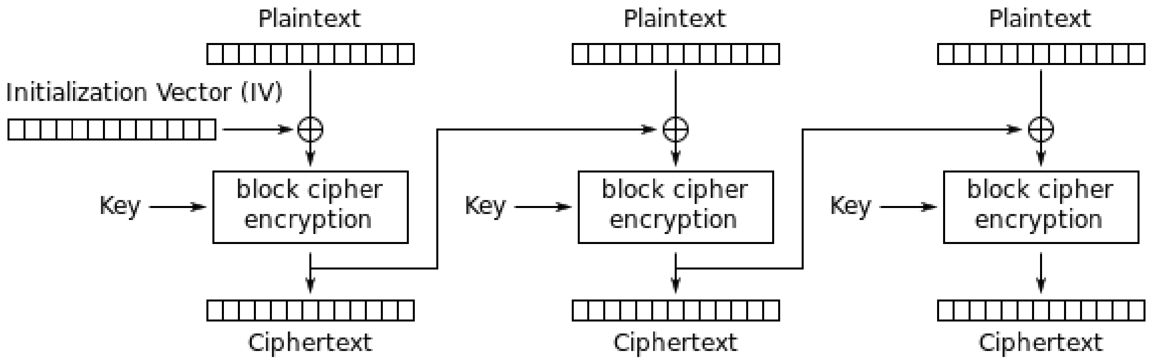
Contents

Block cipher mode for MAC?

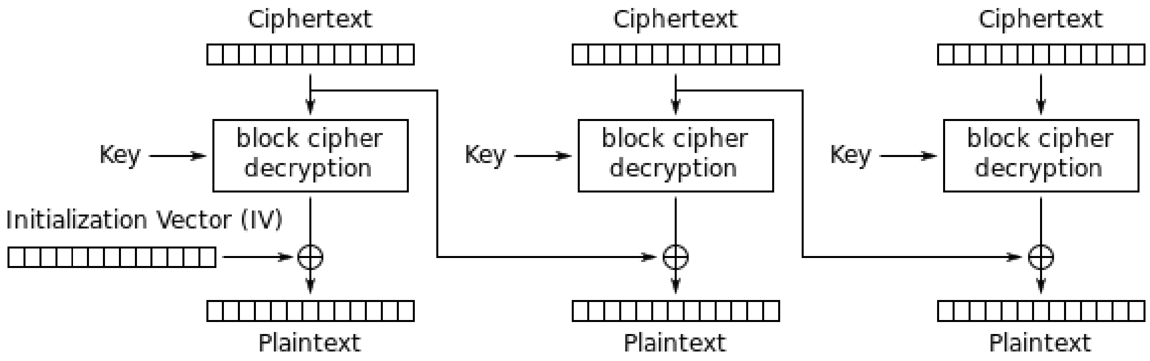
- In CBC mode, changes of any block of a plaintext affect the computation of the next block.
- Then, can we use the result of the final block as MAC?

Reminder: Block cipher operation modes

- CBC mode

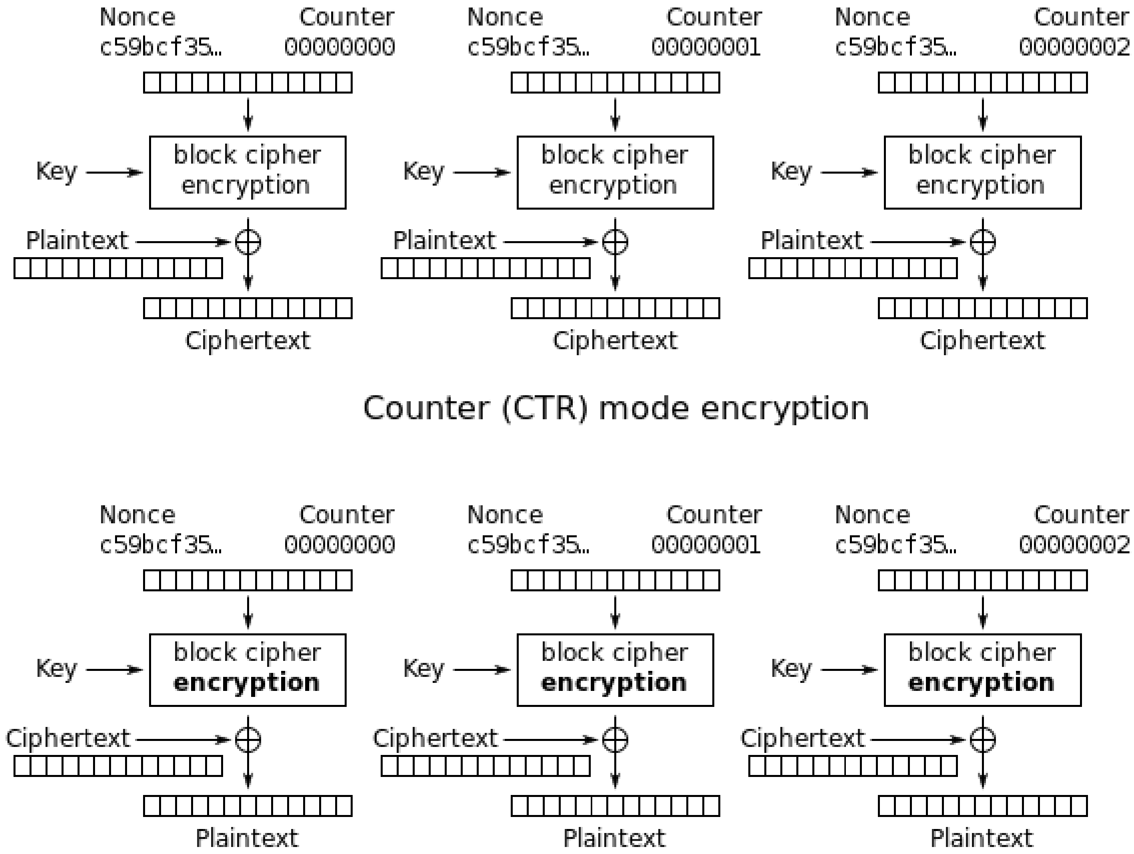


Cipher Block Chaining (CBC) mode encryption



Reminder: Block cipher operation modes

- CTR mode



CBC-MAC Computation

- MAC computation (assuming N blocks)

  $C_0 = E_K(IV \oplus P_0)$,

  $C_1 = E_K(C_0 \oplus P_1)$,

  $C_2 = E_K(C_1 \oplus P_2), \ldots$

  $C_{N-1} = E_K(C_{N-2} \oplus P_{N-1}) = MAC$

- Alice sends plaintext and MAC with IV to Bob.
- Bob does the same computation and verifies that result agrees with MAC
- Note: Bob must know the key K
  - Guarantee message integrity and authentication

Does a CBC-MAC work?

- Suppose Alice has 4 plaintext blocks
- Alice computes

  $\mathbf{C_0} = E_K(IV \oplus P_0)$, $\mathbf{C_1} = E_K(\mathbf{C_0} \oplus P_1)$,

  $\mathbf{C_2} = E_K(\mathbf{C_1} \oplus P_2)$, $\mathbf{C_3} = E_K(\mathbf{C_2} \oplus P_3) = \mathbf{MAC}$
- Alice sends IV, P_0, P_1, P_2, P_3 and **MAC** to Bob
- Suppose an attacker changes P_1 to X
- Bob computes

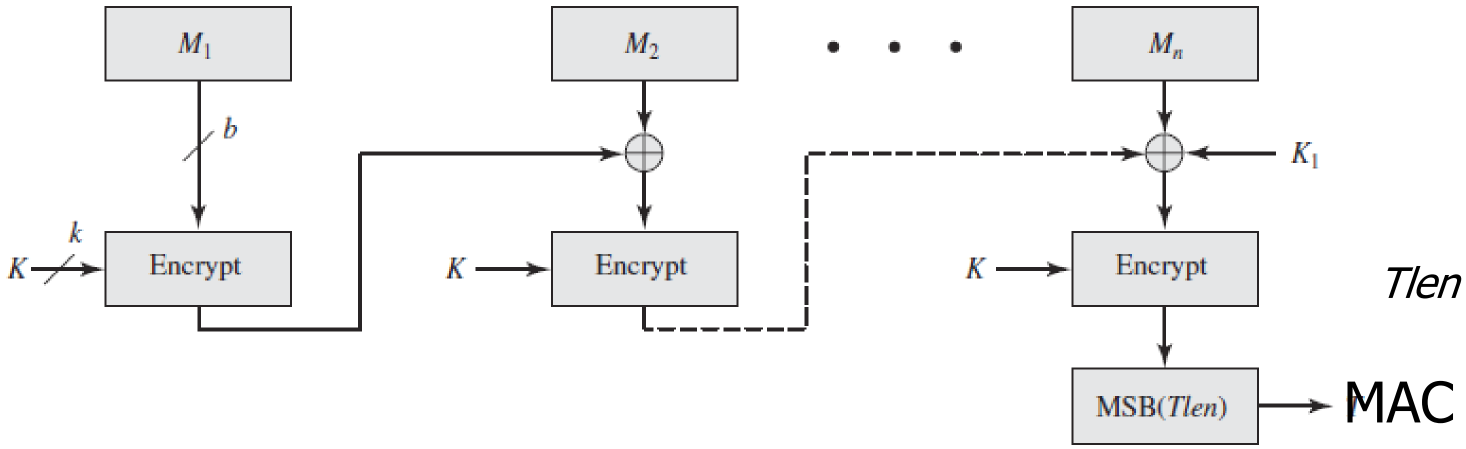
  $\mathbf{C_0} = E_K(IV \oplus P_0)$, $\boldsymbol{C_1} = E_K(\mathbf{C_0} \oplus X)$,

  $\boldsymbol{C_2} = E_K(\boldsymbol{C_1} \oplus P_2)$, $\boldsymbol{C_3} = E_K(\boldsymbol{C_2} \oplus P_3) = \boldsymbol{MAC} \neq \mathbf{MAC}$
- That is, error propagates into **MAC**
- An attacker can't make ***MAC*** == **MAC** without K

Cipher-based Message Authentication(CMAC)

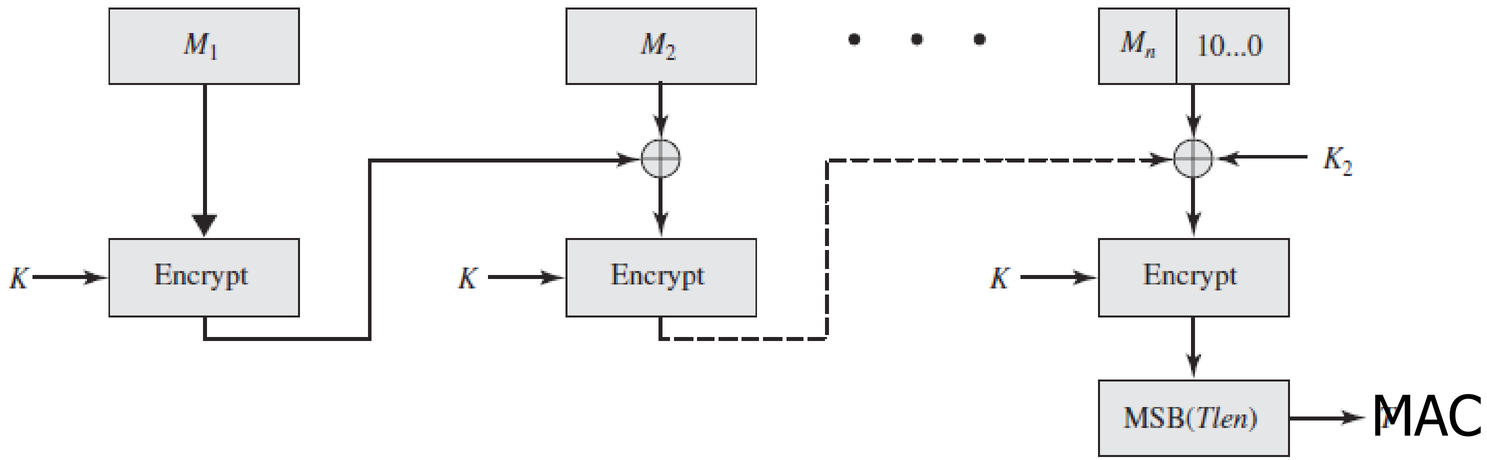
- CBC with a single symmetric key has a limitation.
  - When a message is one block size, P_0, then $MAC = E_K(IV \oplus P_0)$.
  - If an attacker make the following two block size message, $(P_0, P_0 \oplus MAC \oplus IV)$, then the MAC of this message is also MAC.
- So, the CMAC uses two keys: a k-bit encryption key K and a b-bit constant K_1, where b is the cipher block length.

Message length is integer multiple of block size b



Tlen : length of MAC

Message length is not integer multiple of block size b



Authenticated Encryption

- Many applications require both message confidentiality and authentication together.
- Authenticated encryption is to do encryption that simultaneously provide message confidentiality authentication.

Authenticated Encryption methods

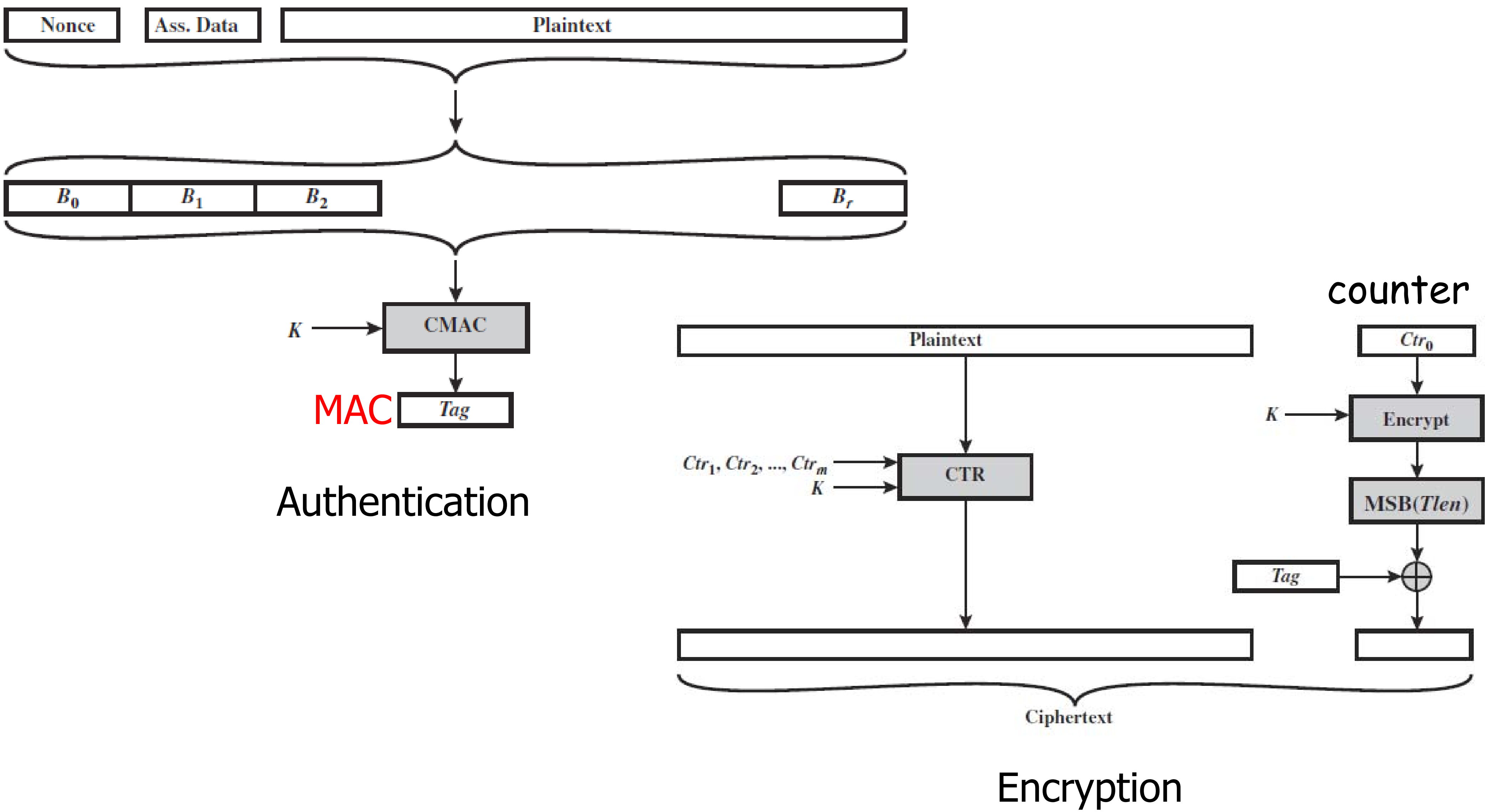
- The simple answer is to do two computations, encryption and MAC, for the same message, using two keys.
- Note: we shouldn’t use a single key for encryption and authentication for CBC.
  - As a simple example, message integrity can’t be verified when we send ciphertext and MAC that are computed from a single key.

Which order of two computations?

- Hashing and then Encryption
- Authentication and then Encryption
  - Using two keys
- Encryption and then Authentication
  - Using two keys
- Encryption and Authentication independently

Counter with CBC-MAC (CCM)

- It is a NIST standard specifically to support IEEE 802.11 WiFi.
- A variation of “encryption and authentication(MAC)” approach.
- Algorithms: AES + CTR + CMAC (authentication)
- A single key is used for both encryption and MAC computation.


Nonce
Ass. Data
Plaintext
B_0
B_1
B_2
B_r
K
CMAC
MAC
Tag
Authentication
counter
Plaintext
Ctr_0
K
Encrypt
$Ctr_1, Ctr_2, ..., Ctr_m$
K
CTR
MSB(Tlen)
Tag
Ciphertext
Encryption

Galois/Counter Mode (GCM)

- As a NIST standard, it is designed for parallel computation.
- Encryption in a variant of CTR mode.
- The standard is also used only for MAC, known as GMAC.
- GCM uses two functions:
  - GHASH : a keyed hash
  - GCTR: CTR mode encryption

H : hash key

$Y_0=0^{128}$

• : multiplication in GF(2128)

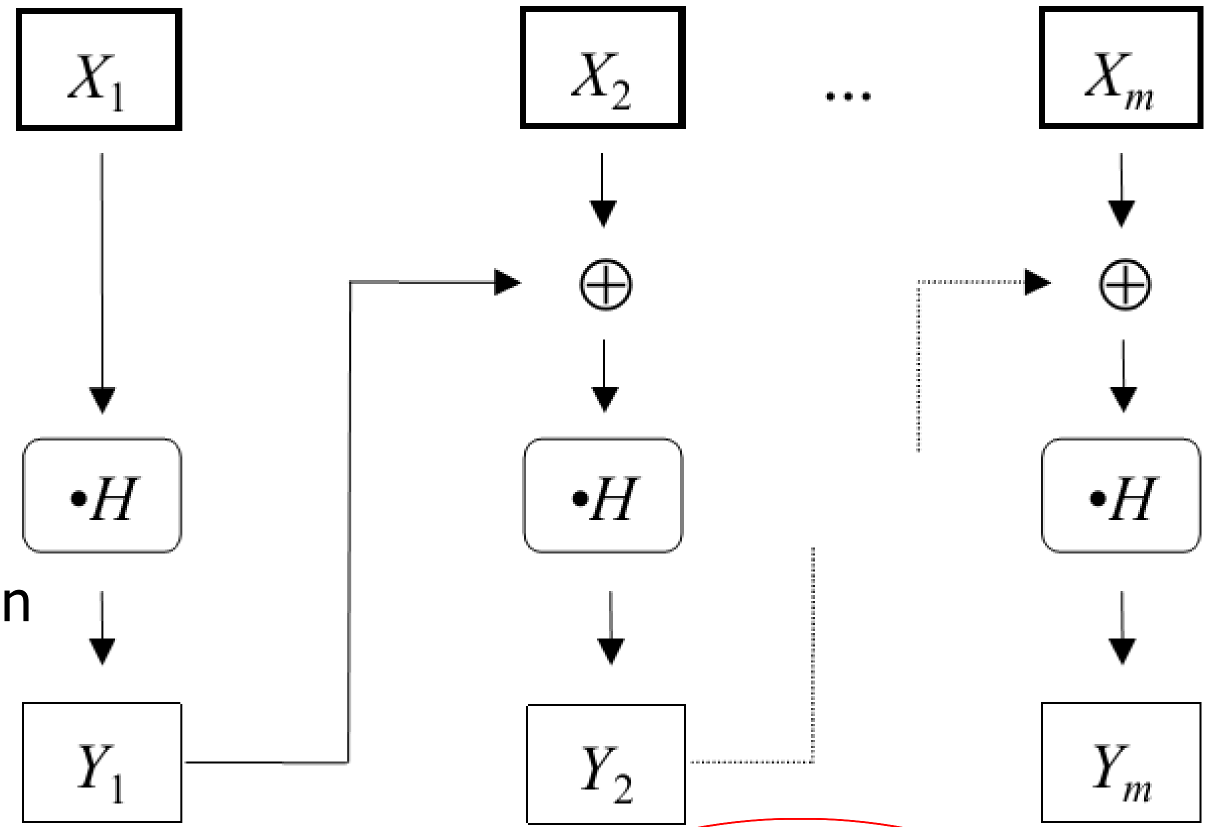


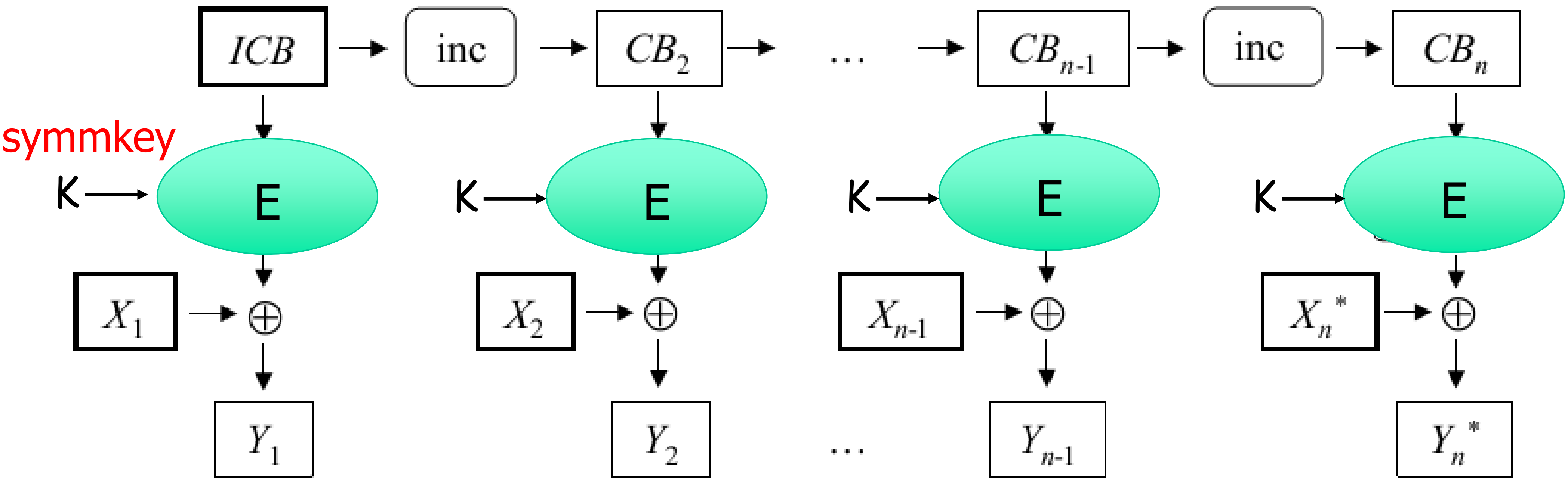
Figure 1: $\text{GHASH}_H(X_1 \| X_2 \| \ldots \| X_m) = Y_m$.

two inputs

outputs: 128 bit MAC

Parallel computation

- GMASHH(X) function can be expressed as
  $(X_1 \bullet H^m) \oplus (X_2 \bullet H^{m-1}) \oplus \ldots \oplus (X_{m-1} \bullet H^2) \oplus (X_m \bullet H^1)$
- If the same hash key is used to authenticate multiple messages, the values H_m can be precalculated one time for use for each message.
- Then the blocks of data $(X_1, \ldots, X_m)$ can be processed in parallel.



Message of arbitrary length

Figure 2: $\mathrm{GCTR}_K(ICB, X_1 \,\|\, X_2 \,\|\, \ldots \,\|\, X_n^*) = Y_1 \,\|\, Y_2 \,\|\, \ldots \,\|\, Y_n^*$.

two inputs

outputs: ciphertext

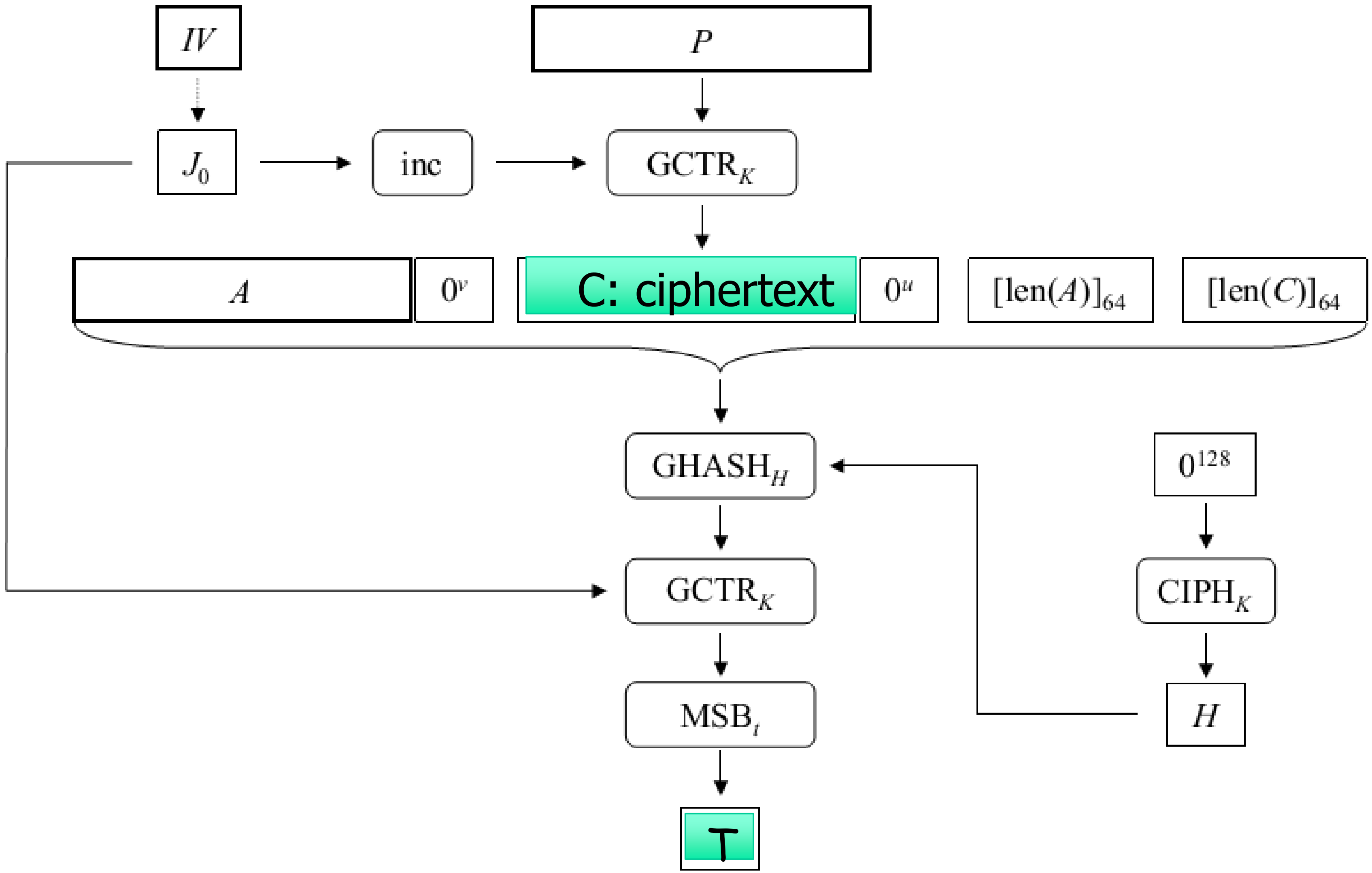
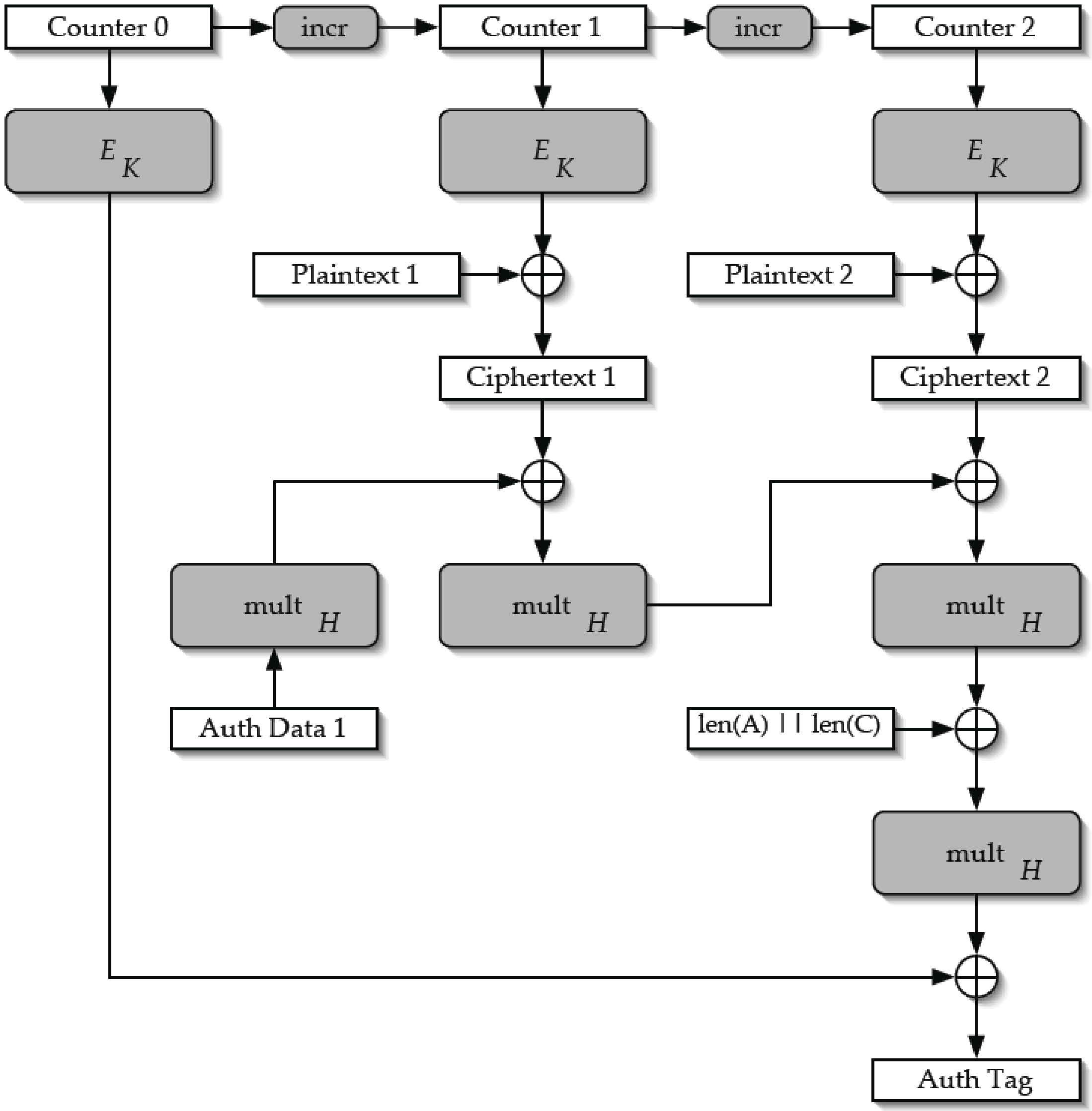


Figure 3 : GCM-AE$_K$ (IV, P, A) = (C, T).



Counter 0
incr
Counter 1
incr
Counter 2
E K
E K
E K
Plaintext 1
Plaintext 2
Ciphertext 1
Ciphertext 2
mult H
mult H
mult H
Auth Data 1
len(A) || len(C)
mult H
Auth Tag