Symmetric Crypto
MAC

Pierre-Alain Fouque

Message Authentication Code (MAC)

- <u>Warning</u>: Encryption does not provide integrity
- <u>Eg</u>: CTR mode ensures confidentiality if the blockcipher used is secure. However, no integrity is guaranteed. (CBC first block)

Alice $\xrightarrow{\quad C_1 \quad\quad\quad\quad C'_1 \quad}$ Bank

Eve

M_1 = « \$200 on Bob's account »

$C_1 = E_K(ctr) \oplus M_1$

M'_1 = « \$2000 on Eve's account »

$C'_1 = C_1 \oplus M_1 \oplus M'_1$

Definition of Message Authentication Code

- Key generation: randomized alg.
  - output: key uniformly distributed
- Tag MAC generation: randomized or deterministic
  - input: $M \in \{0,1\}^*$
  - output: tag $\tau \in \{0,1\}^t \cup \perp : \tau = M_K (M)$
- Verification: deterministic alg.
  - input: tag $\tau \in \{0,1\}^t$ and message M
  - output: bit if the tag is valid for this message s.t.

    for any K and message M, if $\tau = M_K (M)$, then $V_K (\tau, M) = 1$

Security game

Adversary's goals:

1. key recovery attacks
2. forgery: producing a valid MAC for some message M (of his choice, or any)

Adversary's ressources:

1. known message attack: interception of MACs. Adv. knows pair (M, τ) of already tagged messages
2. chosen message attack: Adv. knows the tag of message of his choice (access to a MAC generation alg. adaptively or not)

Security game

Def: Combining an adversary's goal and some ressources

SUF-CMA: strongly inforgeability against chosen message attacks

Challenger $\xleftarrow{\quad M_i \quad}$ Adversary A

$\xrightarrow{\quad \tau_i \quad}$

$\xleftarrow{\quad (M, \tau) \quad}$

$\xrightarrow{\quad 1 \ \text{: valid tag} \quad}$

Adv (A) = Pr (Expérience retourne 1)

Generic Security

1. For a t-bit MAC, advantage (forgery probability) is always at least $1/2^t$

2. Among $2^{t/2}$ MACs, by the birthday paradox, there is a collision between two of them: these collisions can be used to recover the keys ...

MAC vs. Signature

Signatures:

used for vertifying public keys,
guarantee non-repudiation,
same properties than hand-written signature

MACs:

very good performences,
secret-key shared between two users $\Rightarrow$ no non-repudiation,

no public verification

First construction

Let $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^t$ a random function (i.e. outputs are indistinguishable from random values)

MAC construction: For message $M = M_1 \ldots M_m$,

$$= F_K(M_1) \oplus \ldots \oplus F_K(M_m)$$

Is this scheme secure ?

Second Example

Let $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^t$ random function

For message $M = M_1 \ldots M_m$

For $i = 1$ to m, $\quad y_i = F_K(\langle i \rangle, M_i)$

$$= y_1 \oplus \ldots \oplus y_m$$

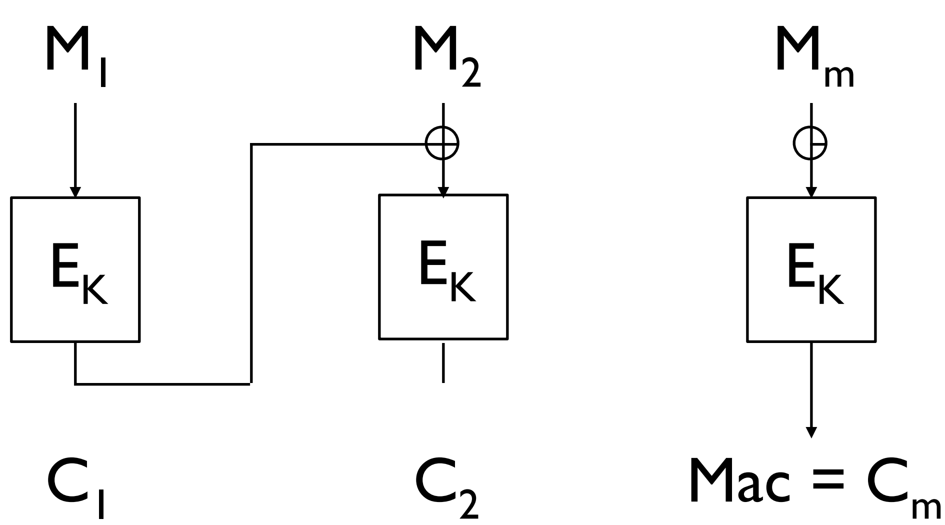
Is this scheme secure ?

unencrypted CBC-MAC

$C_i = E_K\,(M_i \oplus C_{i-1})$

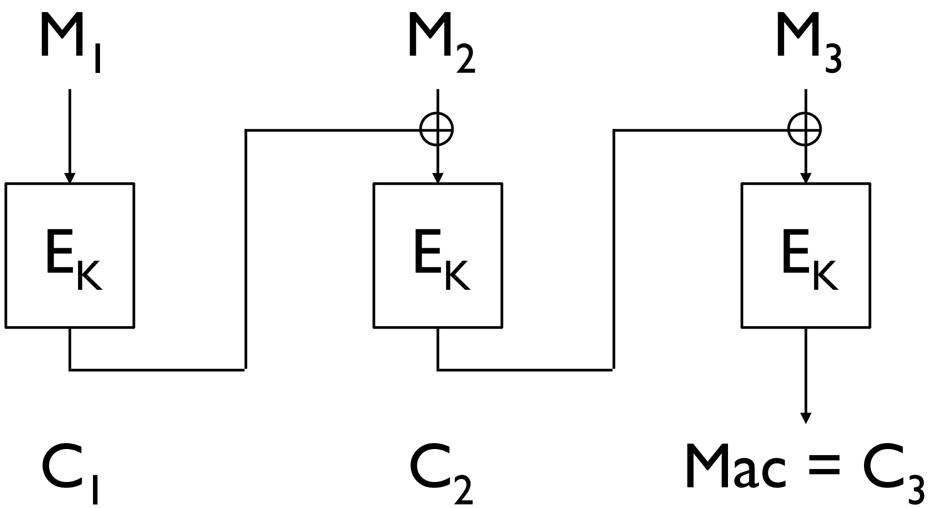
$MAC = C_m$

Secure only for constant length messages

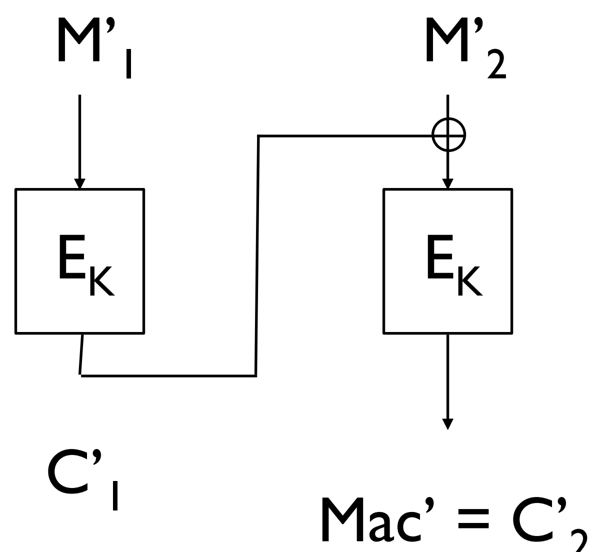


Security CBC-MAC

Let 2 arbitrary messages M and M’



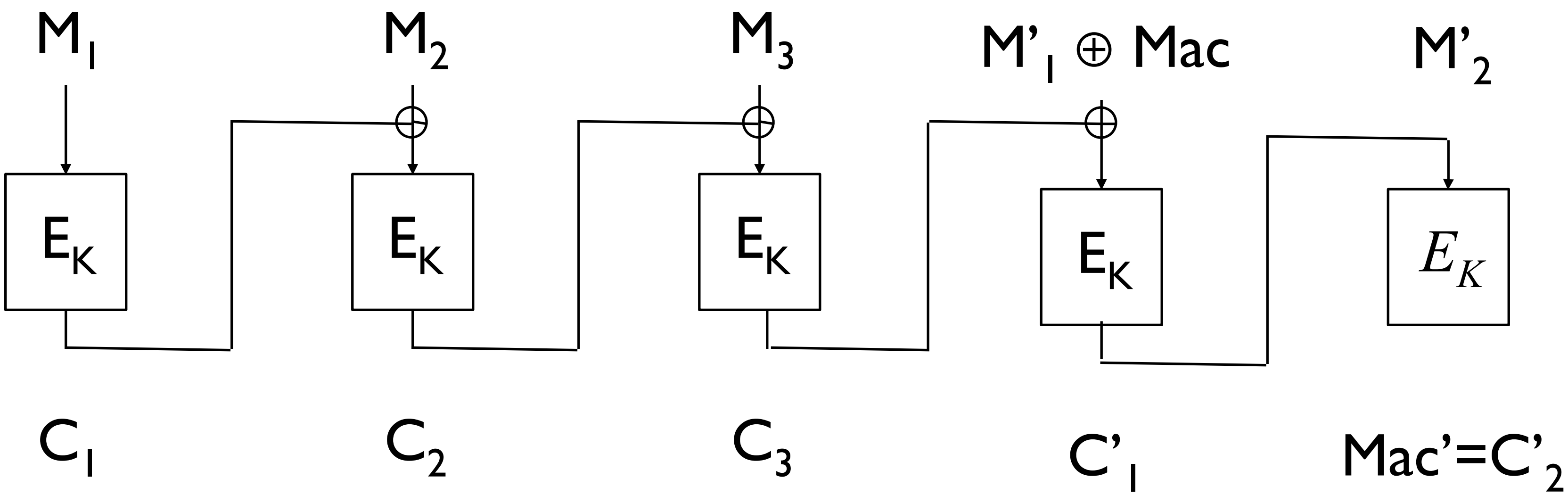
MAC(M) is C_3 = Mac



MAC(M’) is C'_2 = Mac’

unencrypted CBC-MAC

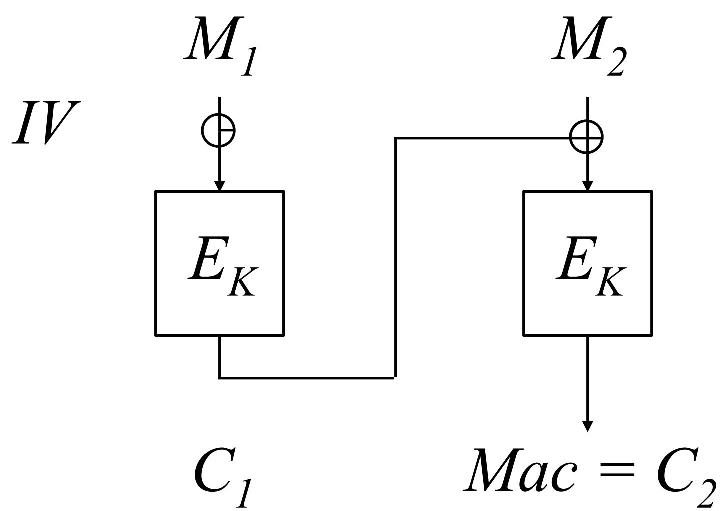
Given MACs of M and M', it is possible to forge MAC of another message



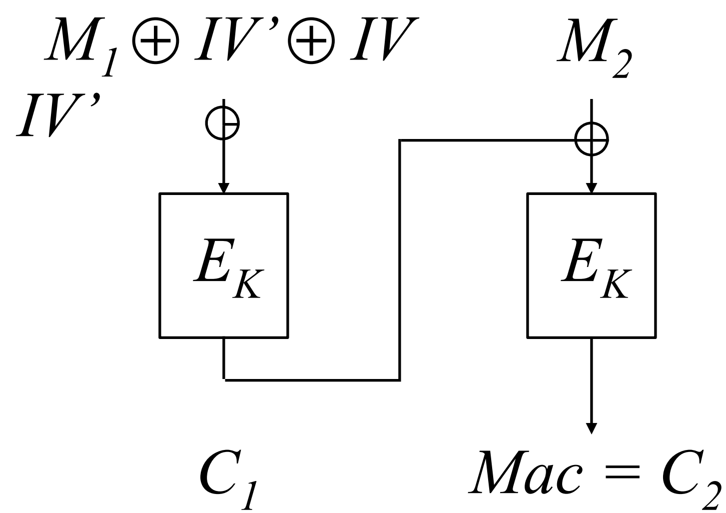
Recovering the secret key is in 2^k MAC computation where k is the bit length of the used key (exhaustive search)

No IV in CBC-MAC

The integrity of the first block is not ensured if an IV is used



(M, IV, Mac)

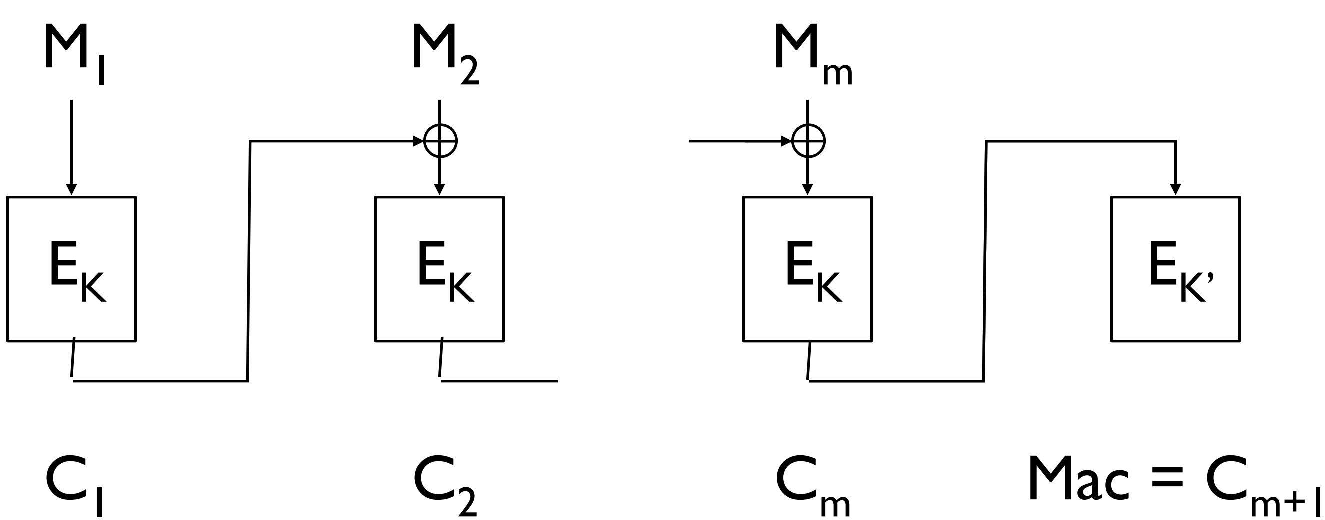


(M', IV', Mac)

Encrypted CBC-MAC (EMAC)

$C_i = E_K\,(M_i \oplus C_{i-1})$ and $MAC = E_{K'}\,(C_m)$

Secure if less than $2^{n/2}$ MACs are computed



Security Analysis

Assume $2^{n/2}$ MACs computed: (M_i, τ_i), $0 \leq i \leq 2^{n/2}$ and $M_i \neq M_j$

Using Birthday Paradox, there exists i,j s.t. $i \neq j$ and $\tau_i = \tau_j$

Ask MAC τ of $M_i \,||\, R$, where R is a random block

<u>Claim</u>: One can forge MAC for message $M_j \,||\, R : \tau$

Hash-based MAC

Consider the following MAC scheme:

$$\text{MAC}_K(M) = \text{H}(K \,||\, M)$$

Is it secure ?

HMAC

$HMAC_K(M) = H(K' \oplus opad, H(K' \oplus ipad, M))$

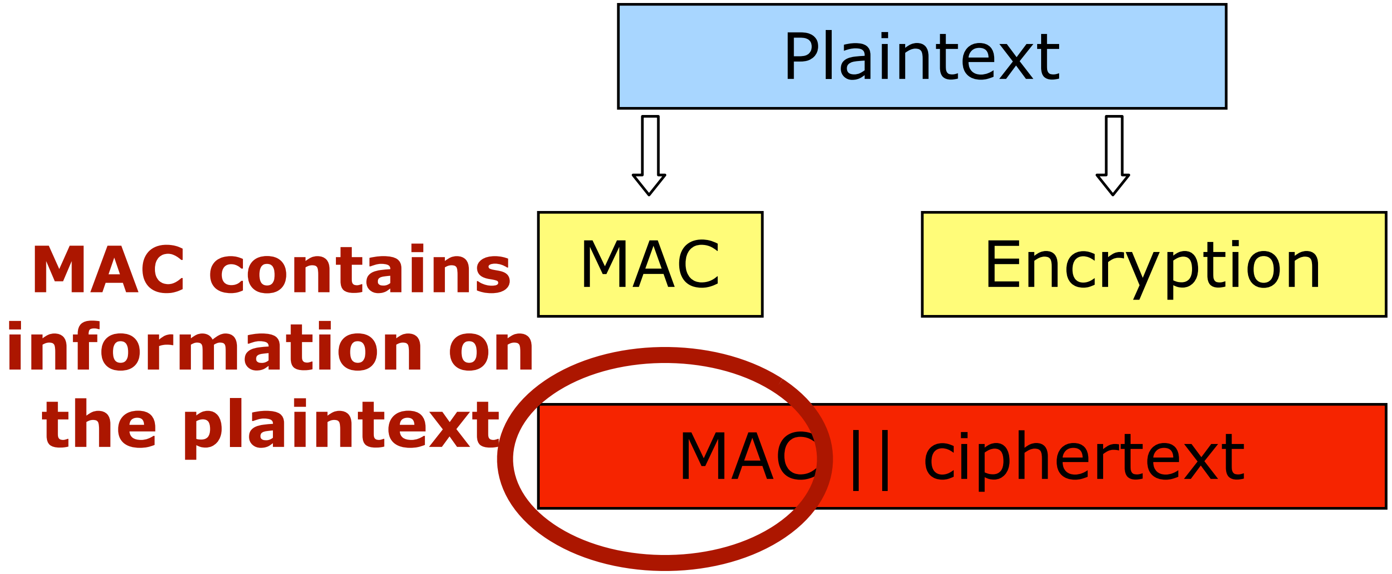
where ipad and opad are constant values:

Encryption and Authentication

- IPSEC: MAC-Then-Encrypt
- SSL/TLS: Encrypt-Then-MAC
- SSH: MAC-And-Encrypt

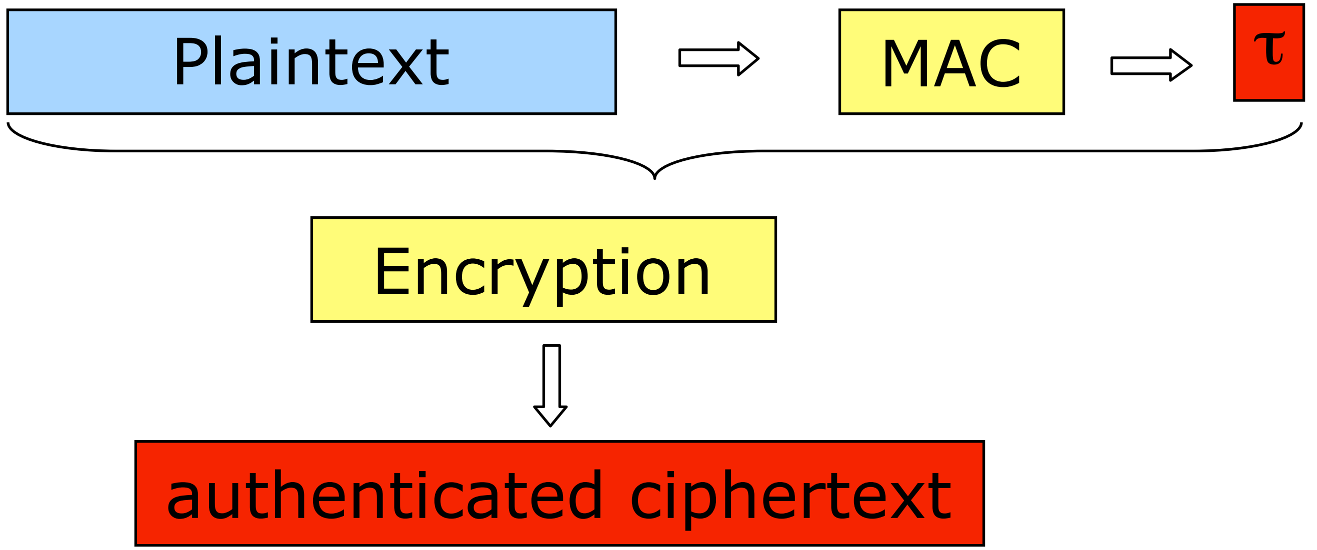
Mac-And-Encrypt

- Non-secure mode of operation
- Confidentiality is not guaranteed



MAC-then-Encrypt

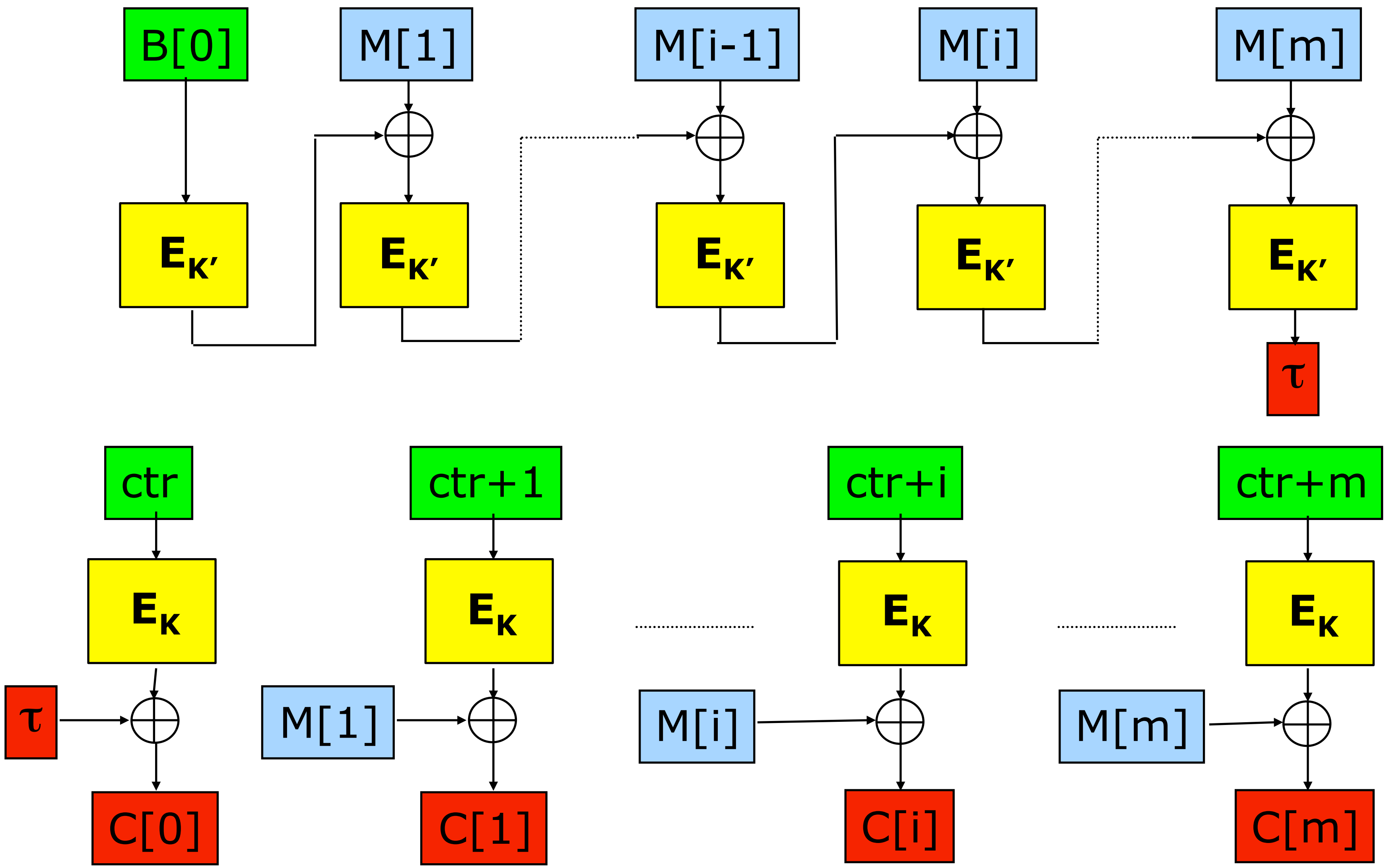
- Non-always secure but it could be
- In practice, one can construct secure scheme



CCM: Mac-then-encrypt

- CCM proposed by Housley, Whiting and Ferguson
- Wifi network
- NIST in 2003, operation mode for AES
- CBC-MAC then CTR
  - associated data
  - security proof

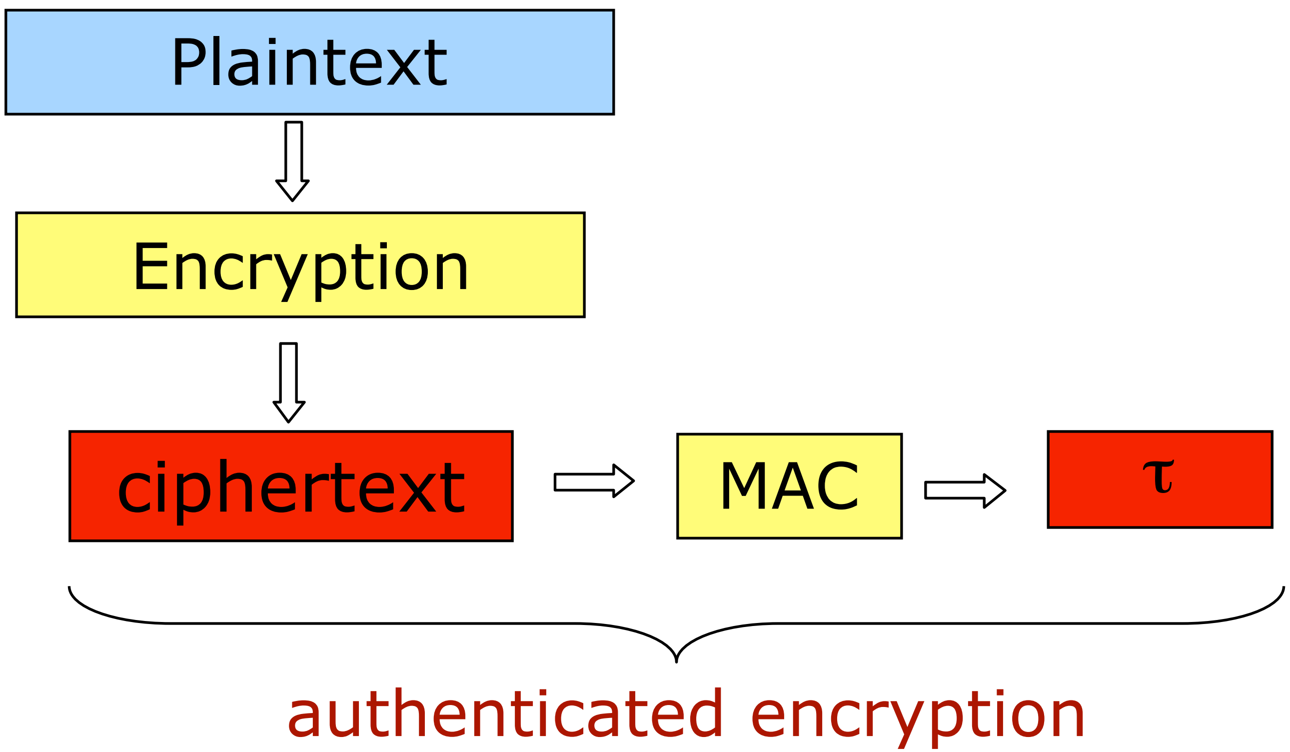
CCM mode



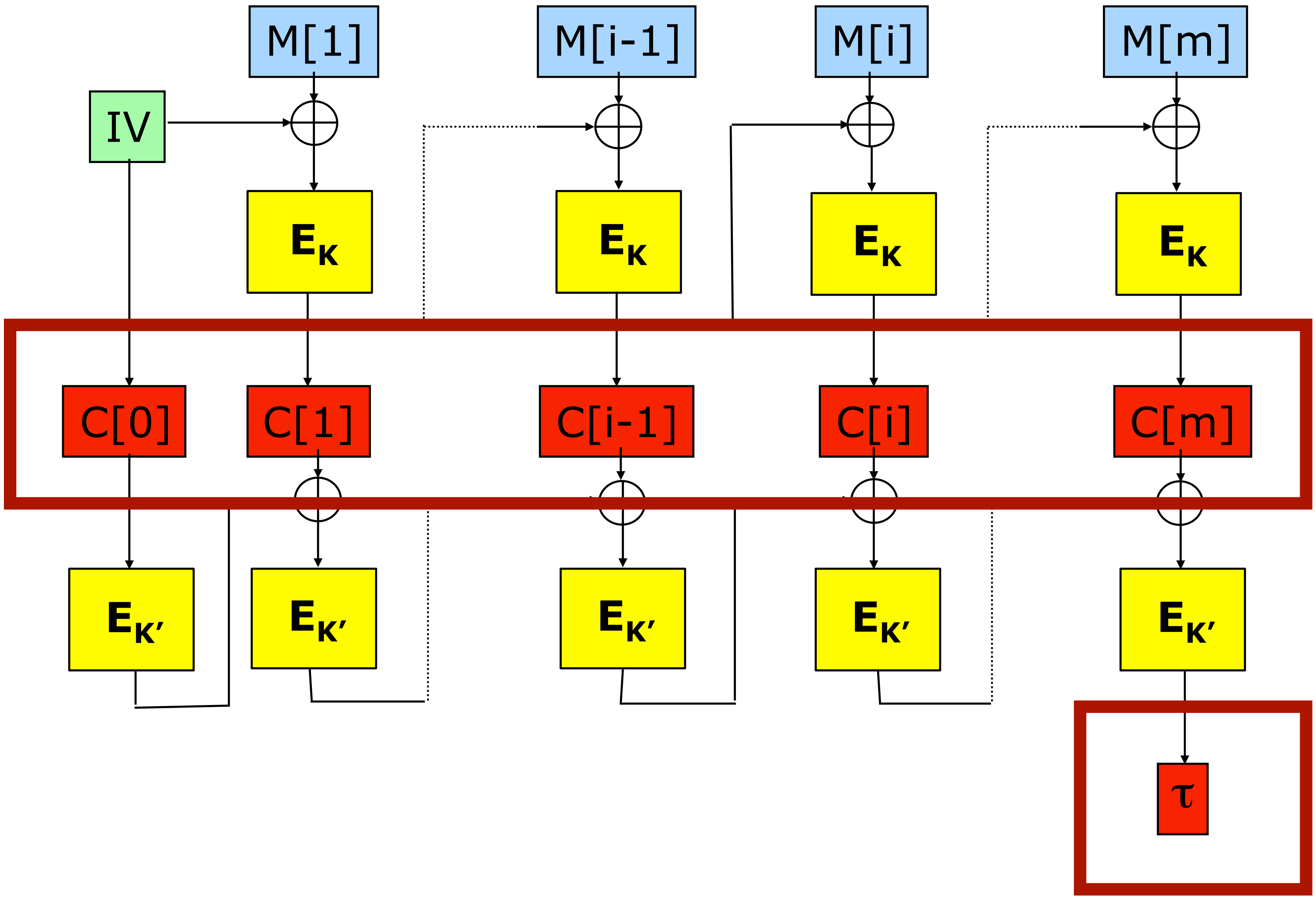
B[0]
M[1]
M[i-1]
M[i]
M[m]
E$_{K'}$
E$_{K'}$
E$_{K'}$
E$_{K'}$
E$_{K'}$
τ
ctr
ctr+1
ctr+i
ctr+m
E$_K$
E$_K$
E$_K$
E$_K$
τ
M[1]
M[i]
M[m]
C[0]
C[1]
C[i]
C[m]

Encrypt-then-MAC

- Secure if the encryption mode is secure and if the MAC is secure



Encrypt-then-MAC



One-pass Mode

- Message is treated once:
  - More efficient : near as efficient as one encryption
  - One key
- Examples : IAPM, IACBC, OCB, …

IACBC mode