Codes d'authentification de messages (MAC) et chiffrement authentifié

Fouque Pierre-Alain

Equipe de Cryptographie

Ecole normale supérieure

Sommaire

Problématique et motivation

Exemples basiques

Autres exemples et attaques de schémas de MACs : CBC-MAC, EMAC et ses variantes

MACs à base de fonctions de hachage : l'exemple d'HMAC

Généralités

Les MACs (Messages Authentication Codes ou Codes d'authentification de messages) assurent l'intégrité du message

Utilisation d'une clé secrète

Primitives utilisées : fonctions de hachage, blocks ciphers, …

Motivation

Le chiffrement ne fournit pas d'intégrité :

Exemple, le mode compteur assure la confidentialité si le block cipher utilisé est sûr. Mais aucune intégrité n'est assurée : on peut intercepter le chiffré et le modifier

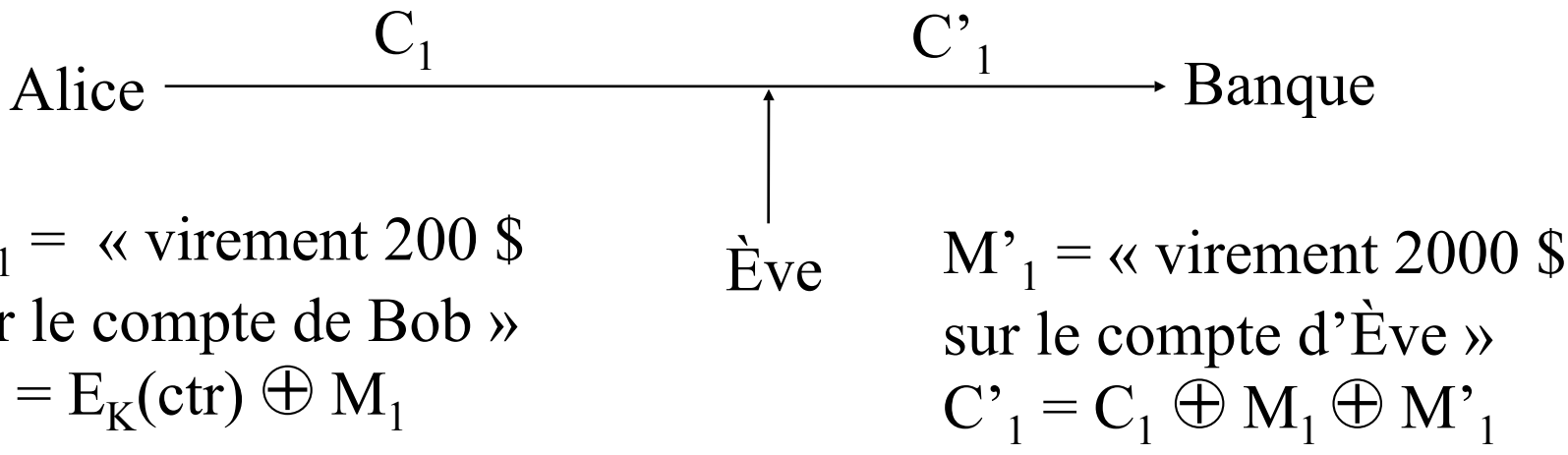


Schéma de MAC

Constitué de 3 algorithmes :

- Génération de clé :
  - algorithme probabiliste
  - retourne une clé aléatoire K prise dans l'ensemble des clés possibles
- Génération de MAC :
  - algorithme déterministe ou probabiliste
  - prend en entrée un clair $M \in \{0,1\}^*$ et
  - retourne un tag $\tau \in \{0,1\}^t \cup \perp : \tau = \mathsf{M}_K(M)$

Schéma de MAC

Vérification : si la génération de MAC est déterministe, cette description est inutile

algorithme déterministe

prend en entrée un tag $\tau \in \{0,1\}^t$, un message M et retourne un bit selon la validité du tag pour ce message :

valide $V_K(C) =$ { 0 si le tag n'est pas ; 1 sinon

Propriété :

Pour toute clé K et tout message M,

si $\tau = M_K(M)$, alors $V_K(\tau, M) = 1$

Buts des attaquants

Intuition : l'adversaire peut produire un MAC valide et le faire passer pour légitime

Formellement, il peut essayer de :

- Retrouver la clé secrète utilisée : cassage total du schéma
- Forger un MAC pour tout message
- Tenter de forger un autre couple (M', t') pour un message M', quelconque ou choisi

Attaques mises en oeuvre

À messages connus : interception de MACs. L'adversaire a accès à des couples (M, τ) de messages déjà authentifiés

À messages choisis : l'adversaire demande le MAC de messages qu'il choisit

accès à un oracle de génération de MACs

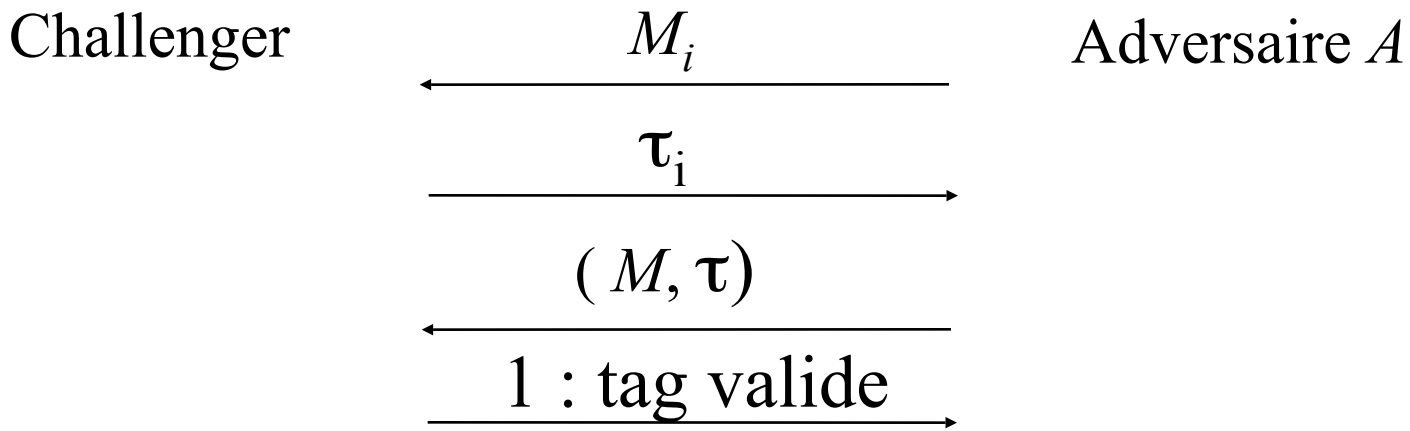
Attaque non adaptative : l'ensemble des messages est choisi a priori

Attaque adaptative : l'adversaire choisit les messages en fonction des réponses de l'oracle

Sécurité d'un MAC

Combinaison du but de l'adversaire et de l'attaque mise en œuvre

Exemple : sécurité SUF-CMA, inforgeabilité contre les attaques à messages choisis



$$\text{Adv}(A) = \Pr(\text{Expérience retourne 1})$$

Sécurité d'un MAC

Si le MAC est sur t bits, la probabilité de forger un MAC valide pour un message est toujours au minimum de $1/2^t$

Parmi $2^{t/2}$ MACs, par le paradoxe des anniversaires, il y a une collision entre deux d'entre eux : ces collisions peuvent souvent être exploitées pour forger

MAC / signature

Signatures :

- utilisées pour certifier des clés,
- assurent la non répudiation,
- mêmes propriétés que la signature manuscrite

MACs :

- très bonnes performances,
- clé secrète partagée entre deux utilisateurs ⇒
- pas de non répudiation,
- pas de vérification publique

Exemple basique

Soit $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^t$ une fonction aléatoire (càd dont les sorties sont indistinguables de valeurs aléatoires)

On propose le schéma de MAC suivant :

Pour tout message $M = M_1 \ldots M_m$,

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

Quelle est la sécurité de ce schéma ?

Attaque

Un adversaire A intercepte un couple (M, τ)

Il veut forger un MAC valide pour un nouveau message

Par exemple, (M || M, 0) est un MAC valide

Si A a accès a deux couples (M_1, τ_1) et (M_2,τ_2), alors (M_1 || M_2 , $\tau_1 \oplus \tau_2$) est aussi valide

Par conséquent ce schéma n'est absolument pas sûr

Autre exemple basique

Soit $F : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^t$ une fonction aléatoire

On propose le schéma de MAC suivant :

Soit $M = M_1 \ldots M_m$ un message

Pour $i = 1$ à m, $\quad y_i = F_K(<i>, M_i)$

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

Quelle est la sécurité de ce schéma ?

Attaque de ce schéma

L'attaque précédente ne s'applique plus

Mais, si l'adversaire a accès aux MACs de $M_1 \| M_2$, $M'_1 \| M_2$ et $M_1 \| M'_2$, notés respectivement τ_1, τ_2 et τ_3, alors il peut forger un MAC pour $M'_1 \| M'_2$: en effet, $(M'_1 \| M'_2, \tau_1 \oplus \tau_2 \oplus \tau_3)$ est valide

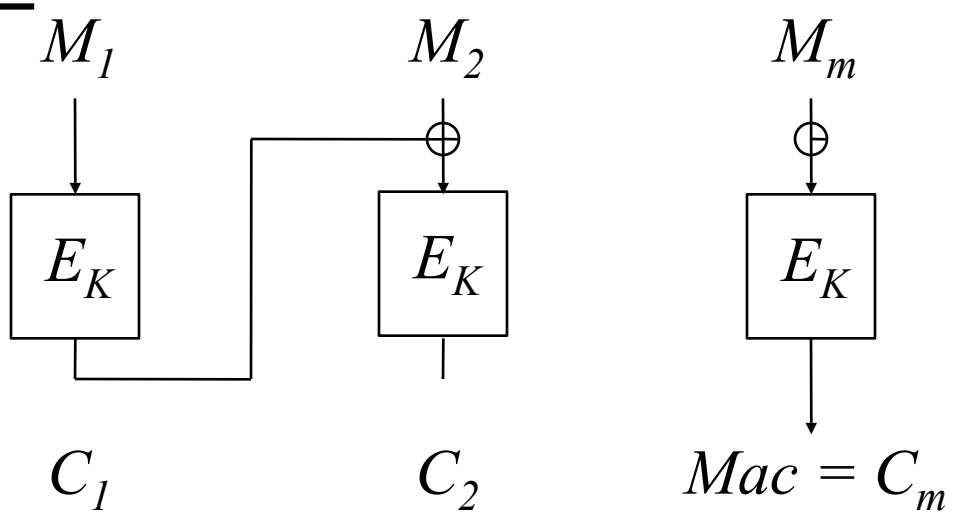
Autres schémas

CBC-MAC «non surchiffré»

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

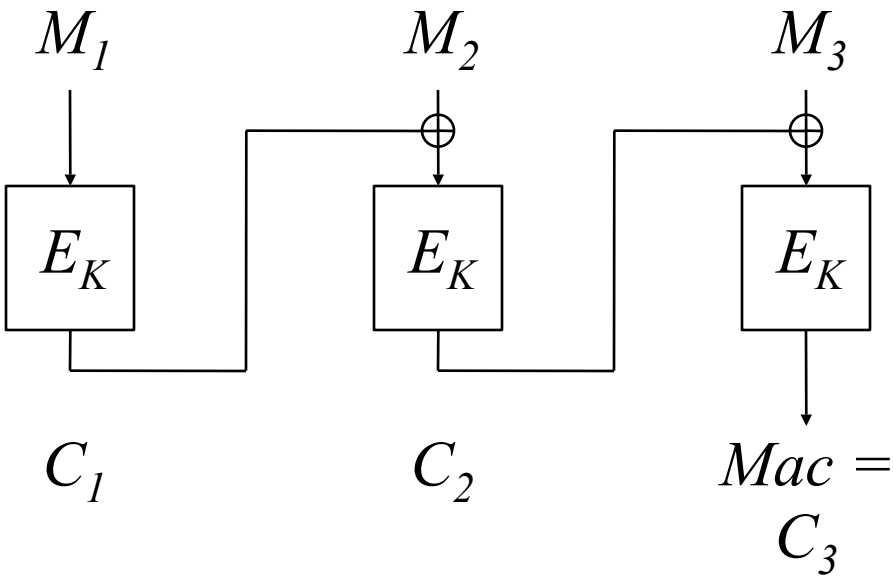
$$MAC = C_m$$

Sûr seulement pour des messages de <u>taille fixe</u>

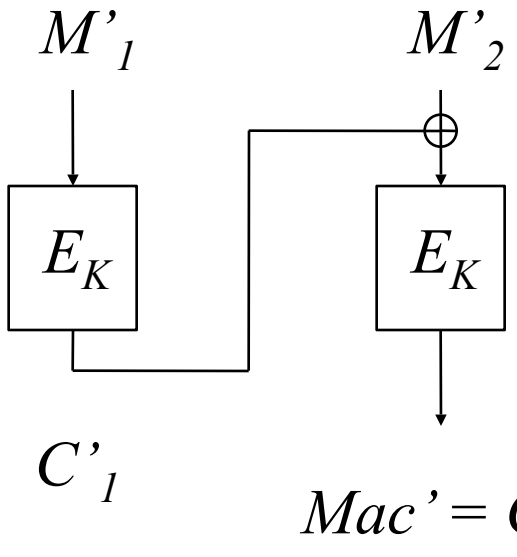


CBC – MAC non surchiffré

Soient 2 messages quelconques M et M'



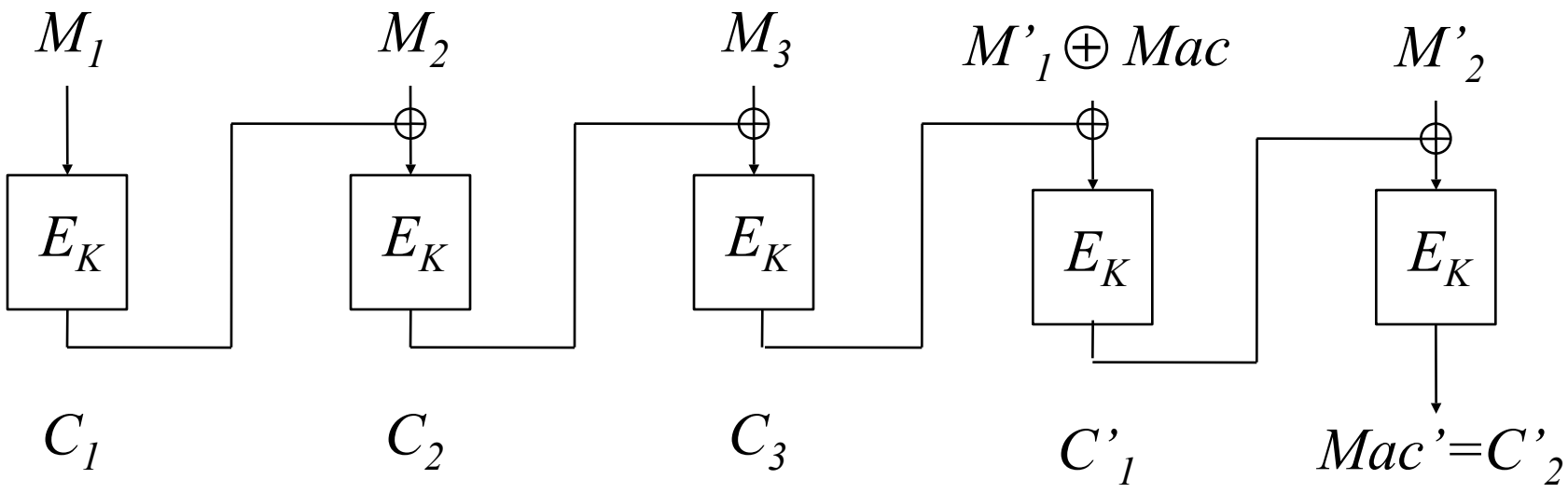
Le MAC de M est C_3 = Mac



Le MAC de M' est C'_2 = Mac'

CBC – MAC non surchiffré

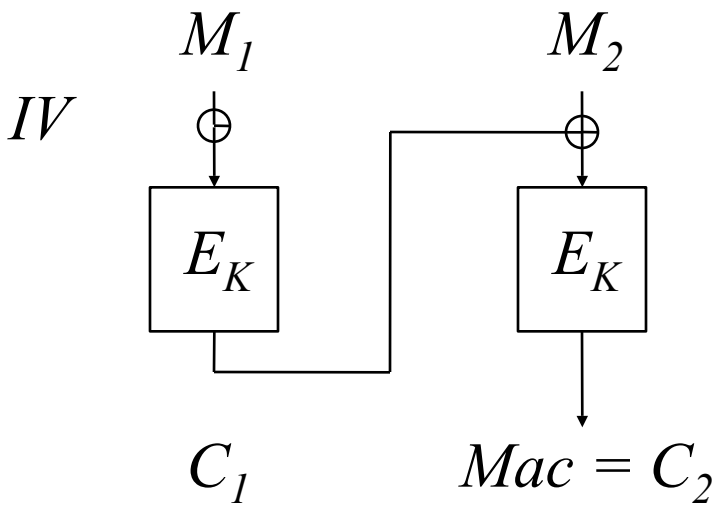
Connaissant les MACs de M et de M', il est possible de forger le MAC d'un autre message



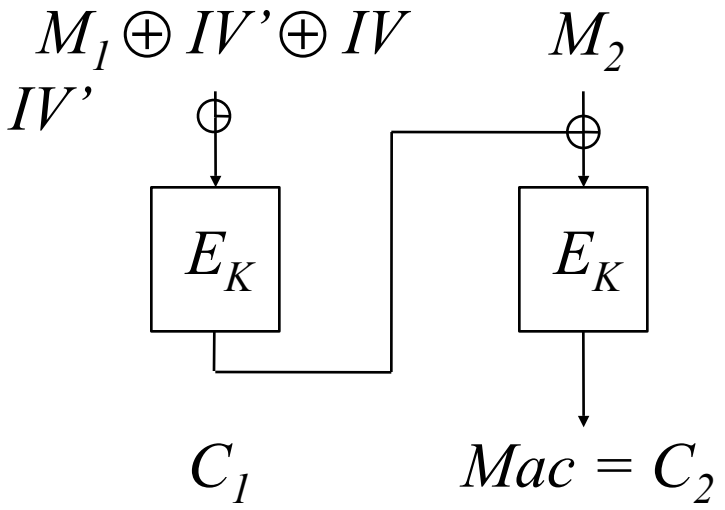
Retrouver la clé sur ce schéma se fait en 2^k où k est la taille en bits de la clé utilisée (recherche exhaustive)

Pas d'IV dans CBC-MAC

L'intégrité du premier bloc n'est pas assurée si une valeur initiale est utilisée



(M, IV, Mac)



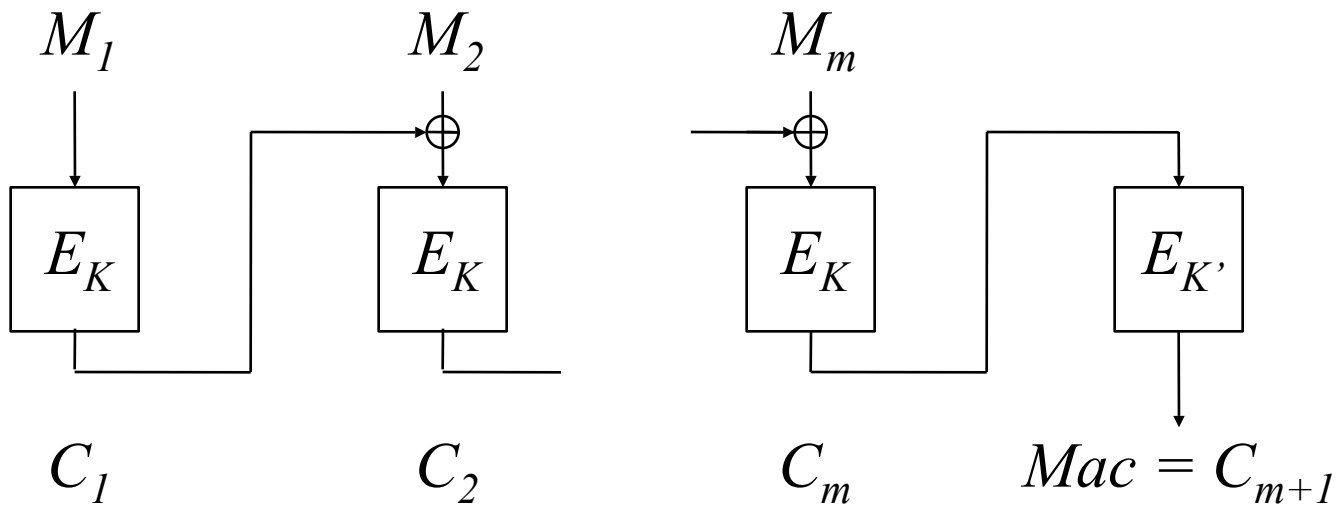
(M', IV', Mac)

EMAC : Encrypted MAC
le CBC – MAC « surchiffré »

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

Sûr si moins de $2^{n/2}$ MACs sont calculés

Les clés peuvent être retrouvées avec 2 recherches exhaustives en 2^k en temps (clés de k bits)



Sécurité d'EMAC

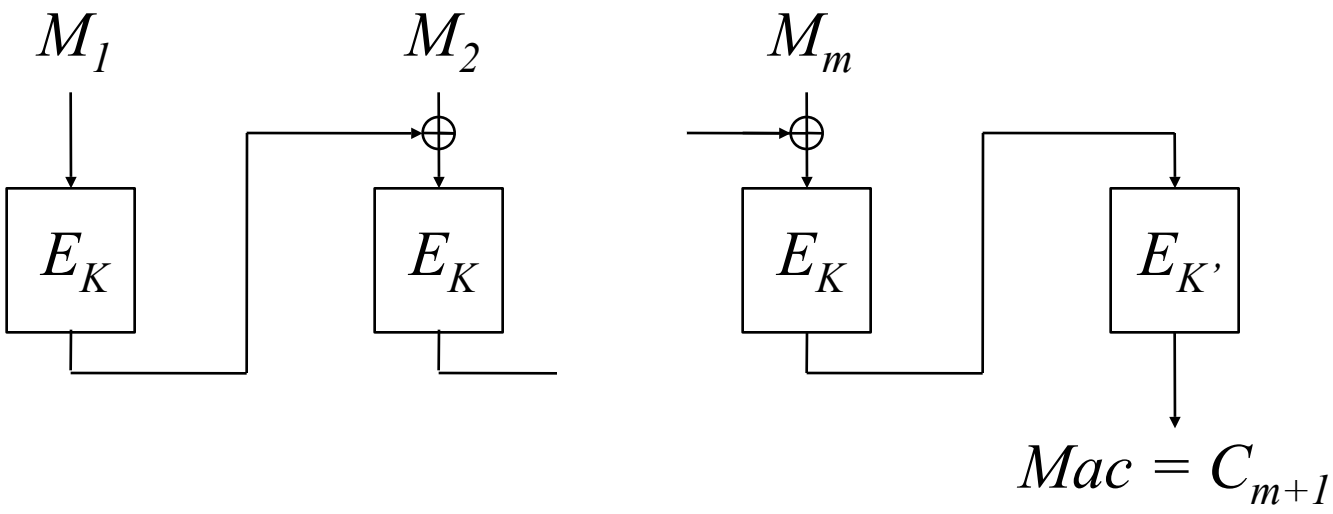
Sécurité prouvée :

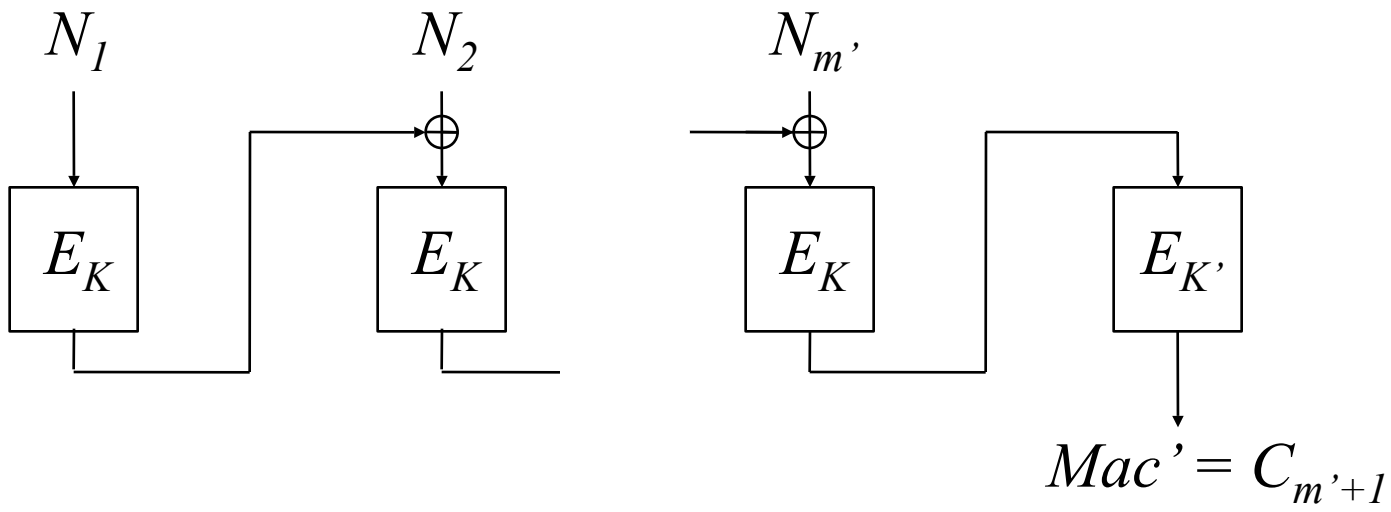
si le block cipher utilisé est une bonne permutation, alors il est impossible de forger un nouveau MAC valide avec une attaque à messages choisis avec probabilité supérieure à $q^2 / 2^{t+1}$, où q est le nombre de MACs demandés

Si $q = 2^{n/2}$, alors aucune sécurité prouvée, il y a même une attaque

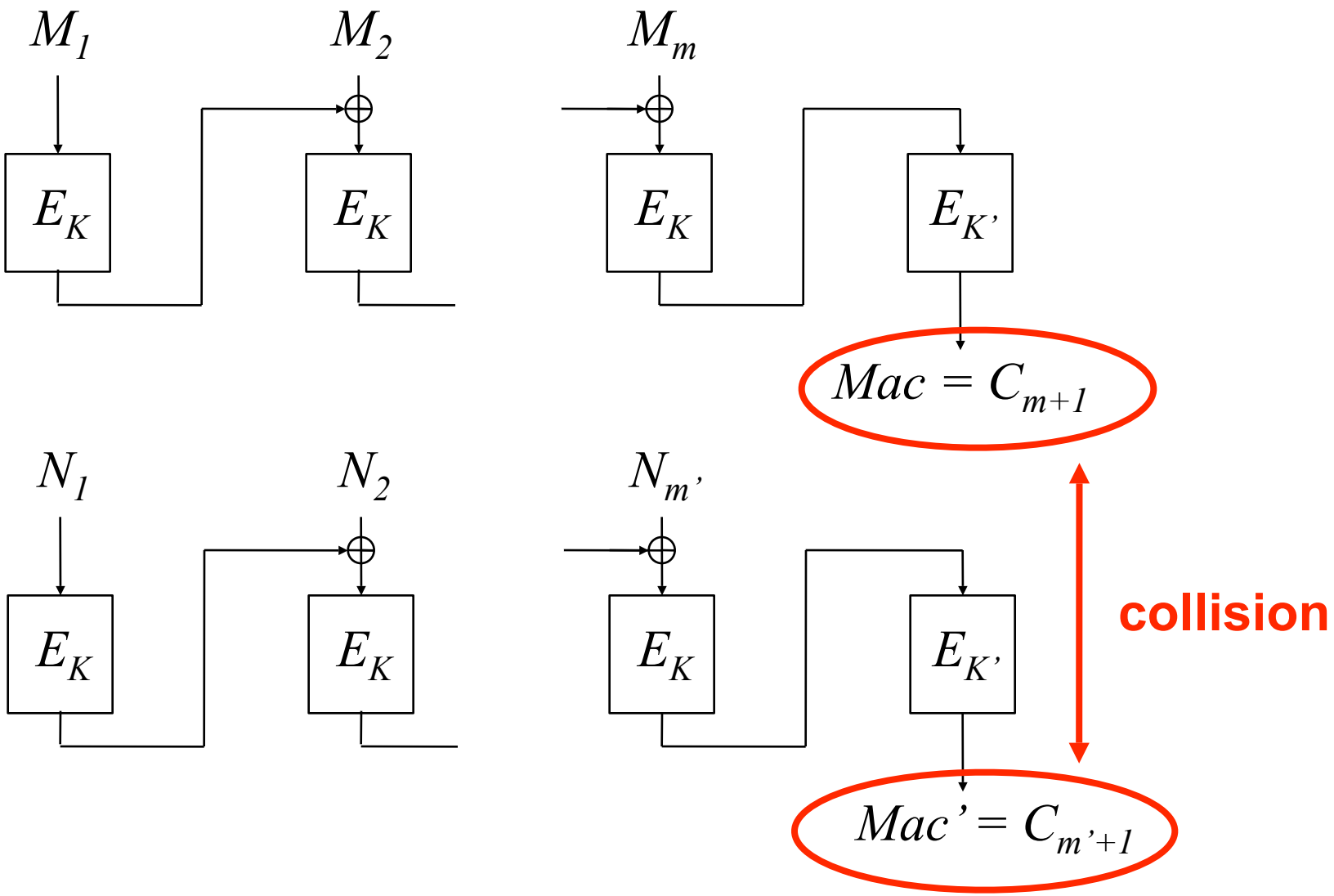
$\Rightarrow$ la preuve est optimale

Collisions sur EMAC

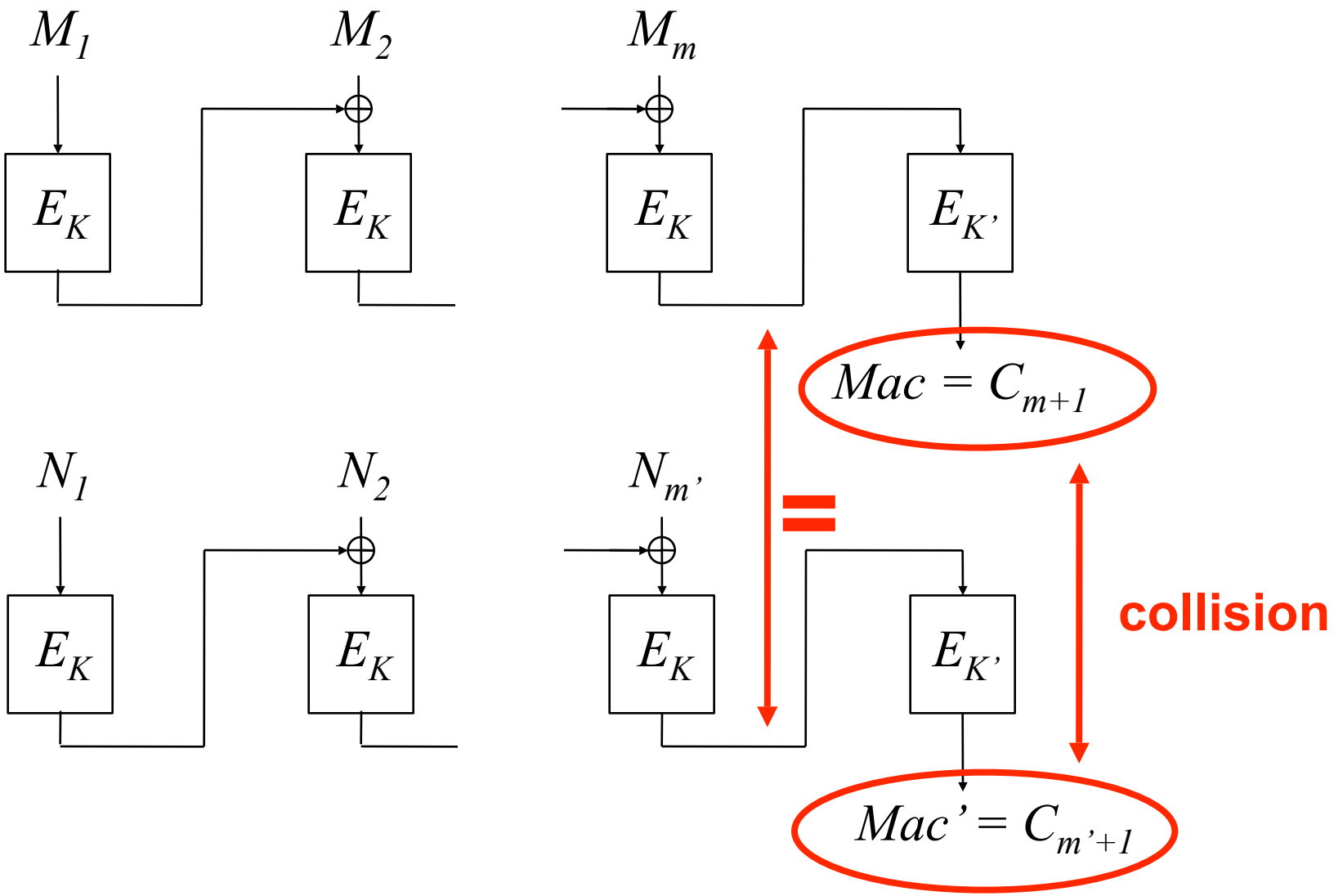




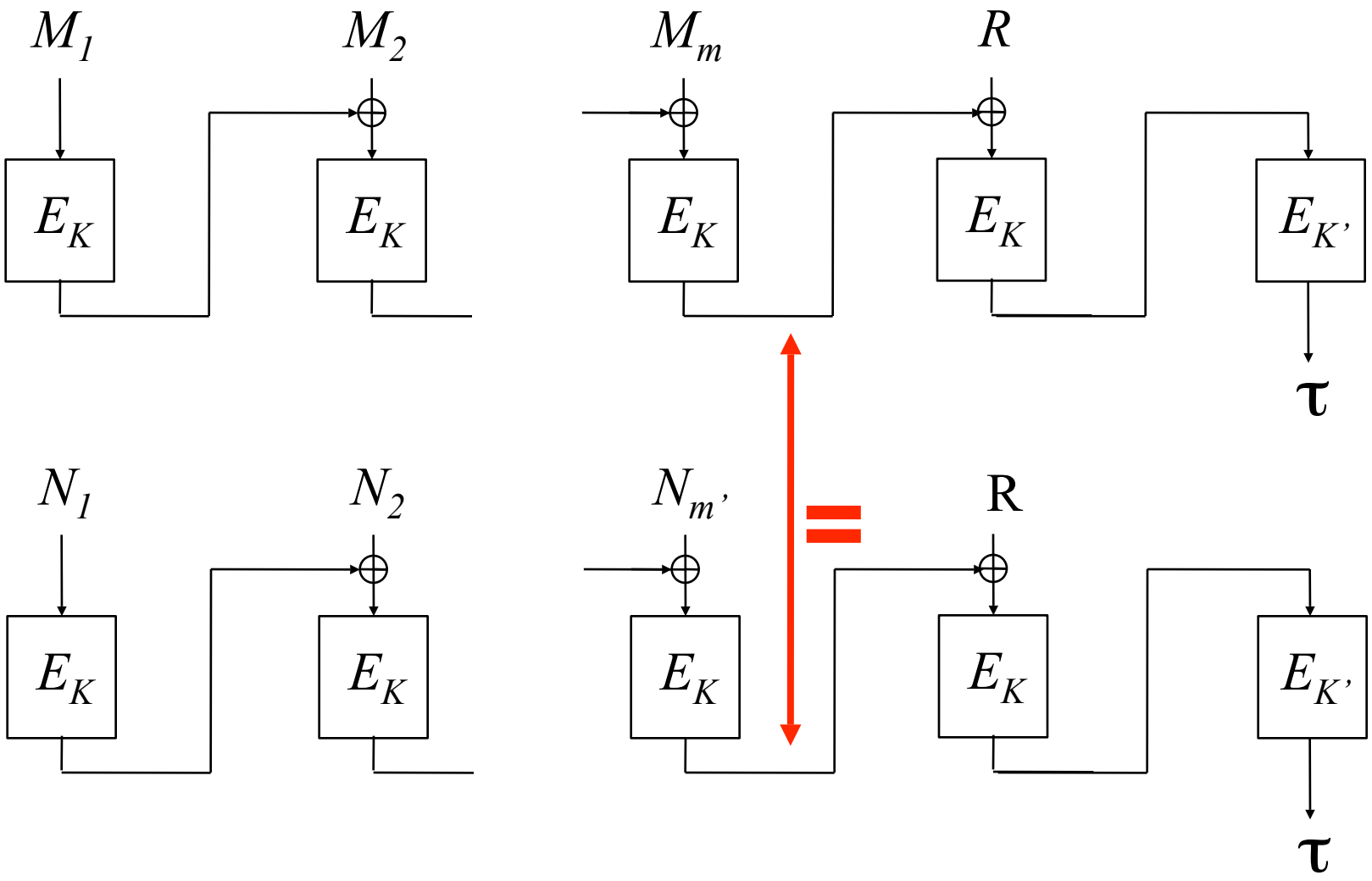
Collisions sur EMAC



Collisions sur EMAC



Collisions sur EMAC



Sécurité d'EMAC

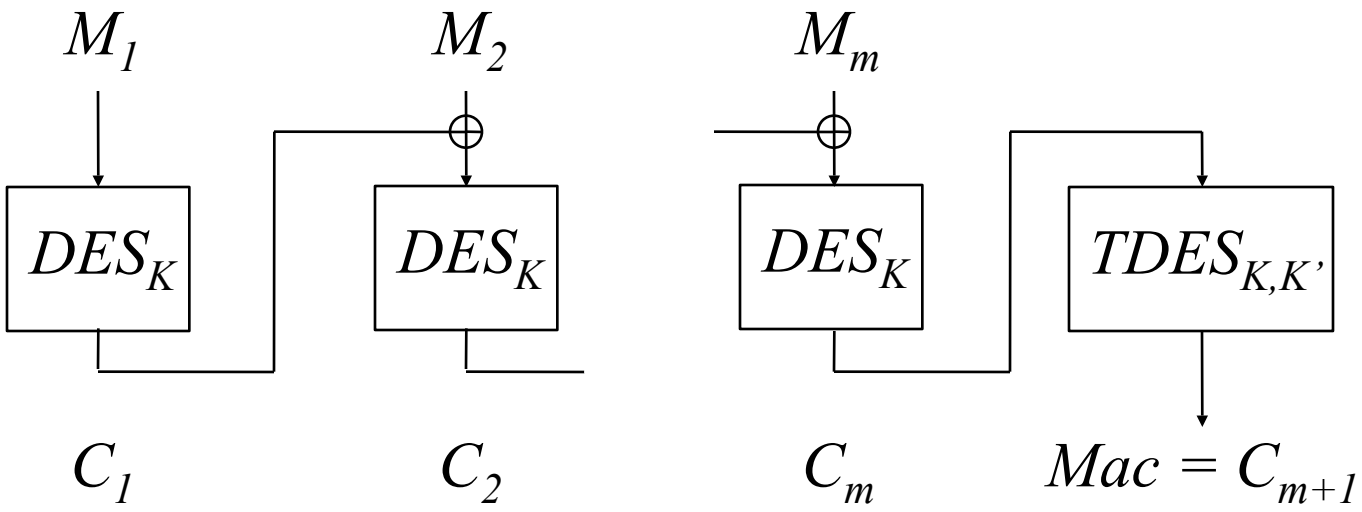
Supposons que $2^{n/2}$ MACs ont été calculés : (M_i , τ_i), $0 \leq i \leq 2^{n/2}$ et $M_i \neq M_j$

Par le paradoxe des anniversaires, il existe i et j tels que $i \neq j$ et $\tau_i = \tau_j$

On demande le MAC τ de $M_i \,||\, R$, où R est une valeur quelconque

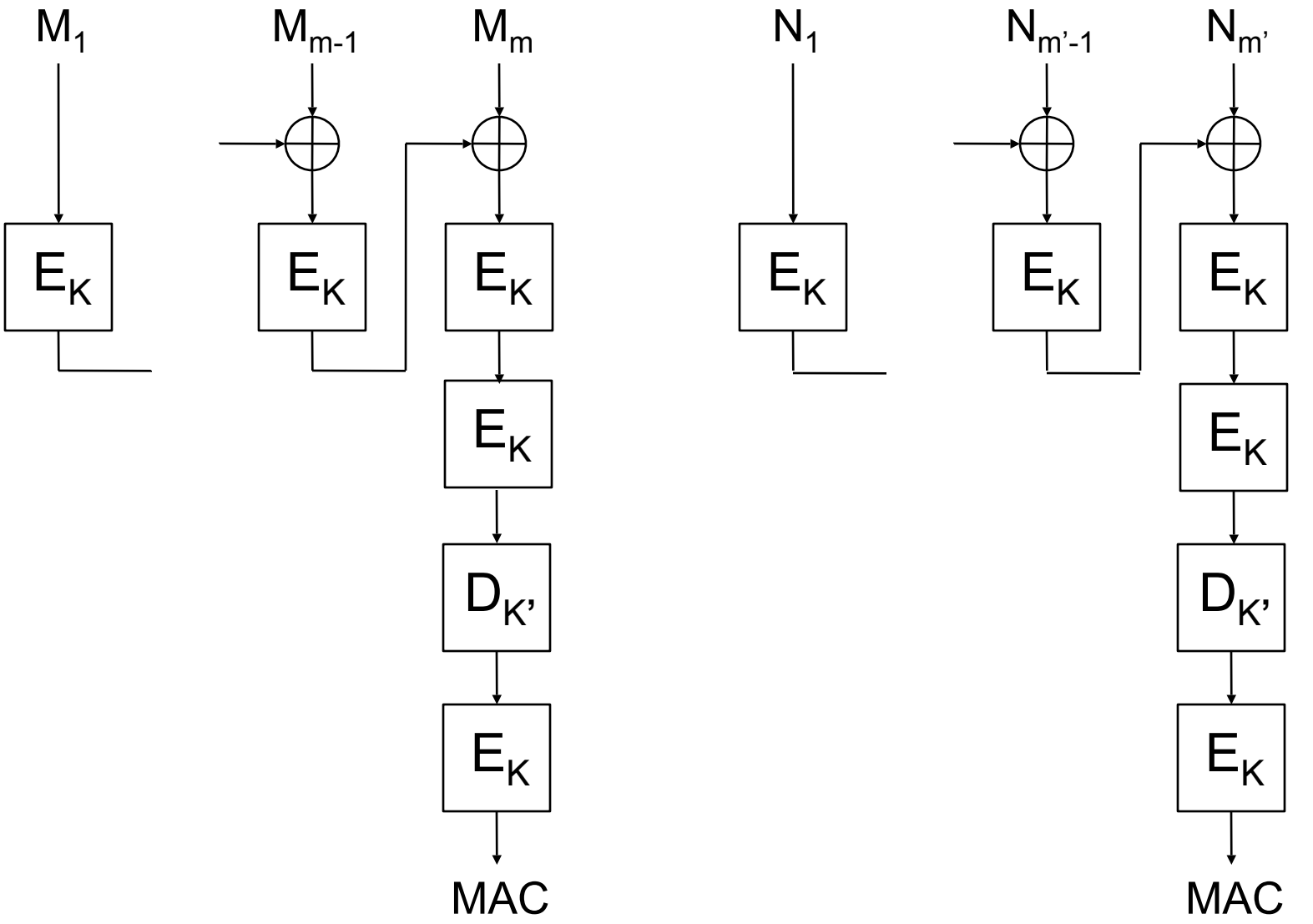
On peut forger le MAC de $M_j \,||\, R : \tau$

Sécurité d'EMAC : retrouver les clés

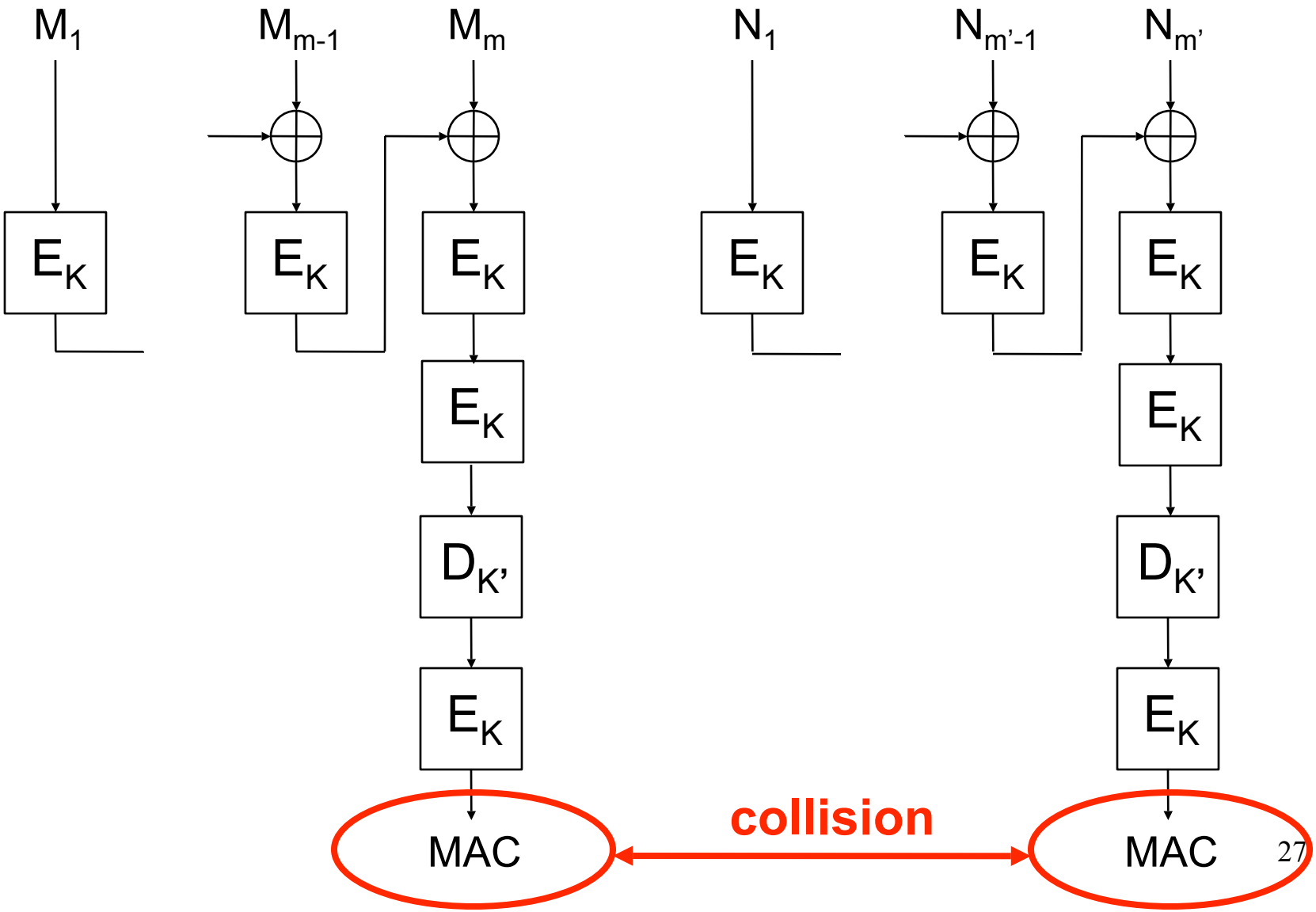


Pour des raisons d'efficacité et de sécurité, on choisit $E = DES$ avec une clé K et $E' = TDES$, avec les clés K et K'. Avec quelle complexité retrouve-t-on les clés K et K' ?

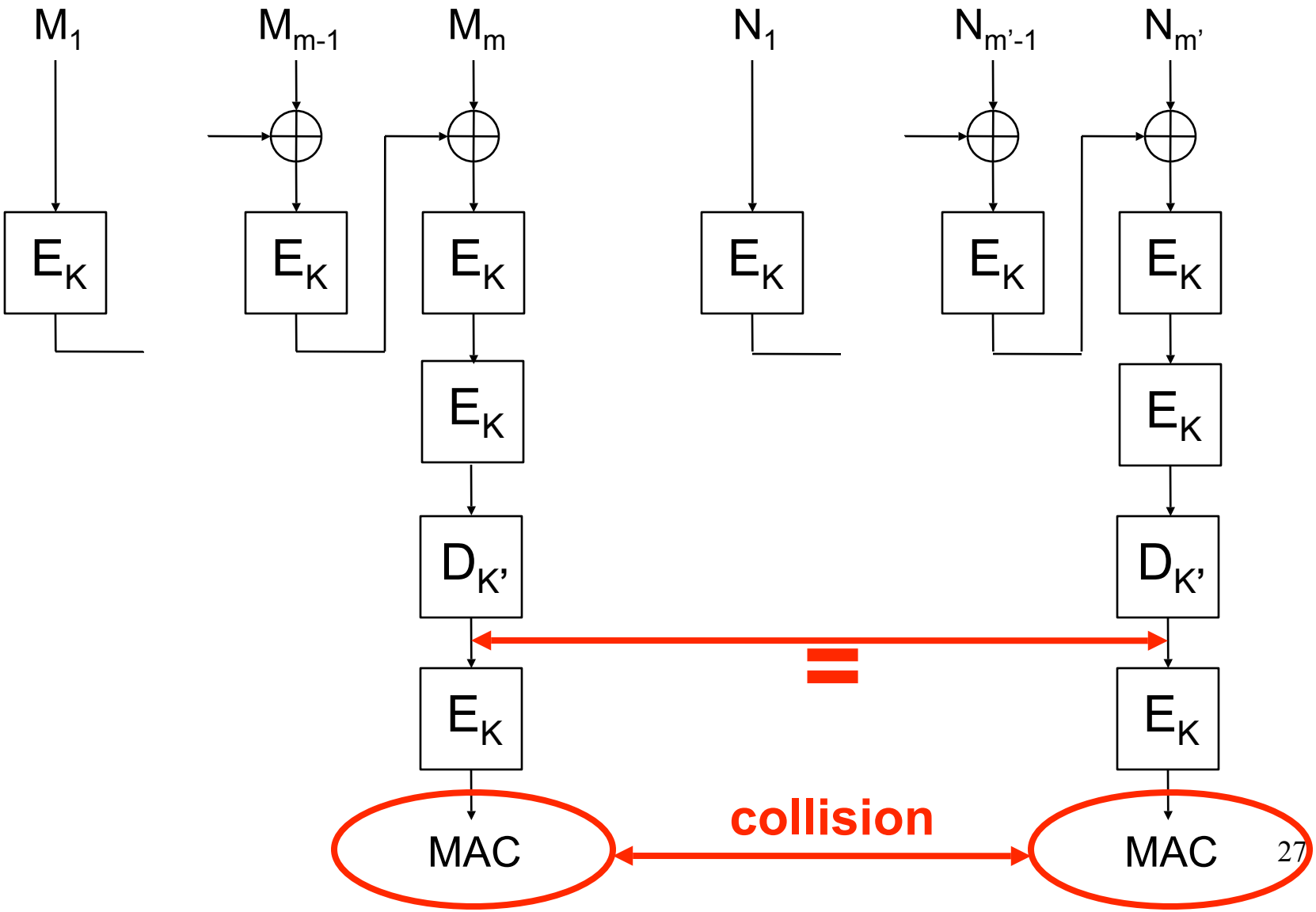
EMAC: DES_K et $TDES_{K,K'}$



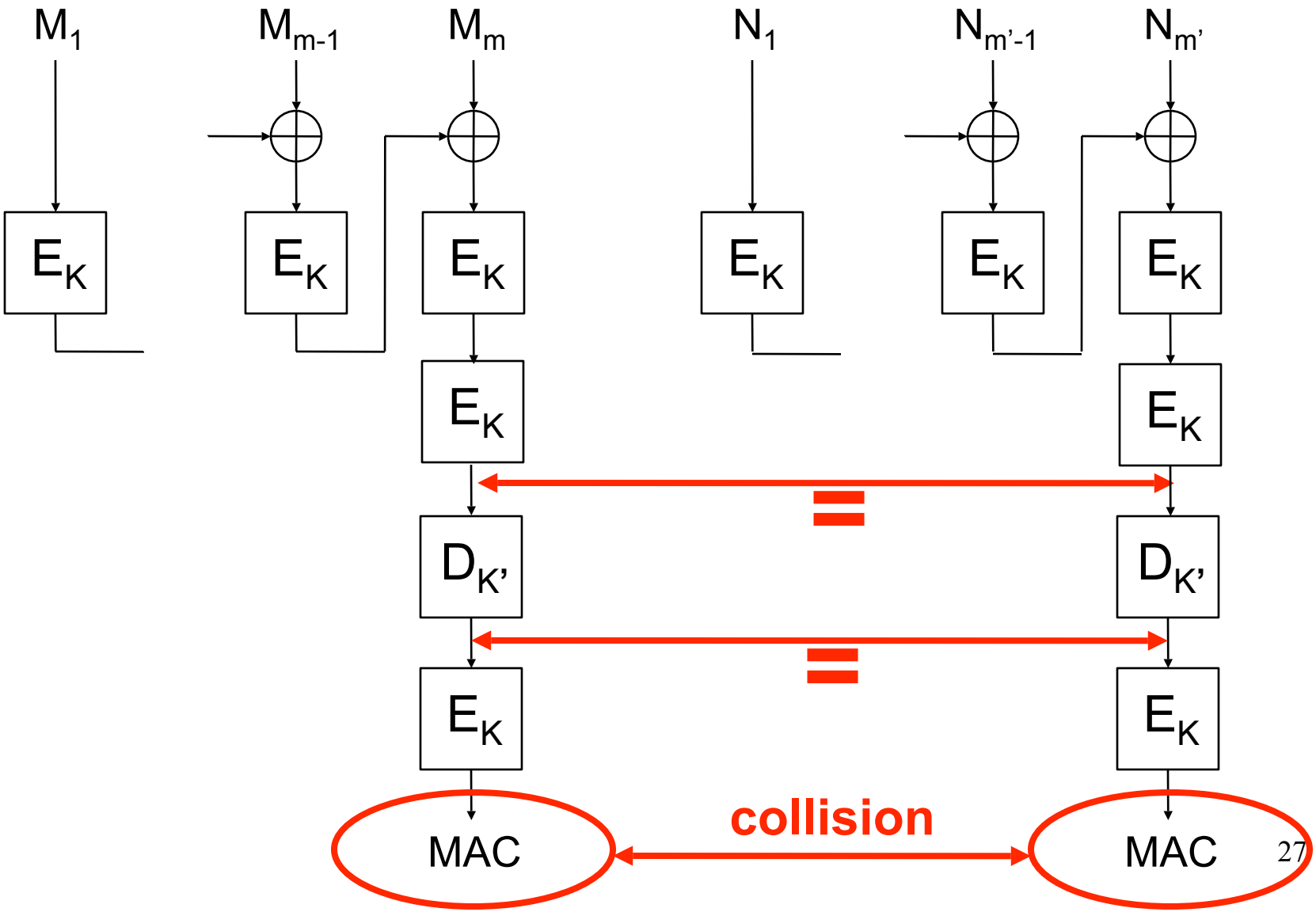
EMAC: DES_K et $TDES_{K,K'}$



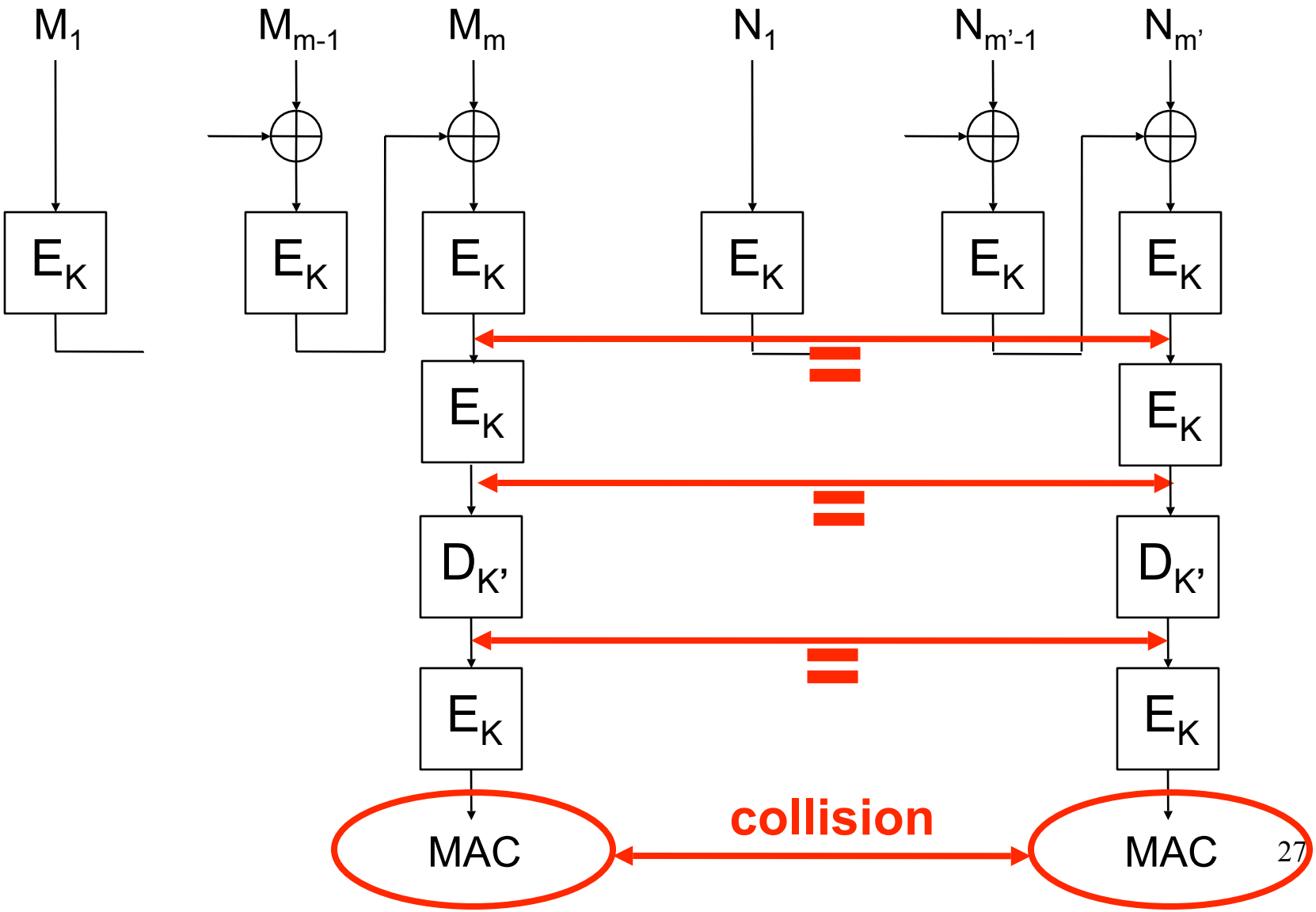
EMAC: DES_K et $TDES_{K,K'}$



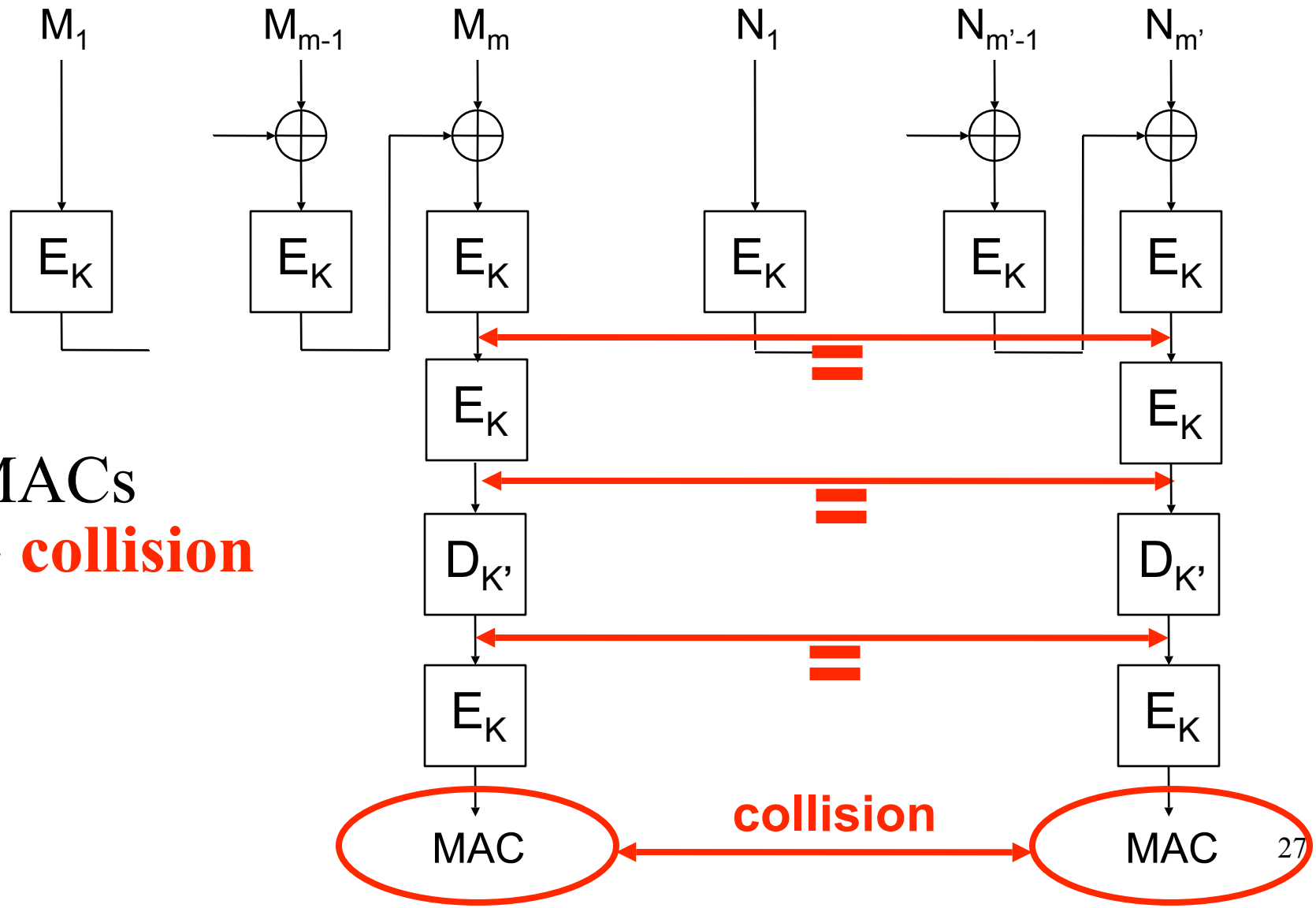
EMAC: DES_K et $TDES_{K,K'}$



EMAC: DES_K et $TDES_{K,K'}$



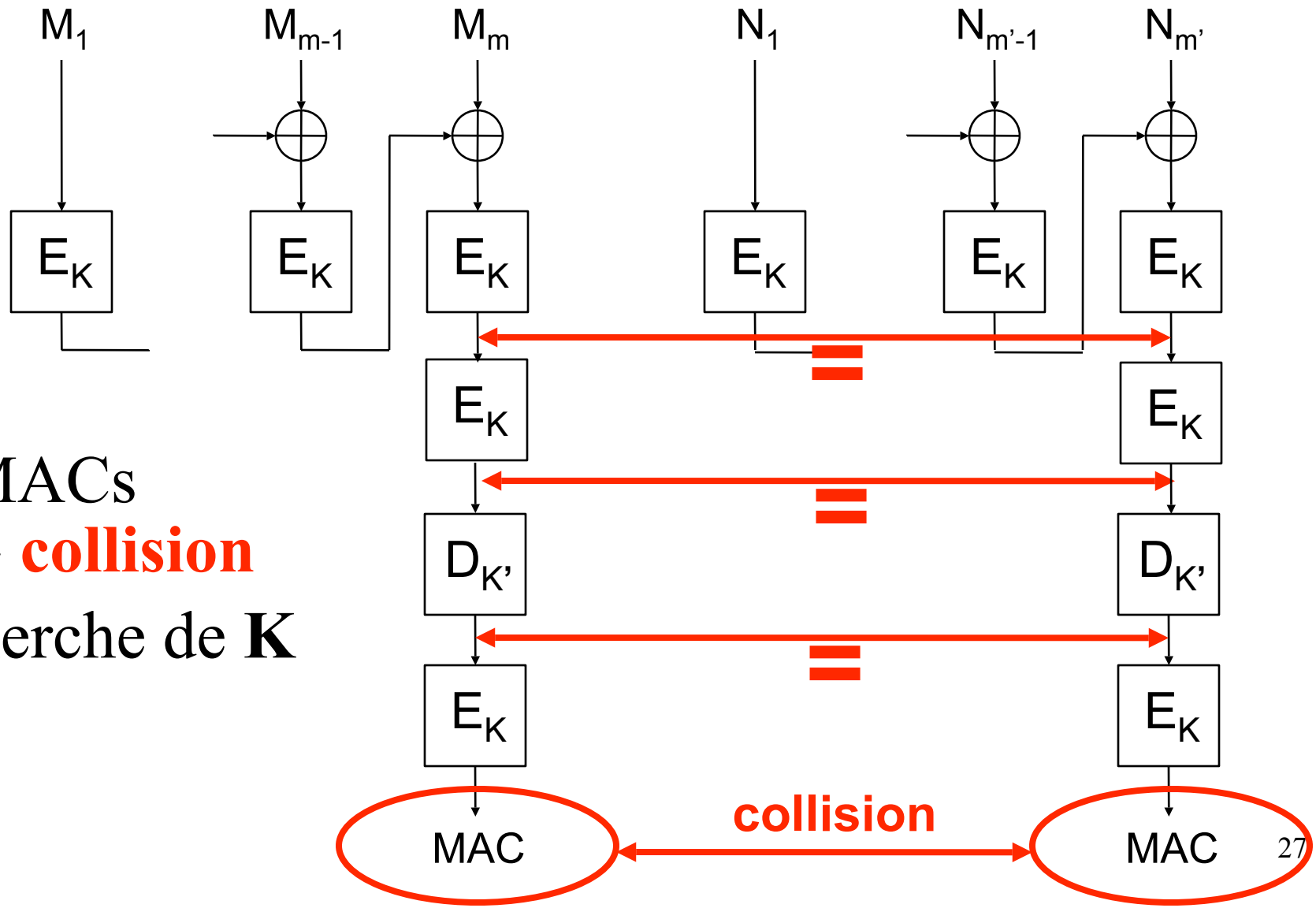
EMAC: DES_K et $TDES_{K,K'}$



2^{32} MACs

→ **collision**

EMAC: DES_K et $TDES_{K,K'}$

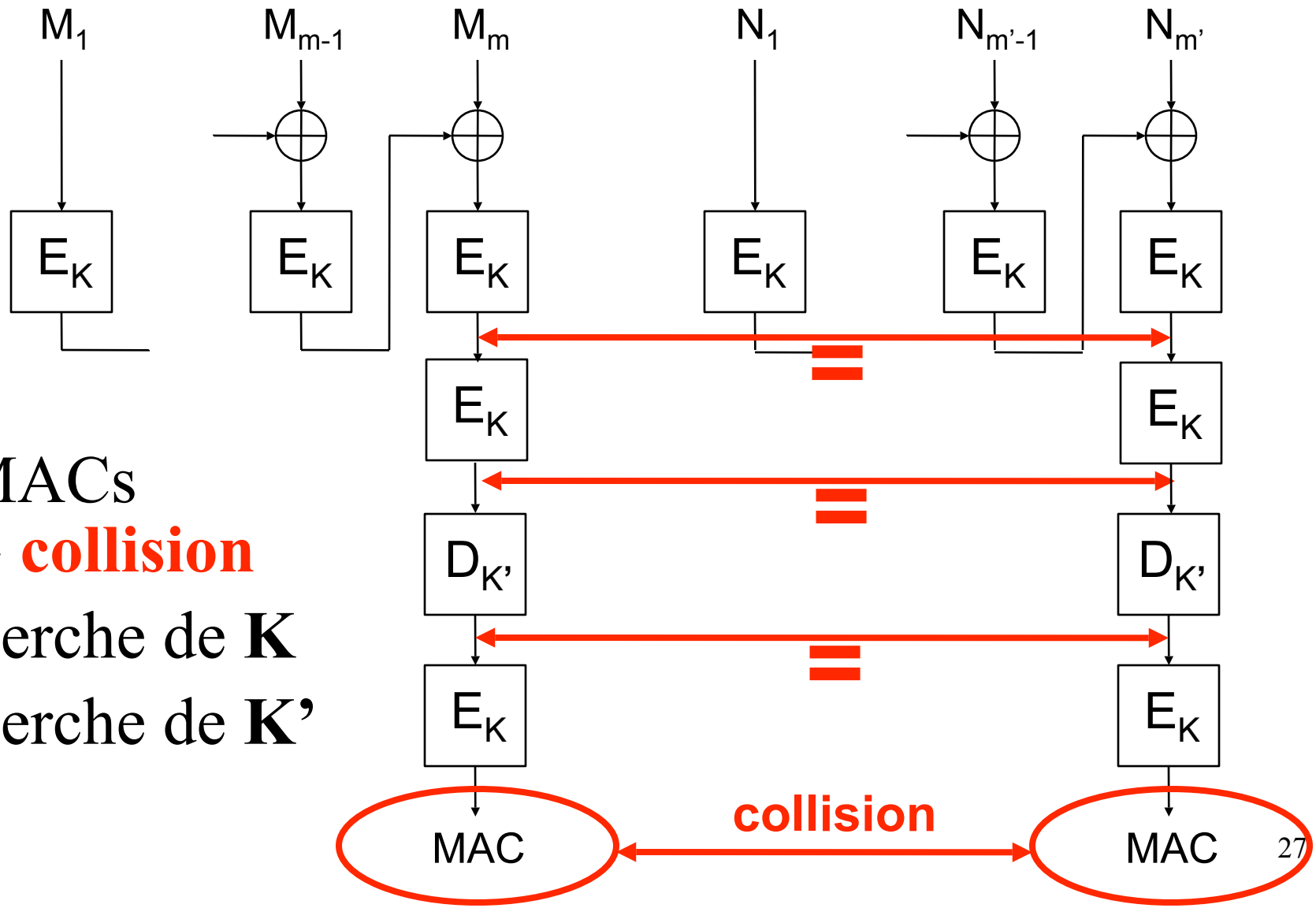


2^{32} MACs

→ **collision**

Recherche de **K**

EMAC: DES_K et $TDES_{K,K'}$



2^{32} MACs

→ **collision**

Recherche de **K**

Recherche de **K'**

EMAC: DES_K et $TDES_{K,K'}$

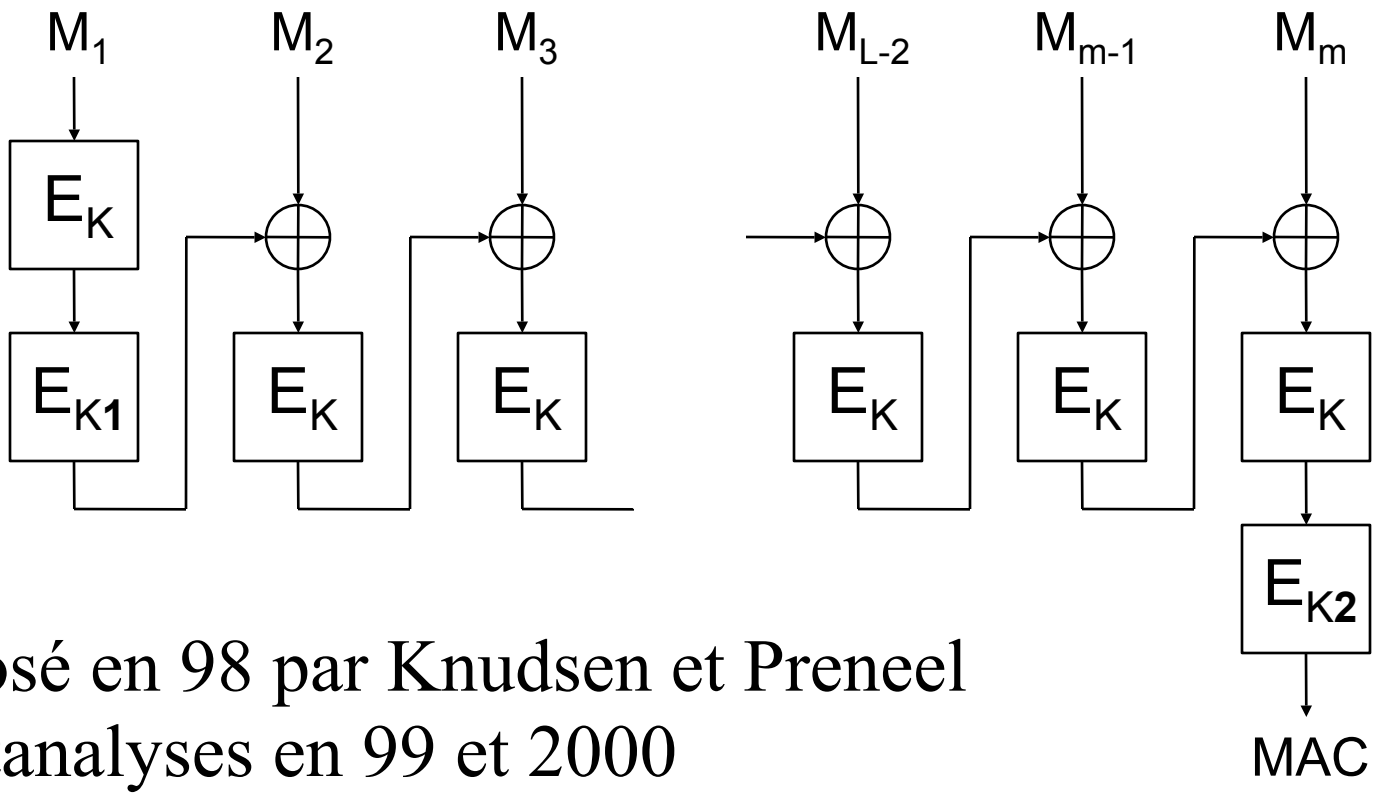
On peut retrouver les clés K et K' avec une attaque en 2^{56} en temps et en mémoire (couples clair / clé)

On génère 2^{32} (M,t), collision sur la sortie avec probabilité supérieure à 1/2 (paradoxe des anniversaires). Comme $TDES$ est une permutation, on a aussi collision en entrée de $TDES$

Puis, en temps 2^{56}, on essaie toutes les clés possibles avec comme condition CBC-MAC(M) = CBC-MAC(N) pour obtenir la clé K de DES

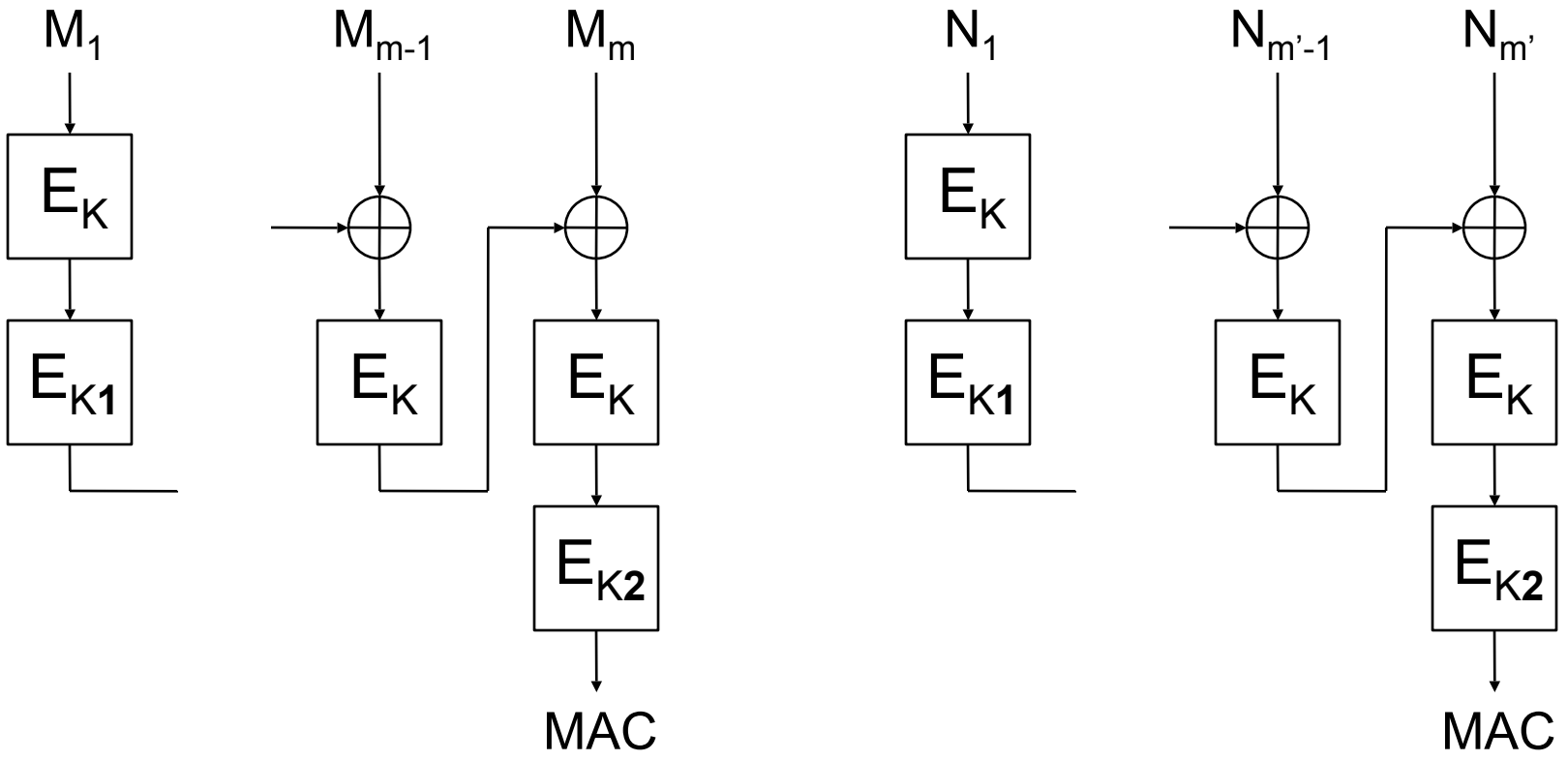
Enfin, si le $TDES$ utilise K et K' comme clés, on fait une recherche exhaustive pour trouver la clé K' en 2^{56}

Variante d'EMAC: MAC DES

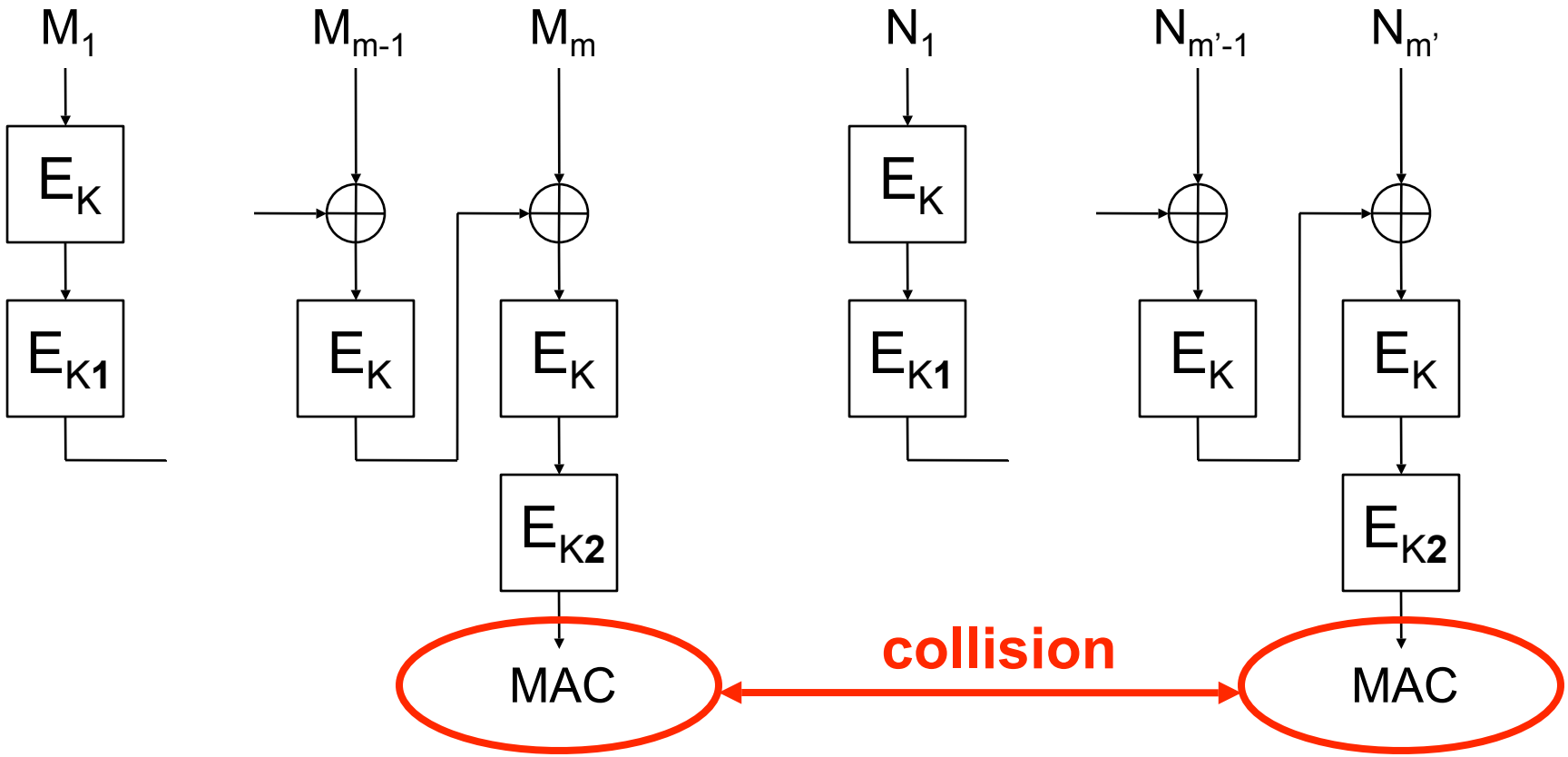


Proposé en 98 par Knudsen et Preneel
Cryptanalyses en 99 et 2000
(Coppersmith et Mitchell, puis
Coppersmith, Knudsen, Mitchell)

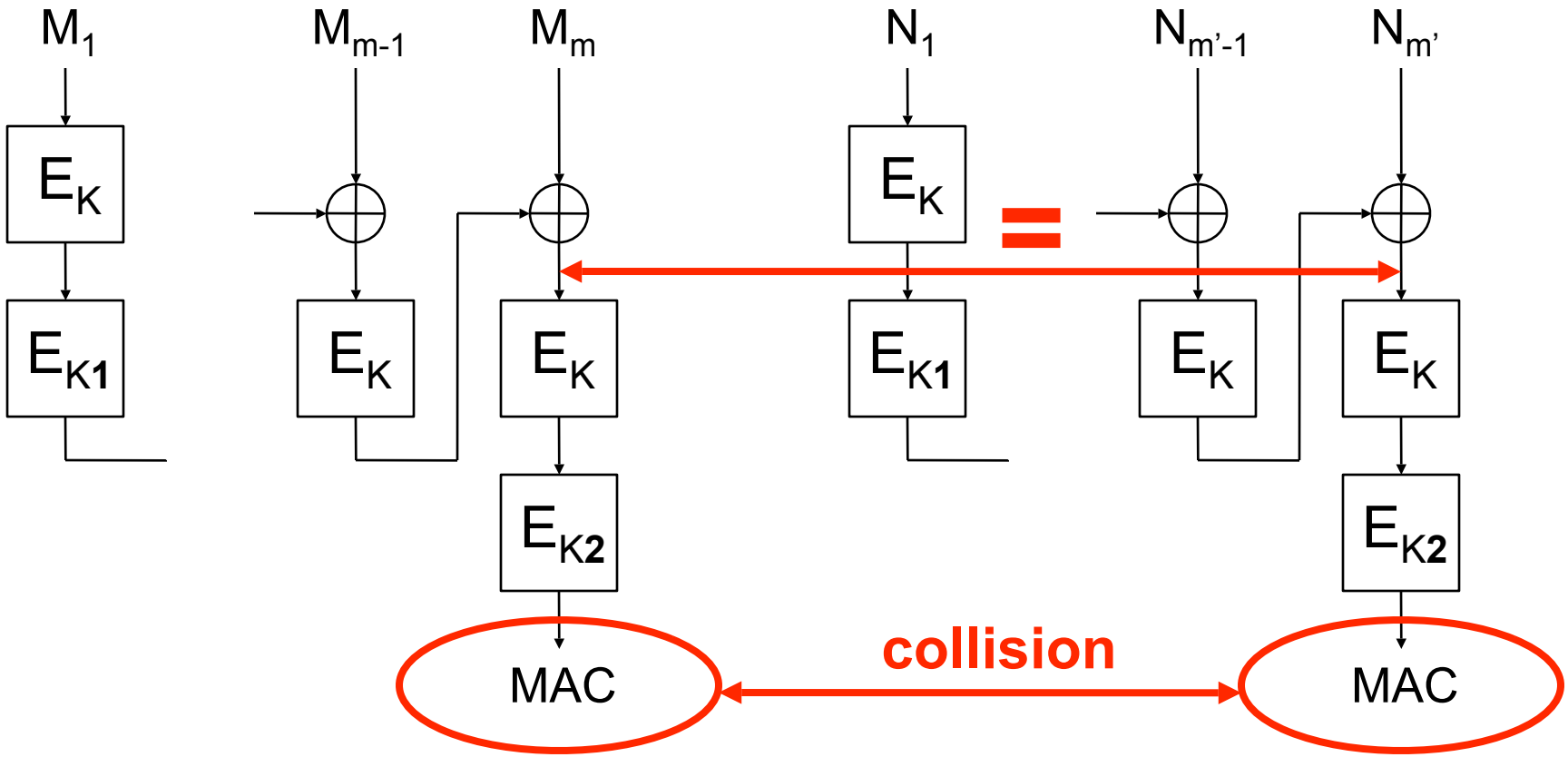
Attaque de MAC DES



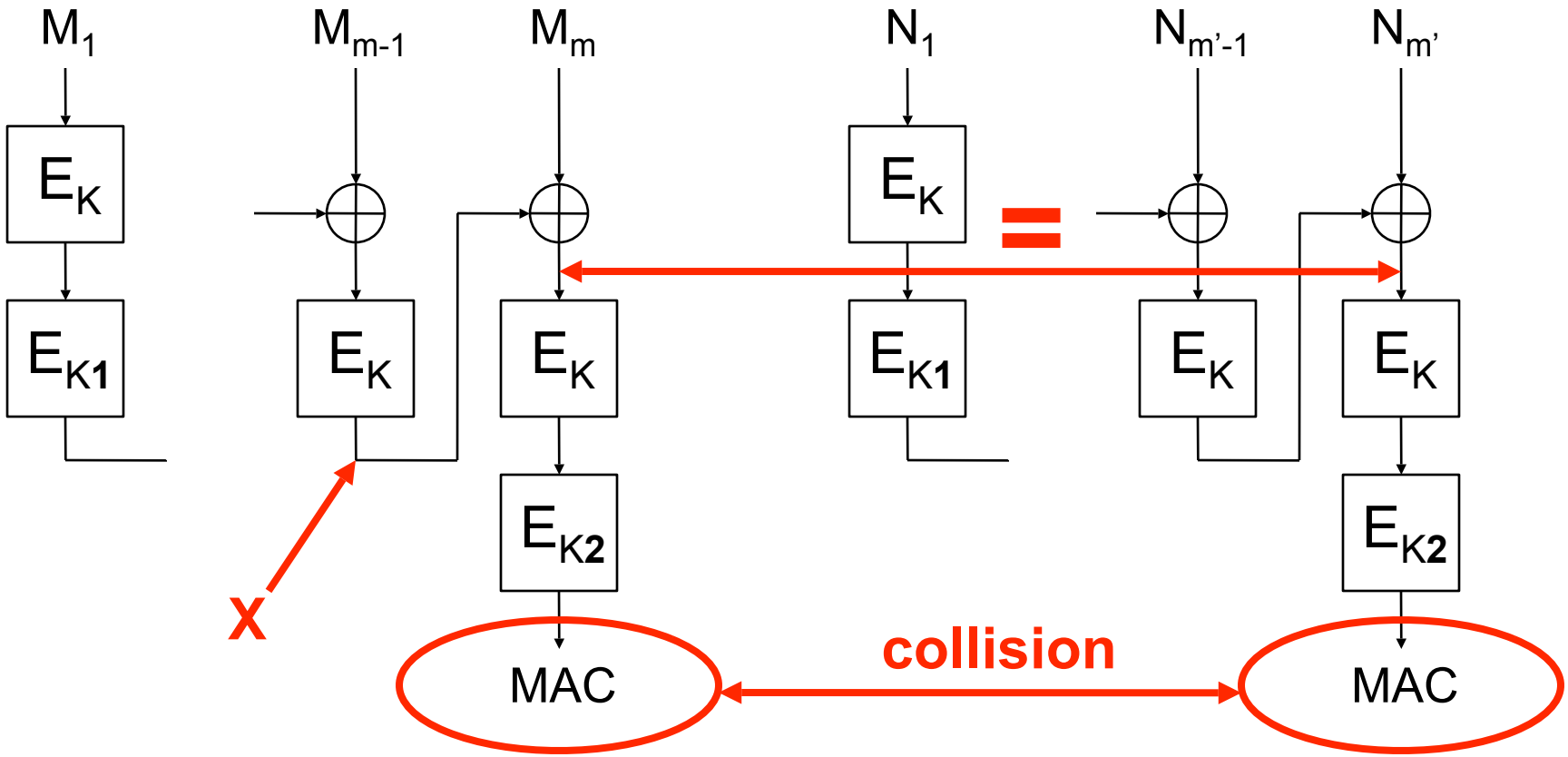
Attaque de MAC DES



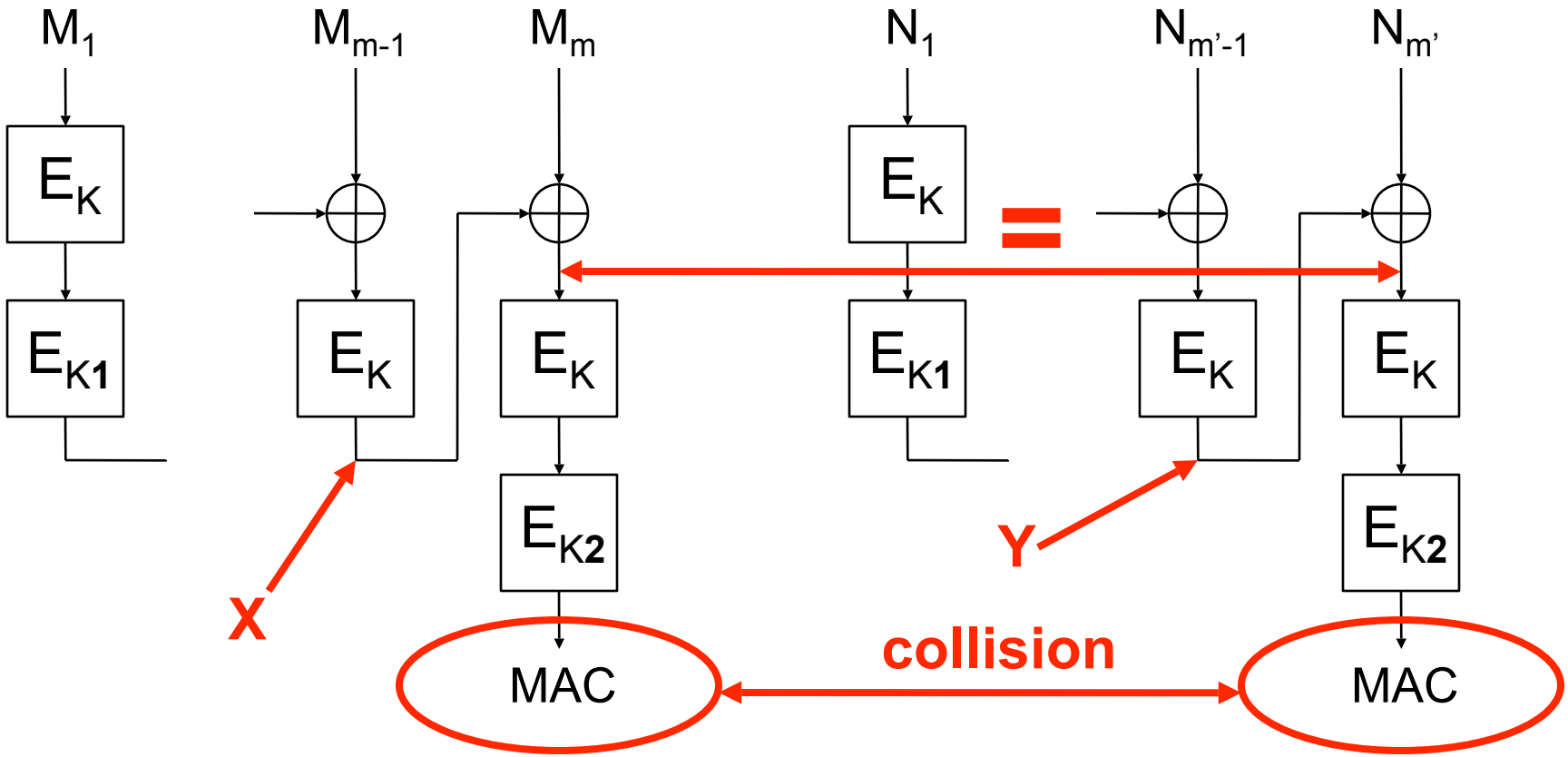
Attaque de MAC DES



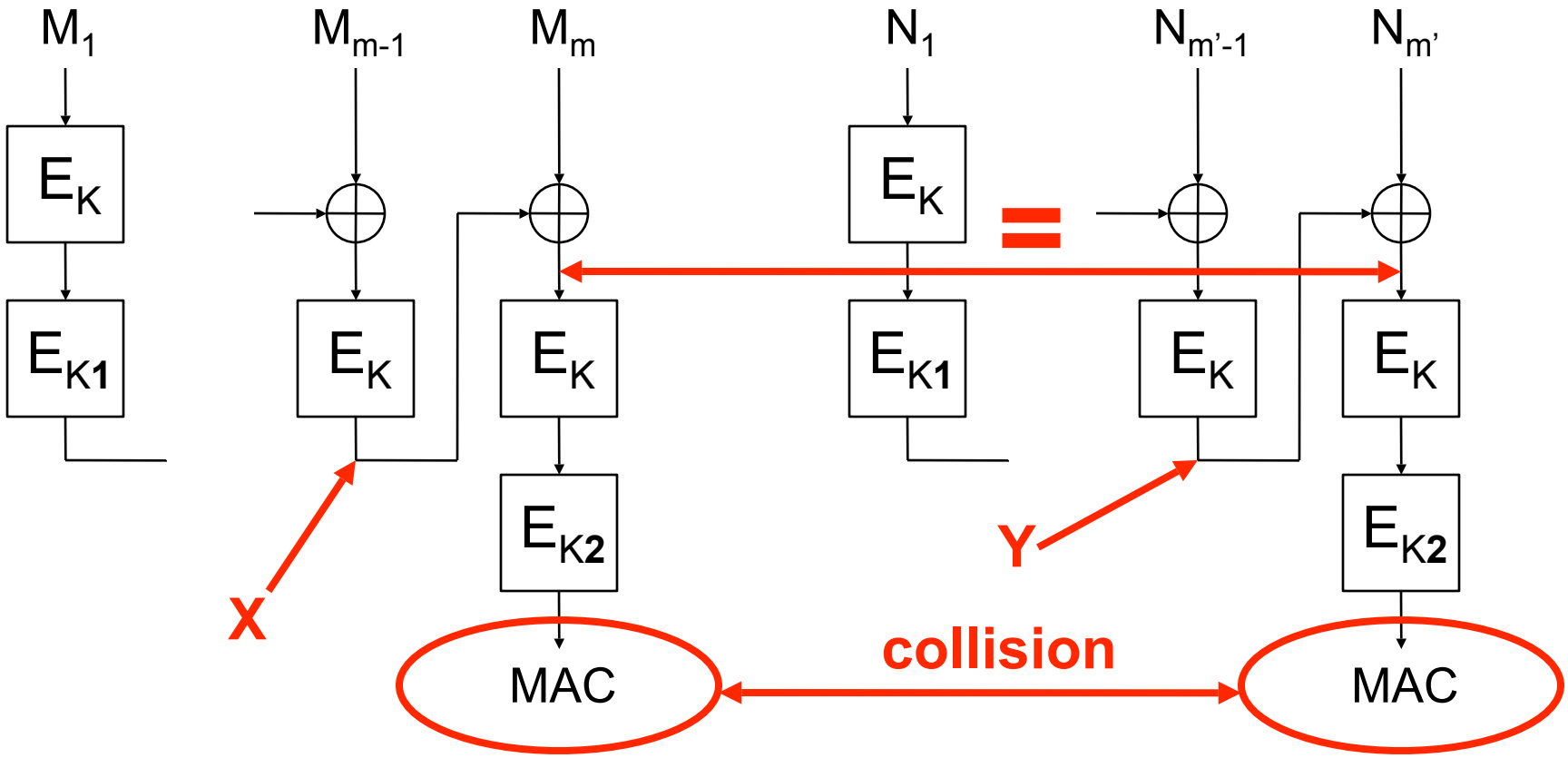
Attaque de MAC DES



Attaque de MAC DES

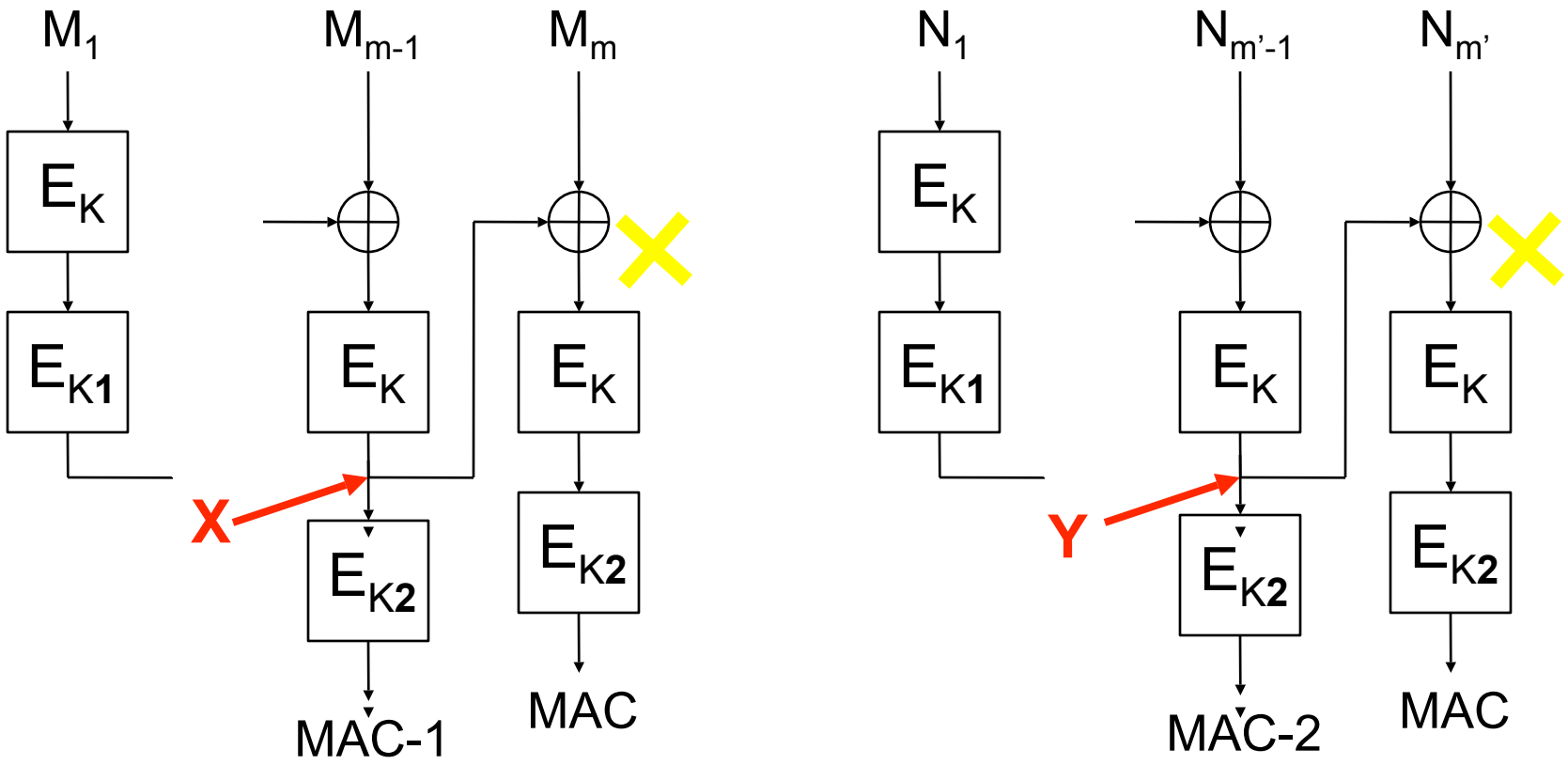


Attaque de MAC DES



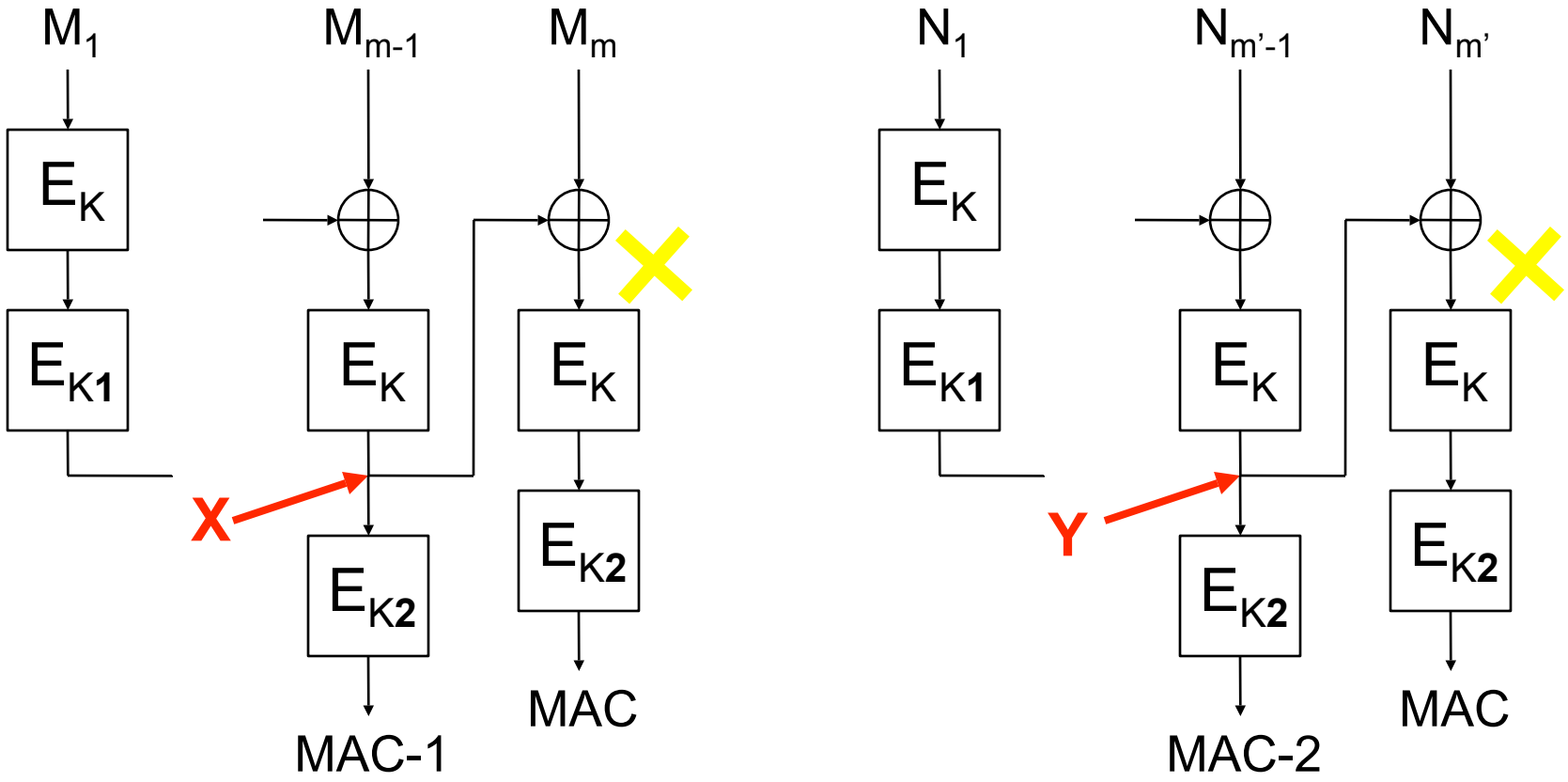
$$X \oplus M_m = Y \oplus N_{m'}$$

Attaque de MAC DES



$$X \oplus M_m = Y \oplus N_{m'}$$

Attaque de MAC DES



$$D_{K2}(MAC\text{-}1) \oplus M_m = D_{K2}(MAC\text{-}2) \oplus N_{m'}$$

Attaque de MAC DES

2^{32} MACs → collision

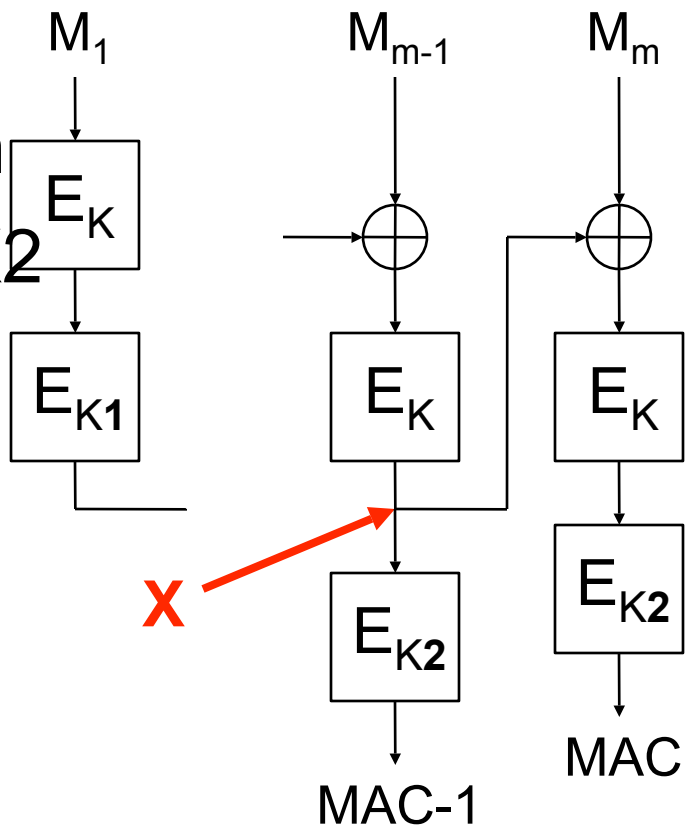
2 messages <u>choisis</u>, condition d'arrêt pour la recherche de K2

$E_K (D_{K2} (MAC\text{-}1) \oplus M_m)$

$= D_{K2} (MAC)$

→ condition d'arrêt pour la recherche de **K**

→ recherche de **K1**



EMAC et MAC DES

Si EMAC utilisé avec DES_K et $DES_{K'}$, on retrouve K et K' avec une attaque en 2^{56} en temps et en mémoire (même attaque que sur le double DES)

Si EMAC utilisé avec DES_K puis $TDES_{K,K'}$, on retrouve K et K' avec 2^{32} messages et deux recherches exhaustives en 2^{56} en temps

On retrouve les 3 clés sur MAC DES avec 2^{32} messages plus 2 choisis et avec trois recherches exhaustives en 2^{56} en temps

Complexité en key recovery

	temps	Mémoire	Couples (M,T)	
			connus	choisis
EMAC avec DES_K , $DES_{K'}$	$2 \cdot 2^{56}$	2^{56} paires (MAC,clé)	2	-
		232	232	-
EMAC avec DES_K, $TDES_{K,K'}$	$2 \cdot 2^{56}$	232	232	-
MAC DES	$3 \cdot 2^{56}$	232	232	2

MACs à base de fonctions de hachage

Définition

Fonction à sens unique $H : \{0,1\}^* \rightarrow \{0,1\}^t$

Sur un message M de longueur arbitraire, la fonction retourne un haché h de taille t fixée

Certaines fonctions prennent une clé en entrée

Exemples : SHA-1, MD5

Propriétés

Propriétés des fonctions de hachage

Résistance aux préimages : étant donné y il est difficile de trouver x tel que $y = H(x)$

Résistance aux collisions : il est difficile de trouver x et x' différents tels que

$$H(x) = H(x')$$

Résistance aux secondes préimages : étant donnés x et y tels que $y = H(x)$, il est difficile de trouver $x' \neq x$ tel que $H(x') = y$

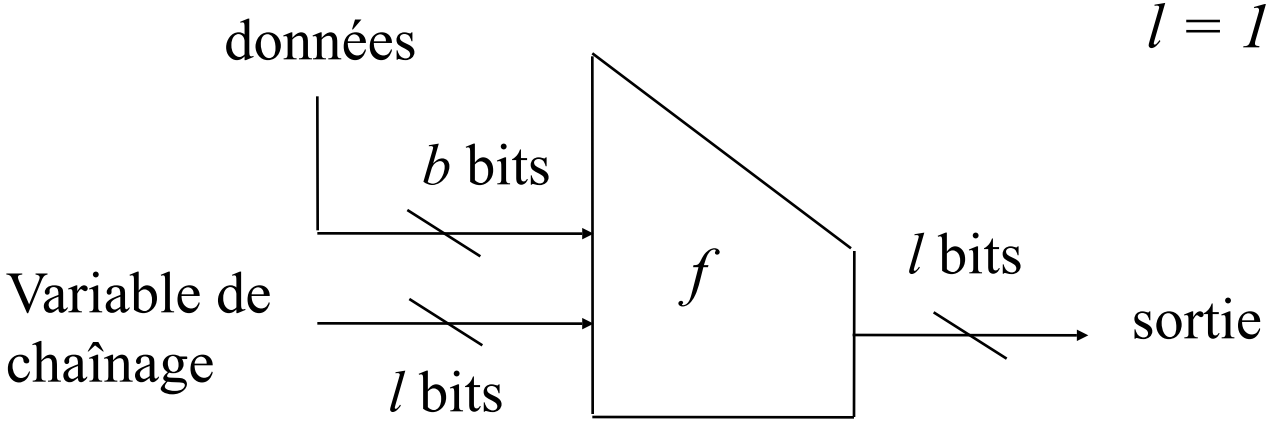
Constructions itérées

Méthode proposée en 1989 par Merkle puis Damgård

SHA-1 et MD5 sont basés sur cette méthode

Primitive utilisée : une fonction de compression f

Pour SHA-1 :
$l = 160$ et $b = 512$

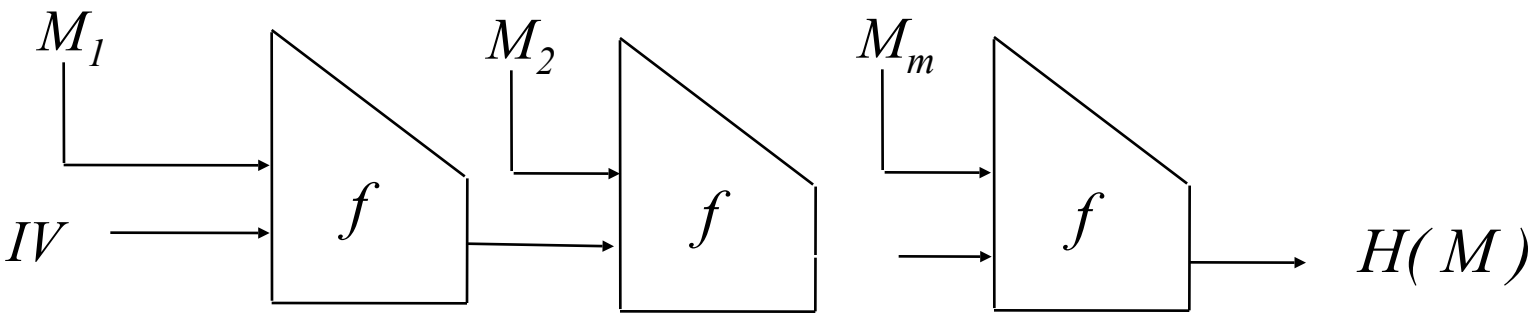


Fonctions de hachage

Message M dont on veut une empreinte

$$M = M_1 \,||\, \ldots \,||\, M_m$$

m blocs de b bits (512 bits pour SHA-1)



« Préfixe » construction

On considère le MAC suivant:

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

Attaque basique :

- On obtient le MAC τ de M
- On peut alors calculer le MAC de n'importe quelle extension de M : il suffit d'utiliser τ comme variable de chainage de la fonction de hachage dans le calcul de H :

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

HMAC

$\text{HMAC}_K(M) = \text{H}(K' \oplus \text{opad}, \text{H}(K' \oplus \text{ipad}, M))$

où ipad et opad sont des constantes:

- Ajouter des 0 à K pour avoir K' de 512 bits
- Calculer $K' \oplus$ ipad
- Concaténer $K' \oplus$ ipad et M
- Appliquer H au résultat pour obtenir T
- Calculer $K' \oplus$ opad
- Concaténer $K' \oplus$ opad et T
- Appliquer H au résultat pour obtenir le MAC

Sécurité de HMAC

Si la fonction de hachage H est résistante aux collisions avec probabilité ε, alors on ne peut pas forger de MACs avec une attaque à messages choisis avec probabilité supérieure à 2ε

Attaque encore possible ici dès que $2^{n/2}$ MACs ont été calculés (car $\varepsilon = 1/ 2^{n/2}$ au mieux)

Conclusions

Importance de la taille des blocs sur l'apparition de collisions

Importance de la taille de chaque clé pour la recherche exhaustive

Ce n'est pas la taille totale des clés qui est importante mais la taille de chacune d'elles

Conclusions

Problème du rejeu : on peut toujours renvoyer un message et son MAC interceptés sur le canal

Comment assurer à la fois confidentialité et intégrité : en une ou deux passes, chiffrement puis MAC, chiffrement et MAC ou MAC puis chiffrement ?

Un simple encodage du message avant le chiffrement peut-il assurer l'intégrité ?

Problème des « secure channels »

Chiffrement authentifié

Problématique

- Comment assurer à la fois confidentialité et intégrité : en une ou deux passes, chiffrement puis MAC, chiffrement et MAC ou MAC puis chiffrement ?
- Un simple encodage du message avant le chiffrement peut-il assurer l'intégrité ?

Chiffrement authentifié

- Les modes de chiffrement authentifié assurent
  - Confidentialité
  - Intégrité
- Combinaison mode de chiffrement et MAC :
  - MAC-And-Encrypt
  - MAC-Then-Encrypt
  - Encrypt-Then-MAC
- Mode dédié en « une seule passe »

Exigences de sécurité

- Confidentialité :
  - Sécurité sémantique contre les attaques à clairs choisis
  - Impossible de distinguer le chiffré de M de celui de M', même avec accès à un oracle de chiffrement
- Intégrité :
  - « Non forgeabilité » contre les attaques à messages choisis
  - Impossible de forger un nouveau chiffré valide

Attaques à chiffrés choisis

- Un schéma assurant à la fois intégrité et confidentialité garantit la sécurité contre les attaques à chiffrés choisis :

  IND-CPA + INT-CTXT $\Rightarrow$ IND-CCA

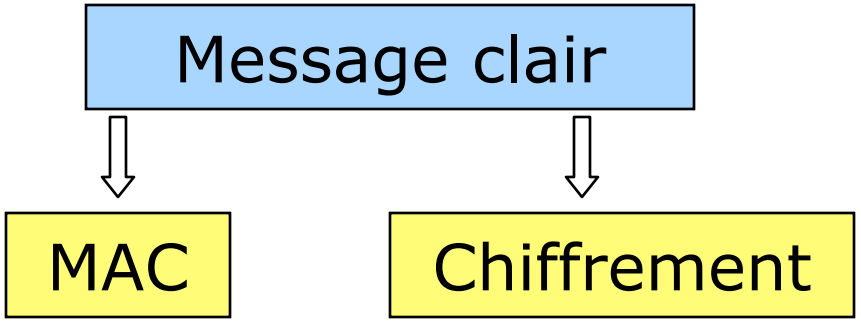
- Il est donc possible d'obtenir la sécurité au sens le plus fort en cryptographie symétrique

MAC-And-Encrypt

- Mode non sûr
- La confidentialité du message n'est pas assurée

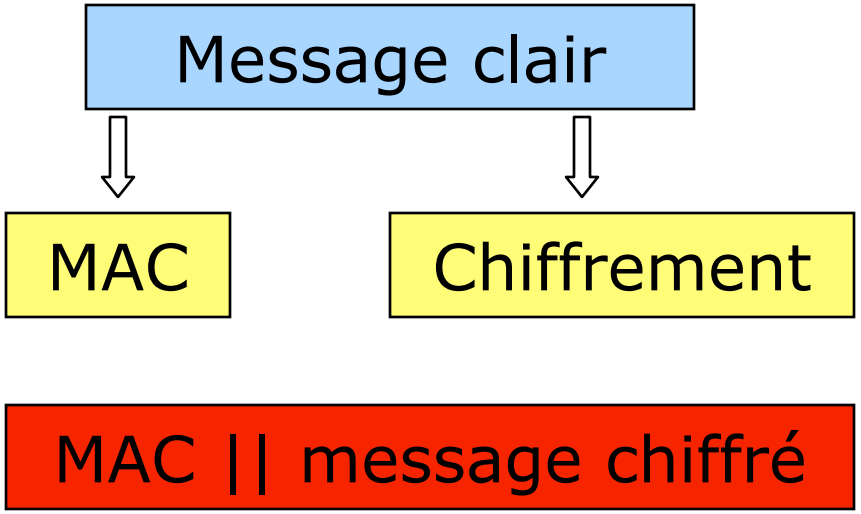
MAC-And-Encrypt

- Mode non sûr
- La confidentialité du message n'est pas assurée



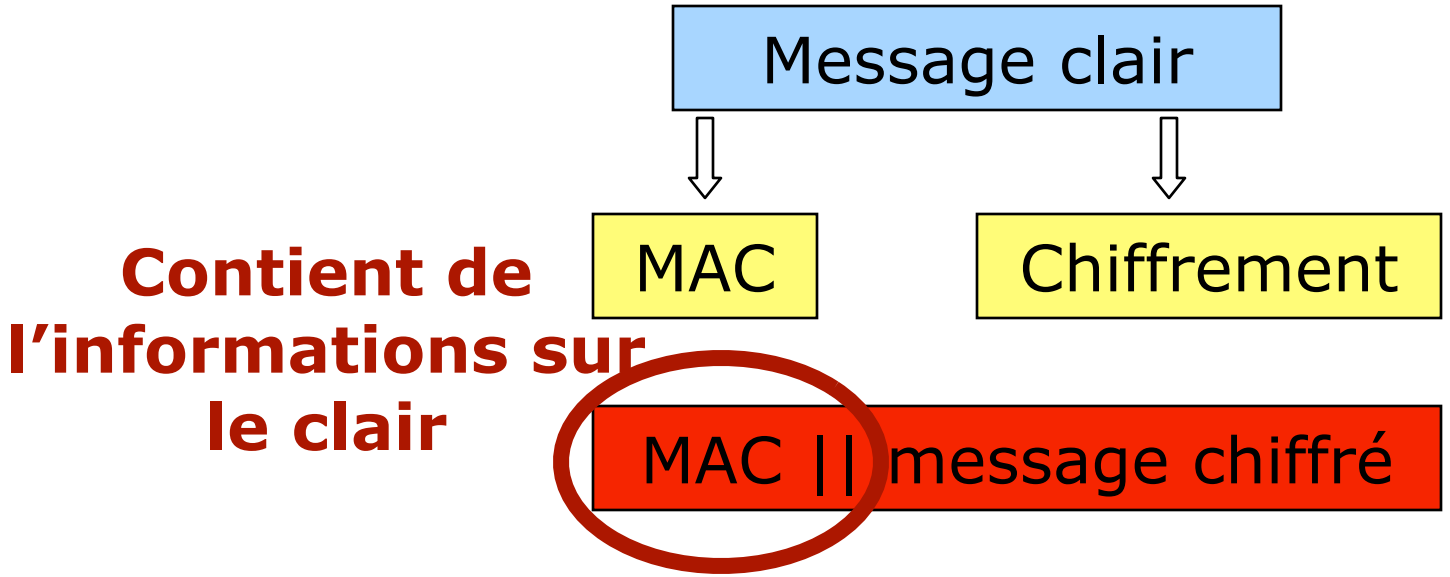
MAC-And-Encrypt

- Mode non sûr
- La confidentialité du message n'est pas assurée



MAC-And-Encrypt

- Mode non sûr
- La confidentialité du message n'est pas assurée

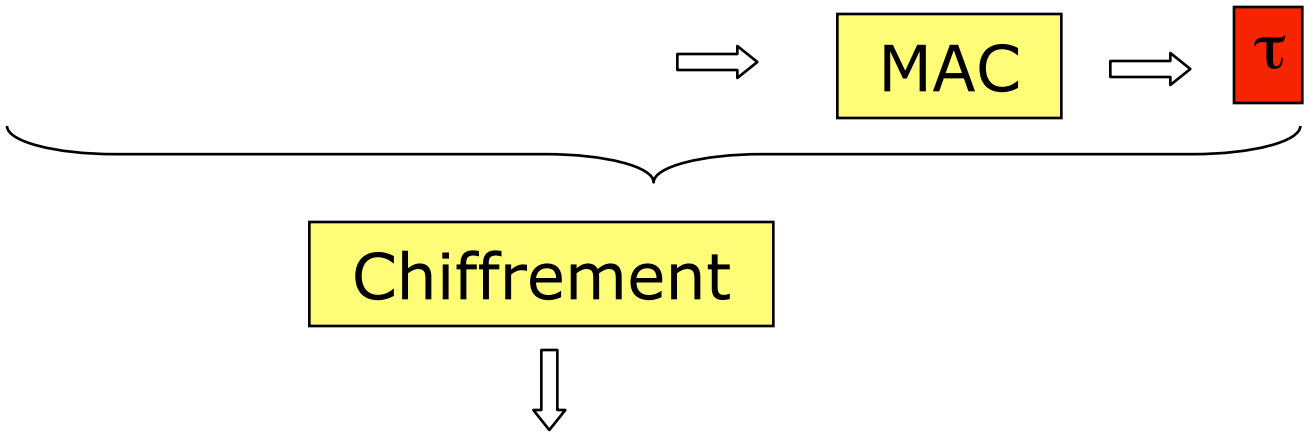


MAC and Encrypt : exemple

- Schéma de MAC sûr : $\tau = MAC_{Km}(M)$
- Schéma de chiffrement sûr : $C=E_{ke}(M)$
- Composition :
  - Message de n bits $M=M_0M_1...M_n$
  - $\tau' = M_0 \,||\, \tau$ et $C = E_{ke}(M)$
  - Le chiffré authentifié est (τ' , C)
- Attaque : pas de confidentialité, un bit du message est transmis en clair

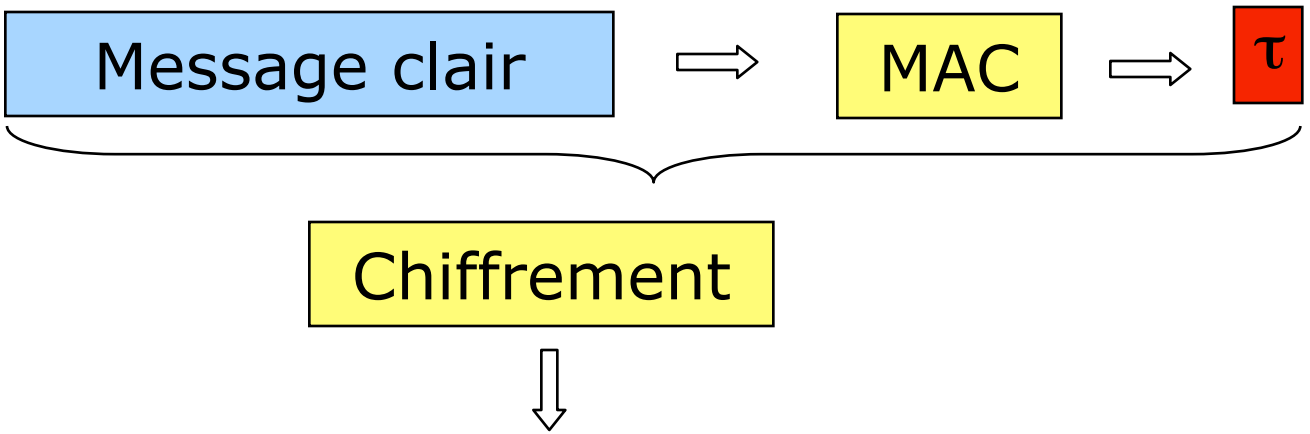
MAC-Then-Encrypt

- Mode non génériquement sûr
- En pratique, on peut construire des schémas sûrs avec ce principe



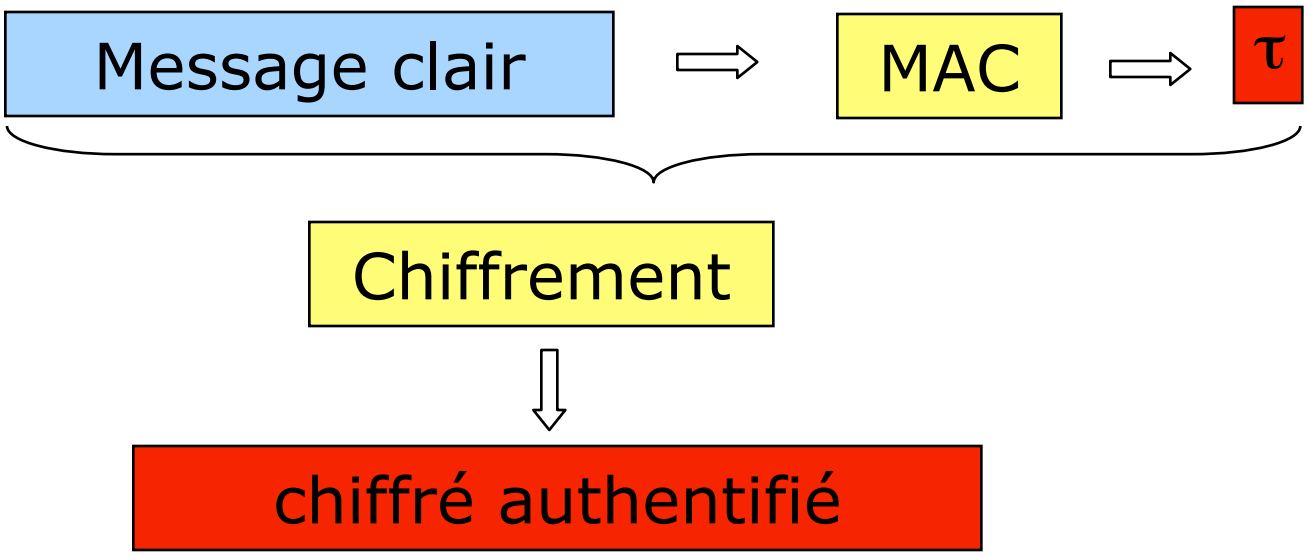
MAC-Then-Encrypt

- Mode non génériquement sûr
- En pratique, on peut construire des schémas sûrs avec ce principe



MAC-Then-Encrypt

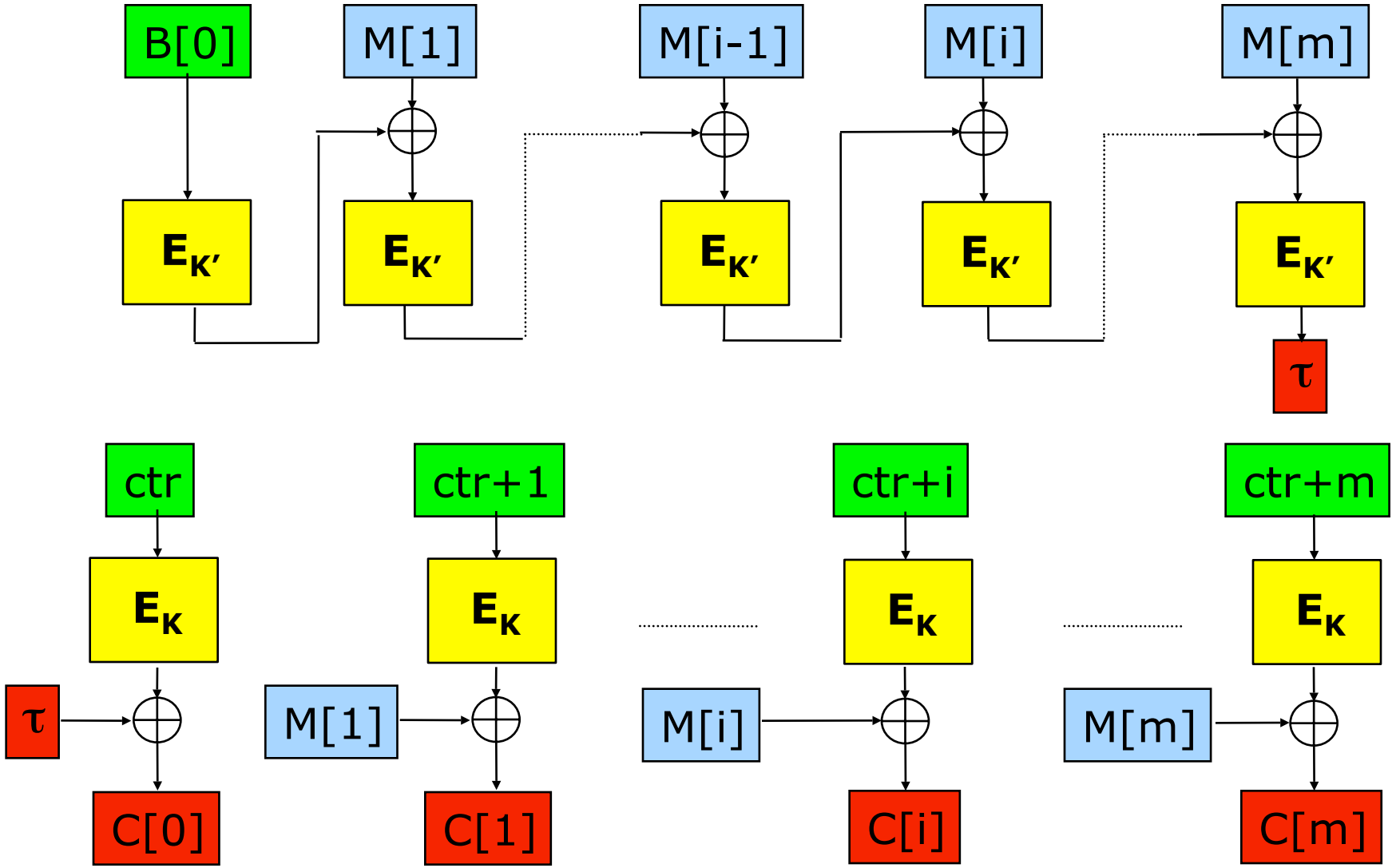
- Mode non génériquement sûr
- En pratique, on peut construire des schémas sûrs avec ce principe



Exemple MAC-Then-Encrypt

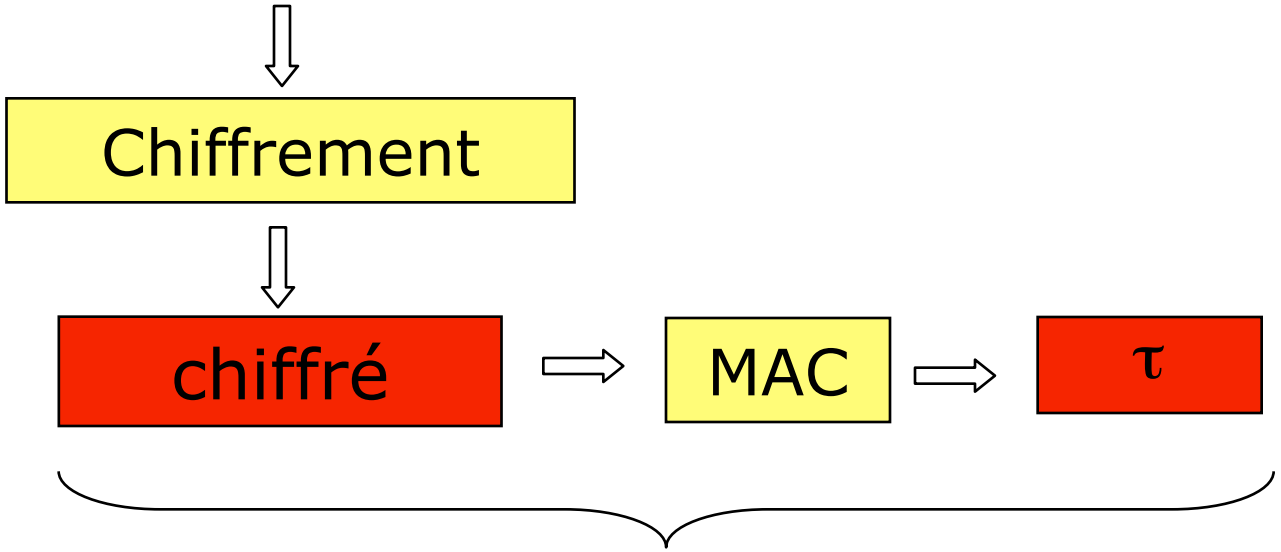
- CCM proposé par Housley, Whiting et Ferguson
- Draft du NIST en 2003, mode d'opération pour l'AES
- CBC-MAC then CTR
  - Gestion des données associées
  - Preuve de sécurité

Mode CCM


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]

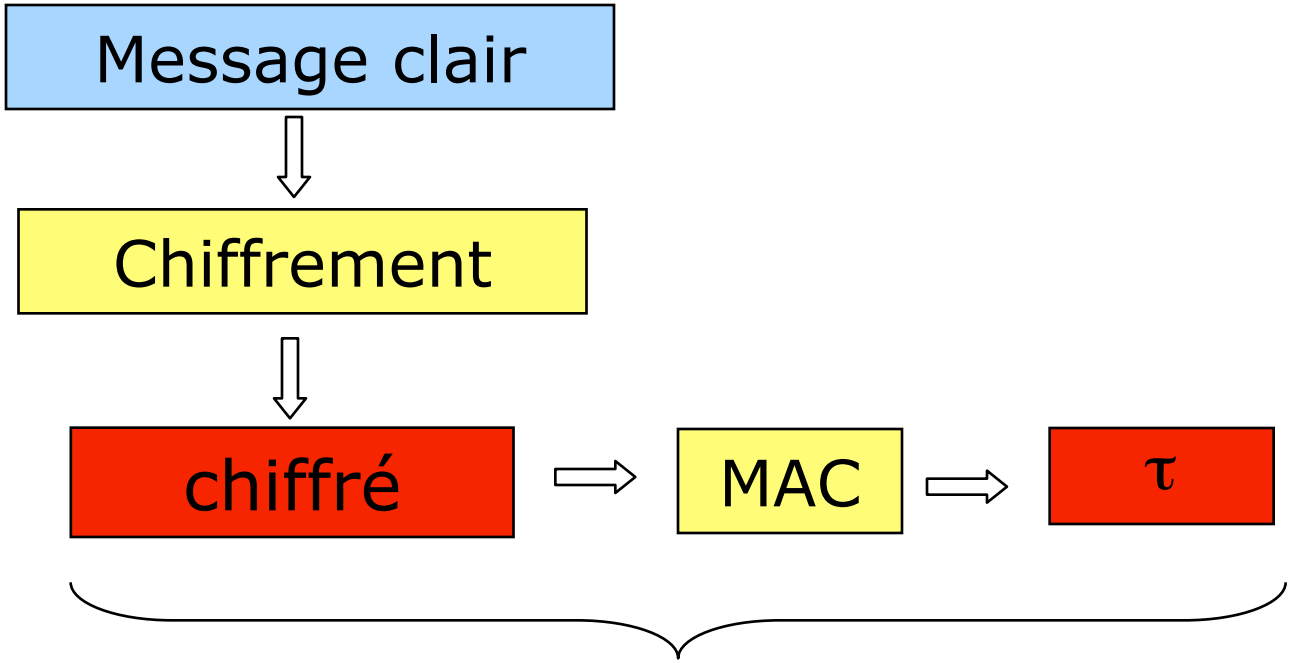
Encrypt-Then-MAC

- Si le mode de chiffrement est « sûr » et si le MAC assure l'intégrité alors cette composition est « sûre »



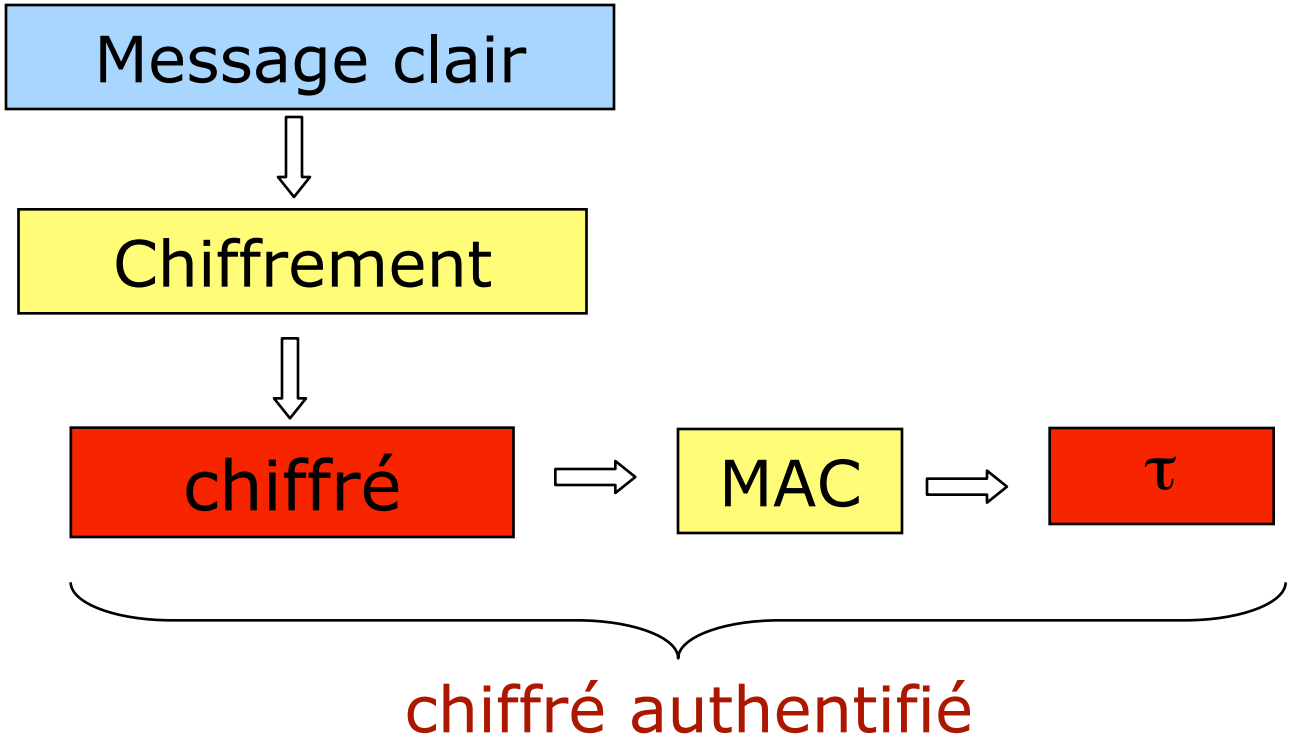
Encrypt-Then-MAC

- Si le mode de chiffrement est « sûr » et si le MAC assure l'intégrité alors cette composition est « sûre »

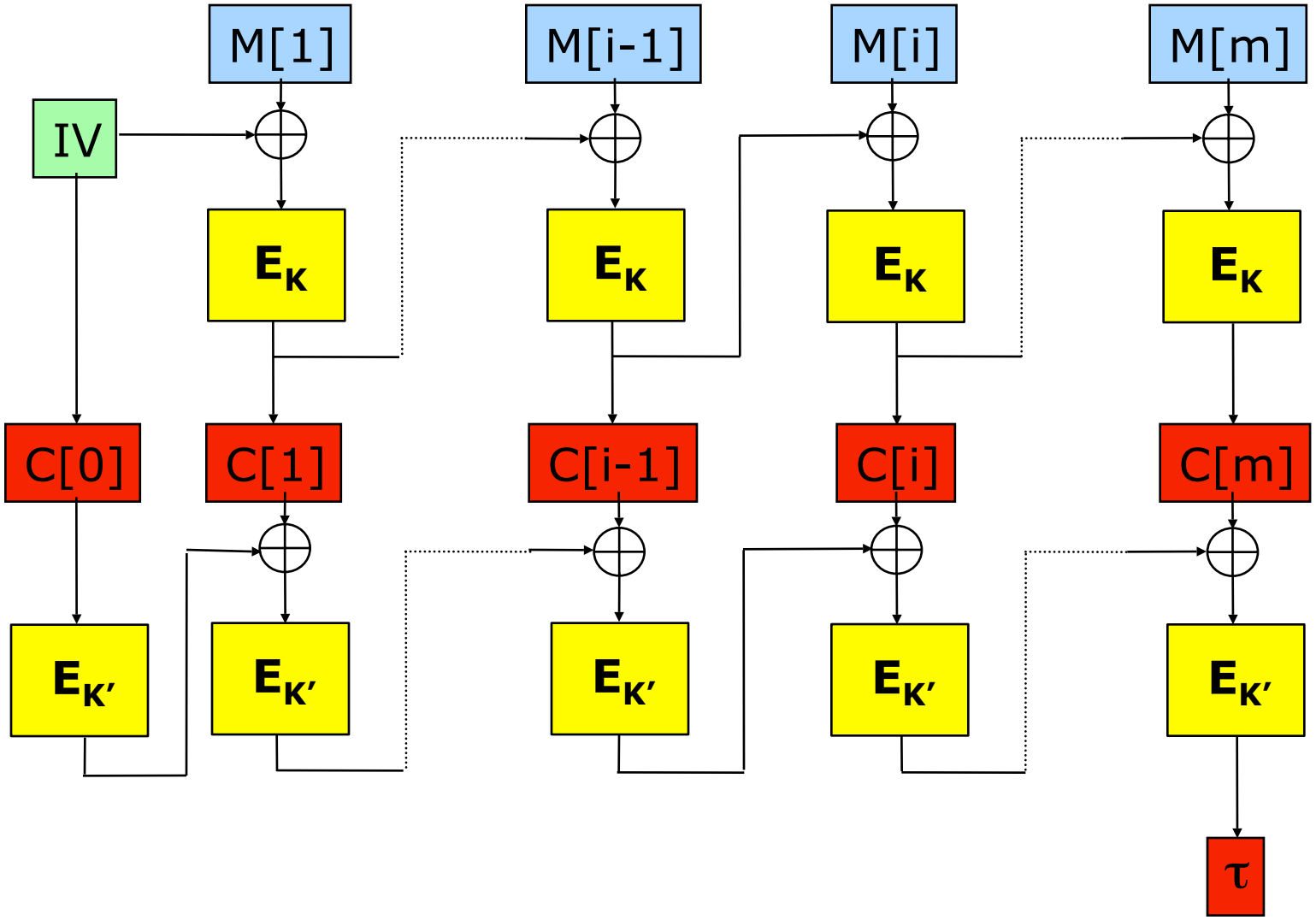


Encrypt-Then-MAC

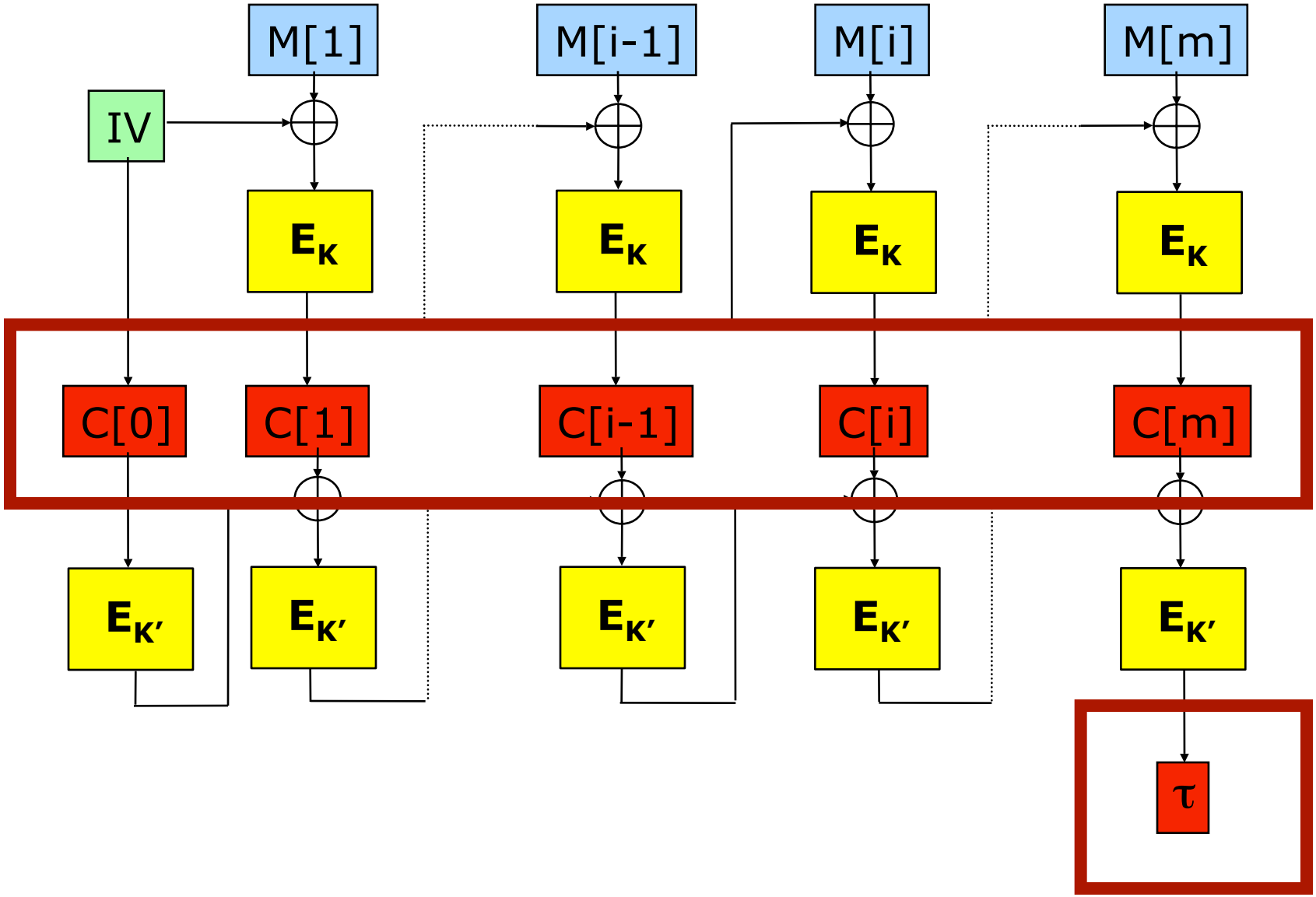
- Si le mode de chiffrement est « sûr » et si le MAC assure l'intégrité alors cette composition est « sûre »



Encrypt-Then-MAC



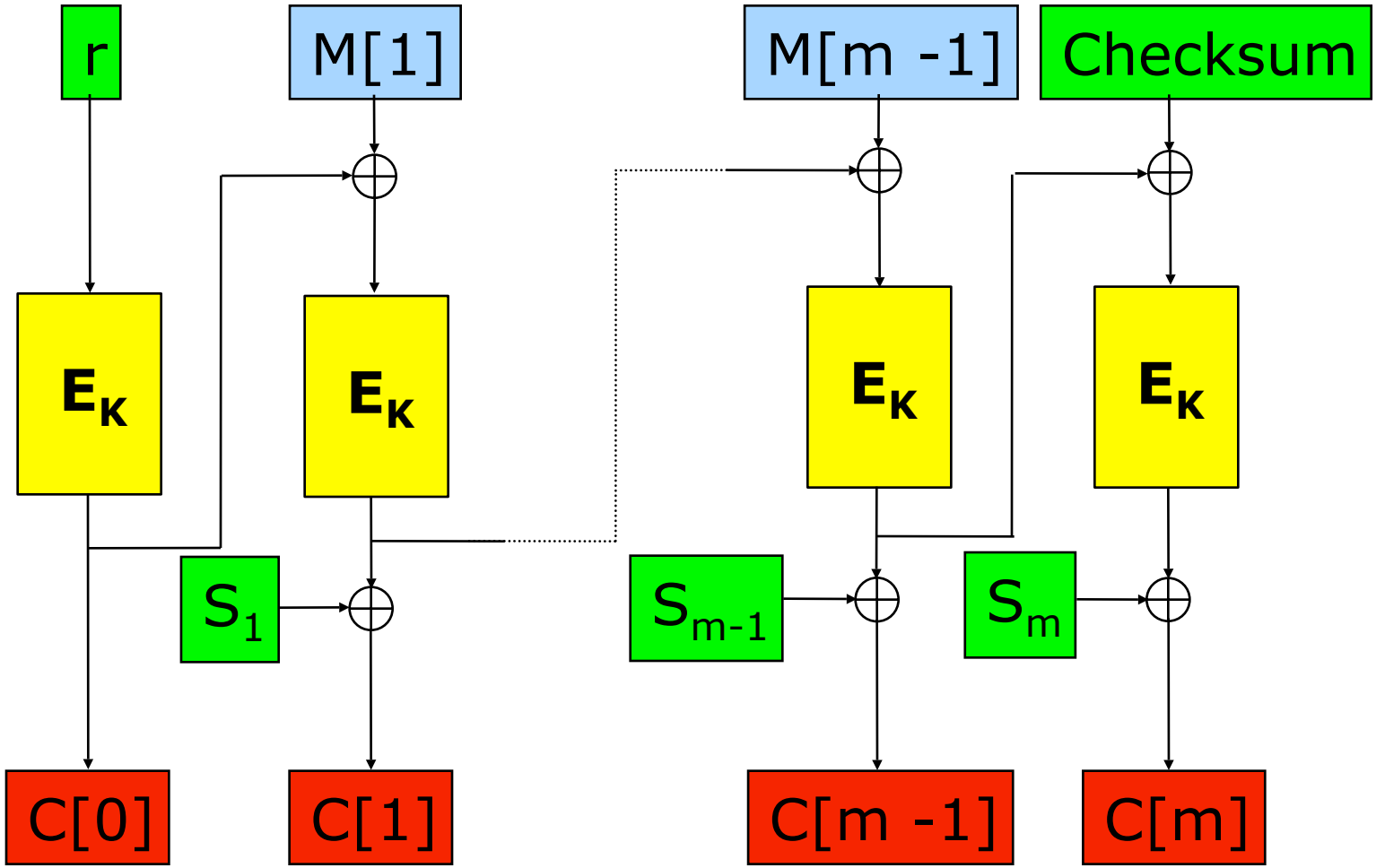
Encrypt-Then-MAC



Modes en une passe

- Le message n'est traité qu'une seule fois :
  - Plus rapide : quasiment aussi efficace qu'un chiffrement seul
  - Une seule clé
- Exemples : IAPM, IACBC, OCB, …

Un exemple : IACBC



IACBC

- Génération des masques (Ha01) :

$$S_i = N \cdot (<2i , r>)$$

où r est une valeur aléatoire

N est une matrice secrète

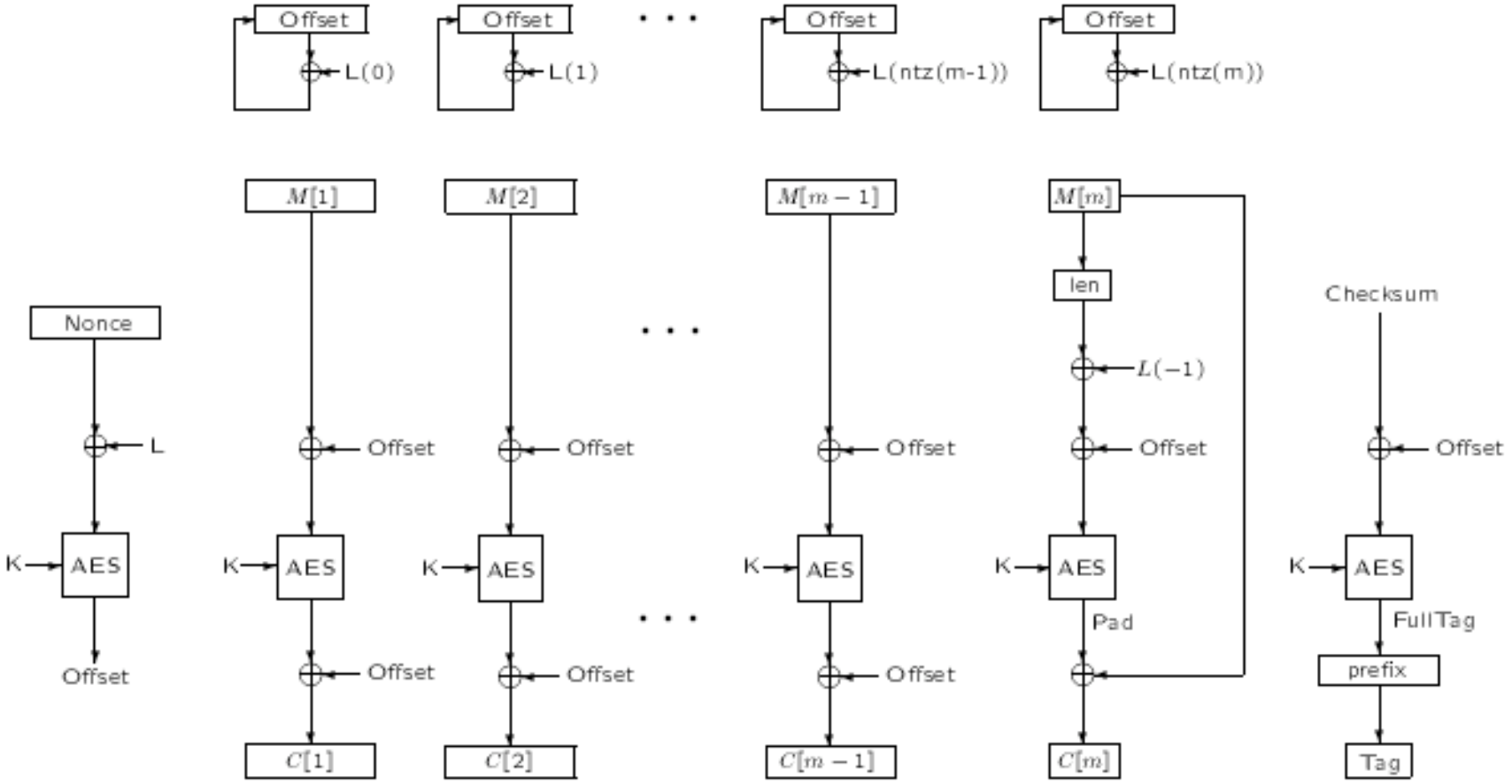
- Les masques sont indépendants deux à deux
- Génération des masques (Jutla01) :
  - Utilisation d'une primitive cryptographique
  - Non nécessaire pour la preuve
- Prouvé sûr IND-CPA et INT-CTXT (donc aussi IND-CCA)

Autre exemple : OCB

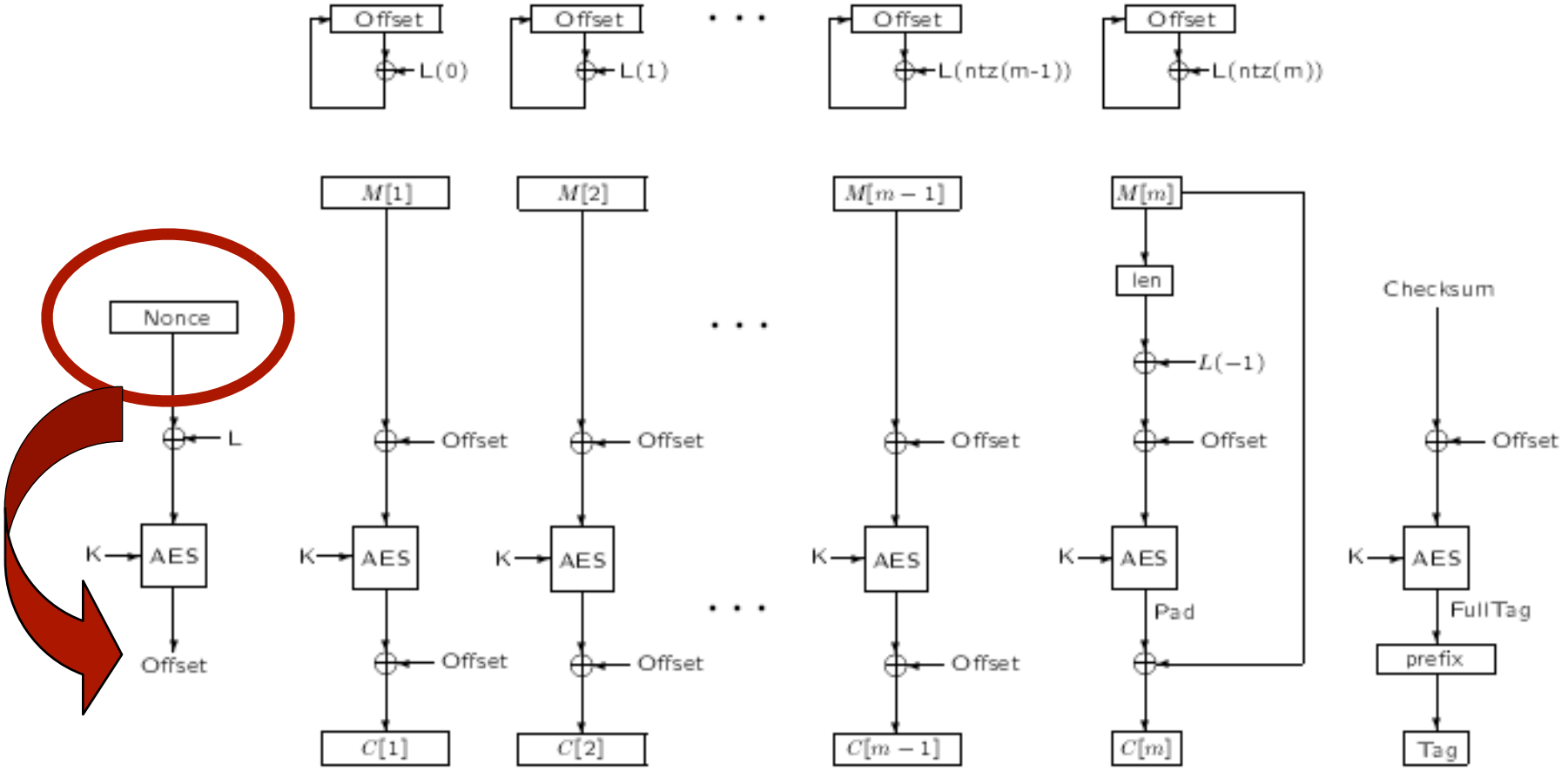
- Proposé par Rogaway, Bellare, Black et Krovetz en 2001
- Prouvé sûr (indistinguabilité contre les attaques à clairs choisis et impossibilité de forger avec une attaque à messages

- Forte inspiration du mode IAPM de Jutla

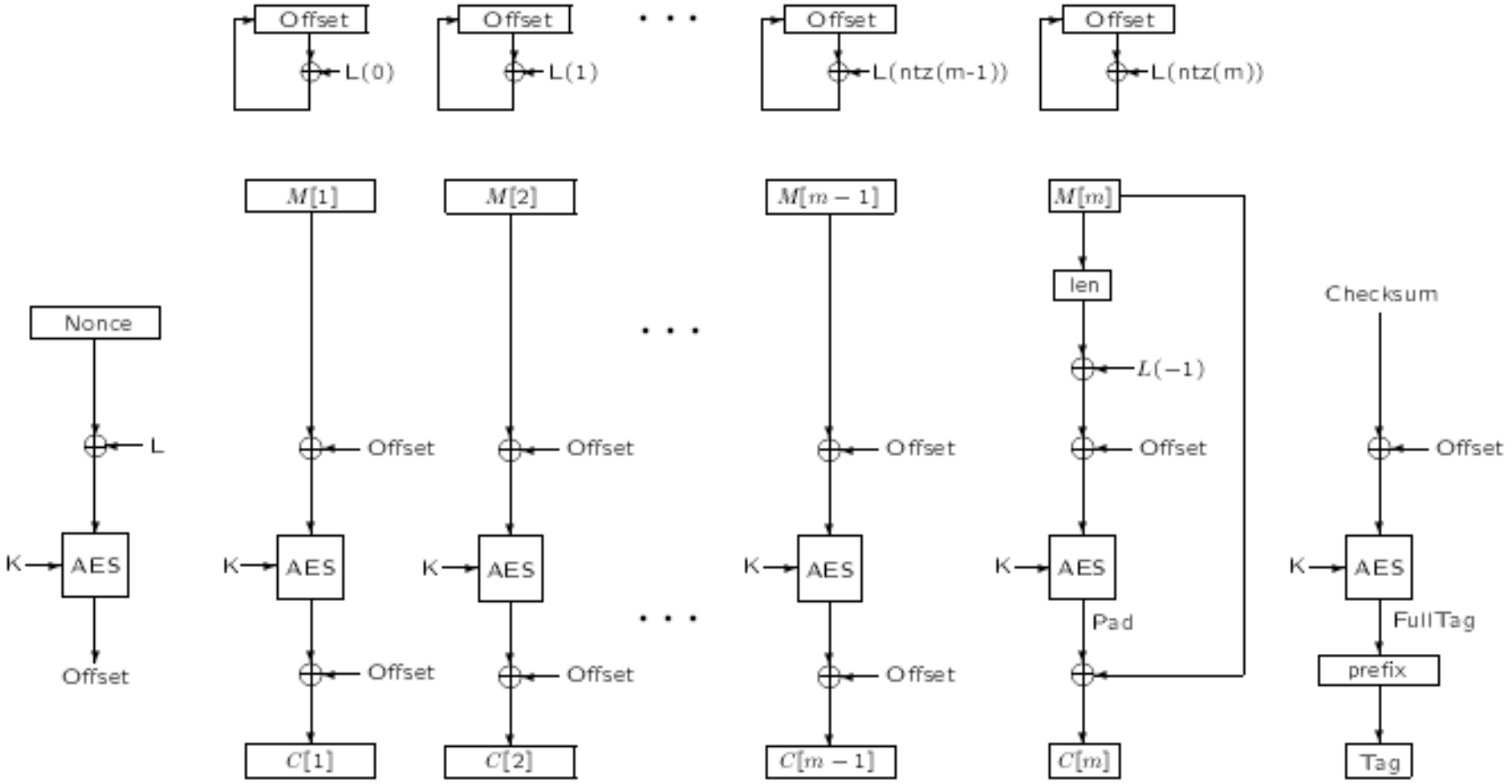
Autre exemple : OCB



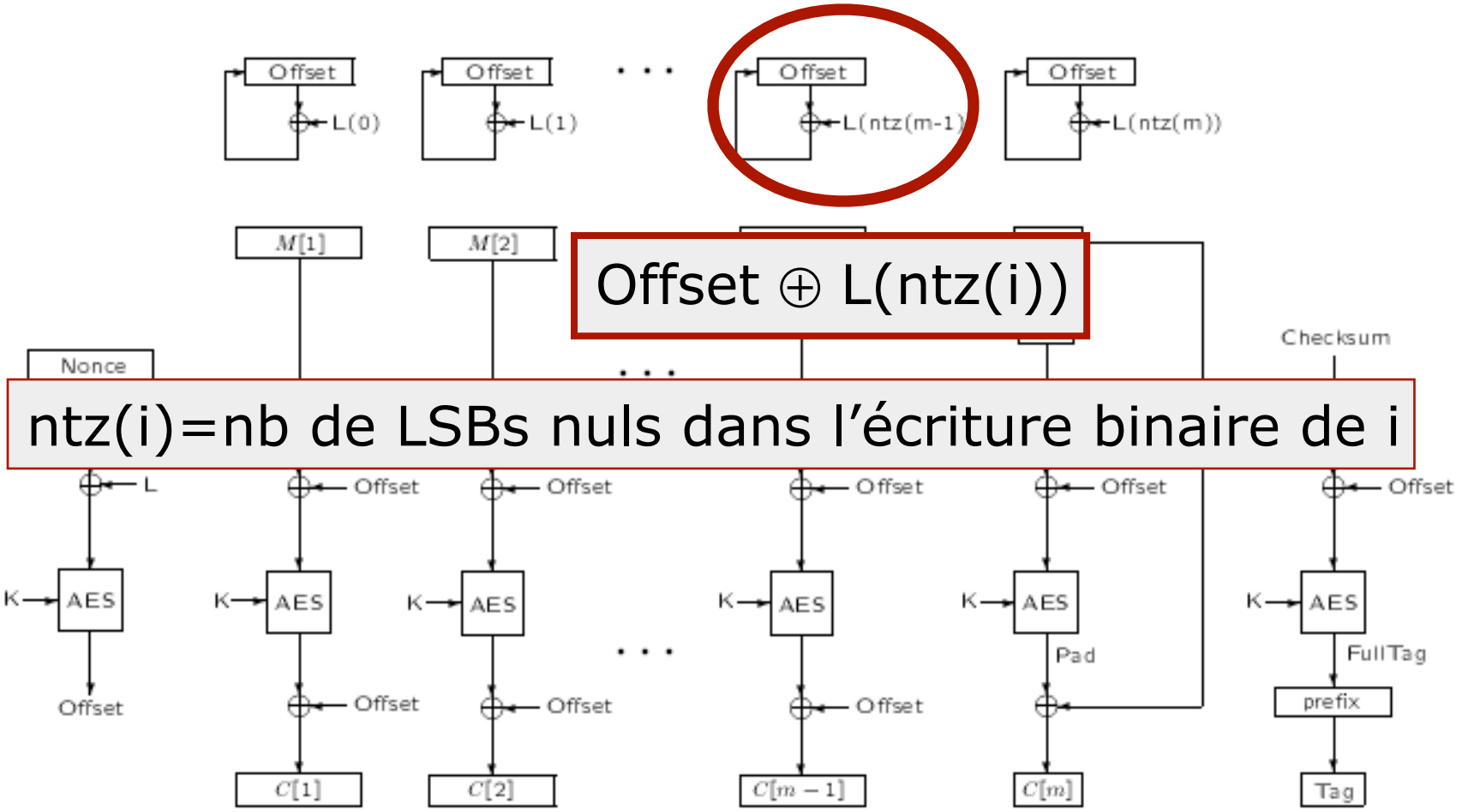
Autre exemple : OCB



Autre exemple : OCB



Autre exemple : OCB



Propriétés

- Une seule clé est utilisée
- Si message de m blocs, m+2 appels au block cipher
- Choix de la valeur initiale :
  - Ne doit pas se répéter
  - Peut être prédictible (compteur)