

# Introduction à la cryptographie

---

Louiza Khati

4A-Partie 1

# ANSSI

---

- Création juillet 2009
- Acteur majeur de la cyber sécurité en France
- Rôles :
  - Favorise le développement de la cyber sécurité en France
  - Apporte son expertise et son assistance aux administrations et aux industriels
  - Encadre et délivre des « visas de sécurité » (via CCN)
  - Forme les citoyens et entreprises (guides)
  - Etc.



# Laboratoire cryptographie

---

- Favorise la recherche dans ce domaine
- Echange sur les différents sujets auprès des acteurs internationaux
- Participe à la mise en place des bonnes pratiques crypto (guides)
- Apporte son expertise (certifications)

# Ce cours

---

- Introduction à la cryptographie
  - Cryptographie : domaine riche et complexe
- Objectifs :
  - Découvrir la cryptographie
  - Donner des intuitions
  - Connaître des exemples de constructions
  - Dépend de vous 😊
- Méthodes :
  - Cours + TD



# Ce cours

---

- Ne pas hésitez à poser des questions
  - Notions inconnues/floues
- Si c'est trop lent, trop rapide
- Répondre aux questions
  - Cours plus interactif → plus agréable!
  - Apprentissage plus rapide!
  - Trouver les réponses sur internet n'apporte rien.

# Règles à suivre

---

- Absences :
  - Récupérer le cours et les notes
  - Attention : questionnaires en début de cours généralement
- Retards :
  - Cours à 8h : 15 min de retard tolérées (qrcode à la pause)
  - Les autres cours : 5 minutes de retard tolérées
  - Si j'arrive après cet horaire, ne pas déranger le cours svp.
- Respect :
  - Pas de nourriture, ni de boissons en classe (trop bruyant)
  - Attention au bavardage !
  - Intervenant et camarades de classe.



# Notation

---

- Note CC (contrôle continu)
  - Participation en classe,
  - TD : je fais mon TD moi-même (je pourrais en discuter avec les camarades par la suite)
  - Questionnaire en ligne
    - Tester vos comptes woodlap et savoir utiliser l'application.
- Examen final
  - Examen papier sur l'ensemble du cours (tout le programme de cryptographie)

# Sondage

---



# Cours précédents

---

- Mécanismes symétriques
  - Primitives de chiffrement par bloc (AES, 3DES, Camellia, etc.)
  - Modes opératoires pour le chiffrement (ECB, CBC)
  - Chiffrement symétrique : mode + primitive de chiffrement par bloc
    - Ex : AES-128-CBC, Camellia-ECB, AES-256-CTR, etc...
- Dans ce cours : Intégrité symétrique
  - A base de primitive de chiffrement par bloc
  - A base de fonctions de hachage

# « Message Authentication Code » (MAC)

---

- Chiffrement symétrique « souvent » malléable
  - Vulnérables aux attaques CCA (chiffrés choisis)
  - Modifications contrôlées par l'adversaire dans le chiffré (CBC/CTR)
- Les MACs :
  - Assure qu'une donnée transmise n'a pas été modifiée sur le canal/ donnée stockée
  - Notion : **intégrité/authenticité des messages**  $\neq$  confidentialité
- Confidentialité et intégrité : besoins de sécurité complémentaires



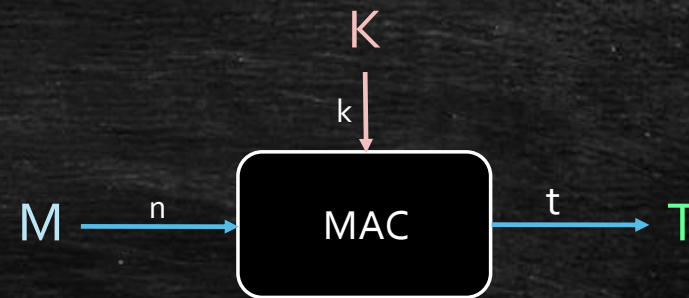
# MAC

---

- «Signature» utilisant de la cryptographie symétrique
  - Garantit **l'intégrité d'une donnée** (avec une clé secrète)
  - Seules les personnes d'un groupe peuvent vérifiées la validité d'un MAC
  - Pas de non-répudiation (car clé partagée dans un groupe)
  - Plus rapide qu'une signature (pratique pour les réseaux)
- Utilise une **clé** symétrique
- Peut-être construit avec un chiffrement par bloc ou une fonction de hachage

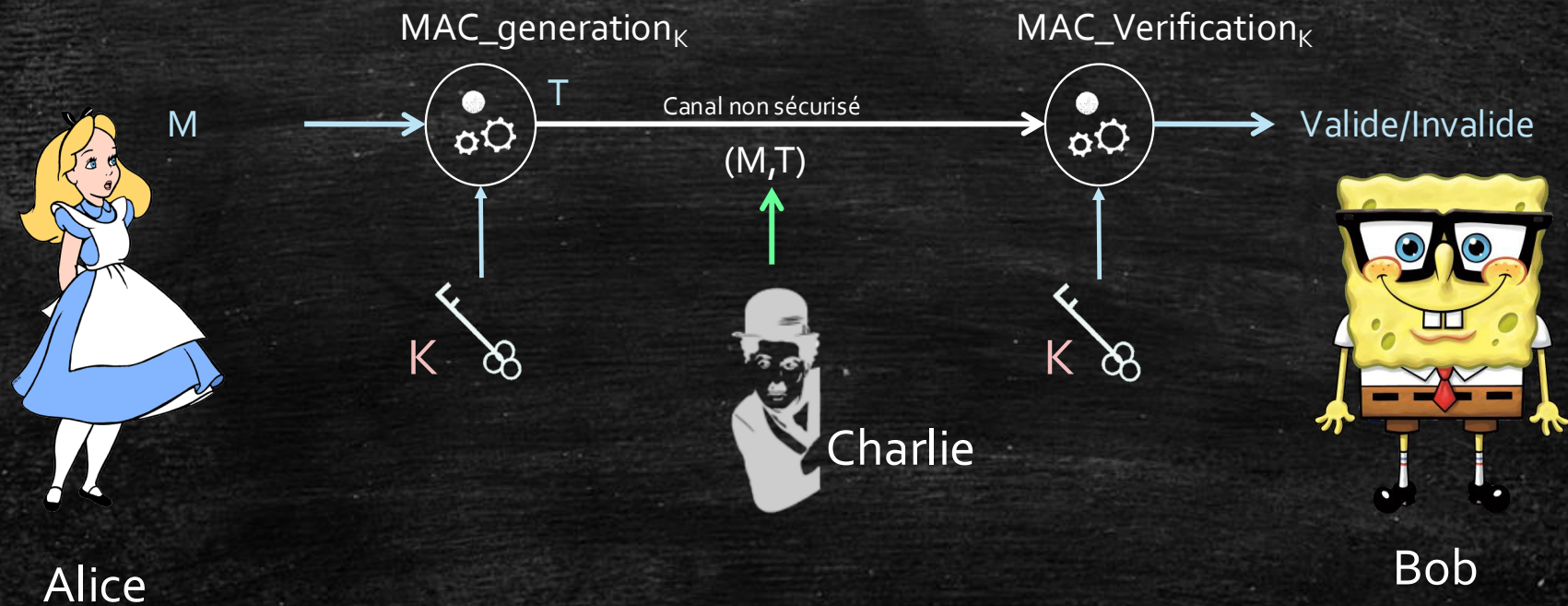
# MAC

- Clé  $K$  de taille  $k > 128$
- Message  $M$  de taille  $n$  quelconque
- Valeur  $T$  :
  - Appelé tag ou MAC
  - Taille constante  $t > 128$  (recommandée)
- MAC :
  - $K \leftarrow \text{keygen}(k)$ ,  $k$  taille de la clé
  - $T \leftarrow \text{MAC\_generation}(K, M)$  déterministe (en général)
  - Valide/Invalide  $\leftarrow \text{Mac\_verification}(K, M, T)$  déterministe





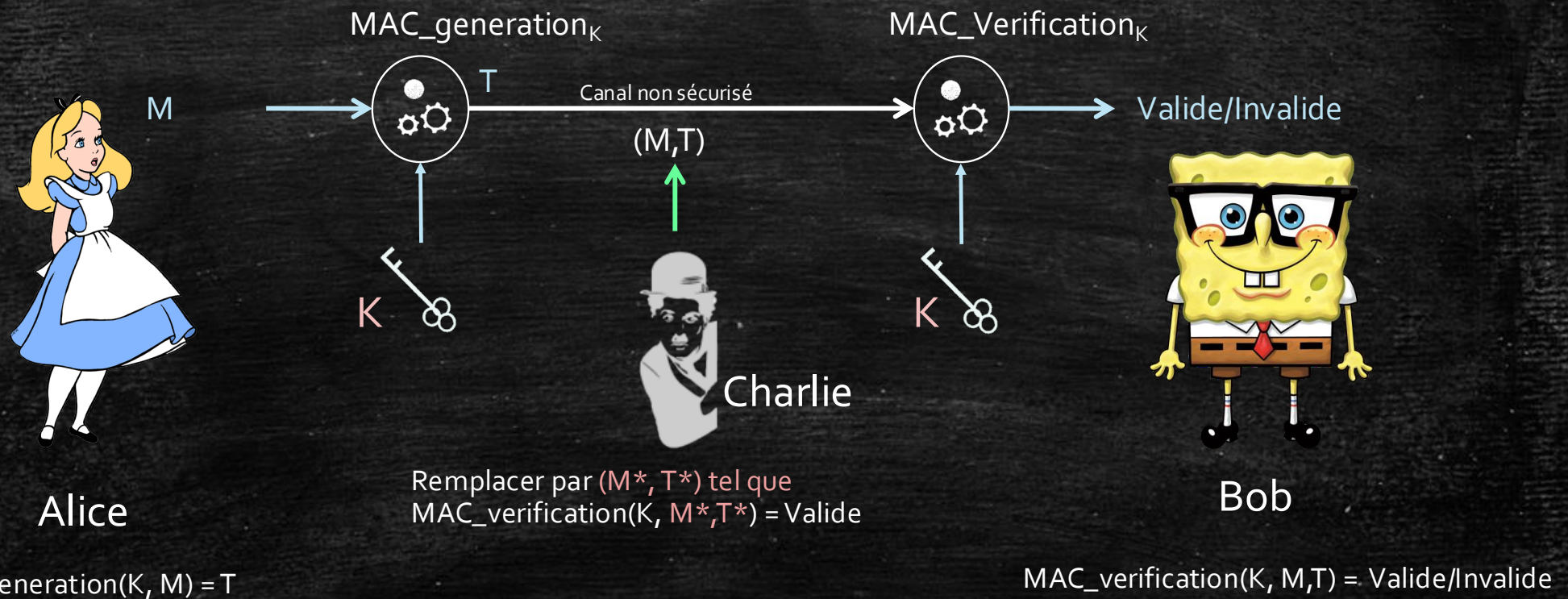
# MAC : authenticité de la donnée



$$\text{MAC\_generation}(K, M) = T$$

$$\text{MAC\_verification}(K, M, T) = \text{Valide/Invalide}$$

# MAC : authenticité de la donnée





## MAC : Modèles d'attaquants

---

- **À messages connus** : l'adversaire a accès à des couples  $(M, T)$  de messages déjà authentifiés (interception de MACs)
- **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

## MAC : Buts de l'adversaire

---

- Retrouver la clé



## MAC : Buts de l'adversaire

---

- Retrouver la clé
- Forger un MAC pour n'importe quel message

## MAC : Buts de l'adversaire

---

- Retrouver la clé
- Forger un MAC pour n'importe quel message
- Forger un MAC pour un message M choisi
- Forger un MAC pour un message M non choisi



## MAC : Buts de l'adversaire


---

- Retrouver la clé
- Forger un MAC pour n'importe quel message
- Forger un MAC pour un message M choisi
- Forger un MAC pour un message M non choisi
- Distinguer un MAC d'une sortie aléatoire


# MAC : Buts de l'adversaire

---

- Retrouver la clé
- Forger un MAC pour n'importe quel message
- Forger un MAC pour un message M choisi
- Forger un MAC pour un message M non choisi
- Distinguer un MAC d'une sortie aléatoire



Attaque de plus en plus simple

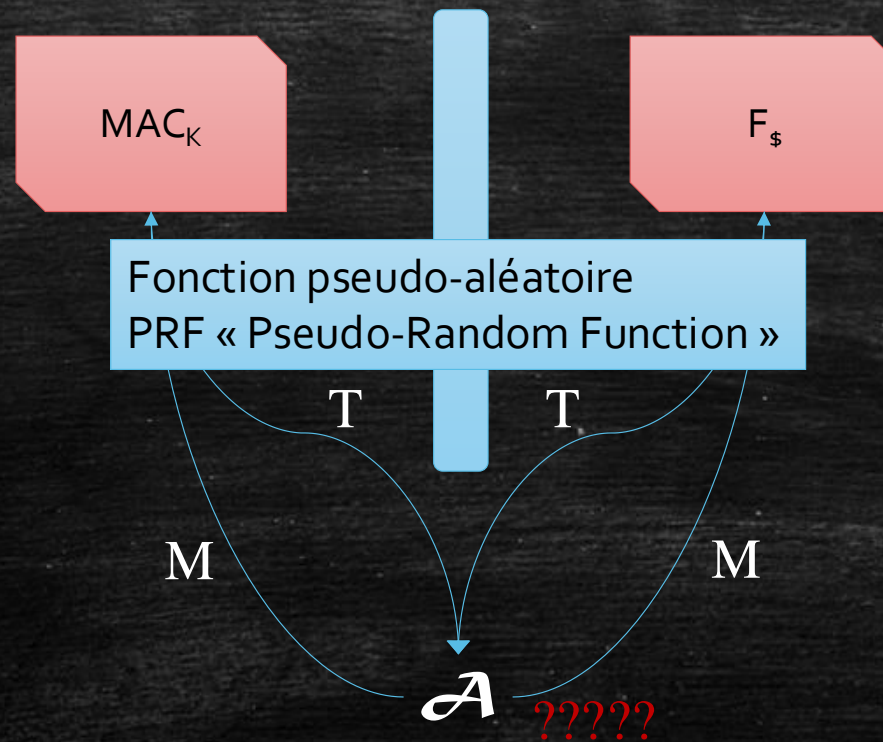


Attaquant de plus en plus fort!

Sécurité maximale



## (MAC : Sécurité)



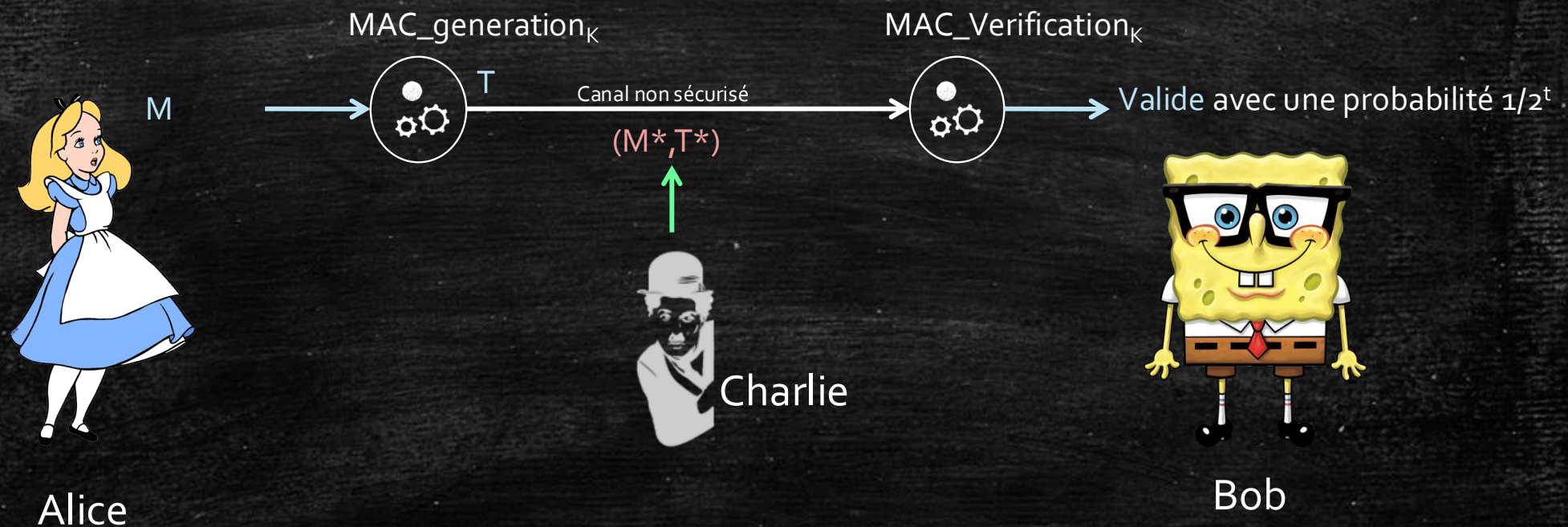
Notion d'indistinguabilité

Sans connaître K, A peut-il distinguer MAC<sub>K</sub> et F<sub>\$</sub> ?



# MAC : Sécurité

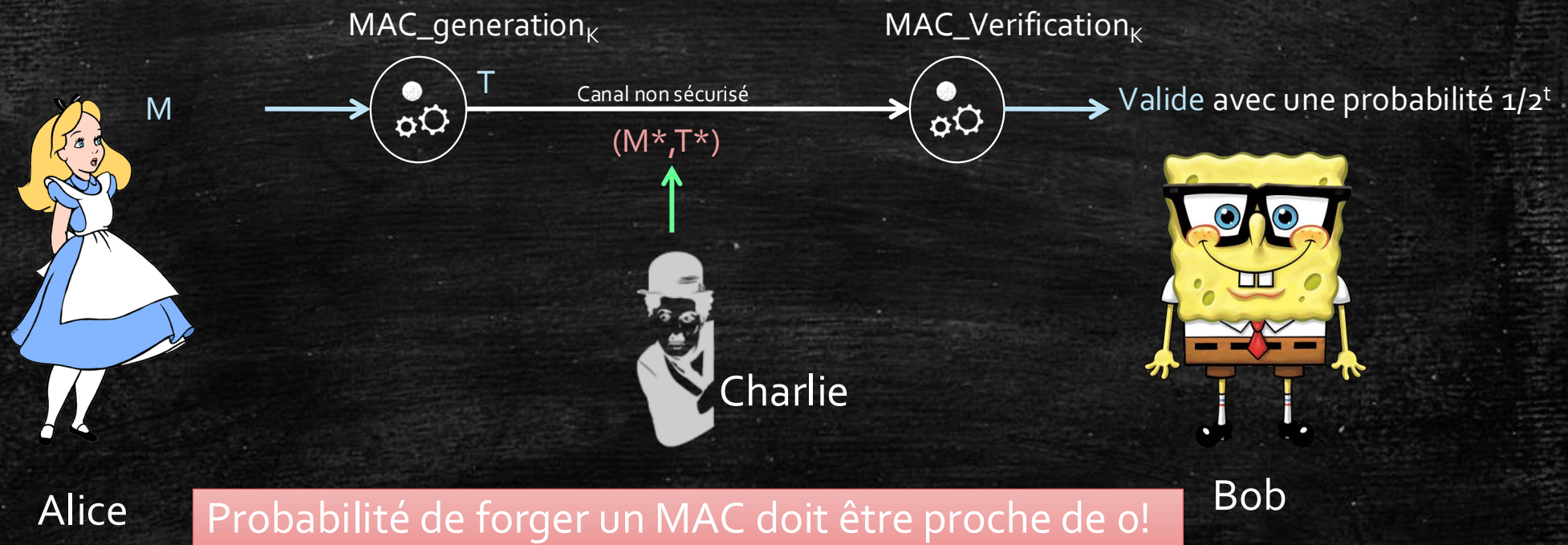
- Probabilité minimale qu'un adversaire produise une contrefaçon :  $1/2^t$





# MAC : Sécurité

- Probabilité minimale qu'un adversaire produise une contrefaçon :  $1/2^t$



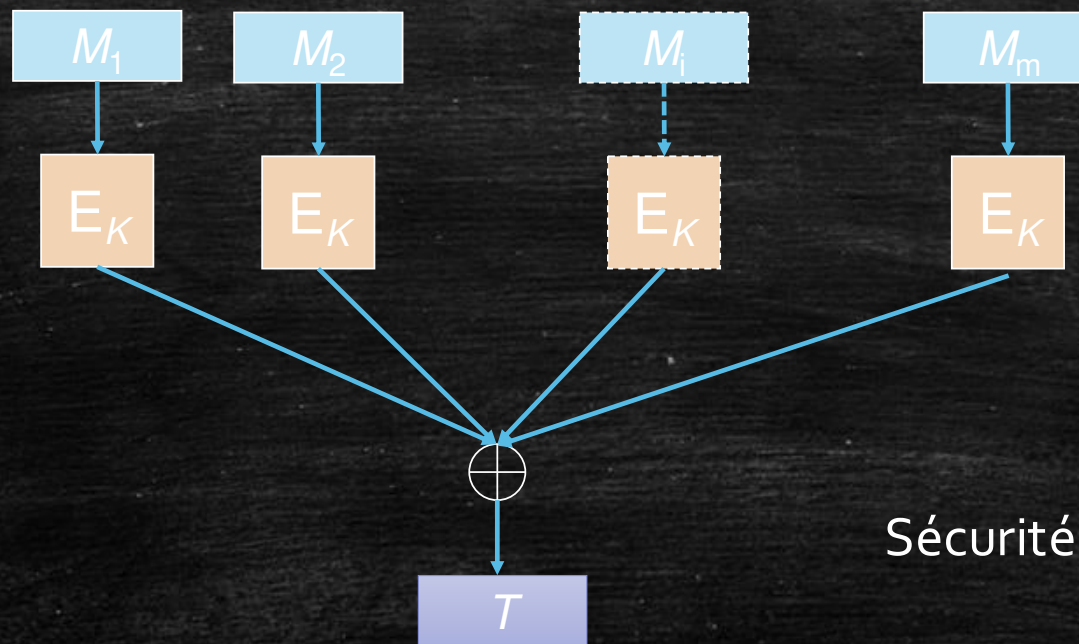
# MACs part 1

---

Basés sur une primitive de chiffrement par bloc

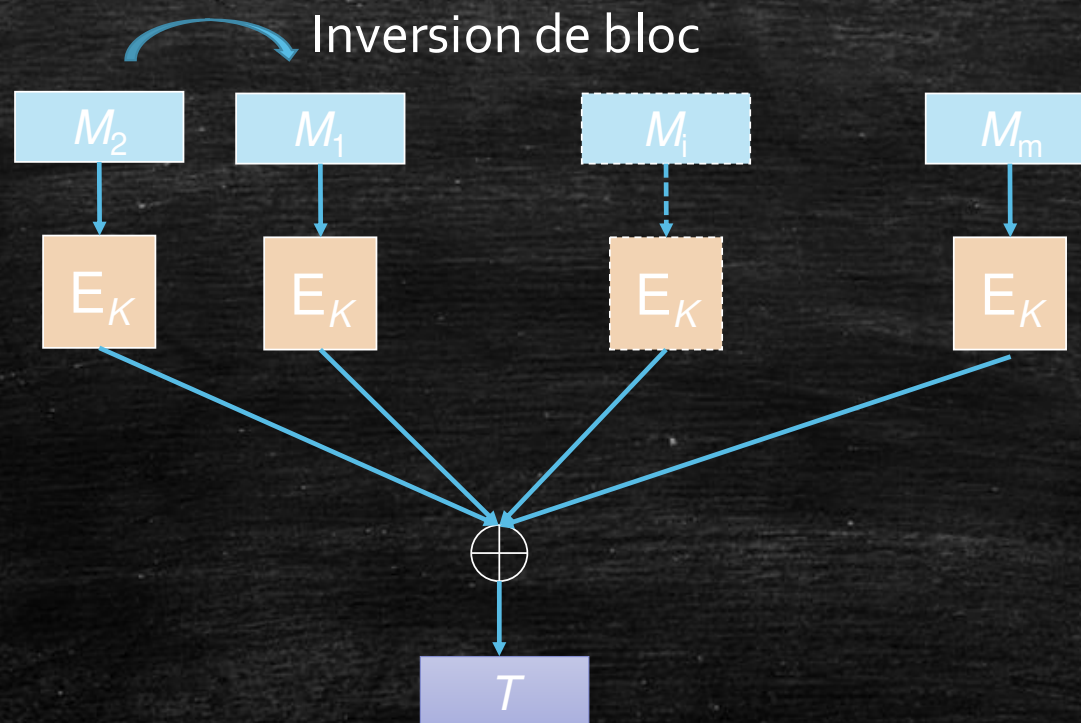


# Un MAC simple



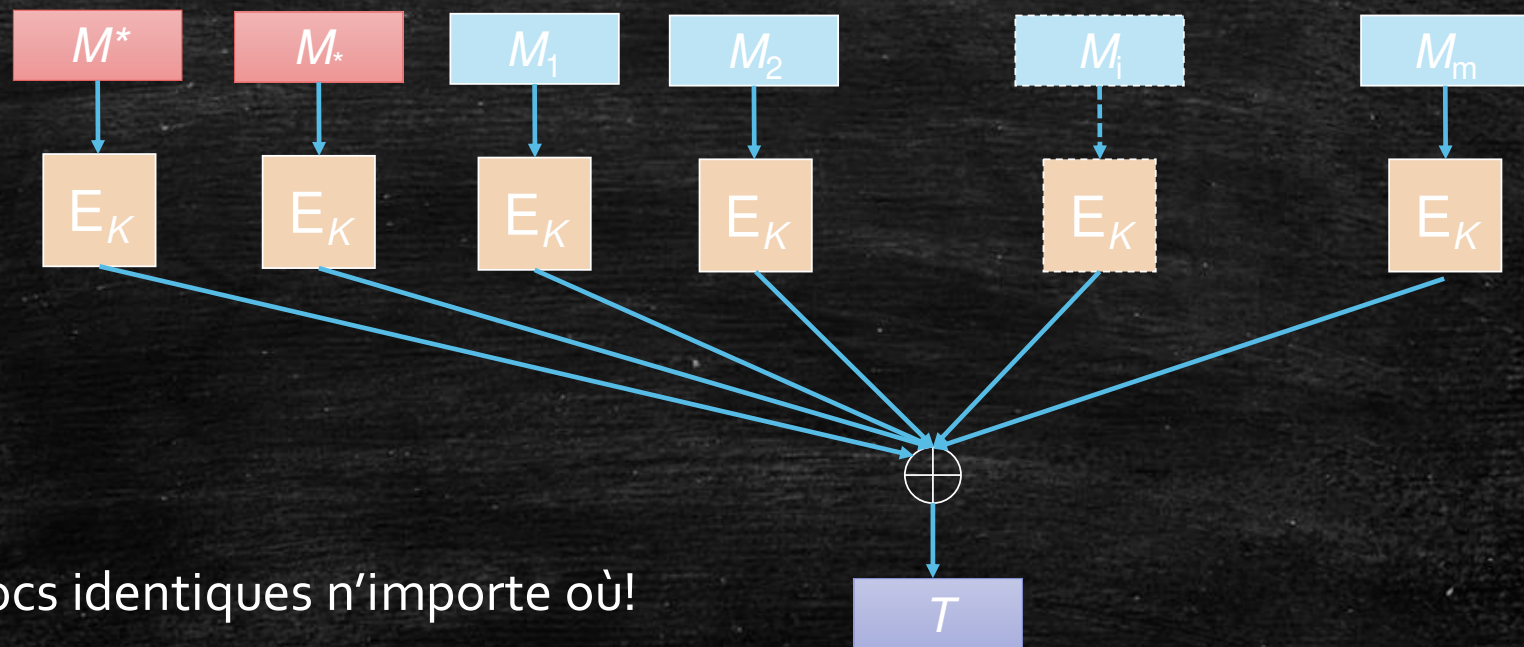
Sécurité de ce MAC ? 

# Un MAC simple



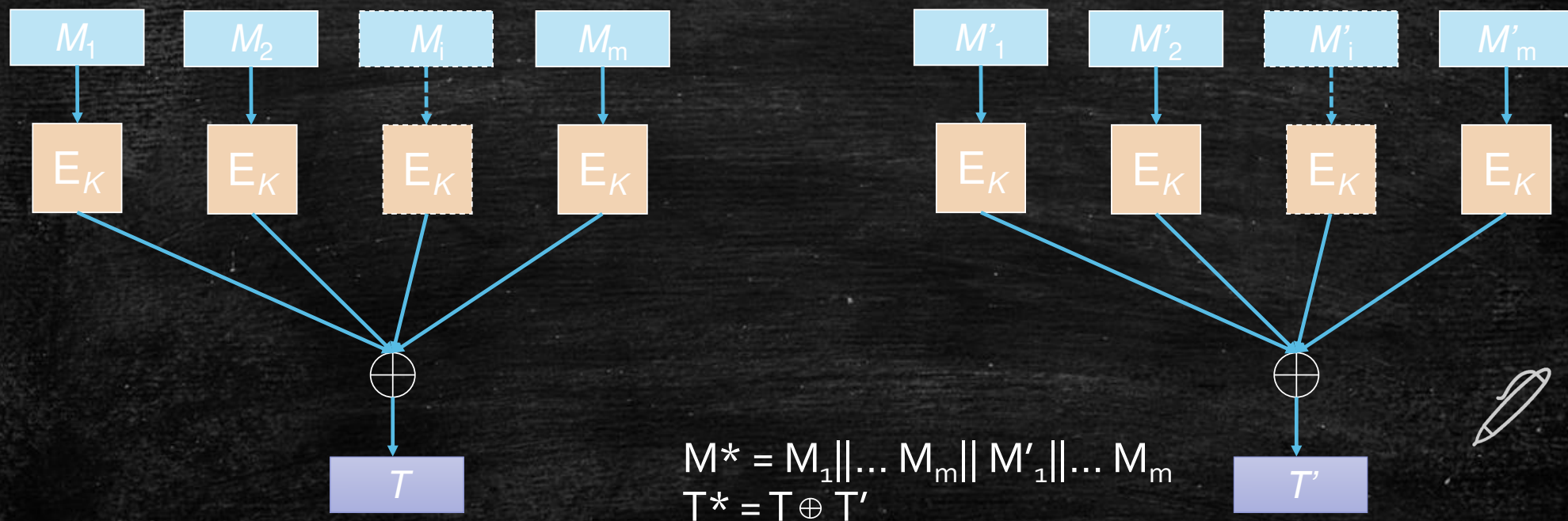


# Un MAC simple



Insertion de blocs identiques n'importe où!

## Un MAC simple





# Un MAC simple

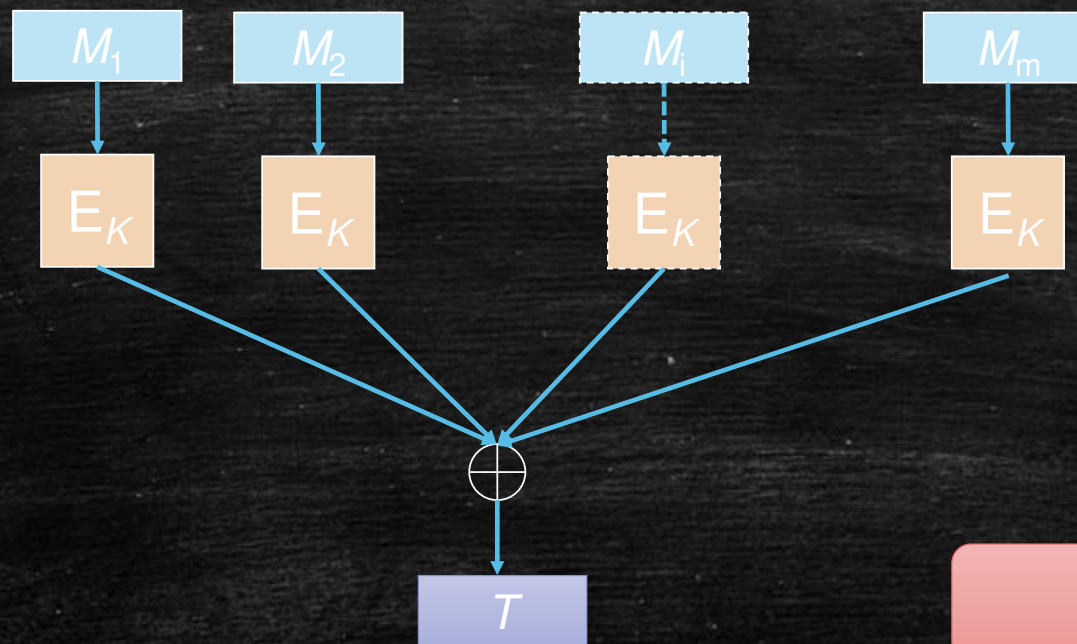


Schéma non sûr!!

# MAC : Exemples

---

- **CBC-MAC**

- Basé sur le chiffrement CBC sans IV

- **EMAC**

- CBC-MAC surchiffré
- Utilisations de deux clés : deux clés dérivées de la même clé maitresse
- Prouvé sûr pour des messages de tailles variables sous des hypothèses raisonnables

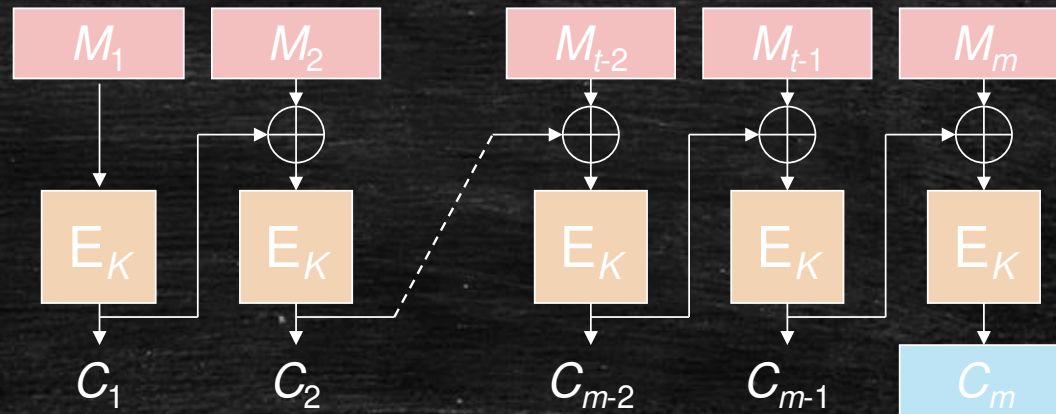
- **HMAC**

- Très utilisé !
- Utilise une **fonction de hachage**
- Prouvé sûr pour des messages de tailles variables sous des hypothèses raisonnables



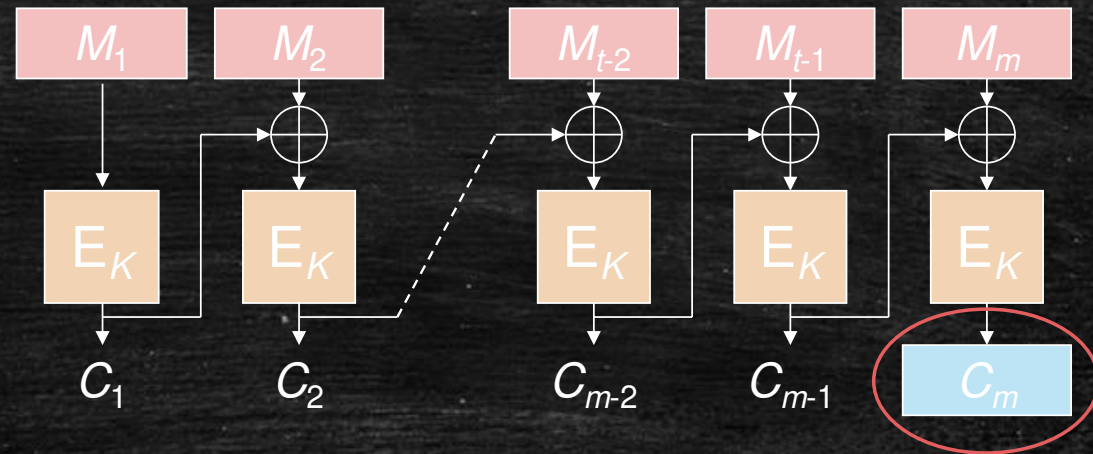
## MAC : CBC-MAC

- CBC
  - Pas d'IV
  - $T = C_m$
  - Valeurs  $C_i$  non publiques
    - $0 < i < m$



## MAC : CBC-MAC

- CBC
  - Pas d'IV
  - $T = C_m$
  - Valeurs  $C_i$  non publiques
    - $0 < i < m$

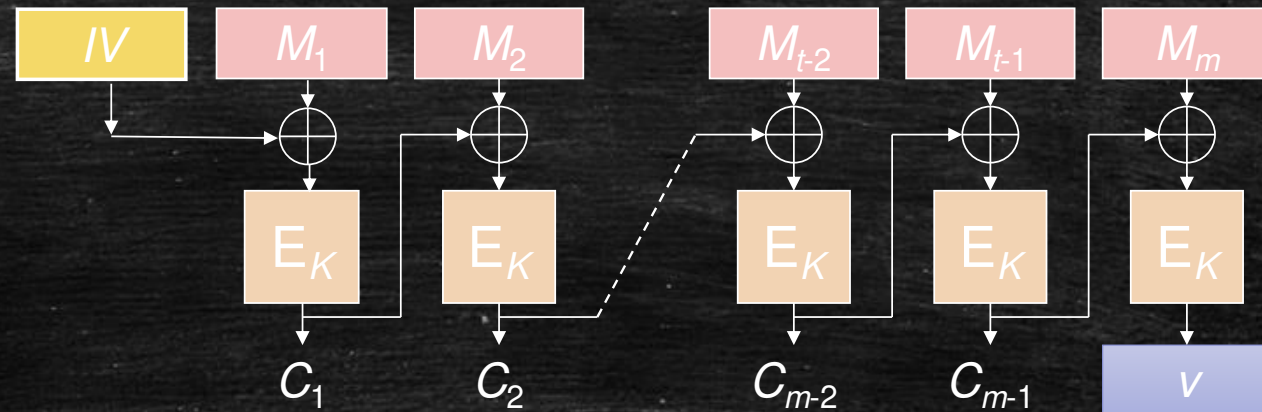


$T = C_m$  et dépend de tous les blocs



## CBC-MAC : IV

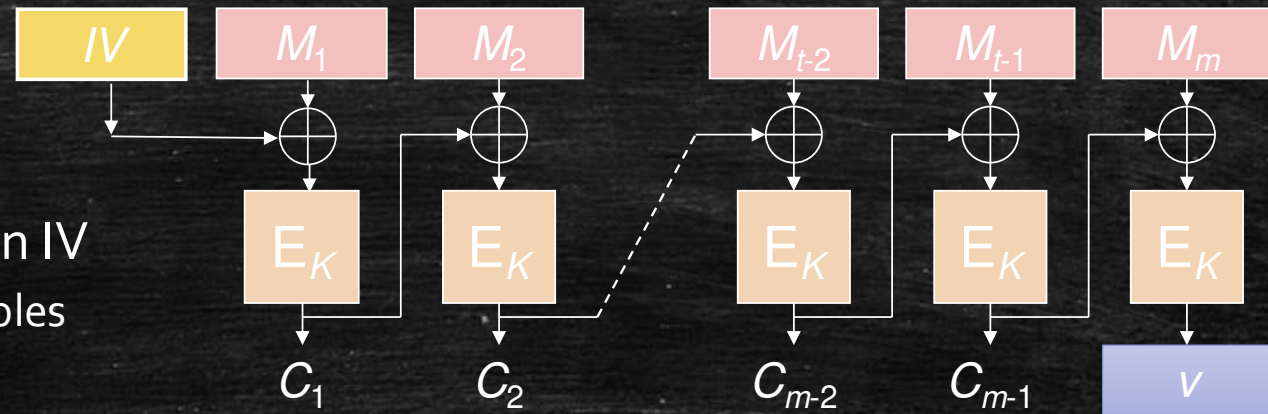
- Si présence d'un IV
  - $T = (IV, v)$



- Possibilité d'utiliser l'IV pour forger
  - Si l'attaquant possède  $(M, (IV, v))$  avec  $M = M_1$  (message composé d'un bloc)
  - Contrefaçon  $(M^*, T^*)$  tel que  $IV' = M_1 + IV + M'_1$ ,  $M^* = M_1$ ,  $T^* = (IV', v)$

## CBC-MAC : Pas d'IV

- Si présence d'un IV
  - Attaques possibles



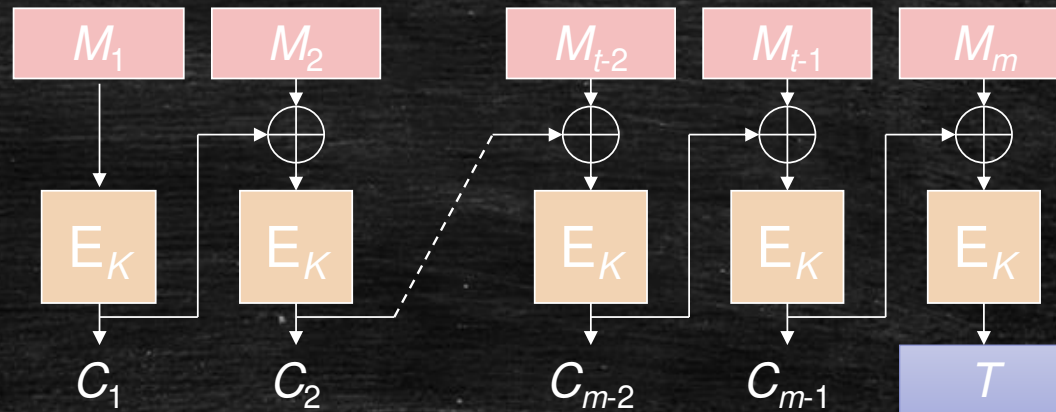
- Pas d'IV dans le cadre de CBC-MAC!



## MAC : CBC-MAC

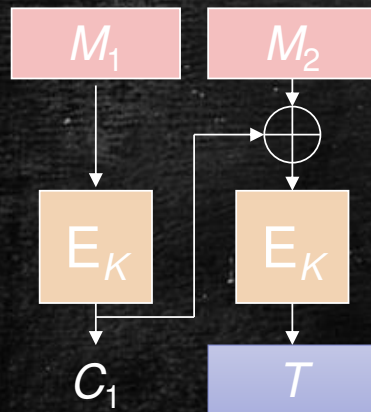
- CBC
  - Pas d'IV
  - $T = C_m$
  - Valeurs  $C_i$  non publiques
    - $0 < i < m$

Sécurité ?

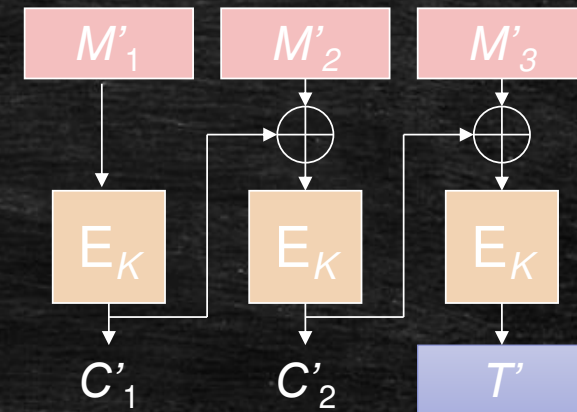


## MAC : CBC-MAC

Message  $M = M_1 \parallel M_2$   
Tag  $T$



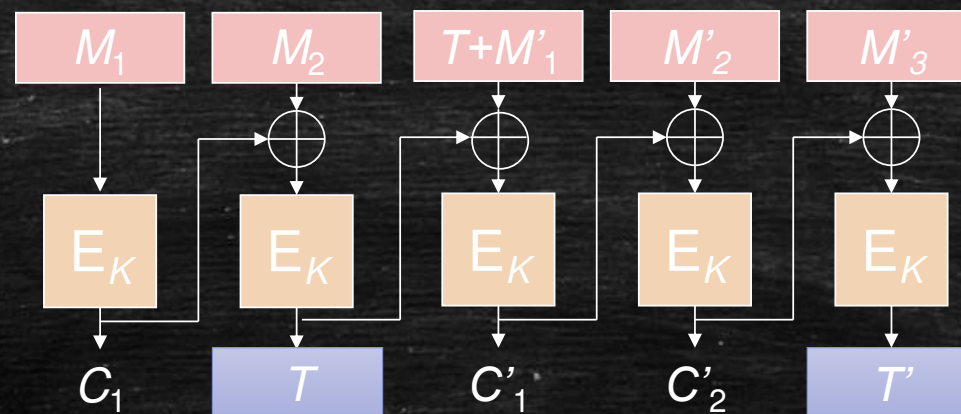
Message  $M' = M'_1 \parallel M'_2 \parallel M'_3$   
Tag  $T'$



Message  $M^* = M_1 \parallel M_2 \parallel T \oplus M'_1 \parallel M'_2 \parallel M'_3$   
Tag  $T^* = T'$



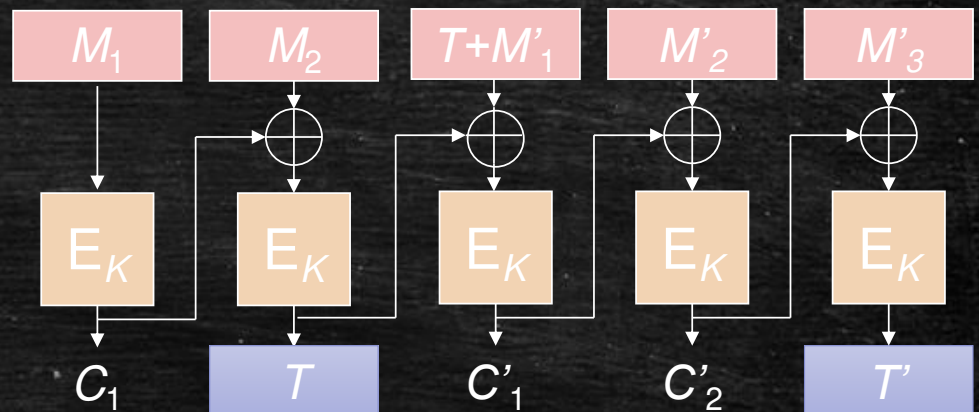
## MAC : CBC-MAC



Message  $M^* = M_1 \parallel M_2 \parallel T+M'_1 \parallel M'_2 \parallel M'_3$   
Tag  $T^* = T'$

## MAC : CBC-MAC

- Pas d'IV
- CBC-MAC sûr pour des messages de taille fixe seulement

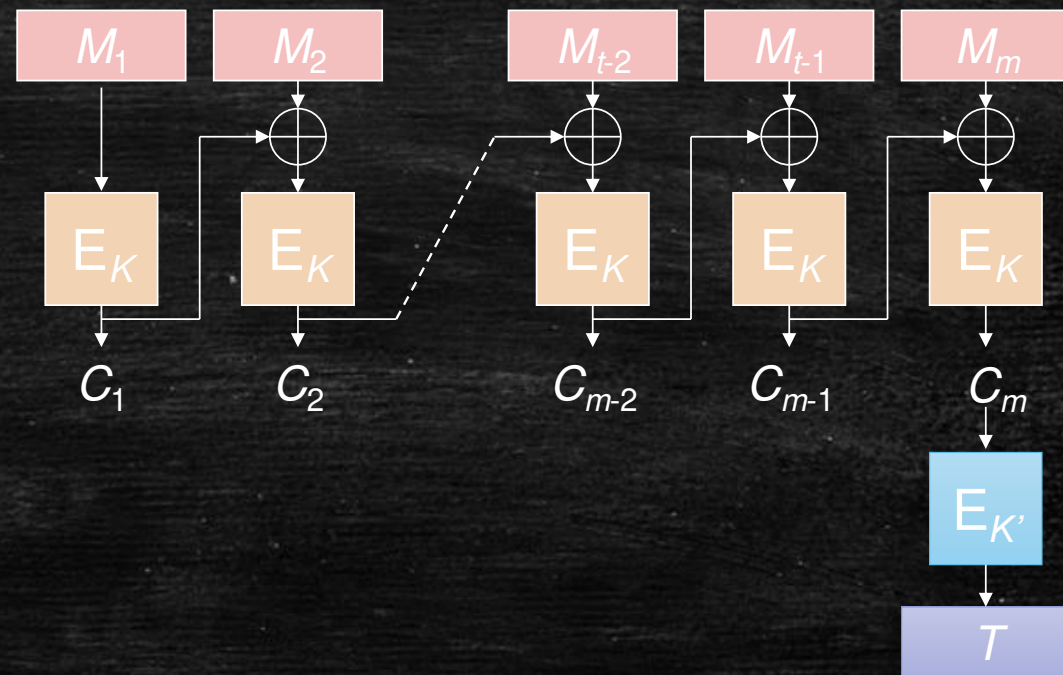


Message  $M^* = M_1 || M_2 || T+M'_1 || M'_2 || M'_3$   
Tag  $T^* = T'$



## MAC : EMAC

- CBC-MAC surchiffré sûr pour des messages de taille variable.
  - Sécurité prouvée :  $q \ll 2^{n/2}$
  - Attaque pour  $q = 2^{n/2}$



# Fonctions de hachage

---



# Fonction de hachage

---

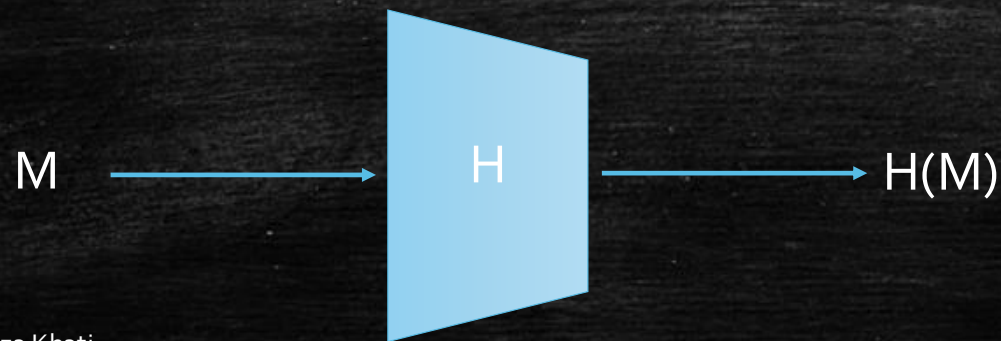
- Une fonction de hachage  $H$  est une fonction :

$$H : \{0,1\}^* \rightarrow \{0,1\}^n$$

$$M \rightarrow H(M)$$

$M$  est un message de taille quelconque

$H(M)$  est appelé « hash », « haché » ou encore « empreinte » **de taille  $n$** .



# Fonction de hachage : Propriétés

---

- Pas de clé
- Pas d'algorithme inverse
- Rapidité du traitement des données
- Répartition des images sur l'ensemble de sortie
- Compression des données





# Fonction de hachage

---

- Une fonction de hachage cryptographique est une fonction de hachage qui compressse de **manière sécurisée** une entrée de longueur arbitraire et une sortie de taille fixe.



# Fonction de hachage : Utilisations

---

- MACs (dans la suite)
- Stockage de mots de passe
  - Quelle propriété intéressante des fonctions de hachage ?
- Générateur d'aléa
- Signature (dans la suite)
- Dérivation de clé
  - Quelle propriété intéressante des fonctions de hachage ?



# Fonctions de hachage

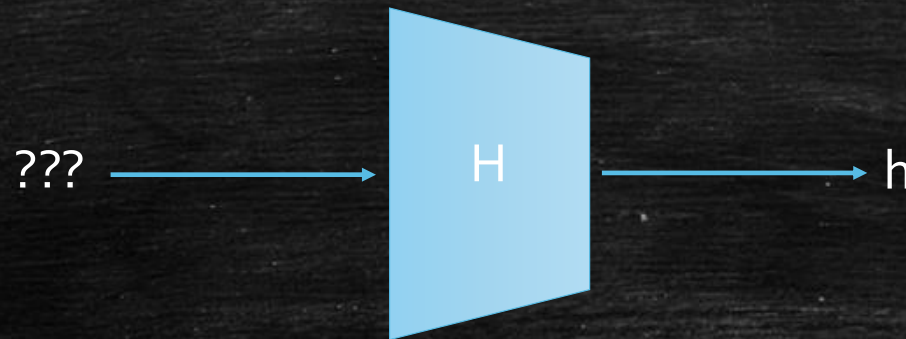
---

- Se comporte comme un « oracle aléatoire »
  - «Comme une fonction aléatoire »
- Pas de clé utilisée --> la fonction est totalement publique
  - Garantir l'intégrité d'une donnée avec une fonction de hachage ? → Pas seule!
- **Propriétés spécifiques aux fonctions de hachage :**
  - Résistance en collisions
  - Résistance en préimages
  - Résistance en seconde préimage

# Propriété de sécurité : préimage

---

- Etant donné  $h \in \{0,1\}^n$ , trouver  $M$  tel que  $H(M) = h$

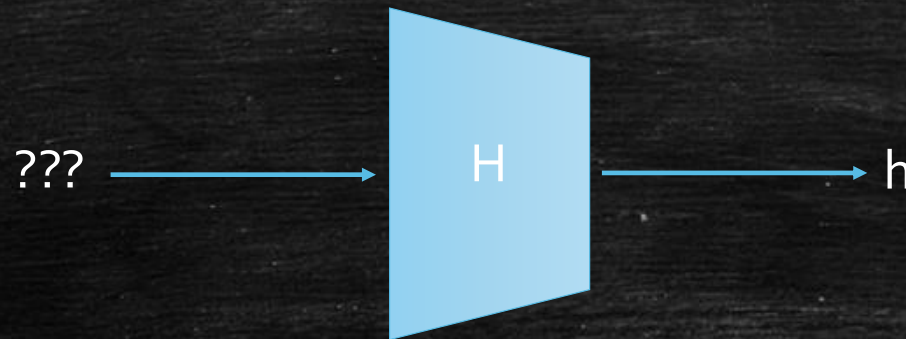




# Propriété de sécurité : préimage

---

- Etant donné  $h \in \{0,1\}^n$ , trouver  $M$  tel que  $H(M) = h$



- Complexité attaque générique : de l'ordre de  $2^n$

# Propriété de sécurité : préimage

---

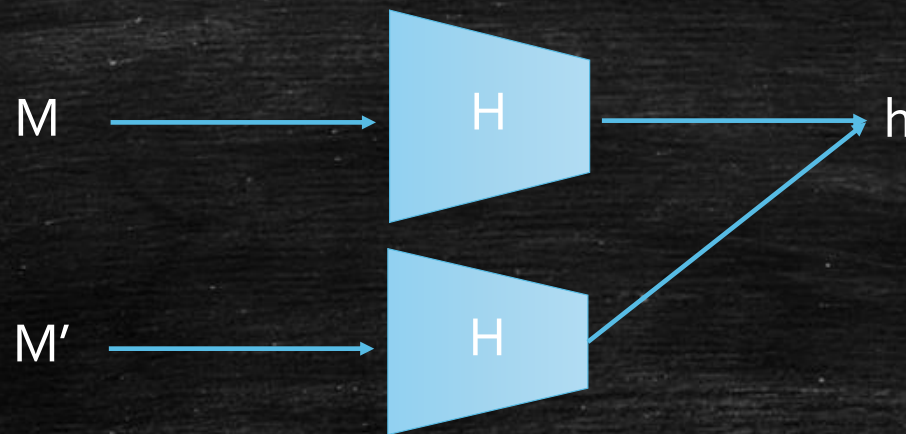
- Entrée  $h$
- Meilleure attaque générique : recherche exhaustive
- Probabilité de trouver une préimage :  $1/2^n$
- Calculer  $H(M)$  pour des messages aléatoires
- Après  $2^n$  messages on s'attend à trouver  $M$  tel que  $H(M) = h$
- Complexité recherche probabiliste :  $O(2^n)$



## Propriété de sécurité : seconde préimage

---

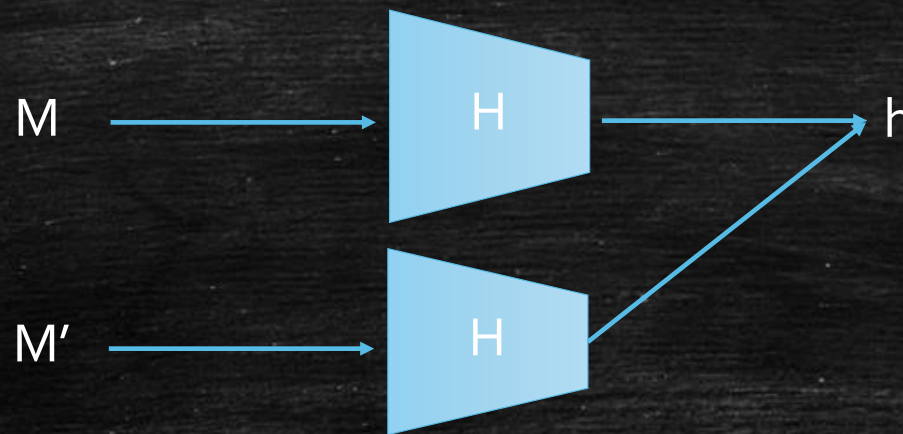
- Etant donné  $M \in \{0,1\}^*$ , trouver  $M' \neq M$  tel que  $H(M) = H(M')$



# Propriété de sécurité : seconde préimage

---

- Etant donné  $M \in \{0,1\}^*$ , trouver  $M' \neq M$  tel que  $H(M) = H(M')$

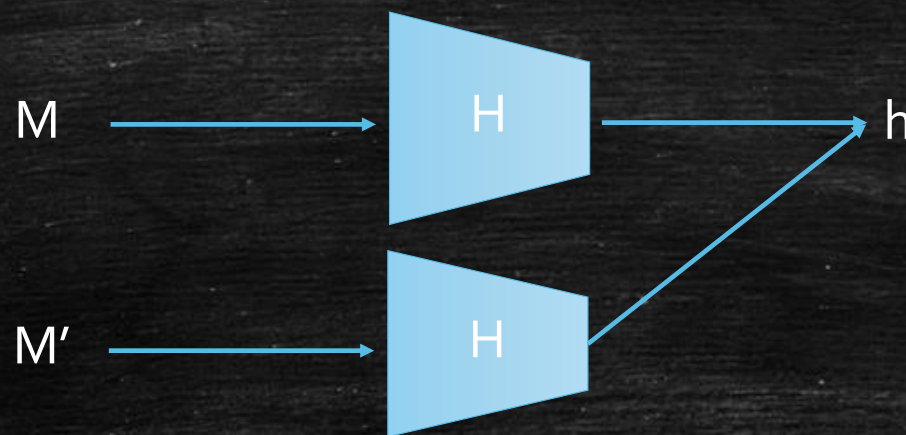


- Complexité attaque générique : de l'ordre de  $2^n$



# Propriété de sécurité : collisions

- Trouver  $M' \neq M$  tel que  $H(M) = H(M')$



- Complexité attaque générique : de l'ordre de  $2^{n/2}$ 
  - **Paradoxe des anniversaires**

# Fonction de hachage : Sécurité

---

- Cryptanalyse :
  - Trouver une attaque plus efficace qu'une attaque générique
  - $2^{n/2}$  calculs de hachés pour les collisions
  - $2^n$  calculs pour les (secondes) préimages
- En pratique : la résistance aux collisions est la plus difficile à obtenir
- La **taille de la sortie** est déterminante!
  - Recommandation guide crypto ANSSI :  **$n \geq 200$**