Notes de cours 6
Cryptographie symétrique -MAC

Dans le cas d'un chiffrement symétrique, si un expéditeur envoie un message à un destinataire, ils possèdent une clé secrète partagée k, et un attaquant qui voit le chiffré a une probabilité négligeable de trouver le message. La question qu'on se pose maintenant est comment s'assurer que le chiffré est bien envoyé par l'expéditeur ?
C'est le principe des MAC *Message Authentification Code*, qui calculent une étiquette t associée à un message m. C'est un schéma de chiffrement symétrique : les deux entités partagent une clé secrète en amont. Cette étiquette est infalsifiable.

1 Message Authentification Code

1.1 Définition

Definition 1. *Un code d'authentification de message (MAC) est un triplet d'algorithmes polynomiaux* ***(Gen, Mac, Vrfy)*** *sur trois ensembles :*
$\mathcal{K}$ *espace des clés,* $\mathcal{M}$ *espace des messages et* $\mathcal{T}$ *espace des étiquettes (tag),*
— ***Gen*** *prend en entrée le paramètre de sécurité n et renvoie une clé* $k \in \mathcal{K}$
— ***Mac*** $\mathcal{K} \times \mathcal{M} \to \mathcal{T}$ *: prend en entrée k, un message* $m \in \mathcal{M}$ *et renvoie une étiquette t*
— ***Vrfy*** $\mathcal{K} \times \mathcal{M} \times \mathcal{T} \to 0$ *ou* 1 *: prend en entrée* k, m *et* t *et renvoie* 1 *si l'étiquette est valide, et* 0 *sinon, tels que pour tout* $k \in \mathcal{K}$*, pour tout* $m \in \mathcal{M}$ *:* $Vrfy(k, m, Mac(k, m)) = 1$.

Remarques :
- Un MAC est à longueur fixée $\ell(n)$ s'il n'accepte que des messages de longueur $\ell(n)$.
- Le MAC traite a priori des messages non chiffrés
- Gen est probabiliste, Mac peut l'être, et Vrfy ne l'est pas
- Si Mac est déterministe, $\text{Vrfy}_k(m, t)$ est simplement le test "$\text{Mac}_k(m) = t$?"

1.2 Sécurité

Intuitivement, un MAC est sûr si un attaquant ne peut pas créer d'étiquette valide. La notion de sécurité est la notion euCMA pour *Existencial Unforgeabilité under chosen message attacks*.

Definition 2 (Sécurité euCMA)**.** *Soit l'expérience de sécurité suivante :*

$\mathcal{C}$		$\mathcal{A}$
Génère $k \leftarrow Gen(n)$		
	$\xleftarrow{m_i}$	*choisit* m_i
	$\xrightarrow{Mac_k(m_i)}$	
		produit un couple (m^*, t^*) *où* $m^* \neq m_i$
		Gagne si $Vrfy_k(m^*, t^*) = 1$.

Un MAC est existentiellement infalsifiable pour une attaque adaptative à clairs choisis *si pour tout attaquant efficace,* $\Pr[Vrfy_k(m, t) = 1] \leq negl(n)$.

2 Construction d'un MAC

2.1 MAC de taille fixe

On peut facilement construire un MAC en utilisant une fonction pseudo aléatoire (PRF) :
— Utiliser une fonction pseudo-aléatoire (PRF) $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$
— Mac : sur l'entrée k, m, renvoie $F_k(m)$

— Vrfy : sur l'entrée k, m, t, teste si $t = F_k(m)$

Theorem 1. *Si F est pseudo-aléatoire, alors $(Mac, Vrfy)$ est un MAC sûr.*

Cela permet de construire un MAC de longueur fixée, qu'en est-il d'un MAC de longueur quelconque ?

Indication : dans les deux exercices qui suivent, on cherche à prouver que des schémas ne sont pas sûrs, pour cela il suffit de réussir une attaque contre la sécurité du schéma. Pour un MAC, réussir l'attaque signifie trouver un couple (m^*, t^*) où m^* est un message et t^* une étiquette valide. Le modèle de sécurité autorise des appels à un oracle : l'attaquant envoie un message m_i et reçoit $Mac_k(m_i)$. L'attaque réussit si pour tout i, $m^* \neq m_i$ et $Vrfy_k(m^*, t^*) = 1$.

Exercise 1.
Soit $F : \{0,1\}^t \times \{0,1\}^n \to \{0,1\}^n$ une fonction pseudo aléatoire sûre (PRF). Montrez que le MAC suivant ne l'est pas : La clé secrète partagée est un élément $k \in \{0,1\}^n$ aléatoire uniforme. Pour authentifier un message $m_1 \| m_2$ avec $|m_1| = |m_2| = n$, le MAC renvoie l'étiquette $(F(k, m_1), F(k, (F(k, m_2))))$.

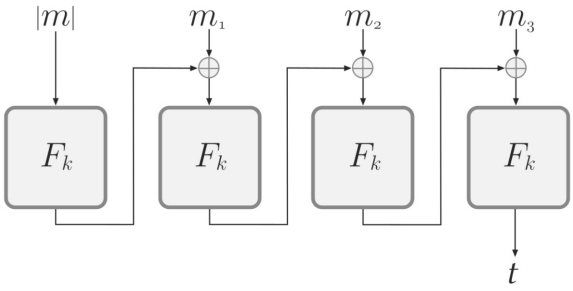
2.2 MAC de longueur quelconque

Une première idée (qui n'est pas bonne) serait de découper le message en plusieurs blocs, et d'appliquer la fonction MAC à chaque bloc indépendamment : pour $m = m_1 || \dots || m_s$, on aurait $t = t_1 || \dots || t_s = \text{Mac}_k(m_1) || \dots || \text{Mac}_k(m_s)$. Cela ne fonctionne pas et donne des problèmes similaires au mode ECB pour le chiffrement. Par exemple, si on a $(m_1^0 || m_2^0, t_1^0 || t_2^0)$ et $(m_1^1 || m_2^1, t_1^1 || t_2^1)$, on crée facilement $(m_1^0 || m_2^1, t_1^0 || t_2^1)$.

Il est donc important d'ajouter de l'information à chaque bloc avant de calculer t_i. Une solution est de calculer : $t_i \leftarrow \text{Mac}_k(r || \ell || i || m_i)$, en ajoutant les éléments suivants :
— numéro i du bloc $\to$ empêche de changer l'ordre,
— longueur ℓ du message $\to$ empêche de tronquer le message,
— identifiant unique r aléatoire $\to$ empêche les recombinaisons.
Cette solution peut-être montré sûre, mais est lourde à mettre en œuvre.

2.2.1 CBC MAC

Definition 3. *Soit F une PRF, on peut définir le mode CBC-MAC selon le schéma ci-dessous :*



Theorem 2. *Si F est pseudo-aléatoire, alors CBC-MAC est sûr pour des messages de longueur fixée.*

Exercise 2.
On rappelle CBC-MAC dans la figure suivante et on considère certaines modifications.

1. On modifie CBC-MAC pour qu'un vecteur d'initialisation IV aléatoire (au lieu de $IV = \mathbf{0}$) soit utilisé chaque fois qu'une étiquette est calculée (et le IV est renvoyé en même temps que le résultat t). Prouvez que cette modification de CBC-MAC ne permet pas d'obtenir un MAC sûr de longueur fixé.

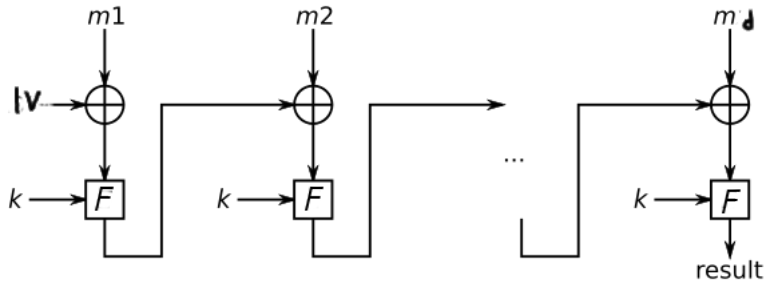


FIGURE 1 – CBC-MAC

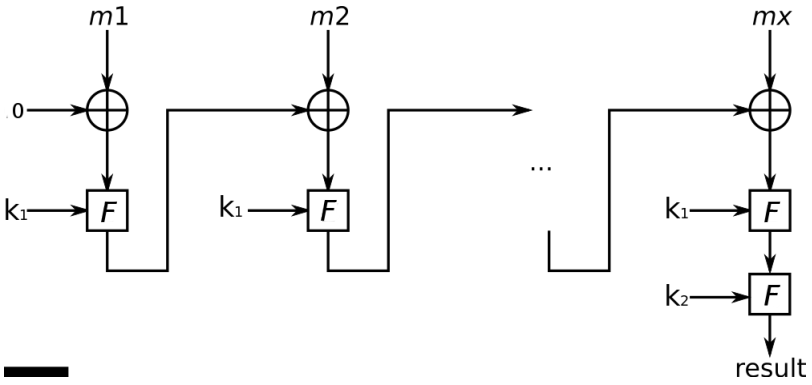


FIGURE 2 – ECBC-MAC

2. On modifie maintenant CBC-MAC pour que toutes les sorties de F soient retournées, au lieu de ne renvoyer que la dernière. Prouvez que cette modification de CBC-MAC ne permet pas d'obtenir un MAC sûr de longueur fixé.

Nous considérons maintenant le schéma ECBC-MAC suivant, soit $F : K \times X \to X$ une fonction pseudo aléatoire, on définie $F_{ECBC} : K^2 \times X^{\leq L} \to X$ comme dans la Figure 2 où k_1 et k_2 sont deux clés indépendantes.
Maintenant si la longueur du message n'est pas un multiple de n, on ajoute un *pad* au dernier bloc : $m = m_1|\ldots|m_{d-1}|(m_d\|\text{pad}(m))$.

1. Montrez qu'il existe un *padding* pour lequel ce schéma n'est pas sûr.

Pour la sécurité du schéma, le *padding* doit être invertible, et en particulier, pour tout message $m_0 \neq m_1$ on doit avoir $\text{pad}(m_0) \neq \text{pad}(m_1)$. La norme ISO est d'ajouter le *pad* $10\cdots0$, et si la longueur du message est un multiple de la longueur des blocs, on ajoute un nouveau bloc $10\cdots0$ de longueur n.

1. Expliquez pourquoi le schéma n'est pas sûr si le *padding* n'ajoute pas de nouveau bloc si la longueur du message est un multiple de la longueur des blocs.

CMAC est une variante de CBC-MAC avec trois clés (k, k_1, k_2). Si la longueur du message n'est pas un multiple de la longueur des blocs, alors on ajoute le *padding* ISO et on XOR ce dernier bloc avec la clé k_1. Si la longueur du message est un multiple de la longueur des blocs, alors on XOR le dernier bloc avec la clé k_2. Ensuite, on applique $F(k, .)$ pour obtenir l'étiquette.

2.2.2 Hash-and-MAC

Une autre solution est d'utiliser une fonction de hachage, pour passer d'un MAC de longueur fixée à un MAC de longueur quelconque. Cela donne la construction suivante :

Definition 4. *Soient (**Mac**, **Vrfy**) un MAC pour les messages de longueur $\ell(n)$ et (Gen, H) une fonction de hachage avec taille de sortie $\ell(n)$. On définit la construction suivante :*

— ***Gen'***(n) *: tire* $k \in \{0,1\}^n$ *uniformément et* $s \leftarrow Gen(n)$
— ***Mac'***$_{k,s}(m)$ *: renvoie* $Mac_k(H^s(m))$
— ***Vrfy'***$_{k,s}(m,t)$ *: renvoie* $Vrfy_k(H^s(m),t)$

Theorem 3. *Si (**Mac**, **Vrfy**) est sûr et (Gen,H) résiste aux collisions, alors (**Gen'**,**Mac'**, **Vrfy'**) est sûr pour les messages de longueur quelconque.*

3 Conclusion

3.1 Chiffrement authentifié

On sait maintenant chiffrer un message et l'authentifier, il y a plusieurs façons de combiner les deux.

Authentifier-puis-chiffrer.
A partir d'un message m, on construit c où $c = \mathrm{Enc}_{k_E}(m||t)$ et $t = \mathrm{Mac}_{k_M}(m)$.
Ce n'est pas sûr, à la réception : $m||t \leftarrow \mathrm{Dec}_{k_E}(c)$ puis $\mathrm{Vrfy}_{k_M}(m,t)$, le déchiffrement peut renvoyer une erreur ce qui permet de faire des attaques par bourrage.

Chiffrer-puis-authentifier (VPN)
A partir d'un message m, on construit (c,t) où $c = \mathrm{Enc}_{k_E}(m)$ et $t = \mathrm{Mac}_{k_M}(c)$.
Cette solution est sûre si le chiffrement et le MAC le sont.

3.2 Sécurité CCA

On peut définir un niveau de sécurité au dessus de la sécurité CPA qui s'appelle la sécurité CCA. L'idée est similaire, mais l'adversaire peut en plus faire des requêtes de déchiffrement.

Definition 5. *Soit les deux expériences* Exp_b *pour* $b \in \{0,1\}$ *:*

$\mathcal{C}$		$\mathcal{A}$
k uniforme		
		Choisit $m_{i,0}, m_{i,1}$,
	$\xleftarrow{m_{i,0},m_{i,1}}$	
calcule $c \leftarrow Enc(k, m_{i,b})$		
	$\xrightarrow{c}$	
	$\xleftarrow{c_i}$	$\forall i, c_i \neq c$
	$\xrightarrow{Dec(k,c_i)}$	
		renvoie b'

L'avantage de l'adversaire est défini par $Adv^{CPA}(\mathcal{A}) = |\Pr[\mathcal{A} \xrightarrow{Exp_0} 1] - \Pr[\mathcal{A} \xrightarrow{Exp_1} 1]|$.
Un schéma de chiffrement symétrique (***Enc, Dec***) *est sûr si pour tout attaquant* $\mathcal{A}$ *efficace, l'avantage de* $\mathcal{A}$ *est négligeable.*