HAI709I - Fondements cryptographiques pour la sécurité

Cours 6 - Securité CCA et chiffrement authentifié

Rocco Mora

13 Octobre, 2025

Université de Montpellier – Faculté des Sciences
M1 informatique, parcours Algo, IASD, Imagine

Attaques par texte chiffré choisi (CCA) et sécurité CCA

Avec la sécurité CPA, nous avons vu la confidentialité contre un adversaire passif (ce qui signifie que $\mathcal{A}$ observe passivement le texte chiffré transmis). Maintenant, $\mathcal{A}$ est actif et peut obtenir le déchiffrement de certains textes chiffrés de son choix !

Il s'agit d'une attaque par texte chiffré choisi [🇬🇧 Chosen-ciphertext attack (CCA)]

Scénario 1 : lors de la bataille de Midway (1942), les cryptanalystes américains ont intercepté un message chiffré provenant des Japonais. Ils ont pu le décoder en partie : l'armée japonaise prévoyait une attaque sur `AF`. Les forces américaines ont envoyé un faux message indiquant que les réserves d'eau étaient faibles. Les Japonais l'ont intercepté et ont envoyé un message à leur supérieur : "`AF` manque d'eau". Les États-Unis ont appris que Dec(`AF`) =Midway.

Attaques par texte chiffré choisi (CCA) et sécurité CCA

Avec la sécurité CPA, nous avons vu la confidentialité contre un adversaire passif (ce qui signifie que $\mathcal{A}$ observe passivement le texte chiffré transmis). Maintenant, $\mathcal{A}$ est actif et peut obtenir le déchiffrement de certains textes chiffrés de son choix !

Il s'agit d'une attaque par texte chiffré choisi [🇬🇧 Chosen-ciphertext attack (CCA)]

Scénario 2 : $\mathcal{A}$ se fait passer pour un client et envoie des messages chiffré à un serveur, qui déchiffre ensuite les textes chiffrés. $\mathcal{A}$ en tire des informations, par exemple si un texte chiffré est déchiffré en un texte en clair mal formé en cas d'erreur.

Attaques par texte chiffré choisi (CCA) et sécurité CCA

Avec la sécurité CPA, nous avons vu la confidentialité contre un adversaire passif (ce qui signifie que $\mathcal{A}$ observe passivement le texte chiffré transmis). Maintenant, $\mathcal{A}$ est actif et peut obtenir le déchiffrement de certains textes chiffrés de son choix !

Il s'agit d'une attaque par texte chiffré choisi [🇬🇧 Chosen-ciphertext attack (CCA)]

Scénario 2 : $\mathcal{A}$ se fait passer pour un client et envoie des messages chiffré à un serveur, qui déchiffre ensuite les textes chiffrés. $\mathcal{A}$ en tire des informations, par exemple si un texte chiffré est déchiffré en un texte en clair mal formé en cas d'erreur.

$\rightarrow$ attaques menées dans la pratique sur des serveurs web lors de sessions TLS.

L'expérience de CCA-securité $\texttt{PrivK}^{cca}_{\mathcal{A},\Pi}(n)$

L'oracle Dec_k est absent dans l'expérience CPA.

1. Une clé k est générée en exécutant $\text{Gen}(1^n)$.
2. $\mathcal{A}$ reçoit en entrée 1^n et l'accès à l'oracle à Enc_k et Dec_k et produit une paire de messages m_0, m_1 avec $|m_0| = |m_1|$.
3. Un bit uniforme $b \in \{0,1\}$ est choisi. Le chiffré $c \leftarrow \text{Enc}_k(m_b)$ est donné à $\mathcal{A}$.
4. $\mathcal{A}$ a toujours accès à l'oracle Enc_k et Dec_k (mais ne peut pas interroger $\text{Dec}_k(c)$) et produit un bit b'.
5. Le résultat de l'expérience est 1 (c'est-à-dire que $\mathcal{A}$ réussit) si $b' = b$, et 0 dans le cas contraire.

CCA-sécurité [🇬🇧 Chosen ciphertext attack (CPA) security]

Un schéma de chiffrement à clé privée $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$ est CCA-sûr [🇬🇧 CCA-secure], si pour tout adversaire PPT $\mathcal{A}$, il existe une fonction négligeable $\texttt{negl}$ telle que, pour tout n,

$$\Pr(\texttt{PrivK}^{cca}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \texttt{negl}(n).$$

Exemple d'attaque par texte chiffré choisi

Aucune des constructions étudiées jusqu'à présent n'est CCA-sûr !

Exemple (construction CPA-sûre déjà vue) : F PRF, choisir $r \in \{0,1\}^n$ uniforme, chiffrer sous la forme $c := (r, s)$ avec $s = F_k(r) \oplus m$, déchiffrer sous la forme $m := F_k(r) \oplus s$.

1. $\mathcal{A}$ choisit m_0, m_1 et reçoit $c = (r, s)$.
2. $\mathcal{A}$ interroge Dec_k sur $(r, s \oplus s') \neq c$ pour une chaîne $s' \in \{0,1\}^n$ non nulle.
3. $\mathcal{A}$ produit $b' = 0$ si le message déchiffré est $m_0 \oplus s'$ et $b' = 1$ s'il est $m_1 \oplus s'$.

La sécurité CCA est très forte (généralement, l'attaquant obtient moins d'informations que tout le texte en clair), mais nous voulons des définitions qui puissent s'appliquer à de nombreux contextes.

Confidentialité + Intégrité

On a examiné séparément les problèmes de confidentialité (à l'aide du chiffrement) et d'intégrité (à l'aide des MACs). On souhaite désormais les résoudre simultanément.

L'expérience de chiffrement infalsifiable $\texttt{Enc-forge}_{\mathcal{A},\Pi}(n)$

1. Une clé k est générée en exécutant $\text{Gen}(1^n)$.
2. L'adversaire $\mathcal{A}$ reçoit en entrée 1^n et un accès oracle à Enc_k et soit $\mathcal{Q}$ l'ensemble des requêtes soumises à l'oracle par $\mathcal{A}$. $\mathcal{A}$ produit en sortie un texte chiffré c. Soit $m := \text{Dec}_k(c)$.
3. Soit $\perp$ une erreur de déchiffrement. Le résultat de l'expérience est 1 (c'est-à-dire que $\mathcal{A}$ réussit) si $m \neq \perp \wedge m \notin \mathcal{Q}$, et 0 dans le cas contraire.

Infalsifiabilité

Un schéma de chiffrement à clé privée Π est infalsifiable [🇬🇧 unforgeable] si, pour tout adversaire PPT $\mathcal{A}$, il existe une fonction négligeable $\texttt{negl}$ telle que

$$\Pr(\texttt{Enc-forge}_{\mathcal{A},\Pi}(n) = 1) \leq \texttt{negl}(n).$$

Schéma de chiffrement authentifié (AE)

Schéma de chiffrement authentifié (AE)

Un schéma de chiffrement à clé privée est appelé schéma de chiffrement authentifié [🇬🇧 authenticated encryption (AE) scheme] s'il est CCA-sûr et infalsifiable.

- Il existe des schémas à clé privée CCA-sûrs qui ne sont pas AE.
- De nombreuses applications du chiffrement à clé privée en présence d'un adversaire actif nécessitent l'intégrité.
- La plupart des constructions CCA-sûrs naturelles sont également infalsifiables et sont à peu près aussi efficaces que les schémas uniquement CCA-sûrs.

En pratique, il n'y a aucune raison d'utiliser un schéma CCA-sûr qui ne soit pas aussi AE.

L'expérience de chiffrement authentifié $\texttt{PrivK}^{ae}_{\mathcal{A},\Pi}(n)$

1. Une clé k est générée en exécutant $\text{Gen}(1^n)$
2. Un bit uniforme $b \in \{0,1\}$ est choisi.
3. L'adversaire $\mathcal{A}$ reçoit l'entrée 1^n et a accès à deux oracles :
   - 3.1 Si $b = 0$, $\mathcal{A}$ a accès à Enc_k et Dec_k.
   - 3.2 Si $b = 1$, $\mathcal{A}$ a accès à Enc^0_k et $\text{Dec}_\perp$, où $\text{Enc}^0_k(m) := \text{Enc}_k(0^{|m|})$ et $\text{Dec}_\perp$ renvoie toujours $\perp$.

   $\mathcal{A}$ n'est pas autorisé à interroger son deuxième oracle sur un texte chiffré c qu'il a précédemment reçu comme réponse de son premier oracle.
4. $\mathcal{A}$ produit un bit b'. Le résultat de l'expérience est 1 ($\mathcal{A}$ réussit) si $b' = b$, sinon 0.

Au lieu d'utiliser deux expériences, on peut *équivalemment* les fusionner en une seule.

Chiffrement authentifié

Un schéma de chiffrement à clé privée Π est un schéma AE si, pour tout adversaire PPT $\mathcal{A}$, il existe une fonction négligeable $\texttt{negl}$ telle que

$$\Pr(\texttt{PrivK}^{ae}_{\mathcal{A},\Pi}(n) = 1) \leq \frac{1}{2} + \texttt{negl}(n).$$

Paradigmes des schémas AE : CPA + MAC

Un chiffrement CPA-sûr et un MAC sûr ne se combinent pas toujours dans un schéma AE

Plusieurs approches pour combiner un schéma de chiffrement CPA-sûr $\Pi_E = (\text{Enc}, \text{Dec})$ et un MAC fortement sûr $\Pi_M = (\text{Mac}, \text{Vrfy})$ avec des clés indépendantes k_E et k_M respectivement :

- Chiffrement et authentification : le chiffrement et l'authentification sont calculés indépendamment en parallèle
- Authentification puis chiffrement : l'étiquette est d'abord calculé, puis le message et l'étiquette sont chiffrés ensemble
- Chiffrement puis authentification : le message est d'abord chiffré, puis l'étiquette du texte chiffré est calculé.

$\rightarrow$ une telle approche peut être toujours sûre, ou parfois vulnerable selon l'instanciation. On va les analyser une par une.

Chiffrer et authentifier

Étant donné m, le texte chiffré est (c, t), où

$$c \leftarrow \mathrm{Enc}_{k_E}(m) \quad \text{et} \quad t \leftarrow \mathrm{Mac}_{k_M}(m).$$

Le destinataire déchiffre c pour récupérer m, en supposant qu'aucune erreur ne s'est produite, il vérifie l'étiquette t. Il affiche m si $\mathrm{Vrfy}_{k_M}(m, t) = 1$, et une erreur dans le cas contraire.

Parfois vulnérable aux attaques CPA :

si le MAC est déterministe, alors l'étiquette d'un message est toujours la même.

$\Rightarrow$ l'espion peut détecter si le même message a été envoyé deux fois.

Problème réel : de nombreux MAC sont déterministes (par exemple, CBC-MAC)

Authentifier-puis-chiffrer

L'étiquette

$$t \leftarrow \mathrm{Mac}_{k_M}(m)$$

est calculée, puis le texte chiffré

$$c \leftarrow \mathrm{Enc}_{k_E}(m||t)$$

est transmis. Le destinataire déchiffre c pour récupérer $m||t$, en supposant qu'aucune erreur ne s'est produite, puis il vérifie l'étiquette t. Il affiche m si $\mathrm{Vrfy}_{k_M}(m,t) = 1$, et une erreur dans le cas contraire.

Parfois vulnérable aux attaques CCA : plus compliqué, par exemple le mode CBC avec remplissage [🇬🇧 padding].

Attaque réelle menée pour certaines configurations TLS

Chiffrer-puis-authentifier

Le message est chiffré comme suit :

$$c \leftarrow \mathrm{Enc}_{k_E}(m)$$

et le texte chiffré est (c, t) avec

$$t \leftarrow \mathrm{Mac}_{k_M}(c).$$

Si $\mathrm{Vrfy}_{k_M}(c, t) = 1$, le destinataire déchiffre c, c-à-d il calcule $\mathrm{Dec}_{k_E}(c)$, et affiche le résultat, sinon il affiche une erreur.

Théorème

Soit Π_E un schéma de chiffrement à clé privée CPA-sûr et Π_M un MAC fortement sûr. Alors la construction ci-dessus est un schéma AE.

Les clés doivent être différentes !

⚠ Si $k = k_E = k_M$, alors le schéma pourrait ne pas être sûr !

Exemple : Considérons l'approche "chiffrer-puis-authentifier" avec

$$\mathrm{Enc}_k(m) = F_k(m||r), \quad \text{pour } m, r \in \{0,1\}^{n/2},$$

où F est une permutation pseudo-aléatoire forte (donc F^{-1} est également une permutation pseudo-aléatoire forte). Le schéma est CCA-sûr.

Définissons

$$\mathrm{Mac}_k(c) = F_k^{-1}(c),$$

ceci est fortement sûr. Cependant,

$$\mathrm{Enc}_k(m), \mathrm{Mac}_k(\mathrm{Enc}_k(m)) = F_k(m||r), F_k^{-1}(F_k(m||r)) = F_k(m||r), m||r,$$

c'est-à-dire le message est révélé en clair.

Quelques schémas AE standardisés

- GCM (mode Galois/compteur) : paradigme “chiffrer-puis-authentifier” avec le mode CTR comme chiffrement et GMAC comme MAC.

  ⚠ Les deux clés ne sont pas indépendantes, mais dans ce cas particulier, cela ne pose pas de problème.

  + hautement parallélisable et extrêmement rapide sur les processeurs modernes grâce aux instructions matérielles lorsqu'il est instancié avec le chiffrement par blocs AES
- CCM (compteur avec CBC-MAC) : authentifier-puis-chiffrer avec le mode CTR comme chiffrement et CBC-MAC comme MAC. La même clé est utilisée pour les 2.
  ⚠ Malgré la clé unique et l'approche “authentifier-puis-chiffrer”, la sécurité du CCM peut être prouvée.
  + facile à mettre en œuvre (simplement un chiffrement par blocs)
  − relativement lent et ne peut pas être entièrement parallélisé
- ChaCha20-Poly1305 : paradigme “chiffrer-puis-authentifier” avec le chiffrement de flux ChaCha20 comme chiffrement et Poly1305 comme MAC, avec la même clé.
  + très rapide dans les logiciels (idéal si instructions matérielles pas disponibles)

Sessions de communication sûrs (Sécurité des réseaux)

Session de communication : intervalle de temps où les parties maintiennent un état.

Soit $\Pi = (\mathrm{Enc}, \mathrm{Dec})$ un schéma AE. Alice et Bob partagent une clé k qu'ils utilisent pendant la session. Ils utilisent tous deux Enc_k pour chiffrer les messages qu'ils souhaitent envoyer et Dec pour déchiffrer les messages reçus.

Attaques lorsqu'un attaquant a le contrôle du réseau.

- **Attaque par réorganisation :** si Alice envoie c_1, c_2, où $c_i = \mathrm{Enc}_k(m_i)$, l'attaquant peut envoyer c_2, c_1 à Bob à la place.
- **Attaques par rejeu :** un attaquant peut rejouer un texte chiffré valide même s'il n'a été envoyé qu'une seule fois.
- **Attaque par suppression de messages :** un attaquant peut supprimer certains des messages envoyés entre Alice et Bob sans que cela soit détecté.
- **Attaque par réflexion :** un attaquant peut prendre c envoyé par Alice à Bob et le renvoyer à Alice.

Comment contrer ces attaques ?

Contre-mesures

Chaque partie dispose de deux compteurs $\texttt{ctr}_{A,B}$ (resp. $\texttt{ctr}_{B,A}$) qui enregistrent le nombre de messages envoyés par Alice à Bob (resp. par Bob à Alice) et sont initialisés à 0. Ils conviennent également d'un bit directionnel $\hat{b}$.

1. Étant donné un message m, Alice calcule le texte chiffré $c \leftarrow \text{Enc}_k(\hat{b}||\texttt{ctr}_{A,B}||m)$ et envoie c à Bob ;
2. Bob déchiffre, si le résultat est $\perp$, il rejette. Sinon, il décompose $c = b||\texttt{ctr}||m$;
3. Si $b = \hat{b}$ et $\texttt{ctr} = \texttt{ctr}_{A,B}$, alors Bob affiche m et incrémente $\texttt{ctr}_{A,B}$. Sinon, il rejette.

+ Les compteurs évitent les **attaques par réorganisation, rejeu et suppression de messages**
+ Lorsque Bob envoie un message à Alice, il utilise $\hat{b} \oplus 1$ à la place de $\hat{b}$. De cette manière, le bit directionnel empêche les **attaques par réflexion**