

8. Etude d'un schéma de chiffrement

8.1 Chiffrement Multiple

Soit $\mathcal{S} = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ un schéma de chiffrement sémantiquement sûr contre les attaques à messages choisis (mais pas forcément contre les attaques à chiffrés choisis). On définit alors le schéma de chiffrement multiple $\mathcal{S}' = (\mathcal{K}', \mathcal{E}', \mathcal{D}')$ de la façon suivante :

- l'algorithme de génération de clés \mathcal{K}' exécute deux fois l'algorithme \mathcal{K} , et fournit ainsi deux paires de clés $(\text{sk}_0, \text{pk}_0)$ et $(\text{sk}_1, \text{pk}_1)$. La clé secrète est alors définie par $\text{sk}' = (\text{sk}_0, \text{sk}_1)$, tandis que la clé publique est définie par $\text{pk}' = (\text{pk}_0, \text{pk}_1)$.
- l'algorithme de chiffrement consiste à chiffrer deux fois le même message sous les deux clés pk_0 et pk_1 . Ainsi, pour chiffrer le message m , on choisit deux aléas r_0 et r_1 , puis on calcule $c_0 = \mathcal{E}_{\text{pk}_0}(m, r_0)$ et $c_1 = \mathcal{E}_{\text{pk}_1}(m, r_1)$, enfin on retourne $c = (c_0, c_1)$.
- l'algorithme de déchiffrement, sur $c = (c_0, c_1)$, déchiffre les deux sous-chiffrés c_0 et c_1 , puis compare les clairs. Plus concrètement, on obtient $m_0 = \mathcal{D}_{\text{sk}_0}(c_0)$ et $m_1 = \mathcal{D}_{\text{sk}_1}(c_1)$ (l'un d'eux peut être égal à \perp en cas d'invalidité). Si $m_0 = m_1$, alors $m = m_0$, sinon $m = \perp$ (qui signifie « chiffré invalide »). Puis on retourne m .

Jeu 0. Le jeu d'attaque contre la **sécurité sémantique du schéma \mathcal{S}'** est le suivant :

- on génère une clé publique aléatoire $\text{pk}' = (\text{pk}_0, \text{pk}_1)$, que l'on transmet à l'attaquant \mathcal{A} ;
- l'attaquant \mathcal{A} choisit deux messages m^0 et m^1 ;
- on choisit un bit b aléatoire, on chiffre m^b en

$$c = \mathcal{E}'_{\text{pk}'}(m^b, r' = (r_0, r_1)) = (c_0 = \mathcal{E}_{\text{pk}_0}(m^b, r_0), c_1 = \mathcal{E}_{\text{pk}_1}(m^b, r_1)) ;$$

- l'attaquant \mathcal{A} retourne son choix b' pour b .

Son avantage utilise l'événement $\mathbf{S} = (b' = b)$, et est défini par

$$\text{Adv}_{\mathcal{S}'}^{\text{ind-cpa}}(\mathcal{A}) = 2 \Pr[\mathbf{S}_0] - 1.$$

Jeu 1 On considère désormais le jeu modifié suivant, avec un attaquant \mathcal{A} contre \mathcal{S}' , où intervient un bit aléatoire u qui définit sur quelle sous-clé on applique le jeu de la **sécurité sémantique du schéma de base \mathcal{S}** :

- on fait générer pk_u , puis on génère le couple $(\text{pk}_{1-u}, \text{sk}_{1-u})$. On définit alors $\text{pk}' = (\text{pk}_0, \text{pk}_1)$, que l'on transmet à l'attaquant \mathcal{A} ;
- l'attaquant \mathcal{A} choisit deux messages m^0 et m^1 ;
- on demande le chiffrement c_u de m_b par pk_u (sans connaître b), on choisit d aléatoire (en espérant que $d = b$), et on génère le *chiffré*

$$c = (c_0, c_1) \quad \text{où} \quad c_{1-u} = \mathcal{E}_{\text{pk}_{1-u}}(m^d, r_{1-u}) ;$$

- l'attaquant \mathcal{A} retourne son choix b' pour b , que l'on transmet.

Q-1. Donner une relation entre $\Pr[\mathbf{S}_1]$ et $\text{Adv}_{\mathcal{S}}^{\text{ind-cpa}}(t + T_{\mathcal{K}} + T_{\mathcal{E}})$, (où $T_{\mathcal{K}}$ est le temps de la génération des clés \mathcal{K} , et $T_{\mathcal{E}}$ est le temps pour évaluer un chiffrement \mathcal{E}).

Rappel : On définit par $\text{Adv}_{\mathcal{S}}^{\text{ind-cpa}}(t)$ l'avantage $\text{Adv}_{\mathcal{S}}^{\text{ind-cpa}}(\mathcal{A})$ maximal sur tous les attaquants \mathcal{A} en temps t .

Q-2. Que peut-on dire du Jeu 1, dans le cas où $d = b$ (et u aléatoire), et donc de la probabilité de l'événement \mathbf{S}_1 conditionné à $b = d$?

Q-3. Que peut-on dire du Jeu 1, dans le cas où $d \neq b$ (et u aléatoire). En constatant qu'aucune information sur u n'est révélée, évaluer la probabilité de l'événement \mathbf{S}_1 conditionné à $b \neq d$?

Q-4. En déduire une borne sur $\text{Adv}_{\mathcal{S}'}^{\text{ind-cpa}}(t)$, en fonction de $\text{Adv}_{\mathcal{S}}^{\text{ind-cpa}}(t + T_{\mathcal{K}} + T_{\mathcal{E}})$.

8.2 Chiffrement El Gamal

- On va alors se concentrer sur le chiffrement El Gamal comme instantiation du schéma \mathcal{S} .
- Données communes : un groupe cyclique $G = \langle g \rangle$ d'ordre premier q . L'élément g est le générateur de référence.
 - \mathcal{K} choisit sk aléatoirement dans \mathbb{Z}_q , et calcule $pk = g^{sk}$.
 - Pour chiffrer un message $m \in G$, avec un aléa $r \in \mathbb{Z}_q$, on calcule $c = (\alpha = g^r, \beta = pk^r \times m)$.
 - Le déchiffrement de $c = (\alpha, \beta)$ se fait en calculant $m = \beta/\alpha^{sk}$.

Q-5. Montrer la sécurité sémantique de ce schéma sous l'hypothèse Diffie-Hellman décisionnelle qu'on rappellera.

8.3 Preuves d'égalité des chiffrés

Notre objectif sera de déchiffrer le chiffrement « Double El Gamal » (chiffrement multiple \mathcal{S}' ci-dessus avec le chiffrement El Gamal comme schéma de base \mathcal{S}) avec une seule clé de déchiffrement (sk_0 ou sk_1), choisie au moment de la génération des clés. Dans ce cas, il faut empêcher un attaquant de générer un chiffré non-valide (avec deux clairs différents). Il suffit pour cela de lui faire ajouter une preuve que les deux sous-chiffrés contiennent le même clair, tout en ne révélant rien sur ce clair.

Considérons le protocole suivant au sujet du chiffré $c = (c_0, c_1)$ où

$$c_0 = (\alpha_0 = g^{r_0}, \beta_0 = pk_0^{r_0} \times m_0) \quad c_1 = (\alpha_1 = g^{r_1}, \beta_1 = pk_1^{r_1} \times m_1)$$

- on choisit $a, b \in \mathbb{Z}_q$, et on calcule $A_0 = g^{a_0}$, $A_1 = g^{a_1}$ et $B = pk_0^{a_0}/pk_1^{a_1}$;
- étant donné un challenge $e \in \mathbb{Z}_q$, on calcule $b_0 = a_0 - r_0e \pmod q$ et $b_1 = a_1 - r_1e \pmod q$, que l'on retourne.

Q-6. Montrer que la vérification de

$$A_0 = g^{b_0} \alpha_0^e \quad A_1 = g^{b_1} \alpha_1^e \quad B = pk_1^{b_1}/pk_0^{b_0} \times (\beta_1/\beta_0)^e$$

conduit à une preuve d'égalité des clairs dans les chiffrés c_0 et c_1 qui est « complète » (*complete*) et « consistante » (*sound*).

Q-7. Montrer que ce protocole est « zero-knowledge » face à un vérifieur honnête (le challenge e est choisi aléatoirement).

Q-8. Décrire la preuve non-interactive d'égalité des clairs, dans le modèle de l'oracle aléatoire.

Q-9. Décrire le schéma de chiffrement « Double El Gamal », qui retourne un chiffré sous forme de triplet (c_0, c_1, c_2) , où (c_0, c_1) est le chiffrement multiple ci-dessus à base de El Gamal, et c_2 est la preuve non-interactive ci-dessus.

8.4 Sécurité contre les attaques à chiffrés choisis

On reprend le jeu 1 décrit ci-dessus, où \mathcal{S} est le chiffrement El Gamal et \mathcal{S}' est le « Double El Gamal », au cours duquel le simulateur connaît sk_{1-u} . On utilise l'attaquant \mathcal{A} à chiffrés choisis contre la sécurité sémantique de \mathcal{S}' pour casser la sécurité sémantique de \mathcal{S} sur la clé pk_u , sans chiffrés choisis :

- on fait générer pk_u , puis on génère le couple (pk_{1-u}, sk_{1-u}) . On définit alors $pk' = (pk_0, pk_1)$, que l'on transmet à l'attaquant \mathcal{A} ;

- l'attaquant \mathcal{A} choisit deux messages m^0 et m^1 ;
- on demande le chiffrement c_u de m_b par pk_u (sans connaître b), on choisit d aléatoire (en espérant que $d = b$), et on génère le *chiffre*

$$c = (c_0, c_1, c_2) \quad \text{où} \quad c_{1-u} = \mathcal{E}_{\text{pk}_{1-u}}(m^d, r_{1-u}) ;$$

- l'attaquant \mathcal{A} retourne son choix b' pour b , que l'on transmet.

Q-10. Expliquer comment on génère c_2 dans notre simulation.

Q-11. Montrer comment on peut simuler l'oracle de déchiffrement $\mathcal{D}'_{\text{sk}'}$. Sans exhiber la probabilité d'erreur, expliquer dans quels cas la simulation peut différer du déchiffrement.

8.5 Déchiffrement distribué

Q-12. Expliquer comment on peut distribuer le déchiffrement de ce « Double El Gamal » parmi deux autorités.

Q-13. Expliquer pourquoi il est toujours sûr contre les attaques à chiffrés choisis, pour un attaquant qui serait l'une des deux autorités de déchiffrement, contrairement à une variante du El Gamal de base qui inclurait de la redondance dans le clair (à vérifier après un déchiffrement préliminaire, avant de retourner le clair).