Février 2021

Le chiffrement fonctionnel pour le produit scalaire

Angélique Lopez, Fivos Reyre, Sid-Ali Zitouni-Terki

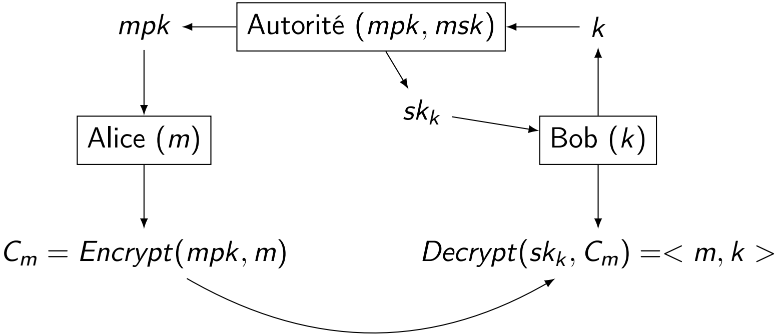
Université de Bordeaux
Master 2 CSI

27 février 2021

Introduction

Chiffrement fonctionnel pour le produit scalaire

Schéma général



université de BORDEAUX

Plan

université de BORDEAUX

Définition formelle

Schéma de **chiffrement fonctionnel** pour une fonctionnalité F :

1. $mpk, msk \leftarrow Setup(1^\lambda)$
2. $sk_k \leftarrow KeyDer(msk, k)$
3. $C_m \leftarrow Encrypt(mpk, m)$
4. $y \leftarrow Decrypt(mpk, C_m, sk_k)$ avec $y = F(k, m)$

Définition formelle

Schéma de **chiffrement fonctionnel** pour une fonctionnalité F :

1. $mpk, msk \leftarrow Setup(1^\lambda)$
2. $sk_k \leftarrow KeyDer(msk, k)$
3. $C_m \leftarrow Encrypt(mpk, m)$
4. $y \leftarrow Decrypt(mpk, C_m, sk_k)$ avec $y = F(k, m)$

Définition formelle

Schéma de **chiffrement fonctionnel** pour une fonctionnalité F :

1. $mpk, msk \leftarrow Setup(1^\lambda)$
2. $sk_k \leftarrow KeyDer(msk,\ k)$
3. $C_m \leftarrow Encrypt(mpk,\ m)$
4. $y \leftarrow Decrypt(mpk,\ C_m,\ sk_k)$ avec $y = F(k, m)$

Définition formelle

Schéma de **chiffrement fonctionnel** pour une fonctionnalité F :

1. $mpk, msk \leftarrow Setup(1^\lambda)$
2. $sk_k \leftarrow KeyDer(msk,\ k)$
3. $C_m \leftarrow Encrypt(mpk,\ m)$
4. $y \leftarrow Decrypt(mpk,\ C_m,\ sk_k)$ avec $y = F(k, m)$

Le cas du produit scalaire

- $F = IP$
- Somme pondérée : $< k, m >= \sum_{i=0}^{l} k_i m_i$

Notion d'indistinguabilité IND-FE-CPA

$\mathrm{Exp}_{\epsilon,\lambda}^{IND-FE-CPA}(\mathcal{A})$

$(mpk, msk) \xleftarrow{\$} Setup(1^\lambda)$

$V \leftarrow \emptyset$

$(m_0, m_1, s) \leftarrow \mathcal{A}_1(mpk)$

$b^* \xleftarrow{\$} \{0,1\}$

$Ct^* \xleftarrow{\$} Encrypt(mpk, m_{b^*})$

$b \leftarrow \mathcal{A}_2(s, Ct^*)$

Si $b \neq b^*$ ou si $\exists\ k \in V,\ F(k, m_0) \neq F(k, m_1)$:

Retourner 0

Retourner 1

$O(k)$:

$V \leftarrow V \cup \{k\}$

$sk_k \xleftarrow{\$} KeyDer(msk, k)$

Retourner sk_k

Sécurité sélective : s-IND-FE-CPA

$\mathrm{Exp}_{\epsilon,\lambda}^{s-IND-FE-CPA}(\mathcal{A})$

$(m_0, m_1, s) \leftarrow \mathcal{A}_1()$

$(mpk, msk) \xleftarrow{\$} Setup(1^\lambda)$

$V \leftarrow \emptyset$

$b^* \xleftarrow{\$} \{0,1\}$

$Ct^* \xleftarrow{\$} Encrypt(mpk, m_{b^*})$

$b \leftarrow \mathcal{A}_2(s, Ct^*)$

Si $b \neq b^*$ ou si $\exists\ k \in V,\ F(k, m_0) \neq F(k, m_1)$:

Retourner 0

Retourner 1

$O(k)$:

$V \leftarrow V \cup \{k\}$

$sk_k \xleftarrow{\$} KeyDer(msk, k)$

Retourner sk_k

Schéma PKE-IP ElGamal

- Schéma PKE-IP

Schéma PKE-IP ElGamal

Application à ElGamal :

1. $Setup(1^\lambda, 1^l)$ retourne :
   $mpk = (pk_1, .., pk_l)$ avec $pk_i = g^{x_i}$
   $msk = (sk_1, ..., sk_l)$ avec $sk_i = x_i$
2. $KeyDer(msk, k)$ retourne $sk_k =< msk, k > = \sum_{i \in [l]} x_i k_i$.
3. $Encrypt(mpk, m)$ retourne $C_m = (c_{m_0}, c_{m_1})$ avec :
   $r \xleftarrow{\$} \mathbb{Z}_q$, $c_{m_0} = g^r$ et $c_{m_1} = \{c_{m_i} = g^{m_i} h_i^r\}_{i=1}^l$
4. $Decrypt(mpk, sk_k, C_m)$ retourne
   $log_g(\frac{\prod_{i \in [l]} c_{m_i}^{k_i}}{c_{m_0}^{sk_k}}) = log_g(g^{<m,k>}) = < m, k >$

université de BORDEAUX

Schéma PKE-IP ElGamal

Application à ElGamal :

1. $Setup(1^\lambda, 1^l)$ retourne :
   $mpk = (pk_1, ..., pk_l)$ avec $pk_i = g^{x_i}$
   $msk = (sk_1, ..., sk_l)$ avec $sk_i = x_i$
2. $KeyDer(msk, k)$ retourne $sk_k =< msk, k > = \sum_{i \in [l]} x_i k_i$.
3. $Encrypt(mpk, m)$ retourne $C_m = (c_{m_0}, c_{m_1})$ avec :
   $r \xleftarrow{\$} \mathbb{Z}_q$, $c_{m_0} = g^r$ et $c_{m_1} = \{c_{m_i} = g^{m_i} h_i^r\}_{i=1}^l$
4. $Decrypt(mpk, sk_k, C_m)$ retourne
   $log_g(\frac{\prod_{i \in [l]} c_{m_i}^{k_i}}{c_{m_0}^{sk_k}}) = log_g(g^{<m,k>}) = < m, k >$

Schéma PKE-IP ElGamal

Application à ElGamal :

1. $Setup(1^\lambda, 1^l)$ retourne :
   $mpk = (pk_1, ..., pk_l)$ avec $pk_i = g^{x_i}$
   $msk = (sk_1, ..., sk_l)$ avec $sk_i = x_i$
2. $KeyDer(msk, k)$ retourne $sk_k =< msk, k > = \sum_{i \in [l]} x_i k_i$.
3. $Encrypt(mpk, m)$ retourne $C_m = (c_{m_0}, c_{m_1})$ avec :
   $r \xleftarrow{\$} \mathbb{Z}_q$, $c_{m_0} = g^r$ et $c_{m_1} = \{c_{m_i} = g^{m_i} h_i^r\}_{i=1}^l$
4. $Decrypt(mpk, sk_k, C_m)$ retourne
   $log_g(\frac{\prod_{i \in [l]} c_{m_i}^{k_i}}{c_{m_0}^{sk_k}}) = log_g(g^{<m,k>}) = < m, k >$

Schéma PKE-IP ElGamal

Application à ElGamal :

1. $Setup(1^\lambda, 1^l)$ retourne :
   $mpk = (pk_1, ..., pk_l)$ avec $pk_i = g^{x_i}$
   $msk = (sk_1, ..., sk_l)$ avec $sk_i = x_i$
2. $KeyDer(msk, k)$ retourne $sk_k =< msk, k > = \sum_{i \in [l]} x_i k_i$.
3. $Encrypt(mpk, m)$ retourne $C_m = (c_{m_0}, c_{m_1})$ avec :
   $r \xleftarrow{\$} \mathbb{Z}_q$, $c_{m_0} = g^r$ et $c_{m_1} = \{c_{m_i} = g^{m_i} h_i^r\}_{i=1}^l$
4. $Decrypt(mpk, sk_k, C_m)$ retourne
   $log_g(\frac{\prod_{i \in [l]} c_{m_i}^{k_i}}{c_{m_0}^{sk_k}}) = log_g(g^{<m,k>}) = < m, k >$

Schéma PKE-IP ElGamal

- Limite : sécurité sélective et logarithme discret coûteux

Amélioration du schéma DDH-IP

- Amélioration : sécurité IND-FE-CPA

Amélioration du schéma DDH-IP

1. $Setup(1^\lambda, 1^l)$ retourne $mpk = (G, g, h, \{h_i\}_{i=1}^l)$ et $msk = \{(s_i, t_i)\}_{i=1}^l$ avec :
   $\{s_i, t_i \xleftarrow{\$} \mathbb{Z}_q\}_{i=1}^l$ et $h_i = g^{s_i} h^{t_i}$
2. $KeyDer(msk, k)$ retourne sk_k avec :
   $$sk_k = (s_k, t_k) = (< s, k >, < t, k >) = (\sum_{i=1}^{l} s_i k_i, \sum_{i=1}^{l} t_i k_i).$$
3. $Encrypt(mpk, m)$ retourne $C_m = (C, D, E_1, ..., E_l)$ avec :
   $(r \xleftarrow{\$} \mathbb{Z}_q)$, $C = g^r$, $D = h^r$ et $\{E_i = g^{m_i} h_i^r\}_{i=1}^l$.
4. $Decrypt(mpk, sk_k, C_m)$ retourne $log_g(E_k)$ avec :
   $$E_k = \frac{\prod_{i=1}^{l} E_i^{k_i}}{C^{s_k} D^{t_k}} = \frac{\prod_{i=1}^{l} g^{m_i + rs_i k_i} h^{rt_i k_i}}{\prod_{i=1}^{l} g^{rs_i k_i} h^{rt_i k_i}} = g^{<k,m>}.$$

Amélioration du schéma DDH-IP

1. *Setup*$(1^\lambda, 1^l)$ retourne $mpk = (G, g, h, \{h_i\}_{i=1}^l)$ et $msk = \{(s_i, t_i)\}_{i=1}^l$ avec :
   $\{s_i, t_i \xleftarrow{\$} \mathbb{Z}_q\}_{i=1}^l$ et $h_i = g^{s_i} h^{t_i}$
2. *KeyDer*$(msk,\ k)$ retourne sk_k avec :
   $$sk_k = (s_k, t_k) = (< s, k >, < t, k >) = (\sum_{i=1}^{l} s_i k_i, \sum_{i=1}^{l} t_i k_i).$$
3. *Encrypt*$(mpk,\ m)$ retourne $C_m = (C, D, E_1, ..., E_l)$ avec :
   $(r \xleftarrow{\$} \mathbb{Z}_q)$, $C = g^r, D = h^r$ et $\{E_i = g^{m_i} h_i^r\}_{i=1}^l$.
4. *Decrypt*$(mpk,\ sk_k,\ C_m)$ retourne $log_g(E_k)$ avec :
   $$E_k = \frac{\prod_{i=1}^{l} E_i^{k_i}}{C^{s_k} D^{t_k}} = \frac{\prod_{i=1}^{l} g^{m_i + rs_i k_i} h^{rt_i k_i}}{\prod_{i=1}^{l} g^{rs_i k_i} h^{rt_i k_i}} = g^{<k,m>}.$$

Amélioration du schéma DDH-IP

1. $Setup(1^\lambda, 1^l)$ retourne $mpk = (G, g, h, \{h_i\}_{i=1}^l)$ et $msk = \{(s_i, t_i)\}_{i=1}^l$ avec :
   $\{s_i, t_i \xleftarrow{\$} \mathbb{Z}_q\}_{i=1}^l$ et $h_i = g^{s_i} h^{t_i}$
2. $KeyDer(msk, k)$ retourne sk_k avec :
   $$sk_k = (s_k, t_k) = (< s, k >, < t, k >) = (\sum_{i=1}^{l} s_i k_i, \sum_{i=1}^{l} t_i k_i).$$
3. $Encrypt(mpk, m)$ retourne $C_m = (C, D, E_1, ..., E_l)$ avec :
   $(r \xleftarrow{\$} \mathbb{Z}_q)$, $C = g^r, D = h^r$ et $\{E_i = g^{m_i} h_i^r\}_{i=1}^l$.
4. $Decrypt(mpk, sk_k, C_m)$ retourne $log_g(E_k)$ avec :
   $$E_k = \frac{\prod_{i=1}^{l} E_i^{k_i}}{C^{s_k} D^{t_k}} = \frac{\prod_{i=1}^{l} g^{m_i + rs_i k_i} h^{rt_i k_i}}{\prod_{i=1}^{l} g^{rs_i k_i} h^{rt_i k_i}} = g^{<k,m>}.$$

Amélioration du schéma DDH-IP

1. $Setup(1^\lambda, 1^l)$ retourne $mpk = (G, g, h, \{h_i\}_{i=1}^l)$ et $msk = \{(s_i, t_i)\}_{i=1}^l$ avec :
   $\{s_i, t_i \xleftarrow{\$} \mathbb{Z}_q\}_{i=1}^l$ et $h_i = g^{s_i} h^{t_i}$
2. $KeyDer(msk, k)$ retourne sk_k avec :
   $$sk_k = (s_k, t_k) = (<s, k>, <t, k>) = (\sum_{i=1}^{l} s_i k_i, \sum_{i=1}^{l} t_i k_i).$$
3. $Encrypt(mpk, m)$ retourne $C_m = (C, D, E_1, ..., E_l)$ avec :
   $(r \xleftarrow{\$} \mathbb{Z}_q)$, $C = g^r, D = h^r$ et $\{E_i = g^{m_i} h_i^r\}_{i=1}^l$.
4. $Decrypt(mpk, sk_k, C_m)$ retourne $log_g(E_k)$ avec :
   $$E_k = \frac{\prod_{i=1}^{l} E_i^{k_i}}{C^{s_k} D^{t_k}} = \frac{\prod_{i=1}^{l} g^{m_i + rs_i k_i} h^{rt_i k_i}}{\prod_{i=1}^{l} g^{rs_i k_i} h^{rt_i k_i}} = g^{<k,m>}.$$

Amélioration du schéma DDH-IP

- Limite : logarithme discret coûteux

Sécurité des constructions étudiées

- Des sécurités différentes
- Preuve par réduction ou séquence de jeux

Sécurité du schéma PKE-IP ElGamal

Challenger $\mathcal{C}$ | Algorithme $\mathcal{B}$ | Attaquant $\mathcal{A}$

$a \xleftarrow{\$} \mathcal{M}$

$\xleftarrow{(0,a)}$

$m \xleftarrow{\$} \{0, a\}$

$C_m = \mathcal{E}.\text{Encrypt}(m, pk)$

$\xrightarrow{C_m}$

$(x_0, x_1) \xleftarrow{\$} \mathcal{A}_1()$

$\xleftarrow{(x_0,x_1)}$

Calcul de mpk à partir de (pk, x_0, x_1)

$b \xleftarrow{\$} \{0,1\}$

Calcul de C_m^* à partir de (mpk, x_b)

$\xrightarrow{C_m^*}$

$\xleftrightarrow{\text{KeyDer}(k)}$

$b^* \xleftarrow{\$} \mathcal{A}_2(mpk, C_m^*)$

$\xleftarrow{b^*}$

$m' = \{0 \text{ si } b = b^*,\ a \text{ sinon}\}$

$\xleftarrow{m'}$

$$
\begin{aligned}
\mathbf{Adv}_{\mathcal{E},\lambda}^{s-IND-CPA}(\mathcal{B}) &= |Pr[\mathbf{Exp}_{\mathcal{E},\lambda}^{s-IND-CPA}(\mathcal{B}) = 1] - \tfrac{1}{2}| \\
&= |Pr[\mathbf{Exp}_{\mathcal{E},\lambda}^{s-IND-CPA}(\mathcal{B}) = 1 \ \cap \ m = 0] \\
&\qquad + Pr[\mathbf{Exp}_{\mathcal{E},\lambda}^{s-IND-CPA}(\mathcal{B}) = 1 \cap m = a] - \tfrac{1}{2}| \\
&= |\tfrac{1}{2} Pr[\mathbf{Exp}_{\mathcal{E},\lambda}^{s-IND-CPA}(\mathcal{B}) = 1 \mid m = 0] \\
&\qquad + \tfrac{1}{2} Pr[\mathbf{Exp}_{\mathcal{E},\lambda}^{s-IND-CPA}(\mathcal{B}) = 1 \mid m = a] - \tfrac{1}{2}| \\
&= |\tfrac{1}{2} Pr[\mathbf{Exp}_{\mathcal{F},\lambda}^{s-IND-FE-CPA}(\mathcal{A}) = 1 \mid m = 0] \\
&\qquad + \frac{1}{2} \times \frac{1}{2} - \frac{1}{2}| \\
&= \tfrac{1}{2} |Pr[\mathbf{Exp}_{\mathcal{F},\lambda}^{s-IND-FE-CPA}(\mathcal{A}) = 1 \mid m = 0] - \tfrac{1}{2}| \\
&= \tfrac{1}{2} \mathbf{Adv}_{\mathcal{F},\lambda}^{s-IND-FE-CPA}(\mathcal{A})
\end{aligned}
$$

Sécurité du schéma DDH-IP

Jeu 0 : Expérience IND-FE-CPA

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} h_i^r$$

Jeu 1 : Modification du chiffré

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

Même distribution pour $C_{m_\beta} = (C, D, \{E_i\}_i)$

$$Pr(S_1) = Pr(S_0)$$

Jeu 2 : Modification du chiffré

$$C = g^r ; D = h^{r+r'} ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

$|Pr(S_2) - Pr(S_1)| = Adv_C^{DDH}(\lambda)$ et $Pr(S_2) = \frac{1}{2}$

Sécurité du schéma DDH-IP

Jeu 0 : Expérience IND-FE-CPA

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} h_i^r$$

Jeu 1 : Modification du chiffré

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

Même distribution pour $C_{m_\beta} = (C, D, \{E_i\}_i)$

$$Pr(S_1) = Pr(S_0)$$

Jeu 2 : Modification du chiffré

$$C = g^r ; D = h^{r+r'} ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

$|Pr(S_2) - Pr(S_1)| = Adv_C^{DDH}(\lambda)$ et $Pr(S_2) = \frac{1}{2}$

Sécurité du schéma DDH-IP

$$\boxed{\text{Jeu 0 : Expérience IND-FE-CPA}}$$

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} h_i^r$$

$$\boxed{\text{Jeu 1 : Modification du chiffré}}$$

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

Même distribution pour $C_{m_\beta} = (C, D, \{E_i\}_i)$

$$Pr(S_1) = Pr(S_0)$$

$$\boxed{\text{Jeu 2 : Modification du chiffré}}$$

$$C = g^r ; D = h^{r+r'} ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

$|Pr(S_2) - Pr(S_1)| = Adv_{\mathcal{C}}^{DDH}(\lambda)$ et $Pr(S_2) = \frac{1}{2}$

Sécurité du schéma DDH-IP

Jeu 2 : Modification du chiffré

$$|Pr(S_2) - Pr(S_1)| = Adv_{\mathcal{C}}^{DDH}(\lambda) \text{ et } Pr(S_2) = \frac{1}{2}$$

$$C = g^r ; D = h^{r+r'} ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

$$\text{Chiffré de } z_\beta = y_\beta + \omega \cdot r' \cdot t \in \mathbb{Z}_q^l$$

$$y^\perp = \{x \in \mathbb{Z}_q^l | < x, y >= 0 \bmod q \}$$

$$X = \left[\frac{X_{top}}{y'^T}\right] \in \mathbb{Z}_q^{l \times l}$$

$$< y', z_\beta >=< y', y_\beta + \omega \cdot r' \cdot t >=< y', u > +\omega \cdot r' \cdot < y', t >$$

Sécurité du schéma DDH-IP

$\boxed{\text{Jeu 0 : Expérience IND-FE-CPA}}$

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} h_i^r$$

$\boxed{\text{Jeu 1 : Modification du chiffré}}$

$$C = g^r ; D = h^r ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

Même distribution pour $C_{m_\beta} = (C, D, \{E_i\}_i)$

$$Pr(S_1) = Pr(S_0)$$

$\boxed{\text{Jeu 2 : Modification du chiffré}}$

$$C = g^r ; D = h^{r+r'} ; E_i = g^{m_{\beta_i}} C^{s_i} D^{t_i}$$

$|Pr(S_2) - Pr(S_1)| = Adv_{\mathcal{C}}^{DDH}(\lambda)$ et $Pr(S_2) = \frac{1}{2}$

$$Adv^{IND-FE-CPA}(\mathcal{A}) = |Pr(S_0) - \frac{1}{2}| = Adv_{\mathcal{C}}^{DDH}(\lambda)$$

université de BORDEAUX

Conclusion

Références :

1. Boneh D., Sahai A., Waters B. (2011) *Functional Encryption : Definitions and Challenges*. In : Ishai Y. (eds) Theory of Cryptography. TCC 2011. Lecture Notes in Computer Science, vol 6597. Springer, Berlin, Heidelberg.
2. Abdalla M., Bourse F., De Caro A., Pointcheval D. (2015) *Simple Functional Encryption Schemes for Inner Products*. In : Katz J. (eds) Public-Key Cryptography – PKC 2015. PKC 2015. Lecture Notes in Computer Science, vol 9020. Springer, Berlin, Heidelberg.
3. Agrawal S., Libert B., Stehlé D. (2016) *Fully Secure Functional Encryption for Inner Products, from Standard Assumptions*. In : Robshaw M., Katz J. (eds) Advances in Cryptology – CRYPTO 2016. CRYPTO 2016. Lecture Notes in Computer Science, vol 9816. Springer, Berlin, Heidelberg.

Merci de votre attention !

Annexe

(1) Choix des messages par $\mathcal{B}$

$\mathcal{B}$ commence par choisir deux messages dans $\mathcal{M} = \mathbb{Z}_q$: a aléatoire dans $\mathcal{M}$ et 0.

(2-7) Obtention du chiffré challenge et des messages de la part de $\mathcal{A}$

$\mathcal{B}$ donne ces deux messages à $\mathcal{C}$ qui lui renvoie une clé publique pk et un chiffré C_t de 0 ou a.
Il demande aussi à $\mathcal{A}$ deux messages $x_{\mathbf{0}}$ et $x_{\mathbf{1}}$ dans $\mathbb{Z}_q^l$.

(8) Fabrication de la clé publique *mpk*

$\mathcal{B}$ trouve une base $(z_{\mathbf{1}}, \dots, z_{l-\mathbf{1}})$ de $(x_{\mathbf{1}} - x_{\mathbf{0}})^{\perp}$. Cette base est utile pour dériver les clefs secrètes. Ainsi, $(x_{\mathbf{1}} - x_{\mathbf{0}}, z_{\mathbf{1}}, \dots, z_{l-\mathbf{1}})$ est une base de $\mathbb{Z}_q^l$ et les vecteurs de la base canonique e_i peuvent être écrits dans cette base. Il existe donc pour tout $i \in \{1, \dots, l\}$ et $j \in \{1, \dots, l-1\}$, $\lambda_{i,j}$ et $\alpha_i \in \mathbb{Z}_q$ tels que :
$e_i = \alpha_i(x_{\mathbf{1}} - x_{\mathbf{0}}) + \sum_j \lambda_{i,j} z_j$.
On a
$$\begin{aligned} x_{\mathbf{1},i} - x_{\mathbf{0},i} &= < x_{\mathbf{1}} - x_{\mathbf{0}}, e_i > = < x_{\mathbf{1}} - x_{\mathbf{0}}, \alpha_i(x_{\mathbf{1}} - x_{\mathbf{0}}) + \textstyle\sum_j \lambda_{i,j} z_j > \\ &= \alpha_i < x_{\mathbf{1}} - x_{\mathbf{0}}, x_{\mathbf{1}} - x_{\mathbf{0}} > + \textstyle\sum_j \lambda_{i,j} < x_{\mathbf{1}} - x_{\mathbf{0}}, z_j > \\ &= \alpha_i ||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}} \end{aligned}$$

Donc $\alpha_i = \frac{(x_{\mathbf{1},i} - x_{\mathbf{0},i})}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}}$ où $||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}} \neq 0$ modulo q pour tout i, puisque $q > l \times B^{\mathbf{2}}$.
Ensuite, $\mathcal{B}$ crée un couple de clefs pour chaque $j \in \{1, \dots, l-1\}$: $(pk_{z_j}, sk_{z_j}) \leftarrow Setup(1^{\lambda})$. À ce stade, le tuple de l clés $(pk, pk_{z_{\mathbf{1}}}, \dots, pk_{z_{l-\mathbf{1}}})$ forme une clé publique maître conforme pour $\mathcal{A}$. Il calcule enfin pour tout $i \in \{1, \dots, l\}$, $pk_i = pk^{\alpha_i} \prod_j pk_{z_j}^{\lambda_{i,j}}$.
On applique donc juste un changement de base sur les pk_{z_j} pour obtenir $mpk = (pk_i)_i$. $\mathcal{B}$ construit donc bien une clé maître publique conforme à l'environnement de $\mathcal{A}$.

(9-11) Fabrication du chiffré challenge

$\mathcal{B}$ prend un bit b aléatoire.
Puis, il pose :

- $ct_{\mathbf{0}}^{*} = ct_{\mathbf{0}}$
  et
- $(ct_i^{*} = ct_{\mathbf{1}}^{\alpha_i} \times \mathcal{E}.E(\gamma_i, 0; r) \times \mathcal{E}.E(\mathbf{1}_H, x_{b,i}; r))_i$

où $C_t = (ct_{\mathbf{0}}, (ct_i)_i)$ est le chiffré de 0 ou a fourni par $\mathcal{C}$.
$\mathcal{B}$ donne $C_t^{*} = (ct_{\mathbf{0}}^{*}, (ct_i^{*})_i)$ à $\mathcal{A}$.

(12) Fabrication des clefs secrètes dérivées

$\mathcal{B}$ fabrique les clefs secrètes dérivées que lui demande $\mathcal{A}$. Par hypothèse, $\mathcal{A}$ ne peut demander des clefs secrètes que pour des vecteurs y tels que $< x_{\mathbf{0}}, y >=< x_{\mathbf{1}}, y >$ soit $< x_{\mathbf{1}} - x_{\mathbf{0}}, y >= 0$ donc $y \in (x_{\mathbf{1}} - x_{\mathbf{0}})^{\perp}$.

Pour fabriquer la clef correspondant à un vecteur y, $\mathcal{B}$ calcule :

$$\begin{aligned} sk_y =< y, msk >= \sum_j y_i sk_i &= \sum_j y_i \alpha_i sk + \sum_i \sum_j y_i \lambda_{i,j} sk_{z_j} \\ &= \sum_j y_i \frac{(x_{\mathbf{1},i} - x_{\mathbf{0},i})}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}} sk + \sum_i \sum_j y_i \lambda_{i,j} sk_{z_j} \end{aligned}$$

Comme $y \in (x_{\mathbf{1}} - x_{\mathbf{0}})^{\perp}$, le premier terme de la somme est nul, $\mathcal{B}$ n'a donc pas besoin de connaître sk pour pouvoir dériver la clé secrète. On a donc : $sk_y = \sum_j (\sum_i y_i \lambda_{i,j}) sk_{z_j}$.
De cette façon, $\mathcal{B}$ simule bien l'environnement de $\mathcal{A}$.

(13-15) Réponse de $\mathcal{B}$ au challenge

Si $\mathcal{A}$ trouve la valeur de b, $\mathcal{B}$ répond 0 à $\mathcal{C}$. Sinon, $\mathcal{B}$ répond 1.

1. Supposons que $\mathcal{C}$ a chiffré 0. Dans ce cas, on a pour tout i :

$$\begin{aligned}
ct_i^* &= ct_{\mathbf{1}}^{\alpha_i} \times \mathcal{E}.E(\gamma_i, 0; r) \times \mathcal{E}.E(1_H, x_{b,i}; r)\\
&= \mathcal{E}.E(pk, 0; r)^{\alpha_i} \times \mathcal{E}.E(\gamma_i, 0; r) \times \mathcal{E}.E(1_H, x_{b,i}; r)\\
&= \mathcal{E}.E(pk, 0; r)^{\alpha_i} \times \mathcal{E}.E(\gamma_i, x_{b,i}; r)\\
&= (\textstyle\prod_{k=\mathbf{1}}^{\alpha_i} \mathcal{E}.E(pk, 0; r)) \times \mathcal{E}.E(\gamma_i, x_{b,i}; r)\\
&= \mathcal{E}.E(pk^{\alpha_i}\gamma_i, 0 + x_{b,i}; r) \qquad \text{grâce à la propriété LCH}\\
&= \mathcal{E}.E(pk_i, x_{b,i}; r)
\end{aligned}$$

Ainsi, Ct^* dans ce cas correspond à un chiffré de la forme :

$$Ct^* = \mathcal{F}.Encrypt(mpk, x_b)$$

$\mathcal{A}$ réussit alors à distinguer x_b avec un avantage ϵ.

2. Supposons que $\mathcal{C}$ a chiffré a. Dans ce cas, on a pour tout i :

$$\begin{aligned}
ct_i^* &= ct_{\mathbf{1}}^{\alpha_i} \times \mathcal{E}.E(\gamma_i, 0; r) \times \mathcal{E}.E(1_H, x_{b,i}; r)\\
&= \mathcal{E}.E(pk, a; r)^{\alpha_i} \times \mathcal{E}.E(\gamma_i, 0; r) \times \mathcal{E}.E(1_H, x_{b,i}; r)\\
&= \mathcal{E}.E(pk, a; r)^{\alpha_i} \times \mathcal{E}.E(\gamma_i, x_{b,i}; r)\\
&= (\textstyle\prod_{k=\mathbf{1}}^{\alpha_i} \mathcal{E}.E(pk, a; r)) \times \mathcal{E}.E(\gamma_i, x_{b,i}; r)\\
&= \mathcal{E}.E(pk^{\alpha_i}\gamma_i, a \times \alpha_i + x_{b,i}; r) \qquad \text{grâce à la propriété LCH}\\
&= \mathcal{E}.E(pk_i, s_i; r)
\end{aligned}$$

Développons s_i :

$$\begin{aligned}
s_i = a \times \alpha_i + x_{b,i} &= a \times \frac{(x_{\mathbf{1},i} - x_{\mathbf{0},i})}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}} + x_{b,i}\\
&= a \times \frac{(x_{\mathbf{1},i} - x_{\mathbf{0},i})}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}} + bx_{\mathbf{1},i} + (1-b)x_{\mathbf{0},i}\\
&= (x_{\mathbf{1},i} - x_{\mathbf{0},i}) \times \frac{a}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}} + b(x_{\mathbf{1},i} - x_{\mathbf{0},i}) + x_{\mathbf{0},i}
\end{aligned}$$

université de BORDEAUX

$$
\begin{aligned}
&= (x_{\mathbf{1},i} - x_{\mathbf{0},i}) \times (\tfrac{a}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}} + b) + x_{\mathbf{0},i} \\
&= (x_{\mathbf{1},i} - x_{\mathbf{0},i}) \times u + x_{\mathbf{0},i} \qquad \text{où } u = (\tfrac{a}{||x_{\mathbf{1}} - x_{\mathbf{0}}||^{\mathbf{2}}} + b) \\
&= u x_{\mathbf{1},i} + (1-u) x_{\mathbf{0},i}
\end{aligned}
$$

Ainsi, Ct^* dans ce cas correspond à un chiffré de la forme :

$$Ct^* = \mathcal{F}.Encrypt(mpk, s)$$

où $s = ux_{\mathbf{1}} + (1-u)x_{\mathbf{0}}$. Puisque a est choisi aléatoirement, u est donc également un élément aléatoire de $\mathcal{M}$. $s = ux_{\mathbf{1}} + (1-u)x_{\mathbf{0}}$ est alors une combinaison linéaire aléatoire de $x_{\mathbf{0}}$ et $x_{\mathbf{1}}$, qui cache donc b. L'avantage de $\mathcal{A}$ sur le chiffré de ce message est alors 0.

Sécurité du schéma DDH-IP I

Jeu 1 :

$$E_i = g^{y_{\beta,i}} C^{s_i} D^{t_i}$$

On peut observer que $C_{y_\beta} = (C, D, E_1, \cdots, E_l)$ a la même distribution que dans le jeu 0, car on a :

$\{C^{s_i} D^{t_i} \mid r \xleftarrow{\$} \mathbb{Z}_q\} = \{(g^r)^{s_i}(h^r)^{t_i} \mid r \xleftarrow{\$} \mathbb{Z}_q\} = \{(g^{s_i} h^{t_i})^r \mid r \xleftarrow{\$} \mathbb{Z}_q\} = \{h_i^r \mid r \xleftarrow{\$} \mathbb{Z}_q\}$.

Jeu 2 :

On peut construire un algorithme $\mathcal{C}$ qui prend en entrée un triplet (X, Y, Z) et qui attaque DDH en utilisant $\mathcal{A}$. Pour cela, il suffit de prendre : $(h, C, D) = (X, Y, Z)$ et de construire un chiffré challenge comme dans les jeux 1 et 2. Plus spéciquement, $\mathcal{C}$ crée deux clefs *mpk* et *msk*, demande deux messages clairs y_0, y_1 à $\mathcal{A}$, tire β aléatoire dans $\{0, 1\}$ et fabrique un chiffré de y_β comme ce qui suit. $\mathcal{C}$ pose $C = Y$, $D = Z$ et calcule pour tout $i \in \{1, \ldots, l\}$, $E_i = X^{y_{\beta,i}} C^{s_i} D^{t_i}$. Le chiffré challenge est alors : $C_{y\beta} = (C, D, E_1, \cdots, E_l)$, comme précédemment. $\mathcal{A}$ répond β' et $\mathcal{C}$ retourne 1 si $\beta = \beta'$ et 0 sinon.

Sécurité du schéma DDH-IP II

Dans ce cas, $Adv_{\mathcal{C}}^{DDH}(\lambda) = |Pr(\text{C retourne 1 et le triplet est un triplet DDH}) - Pr(\text{C retourne 1 et le triplet est un triplet aléatoire})| = |Pr[S_1] - Pr[S_2]|$.
Ainsi, comme sous l'hypothèse DDH $Adv_{\mathcal{C}}^{DDH}(\lambda)$ est négligeable, $|Pr[S_2] - Pr[S_1]|$ est aussi négligeable.

$z_\beta = (y_{\beta,1} + \omega \cdot r' \cdot t_1, \ldots, y_{\beta,l} + \omega \cdot r' \cdot t_l) = y_\beta + \omega \cdot r' \cdot t \in \mathbb{Z}_q^l$.

Il suffit donc de montrer que z_β ne révèle rien sur β. Pour cela, on pose $y = y_0 - y_1$ la différence entre les deux messages que $\mathcal{A}$ donne au début de l'expérience et on prend une base de $y^\perp$. On pose alors $X_{top} \in \mathbb{Z}_q^{(l-1)\times l}$, la matrice dont les lignes sont les vecteurs de cette base. Soit $y' \in \mathbb{Z}_q^l \backslash y^\perp$. On construit alors X la matrice suivante :

$$X = \begin{bmatrix} X_{top} \\ \hline y'^T \end{bmatrix} \in \mathbb{Z}_q^{l\times l}$$

Par construction, $X_{top} \cdot y = 0_{\mathbb{Z}^l}$ donc $X_{top} \cdot y_0 = X_{top} \cdot y_1$ et $X_{top} \cdot z_\beta$ est alors indépendant de β et n'en révèle aucune information. La dernière ligne de $X_{top} \cdot z_\beta$ est la suivante, dans $\mathbb{Z}_q$:

$$< y', z_\beta > = < y', y_\beta + \omega \cdot r' \cdot t > = < y', u > + \omega \cdot r' \cdot < y', t >$$

Sécurité du schéma DDH-IP III

Soit une clé maître privée $msk_{\mathbf{0}} = (s_{\mathbf{0}}, t_{\mathbf{0}}) = ((s_{\mathbf{0},\mathbf{1}}, \ldots, s_{\mathbf{0},l}), (t_{\mathbf{0},\mathbf{1}}, \ldots, t_{\mathbf{0},l})) \in \mathbb{Z}_q^l \cdot \mathbb{Z}_q^l$. Soit une requête de clé privée, le vecteur x. On a alors $sk_x = (< s_{\mathbf{0}}, x >, < t_{\mathbf{0}}, x >)$. Construisons l'ensemble de tous les couples (s, t) pour lesquels on a des clés secrètes égales à sk_x (toujours pour la requête x). En réalité, on s'intéresse principalement à t (s dépendra de t) :
$x \in y^{\perp}$ donc $< x, y >= 0$ et on a $< t, y >=< t_{\mathbf{0}}, y > +\mu < x, y >$ pour tout μ dans $\mathbb{Z}_q$ donc on obtient l'ensemble $\{t \in \mathbb{Z}_q^l \mid t = t_{\mathbf{0}} + \mu y \mod q\}$.

Ainsi, $\omega r' t < y', t > \mod q = \omega r' < y', t_{\mathbf{0}} + \mu y > \mod q$
$= \omega r'(< y', t_{\mathbf{0}} > +\mu < y', y >) \mod q$
Par construction, $< y', y >\neq 0$ donc la distribution de $\mu < y', y >$ est uniforme. r' est choisi uniformément dans $\mathbb{Z}_q$ donc il est non nul avec probabilité valant $1 - \frac{\mathbf{1}}{q}$. Finalement, $\omega \cdot r' \cdot t < y', t > \mod q$ est donc uniforme.
Donc $Pr(S_{\mathbf{2}}) = \frac{\mathbf{1}}{\mathbf{2}}$.

En cumulant les majorations obtenues dans les différents jeux, on obtient :

$Adv_{\mathcal{A}}^{IND-FE-CPA} = |Pr(S_{\mathbf{0}}) - 1/2| = |Pr(S_{\mathbf{1}}) - 1/2|$ donc

$Adv_{\mathcal{A}}^{IND-FE-CPA} = |Pr(S_{\mathbf{1}}) - Pr(S_{\mathbf{2}})| \leq Adv_{\mathcal{C}}^{DDH}(\lambda)$ qui est négligeable sous l'hypothèse DDH.