Traçabilité publique dans le traçage de traîtres

Sommaire

À Eurocrypt '02, Kiayias et Yung [73] ont proposé un schéma de traçage de traîtres à taux de transmission constant (noté KY dans la suite) : le taux de texte chiffré (le ratio entre la taille du chiffré et celle du clair) est 3, le taux de la clef de chiffrement (le ratio entre la taille de la clef de chiffrement et celle du clair) est 4 et le taux de la clef de l'usager (le ratio entre la taille de la clef d'usager et celle du clair) est 2.

Dans ce chapitre, nous apportons quelques propositions qui améliorent ces taux : le taux de texte chiffré est réduit au taux optimal de 1 (asymptotiquement) ; le taux de la clef de chiffrement est réduit à 1 et le taux de la clef de l'usager reste inchangé. Remarquons

que ces taux de transmission sont considérés dans le cas de plusieurs usagers, lorsque le nombre d'usagers et la taille du texte clair sont grands. Ces améliorations sont obtenues tout en préservant deux propriétés extrêmement précieuses pour un schéma de traçage de traîtres, comme dans le schéma KY :

- diffusion de données chiffrées à *clef publique*, où une tierce personne est capable d'envoyer des messages confidentiels aux abonnés ;
- traçage de traîtres efficace en *boîte noire* où la procédure de traçage peut s'effectuer sans ouvrir le décodeur pirate.

De plus, nous proposons une nouvelle fonctionnalité intéressante : la *traçabilité publique*. Dans les schémas précédents, on a besoin d'informations privées (importantes) pour retrouver les traîtres. Dans notre schéma, le centre peut publier des informations de telle sorte que n'importe qui est capable d'exécuter la procédure de traçage, au moins dans la phase la plus coûteuse d'interaction avec le décodeur pirate.

8.1 Hypothèses calculatoires

Dans ce qui suit, on utilise le terme « couplage » pour signifier les couplages modifiés de Weil ou de Tate qui satisfont les propriétés d'une application bilinéaire admissible. Dans cette section, on considère des groupes $\mathcal{G}_1$ et $\mathcal{G}_2$ sur lesquels il existe un couplage $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$. .

8.1.1 Hypothèses classiques et variantes

Nous rappelons quelques problèmes bien connus dans les groupes dotés d'un couplage , tels que le problème calculatoire bilinéaire de Diffie-Hellman. Nous proposons également quelques nouveaux problèmes.

Tout d'abord, on rappelle quelques problèmes classiques dans le groupe $\mathcal{G}_1$.

CDH – le problème calculatoire Diffie-Hellman dans $\mathcal{G}_1$:
Étant donné (P, aP, bP), où $a, b \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, calculer abP.

CBDH^1 – le problème calculatoire Diffie-Hellman bilinéaire dans $\mathcal{G}_1$:
Étant donné (P, aP, bP, cP), où $a, b, c \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, calculer $abcP$.

DBDH^1 – le problème décisionnel Diffie-Hellman bilinéaire dans $\mathcal{G}_1$:
Étant donné (P, aP, bP, cP, U), où $a, b, c \overset{R}{\leftarrow} \mathbb{Z}_q^\star$ et $U \overset{R}{\leftarrow} \mathcal{G}_1$, décider si $U = abcP$.

Nous introduisons maintenant deux versions modifiées des deux problèmes bilinéaires Diffie-Hellman. Elles sont, en effet, des cas particuliers, où $b = c$. Nous prouvons donc quelques relations entre ces problèmes et le problème usuel CDH .

$\mathsf{CBDH}^1\text{–}\mathsf{M}$ – le problème calculatoire Diffie-Hellman bilinéaire modifié dans $\mathcal{G}_1$:
Étant donné (P, aP, bP), où $a, b \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, calculer ab^2P.

$\mathsf{DBDH}^1-\mathsf{M}$ – le problème décisionnel Diffie-Hellman bilinéaire modifié dans $\mathcal{G}_1$:
Étant donné (P, aP, bP, U), où $a, b \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ et $U \stackrel{R}{\leftarrow} \mathcal{G}_1$, décider si $U = ab^2P$.

Proposition 51 *Le problème* $\mathsf{CBDH}^1-\mathsf{M}$ *est au moins aussi difficile que le problème* CBDH^1*, qui est au moins aussi difficile que le problème* CDH *:*

$$(\mathsf{Succ}_{\mathcal{G}_1}^{\mathsf{CBDH}^1-\mathsf{M}}(t))^2 \leq \mathsf{Succ}_{\mathcal{G}_1}^{\mathsf{CBDH}^1}(2t + 3T_e) \leq \mathsf{Succ}_{\mathcal{G}_1}^{\mathsf{CDH}}(2t + 4T_e),$$

où T_e *est le temps de calcul d'une exponentielle dans le groupe* $\mathcal{G}_1$*.*

Preuve. On montre d'abord l'inégalité de gauche. Considérons un attaquant $\mathcal{A}$ qui peut résoudre le problème $\mathsf{CBDH}^1-\mathsf{M}$ avec au moins une probabilité de succès ε. On peut alors construire un algorithme $\mathcal{B}$ qui résout le problème CBDH^1 avec au moins une probabilité de succès ε^2. Une instance aléatoire $(P, A = aP, B = bP))$ du problème CBDH^1 est donnée comme entrée à $\mathcal{B}$. $\mathcal{B}$ trouve la solution $D = abcP$ en interagissant avec $\mathcal{A}$ de la manière suivante :

Étape 1 : $\mathcal{B}$ calcule $X_1 = A - B = (a - b)P$ et envoie αC et X_1 à $\mathcal{A}$ où α est un aléa $\alpha \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$. $\mathcal{A}$ retourne donc $Y_1 = \alpha c(a - b)^2P$ avec probabilité ε.

Étape 2 : $\mathcal{B}$ calcule $X_2 = A + B = (a + b)P$ et envoie βC et X_2 à $\mathcal{A}$ où β est un aléa $\beta \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$. $\mathcal{A}$ retourne donc $Y_2 = \beta c(a + b)^2P$ avec probabilité ε.

Étape 3 : L'algorithme $\mathcal{B}$ retourne

$$\begin{aligned} D &= (4^{-1} \bmod q)(\beta^{-1}Y_2 - \alpha^{-1}Y_1)P \\ &= 4^{-1}c\left((a + b)^2 - (a - b)^2\right)P = 4^{-1}c(4ab)P \\ &= abcP. \end{aligned}$$

Comme a et b sont deux aléas indépendants, $a + b$ et $a - b$ le sont aussi. Par conséquent, $\mathcal{B}$ retourne une solution correcte avec une probabilité plus grande que ε^2.

On démontre maintenant l'inégalité de droite.

Considérons un attaquant $\mathcal{A}$ qui peut résoudre le problème CBDH^1 avec une probabilité de succès ε. On peut alors construire un algorithme $\mathcal{B}$ qui résout le problème CDH avec une probabilité de succès ε. Une instance aléatoire $(P, A = aP, B = bP, C = cP))$ du problème CDH est donnée comme entrée à $\mathcal{B}$. $\mathcal{B}$ trouve la solution $D = abP$ en interagissant avec $\mathcal{A}$, de la manière suivante :

Étape 1 : $\mathcal{B}$ choisit aléatoirement $c \stackrel{R}{\leftarrow} \mathbb{Z}_q$ et envoie $(P, A, B, C = cP)$ à $\mathcal{A}$. $\mathcal{A}$ retourne donc $U = abcP$ avec probabilité ε.

Étape 2 : $\mathcal{B}$ retourne $D = c^{-1}U$.

On voit facilement que $\mathcal{B}$ retourne une solution correcte avec une probabilité plus grande que ε. □

8.1.2 Nouveaux problèmes fondés sur les couplages

Nous rappelons maintenant les problèmes Diffie-Hellman bilinéaires dans les groupes $\mathcal{G}_1$ et $\mathcal{G}_2$.

CBDH^2 – Le problème calculatoire Diffie-Hellman bilinéaire :
Étant donné (P, aP, bP, cP), où $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$.
Calculer g^{abc}.

DBDH^2 – Le problème décisionnel Diffie-Hellman bilinéaire :
Étant donné (P, aP, bP, cP, Z), où $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ et $Z \stackrel{R}{\leftarrow} \mathcal{G}_2$. Décider si $Z = g^{abc}$.

$\mathsf{CBDH}^2{-}\mathsf{E}$ – Le problème calculatoire Diffie-Hellman bilinéaire étendu :
Étant donné (P, aP, bP, cP, ab^2P), où $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$. Calculer g^{cb^2}.

$\mathsf{DBDH}^2{-}\mathsf{E}$ – Le problème décisionnel Diffie-Hellman bilinéaire étendu :
Étant donné $(P, aP, bP, cP, ab^2P, Z)$, où $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$ et $Z \stackrel{R}{\leftarrow} \mathcal{G}_2$.
Décider si $Z = g^{cb^2}$.

Nous introduisons également une variante du problème CBDH^2 avec le but de conforter la difficulté du problème de $\mathsf{CBDH}^2{-}\mathsf{E}$

$\mathsf{CBDH}^2{-}\mathsf{V}$ – Une variante du problème calculatoire Diffie-Hellman bilinéaire :
Étant donné $(P, aP, bP, cP, a(a^2 - b^2)P, b(a^2 - b^2)P)$, où $a, b, c \stackrel{R}{\leftarrow} \mathbb{Z}_q^\star$.
Calculer g^{abc}.

Proposition 52 *Le problème* $\mathsf{CBDH}^2{-}\mathsf{E}$ *est au moins aussi difficile que le problème* $\mathsf{CBDH}^2{-}\mathsf{V}$ *:*

$$(\mathsf{Succ}^{\mathsf{CBDH}^2-\mathsf{E}}_{e,\mathcal{G}_1,\mathcal{G}_2}(t))^2 \leq \mathsf{Succ}^{\mathsf{CBDH}^2-\mathsf{V}}_{e,\mathcal{G}_1,\mathcal{G}_2}(t + T_e),$$

où T_e est le temps de calcul d'une exponentielle dans le groupe $\mathcal{G}_2$.

Preuve. Considérons un attaquant $\mathcal{A}$ qui peut résoudre le problème $\mathsf{CBDH}^2{-}\mathsf{E}$ avec une probabilité de succès ε. On peut alors construire un algorithme $\mathcal{B}$ qui résout le problème $\mathsf{CBDH}^2{-}\mathsf{V}$ avec une probabilité de succès ε^2. Une instance aléatoire $(P, A = aP, B = bP, C = cP, U = a(a^2-b^2)P, V = b(a^2-b^2)P)$ du problème $\mathsf{CBDH}^2{-}\mathsf{V}$ est donnée comme entrée à $\mathcal{B}$. $\mathcal{B}$ trouve la solution $Z = g^{abc}$, en interagissant avec $\mathcal{A}$, de la manière suivante :

Étape 1 : $\mathcal{B}$ calcule $xP = A + B = (a + b)P$, $yP = A - B = (a - b)P$, $zP = cP$ et $xy^2P = U - V$. $\mathcal{B}$ envoie (xP, yP, zP, xy^2P) à $\mathcal{A}$. $\mathcal{A}$ retourne $Z_1 = g^{zy^2} = g^{c(a-b)^2}$ avec probabilité ε.

Étape 2 : $\mathcal{B}$ calcule $yx^2P = U + V$ et envoie (yP, xP, zP, yx^2P) à $\mathcal{A}$. $\mathcal{A}$ retourne $Z_2 = g^{zx^2} = g^{c(a+b)^2}$ avec probabilité ε.

Étape 3 : $\mathcal{B}$ retourne $Z = (Z_2/Z_1)^{4^{-1} \bmod q}$.

Du fait que x, y, z sont des aléas indépendants, on peut facilement rendre aléatoires les instances envoyées à $\mathcal{A}$. On voit donc que $\mathcal{B}$ retourne une solution correcte avec probabilité ε^2. □

Problèmes mixtes

Nous introduisons quelques nouveaux problèmes qui concernent des éléments aussi bien de $\mathcal{G}_1$ que de $\mathcal{G}_2$.

MCDH – le problème calculatoire Diffie-Hellman mixte :
Étant donné (P, aP, a^2P, g^b), où $a, b \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, et $g = e(P, P)$.
Calculer g^{ba^2}.

MDDH – le problème décisionnel Diffie-Hellman mixte :
Étant donné (P, aP, a^2P, g^b, Z), où $a, b \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, $Z \overset{R}{\leftarrow} \mathcal{G}_2$, Et $g = e(P, P)$.
Décider si $Z = g^{ba^2}$.

Convention 53 *Pour un problème X, on appelle « l'hypothèse X » l'hypothèse selon laquelle X est difficile à résoudre. Autrement dit, il n'existe pas d'algorithme polynomial probabiliste qui peut résoudre X avec une probabilité non négligeable.*

8.2 Schéma à deux usagers

8.2.1 Discussion sur l'utilisation des hypothèses

Nous avons introduit plusieurs nouveaux problèmes qui servent aux analyses de sécurité. Avant d'étudier ces problèmes de sécurité, résumons d'abord notre choix d'utilisation des hypothèses selon le niveau de sécurité.

Traçage de traîtres

Considérons d'abord la sécurité sémantique du schéma de chiffrement, *i.e.* la confidentialité vis-à-vis des non-abonnée. Dans le modèle de l'oracle aléatoire, la sécurité est assurée sous l'hypothèse MCDH. Dans le modèle standard, la sécurité est assurée sous une hypothèse plus forte : l'hypothèse MDDH.

En ce qui concerne le traçage de traîtres, on considère d'abord une propriété plus faible : la *non-incrimination*, *i.e.* un traître ne peut pas incriminer un innocent. On montre que, sous l'hypothèse CDH, notre schéma atteint cette propriété. La propriété de traçabilité de traîtres, *i.e.* dénoncer un véritable traître sans se tromper, est assurée sous l'hypothèse $\mathsf{DBDH}^1{-}\mathsf{M}$.

En un mot, notre schéma atteint les niveaux de sécurité classiques de traçage de traîtres sous les hypothèses MDDH et $\mathsf{DBDH}^1{-}\mathsf{M}$.

Traçabilité publique

Notre schéma apporte la nouvelle propriété intéressante de *traçabilité publique*. Cependant, pour l'obtenir, nous avons besoin d'hypothèses plus fortes car l'attaquant dispose de beaucoup plus d'informations.

Pour la sécurité sémantique du schéma de chiffrement, dans le modèle de l'oracle aléatoire, l'hypothèse $\mathsf{CBDH}^2-\mathsf{E}$, une petite extension de l'hypothèse classique CBDH^2 (voir la proposition 52), est exigée. Dans le modèle standard, la sécurité est fondée sur l'hypothèse $\mathsf{DBDH}^2-\mathsf{E}$.

Au sujet du traçage de traîtres, le traçage d'un traître est assurée sous l'hypothèse $\mathsf{DBDH}^1-\mathsf{M}$ (ce qui implique la non-incrimination).

Conclusion

Notre schéma apporte la propriété de traçabilité publique fondée essentiellement sur trois nouvelles hypothèses : MDDH pour la sécurité du chiffrement, $\mathsf{DBDH}^1-\mathsf{M}$ pour la propriété de traçage de traître, et $\mathsf{DBDH}^2-\mathsf{E}$ pour la traçabilité publique en boîte noire.

8.2.2 Schéma de Kiayias-Yung

Notre construction est inspirée du schéma de Kiayias et Yung [73], qui est, à son tour, considéré comme un cas particulier du schéma de Boneh et Franklin [22]. Nous allons modifier le schéma KY pour rendre possible l'utilisation des couplages.

Tout d'abord, rappelons le schéma KY.

Implémentation : Étant donné un paramètre de sécurité $\kappa \in \mathbb{Z}$, le schéma de traçage de traîtres fonctionne de la manière suivante :

Étape 1 : Générer un nombre premier q de κ bits et un groupe $\mathcal{G}$ d'ordre q. Choisir aléatoirement un générateur $g \in \mathcal{G}$.

Étape 2 : Prendre des éléments aléatoires $a, z \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, et poser $Q = g^a$, $Z = g^z$.

Clef secrète du centre : le couple (a, z)

Clef de chiffrement : $\mathsf{pk} = (g, Q, Z)$

Clef d'usager : L'usager u_b (pour $b \in \{0, 1\}$, car on se focalise dans un premier temps sur le cas à deux usagers), est associé à une « représentation » $k_b = (\alpha_b, \beta_b)$ de g^z dans la base (g, g^a), *i.e.* l'autorité choisit deux vecteurs (α_0, β_0) et (α_1, β_1) dans $\mathbb{Z}_q^2$ tels que $\alpha_b + a\beta_b = z \bmod q$ pour tout $b \in \{0, 1\}$. Ces deux vecteurs sont choisis pour être linéairement indépendants. L'ensemble des clefs possibles est :

$$\mathcal{K}_{\mathsf{pk}} = \{(\alpha, \beta) | \alpha + a\beta = z \bmod q\}.$$

Algorithme de chiffrement : L'algorithme de chiffrement génère un aléa $k \in \mathbb{Z}_q$ et retourne un texte chiffré (c_1, c_2, d) dans $\mathcal{G}^3$: on a le texte clair m, supposé être dans le groupe $\mathcal{G}$, le centre calcule $C = (c_1 = g^k, c_2 = Q^k, d = m \cdot Z^k)$. On dit que le triplet $(c_1, c_2, d) \in \mathcal{G}^3$ est un texte chiffré valide s'il existe $k \in \mathbb{Z}_q$ tel que $c_1 = g^k$ et $c_2 = Q^k$. Sinon, le texte chiffré est invalide.

Algorithme de déchiffrement : Sur le texte chiffré (c_1, c_2, d), l'usager u_b calcule :

$$Z^k = c_1^\alpha \cdot c_2^\beta \quad \text{et} \quad m = d/Z^k.$$

8.2.3 Notre construction

Dans cette section, nous montrons comment on peut utiliser les applications bilinéaires pour améliorer ce schéma. Plus précisément, nous introduisons la notion de traçage de traîtres à clef publique avec « *sous-clef de déchiffrement équivalente* » : l'autorité génère pour chaque usager une clef et une *sous-clef de déchiffrement équivalente* qui lui correspond. L'autorité conserve la clef d'usager et ne donne à l'usager que la seule sous-clef (de déchiffrement équivalente) qui est suffisante pour le déchiffrement. La clef d'usager sera ultérieurement utilisée pour le traçage.

Implémentation : Étant donné un paramètre de sécurité $\kappa \in \mathbb{Z}$, le schéma de traçage de traîtres fonctionne de la façon suivante :

Étape 1 : Générer un nombre premier q de κ bits et deux groupes $\mathcal{G}_1$ et $\mathcal{G}_2$ d'ordre q ainsi qu'une application bilinéaire admissible $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$. Choisir aléatoirement un générateur $P \xleftarrow{R} \mathcal{G}_1$ et poser $g = e(P, P)$, un générateur du groupe $\mathcal{G}_2$.

Étape 2 : Choisir des éléments aléatoires $a, z \xleftarrow{R} \mathbb{Z}_q^\star$, et poser $Q = aP$, $Z = g^z$.

Étape 3 : Choisir une fonction $H : \mathcal{G}_1 \to \mathcal{M}$. L'analyse de sécurité considère H comme un oracle aléatoire ou une fonction de hachage universelle (avec le « leftover-hash-lemma » [64, 65]).

L'espace des messages est $\mathcal{M} = \{0,1\}^{\kappa'}$ (la taille κ' du masque dépend du modèle utilisé : fonction de hachage universelle ou oracle aléatoire). L'espace des textes chiffrés est $\mathcal{G}_1^\star \times \mathcal{G}_1^\star \times \{0,1\}^{\kappa'}$. Les paramètres du système sont $\mathsf{params} = (q, \mathcal{G}_1, \mathcal{G}_2, e, P, H)$.

Clef secrète du centre : le couple (a, z).

Clef de chiffrement : le multiplet $\mathsf{pk} = (g, Q, Z)$.

Clef d'usager : l'usager u_b (pour $b \in \{0,1\}$) est associé à une « représentation » $k_b = (\alpha_b, \beta_b)$ de g^z dans la base (g, g^a). L'ensemble des clefs possibles est :

$$\mathcal{K}_{\mathsf{pk}} = \{(\alpha, \beta) | \alpha + a\beta = z \bmod q\}$$

Remarquons que l'autorité génère ces clefs pour les usagers mais il ne les leur donne pas. Seule une sous-clef de déchiffrement équivalente est fournie à chaque usager. Cette sous-clef équivalente est décrite ci-dessous.

Sous-clef de déchiffrement équivalente : l'usager u_b (pour $b \in \{0,1\}$) reçoit une sous-clef de déchiffrement équivalente $\Pi(k_b) = (\alpha_b, \pi_b = \beta_b P)$. L'ensemble de ces clefs est :

$$\Pi_{\mathsf{pk}} = \{(\alpha, \pi = \beta P) | (\alpha, \beta) \in \mathcal{K}_{\mathsf{pk}}\}.$$

Algorithme de chiffrement : L'algorithme de chiffrement génère un aléa $k \in \mathbb{Z}_q$ et retourne un texte chiffré (c_1, c_2, d) dans $\mathcal{G}_1 \times \mathcal{G}_1 \times \mathcal{M}$: sur le texte clair $m \in \mathcal{M}$; le centre calcule $C = (c_1 = kP, c_2 = k^2 Q, d = m \oplus H(Z^{k^2}))$. On dit que $(c_1, c_2, d) \in \mathcal{G}_1 \times \mathcal{G}_1 \times \mathcal{M}$ est un texte chiffré valide s'il existe $k \in \mathbb{Z}_q$ tel que $c_1 = kP$ et $c_2 = k^2 Q$. Sinon, il est dit invalide.

Algorithme de déchiffrement : Sur un texte chiffré (c_1, c_2, d), l'usager u_b calcule, grâce à sa sous-clef de déchiffrement équivalente $\Pi(k_b) = (\alpha_b, \pi_b)$,

$$Z^{k^2} = e(\alpha_b c_1, c_1) \cdot e(\pi_b, c_2) \quad \text{et} \quad m = d \oplus H(Z^{k^2}).$$

8.2.4 Raisonnement

Tout d'abord, il est naturel de se poser la question : « Peut-on chiffrer le message par $c_1 = kP$, $c_2 = kQ$ et $d = m \oplus H(Z^k)$, et donner à chaque usager une clef $k_b = (\alpha_b, \beta_b)$ (telle que $\alpha_b + a\beta_b = z \bmod q$) ? ». Ce schéma serait plus simple et serait, en effet, une transformation directe du schéma KY en utilisant les couplages. Cependant, de la même façon que l'attaque contre le schéma TSZ, présentée dans le chapitre précédent, l'attaquant pourrait profiter des propriétés des couplages pour casser le schéma. En effet, même s'il ne peut pas produire une nouvelle clef, il peut produire et distribuer un décodeur anonyme : $(X = \alpha P - uQ, Y = \beta P + uP)$, où u est aléatoirement choisi dans $\mathbb{Z}_q$. Alors, n'importe qui peut utiliser ce décodeur pour retrouver $Z^k = e(X, c_1) \cdot e(Y, c_2)$ et comme u est aléatoirement choisi, l'autorité ne peut pas tracer le traître.

Dans notre schéma, nous prouvons l'impossibilité d'un tel attaquant : l'usager ne dispose pas de clef de la forme (α, β), mais seulement une sous-clef de déchiffrement équivalente de forme $(\alpha, \beta P)$. Ceci entraîne que même si deux usagers s'entendent pour tricher, ils ne peuvent rien apprendre de la clef secrète (a, z). On va voir que cette propriété est primordiale pour améliorer le résultat dans le cas de plusieurs usagers.

8.2.5 Sécurité du schéma de chiffrement

Avant de considérer les propriétés spécifiques du traçage de traîtres, nous étudions la sécurité du schéma de chiffrement. Nous montrons que si on considère la fonction H comme un oracle aléatoire, la sécurité sémantique du chiffrement sera fondée sur l'hypothèse MCDH et, si on considère la fonction H comme aléatoirement choisie dans une famille de fonctions de hachage universelles [64, 65], la sécurité sémantique du chiffrement sera fondée sur l'hypothèse MDDH.

Théorème 54 *Considérons H comme un oracle aléatoire. Le schéma de chiffrement est sémantiquement sûr sous l'hypothèse* MCDH.

Preuve. Considérons un attaquant $\mathcal{A}$ contre le schéma de chiffrement, selon une attaque IND-CPA. Supposons que $\mathcal{A}$ peut avoir un avantage ε, alors on peut résoudre le problème MCDH avec une probabilité de succès ε/q_H, où q_H est le nombre de requêtes que $\mathcal{A}$ soumet à l'oracle H. En effet, on va construire un algorithme $\mathcal{B}$ qui, en utilisant $\mathcal{A}$ comme un sous-programme, résout le problème MCDH.

Une instance aléatoire du problème MCDH $(P, A = kP, B = k^2P, C = g^z))$ est donnée à $\mathcal{B}$ et $\mathcal{B}$ en trouve la solution $D = g^{zk^2}$ de la manière suivante :

Implémentation : $\mathcal{B}$ choisit aléatoirement a, définit la clef publique $\mathsf{pk} = (g, Q = aP, Z = C)$ et transmet pk à $\mathcal{A}$.

Requête-H : $\mathcal{B}$ maintient une liste $\mathcal{H}$-List pour répondre aux requêtes de l'oracle H à l'attaquant $\mathcal{A}$.

Challengeur : Quand $\mathcal{A}$ retourne deux candidats m_0, m_1, $\mathcal{B}$ prend un élément aléatoire $d \in \mathcal{M}$ et transmet (A, aB, d) comme challenge à $\mathcal{A}$. Ceci est une simulation parfaite sauf si g^{zk^2} a déjà été demandé à l'oracle H. Dans cette simulation, $\mathcal{A}$ a un avantage nul car d est indépendant de m_0, m_1.

Choix : Quand l'attaquant $\mathcal{A}$ devine $b' \in \{0, 1\}$, l'algorithme $\mathcal{B}$ prend aléatoirement (X, h) dans $\mathcal{H}$-List et retourne X.

On voit facilement que l'algorithme $\mathcal{B}$ retourne une solution correcte $X = D$ avec probabilité ε/q_H : la seule façon pour $\mathcal{A}$ d'avoir un certain avantage sur le schéma est de demander D à H. □

Théorème 55 *Considérons H une fonction aléatoirement choisie dans une famille de fonctions de hachage universelles. Le schéma est sémantiquement sûr sous l'hypothèse* MDDH.

Preuve. Considérons un attaquant $\mathcal{A}$ contre le schéma de chiffrement, selon une attaque IND-CPA. Supposons que $\mathcal{A}$ peut avoir un avantage ε, alors on peut résoudre le problème MDDH avec probabilité $\varepsilon/2$. En effet, on va construire un algorithme $\mathcal{B}$ qui, en utilisant $\mathcal{A}$ comme un sous-programme, résout le problème MDDH.

Une instance aléatoire du problème MDDH $(P, A = kP, B = k^2P, C = g^z, U))$ où U, qui est soit la solution du problème MCDH soit aléatoirement choisi dans $\mathcal{G}_2$, est donnée à $\mathcal{B}$. $\mathcal{B}$ distingue ces deux cas de la façon suivante :

Implémentation : $\mathcal{B}$ choisit aléatoirement a, pose la clef publique $\mathsf{pk} = (g, Q = aP, Z = C)$ et transmet pk à $\mathcal{A}$.

Challenger : Quand $\mathcal{A}$ retourne deux candidats m_0, m_1, $\mathcal{B}$ prend un bit aléatoire $b \in \{0, 1\}$ et transmet $(A, aB, d = m_b \oplus H(U))$ comme challenge à $\mathcal{A}$.

Choix : Quand l'attaquant $\mathcal{A}$ devine $b' \in \{0,1\}$, l'algorithme $\mathcal{B}$ retourne 1 (*i.e.* U est la solution au problème MCDH) si $b = b'$ et 0 sinon.

On observe que si U est la solution au problème MCDH, alors le texte chiffré C est un chiffrement de m_b. Sinon, comme H est aléatoirement choisie dans une famille de fonctions de hachage universelles, C est le chiffrement d'un message aléatoire, b est donc égal à b' avec probabilité 1/2. Par un argument classique, on trouve que l'attaquant $\mathcal{B}$ dispose d'un avantage de $\varepsilon/2$ pour résoudre le problème MDDH. □

8.2.6 Non-incrimination

Dans plusieurs situations, on se contente de la propriété de non-incrimination, *i.e.* un traître, en utilisant les informations de sa sous-clef de déchiffrement équivalente, ne peut fabriquer un décodeur pirate correspondant à la sous-clef d'un usager innocent. Évidemment, cette propriété est plus faible que la propriété de traçage de traîtres.

Théorème 56 *Étant donné la clef de chiffrement et une sous-clef de déchiffrement équivalente* $(\alpha, \pi) \in \Pi_{\mathsf{pk}}$*, il est calculatoirement difficile de fabriquer une autre sous-clef de déchiffrement équivalente dans* Π_{pk} *sous l'hypothèse* CDH*.*

Preuve. Considérons un attaquant $\mathcal{A}$, un traître, dont le but est d'incriminer un usager innocent. Étant donné une sous-clef de déchiffrement équivalente $(\alpha, \pi) \in \Pi_{\mathsf{pk}}$, supposons que $\mathcal{A}$ puisse construire une autre sous-clef de déchiffrement équivalente. Alors, on peut résoudre le problème inverse de Diffie-Hellman (*i.e.* à partir de (P, aP), calculer $a^{-1}P$) qui est connu pour être équivalent au problème classique Diffie-Hellman CDH. En effet, on va construire un algorithme $\mathcal{B}$ qui, en utilisant $\mathcal{A}$ comme un sous-programme, résout le problème inverse de CDH.

Une instance aléatoire du problème inverse de CDH $(P, A = aP)$ est donnée à $\mathcal{B}$ et $\mathcal{B}$ en trouve la solution $B = a^{-1}P$ de la manière suivante :

Implémentation : $\mathcal{B}$ choisit aléatoirement $\alpha \overset{R}{\leftarrow} \mathbb{Z}_q^\star$ et $\pi \overset{R}{\leftarrow} \mathcal{G}_1$, puis calcule $Z = e(P, \alpha P)e(A, \pi)$. Finalement, $\mathcal{B}$ définit la clef publique $\mathsf{pk} = (g, A, Z)$ et transmet pk ainsi que la sous-clef de déchiffrement équivalente (α, π) à $\mathcal{A}$.

Attaque : $\mathcal{A}$ retourne une autre sous-clef de déchiffrement équivalente $(\tilde{\alpha}, \tilde{\pi})$.

Solution : $\mathcal{B}$ calcule $c = \tilde{\alpha} - \alpha$ et retourne $B = (c^{-1} \bmod q)(\pi - \tilde{\pi})$.

Puisque $(\tilde{\alpha}, \tilde{\pi})$ est une nouvelle sous-clef de déchiffrement équivalente, pour tout texte chiffré (c_1, c_2, d),

$$\begin{aligned} e(\alpha c_1, c_1) \cdot e(\pi, c_2) &= e(\tilde{\alpha} c_1, c_1) \cdot e(\tilde{\pi}, c_2) \\ \iff e(\alpha k P, kP) \cdot e(\pi, ak^2 P) &= e(\tilde{\alpha} k P, kP) \cdot e(\tilde{\pi}, ak^2 P) \\ \iff e(\alpha P, P) \cdot e(\pi, aP) &= e(\tilde{\alpha} P, P) \cdot e(\tilde{\pi}, aP) \\ \iff e((\tilde{\alpha} - \alpha)P, P) &= e(\pi - \tilde{\pi}, aP). \end{aligned}$$

Du fait que $(\alpha, \pi), (\tilde{\alpha}, \tilde{\pi}) \in \Pi_{\mathsf{pk}}$, on voit que π et $\tilde{\pi}$ sont dans le groupe $\mathcal{G}_1$ généré par P. Alors, il existe b tel que $\pi - \tilde{\pi} = bP$ pour b quelconque. On obtient alors $c = \tilde{\alpha} - \alpha = ab$ et donc :

$$B = (c^{-1} \bmod q)(\pi - \tilde{\pi}) = (ab)^{-1}bP = a^{-1}P.$$

□

8.2.7 Traçage de traîtres en boîte noire

Pour des raisons pratiques, dans le traçage de traîtres, il est important que le décodeur pirate n'ait pas besoin d'être ouvert. Nous prouvons en effet que, pour notre schéma, nous pouvons faire le traçage de traîtres en boîte noire contre une coalition de deux usagers sous l'hypothèse $\mathsf{DBDH}^1-\mathsf{M}$. Nous construisons, de plus, un algorithme de traçage. Le résultat de sécurité est obtenu dans le modèle standard, où la fonction H est aléatoirement choisie dans une famille de fonctions de hachage universelles.

Théorème 57 *Étant donné une clef de chiffrement* pk *et une sous-clef de déchiffrement équivalente* $(\alpha, \pi = \beta P) \in \Pi_{\mathsf{pk}}$*, supposons qu'il existe un attaquant* $\mathcal{A}$ *qui peut produire un simulateur de déchiffrement* $\mathcal{S}$ *tel que : quand on lui donne un texte chiffré valable, il retourne correctement le texte clair correspondant ; quand on lui donne un texte chiffré « randomisé » de la forme* (kP, ak'^2P, d)*, où* $k, k' \overset{R}{\leftarrow} \mathbb{Z}_q$ *et* $d \overset{R}{\leftarrow} \mathcal{M}$*, il retourne une valeur différente de* $d \oplus H(g^{\alpha k^2 + a\beta k'^2})$ *avec probabilité* ε*. Alors, on peut résoudre le problème* $\mathsf{DBDH}^1-\mathsf{M}$ *avec probabilité* $\frac{1}{2} + \varepsilon/4$.

Preuve. On va construire un algorithme $\mathcal{B}$ qui, en utilisant $\mathcal{A}$ comme une sous-programme, résout le problème $\mathsf{DBDH}^1-\mathsf{M}$.

Une instance aléatoire du problème $\mathsf{DBDH}^1-\mathsf{M}$ $(P, A = aP, B = kP, X)$ (où $a, k \overset{R}{\leftarrow} \mathbb{Z}_q^\star$), où X est soit la solution du problème $\mathsf{DBDH}^1-\mathsf{M}$, soit aléatoirement choisi dans $\mathcal{G}_1$, est donnée à $\mathcal{B}$. $\mathcal{B}$ distingue ces deux cas de la façon suivante :

Implémentation : $\mathcal{B}$ choisit aléatoirement $\alpha \overset{R}{\leftarrow} \mathbb{Z}_q^\star$ et $\pi \overset{R}{\leftarrow} \mathcal{G}_1$, pose $Q = A$ et calcule $Z = g^\alpha \cdot e(Q, \pi)$. Puis, $\mathcal{B}$ définit la clef publique $\mathsf{pk} = (g, Q, Z)$ et transmet pk ainsi que la sous-clef de déchiffrement équivalente (α, π) à $\mathcal{A}$.

Challenge : $\mathcal{B}$ choisit aléatoirement $d \in \mathcal{M}$, construit un texte chiffré $(c_1 = B, c_2 = X, d)$ et le transmet à $\mathcal{A}$. Grâce aux choix aléatoires de (B, X, d) et de la fonction de hachage H dans la famille de fonctions de hachage universelles, le challenge (c_1, c_2, d) est un texte chiffré aléatoire.

Décision : Si l'attaquant $\mathcal{A}$ retourne $d \oplus H(e(\alpha c_1, c_1) \cdot e(\pi, X))$, $\mathcal{B}$ retourne aléatoirement 1 ou 0. Sinon $\mathcal{B}$ retourne 0 (X n'est probablement pas égal à ak^2P).

On considère le cas où $X = ak^2P$, le texte chiffré (kP, X, d) est un texte chiffré valide. Sous l'hypothèse du théorème sur l'attaquant $\mathcal{A}$, il retourne correctement le texte clair

$m = d \oplus H(e(\alpha c_1, c_1) \cdot e(\pi, X))$. Dans ce cas, l'algorithme $\mathcal{B}$ retourne aléatoirement 1 ou 0 et la probabilité que $\mathcal{B}$ donne une décision correcte est donc $1/2$.

Dans le cas où $X \neq ak^2P$, le texte chiffré (kP, X, d) est aléatoire et invalide. Par l'hypothèse du théorème sur l'attaquant $\mathcal{A}$, il retourne une valeur différente du texte clair espéré $d \oplus H(g^{\alpha k^2 + a\beta k'^2})$ avec probabilité ε. Si $\mathcal{A}$ retourne un texte clair différent de ce texte clair espéré, $\mathcal{B}$ sait répondre correctement que X n'est pas égal à ak^2P. Si $\mathcal{A}$ retourne le texte clair espéré (la probabilité de cet événement est au plus $1 - \varepsilon$), $\mathcal{B}$ répond aléatoirement 1 ou 0. Alors, la probabilité que $\mathcal{B}$ retourne une décision correcte est $\varepsilon + (1 - \varepsilon) \cdot 1/2 = 1/2 + \varepsilon/2$.

En combinant ces deux cas, on voit facilement que $\mathcal{B}$ peut retourner une décision correcte au problème $\mathsf{DBDH}^1{-}\mathsf{M}$ avec probabilité $1/2 + \varepsilon/4$. □

Intuitivement, ce théorème montre qu'un texte chiffré « randomisé », et donc invalide, et un texte chiffré valide sont indistinguables. Par conséquent, étant donné l'accès en boîte noire à un simulateur de déchiffrement $\mathcal{S}$ produit par un des deux usagers, on peut retrouver son fabriquant : on choisit aléatoirement $k, k' \overset{R}{\leftarrow} \mathbb{Z}_q^\star$ (on suppose que $k \neq k'$), et on pose $u_0 = \alpha_0 k^2 + a\beta_0 k'^2$ et $u_1 = \alpha_1 k^2 + a\beta_1 k'^2$. Avec une forte probabilité (plus grande que $1 - 2/q$), u_0 est différent de u_1. On soumet le texte chiffré randomisé (kP, ak'^2P, d) à $\mathcal{S}$. Si la sortie de $\mathcal{S}$ est d/g^{u_0} alors on peut déclarer que u_0 est le traître. Si la sortie est d/g^{u_1} alors u_1 est le traître. Sinon, *i.e.* la sortie est différente de ces deux valeurs, on conclut que les deux usagers se sont coalisés, d'où le corollaire suivant.

Corollaire 58 *Dans le schéma décrit ci-dessus, il est possible de faire le traçage de traîtres en boîte noire contre des attaquants actifs.*

8.2.8 Traçabilité publique

Maintenant, on considère une propriété additionnelle relativement intéressante : la traçabilité publique. Dans la section précédente, pour réaliser la procédure de traçage de traîtres en boîte noire, les deux clefs d'usagers (α_0, β_0) et (α_1, β_1) sont utilisées. Cependant, les sous-clefs de déchiffrement équivalentes suffisent pour le faire, voire moins : $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$ suffisent. En effet, à partir de $k, k' \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, on n'a pas vraiment besoin de u_0, u_1, mais seulement de g^{u_0} et g^{u_1} pour faire le traçage. Les valeurs de g^{u_0} et g^{u_1} peuvent être calculées de la manière suivante :

$$\begin{aligned} g^{u_0} &= e(\alpha_0 P, k^2 P) \cdot e(Q, k'^2(\beta_0 P)) \ ; \\ g^{u_1} &= e(\alpha_1 P, k^2 P) \cdot e(Q, k'^2(\beta_1 P)). \end{aligned}$$

Alors, si on rend publique les *informations de traçage* $(\alpha_0 P, \beta_0 P)$ et $(\alpha_1 P, \beta_1 P)$, n'importe qui peut faire le traçage et on peut donc alléger le travail du centre. Plus précisément, le traçage peut être délégué à une tierce personne et cette délégation n'exige aucune confiance en cette tierce partie, car les informations données n'aident ni à apprendre

des informations sur la clef secrète (a, z), ni à construire un décodeur pirate (sous une hypothèse additionnelle calculatoire).

Nous montrons ces propriétés de sécurité avec les théorèmes suivants.

Théorème 59 *Supposons que l'information de traçage soit rendue publique le schéma de chiffrement est sémantiquement sûr : dans le modèle de l'oracle aléatoire, la sécurité est fondée sur l'hypothèse* $\mathsf{CBDH}^2-\mathsf{E}$*, alors que dans le modèle standard, elle est fondée sur l'hypothèse* $\mathsf{DBDH}^2-\mathsf{E}$*.*

Preuve. On focalise sur le cas où la fonction H est aléatoirement choisie dans une famille de fonctions de hachage universelles. La preuve pour le cas où H est considérée comme un oracle aléatoire utilise la même technique que la preuve du théorème 54. Considérons un attaquant $\mathcal{A}$ contre le schéma de chiffrement selon une attaque IND-CPA. Supposons que $\mathcal{A}$ dispose d'un avantage ε, alors on peut résoudre le problème $\mathsf{DBDH}^2-\mathsf{E}$ avec probabilité $\varepsilon/2$. En effet, on construit un algorithme $\mathcal{B}$ qui, en utilisant $\mathcal{A}$ comme sous-programme, résout le problème $\mathsf{DBDH}^2-\mathsf{E}$.

Une instance aléatoire du problème $\mathsf{DBDH}^2-\mathsf{E}$ $(P, A = aP, B = kP, C = zP, D = ak^2P, U)$ (où $a, k \overset{R}{\leftarrow} \mathbb{Z}_q^\star$), où U est soit la solution du problème $\mathsf{CBDH}^2-\mathsf{E}$ soit aléatoirement choisi dans $\mathcal{G}_2$, est donnée à $\mathcal{B}$. $\mathcal{B}$ distingue ces deux cas de la façon suivante :

Implémentation : $\mathcal{B}$ choisit aléatoirement β_0, β_1 et calcule :

$$\alpha_0 P = zP - \beta_0 Q \qquad \alpha_1 P = zP - \beta_1 Q.$$

$\mathcal{B}$ définit la clef publique $\mathsf{pk} = (g, Q = A, Z = g^z = e(C, P))$. Il transmet pk et l'information de traçage $(\alpha_0 P, \beta_0 P, \alpha_1 P, \beta_1 P)$ à $\mathcal{A}$.

Challenger : Quand $\mathcal{A}$ retourne deux candidats m_0, m_1, $\mathcal{B}$ prend un bit aléatoire $b \in \{0, 1\}$ et transmet $(A, D, d = m_b \oplus H(U))$ comme challenge à $\mathcal{A}$.

Choix : Quand l'attaquant $\mathcal{A}$ devine $b' \in \{0, 1\}$, l'algorithme $\mathcal{B}$ retourne 1 (*i.e.* U est la solution au problème $\mathsf{DBDH}^2-\mathsf{E}$) si $b = b'$ et 0 sinon.

On observe que si U est la solution au problème $\mathsf{CBDH}^2-\mathsf{E}$, alors le texte chiffré C est un chiffrement de m_b. Sinon, comme H est aléatoirement choisie dans une famille de fonctions de hachage universelles, C est un chiffrement d'un message aléatoire, b est donc égal à b' avec probabilité $1/2$. Par un argument classique, on trouve que l'attaquant $\mathcal{B}$ dispose d'un avantage $\varepsilon/2$ pour résoudre le problème $\mathsf{CBDH}^2-\mathsf{E}$. □

Théorème 60 *Étant donné une clef de chiffrement* pk*, une sous-clef de déchiffrement équivalente* $(\alpha, \pi = \beta P) \in \Pi_{\mathsf{pk}}$ *et l'information publique de traçage* $(\alpha_0 P, \beta_0 P, \alpha_1 P, \beta_1 P)$*. Supposons qu'il existe un attaquant* $\mathcal{A}$ *qui peut fabriquer un simulateur de déchiffrement* $\mathcal{S}$ *tel que : quand on lui donne un texte chiffré valable, il retourne correctement le texte clair correspondant ; quand on lui donne un texte chiffré « randomisé » de la forme* (kP, ak'^2P, d)*, où* $k, k' \overset{R}{\leftarrow} \mathbb{Z}_q, d \overset{R}{\leftarrow} \mathcal{M}$*, il retourne une valeur différente de* $d \oplus H(g^{\alpha k^2 + a\beta k'^2})$ *avec probabilité* ε*. Alors, on peut résoudre le problème* $\mathsf{DBDH}^1-\mathsf{M}$ *avec probabilité* $\frac{1}{2} + \varepsilon/4$*.*

Preuve. On construit un algorithme $\mathcal{B}$ qui, en utilisant $\mathcal{A}$ comme sous-programme, résout le problème $\mathsf{DBDH}^1-\mathsf{M}$.

Une instance aléatoire du problème $\mathsf{DBDH}^1-\mathsf{M}$ $(P, A = aP, B = kP, X)$ (où $a, k \overset{R}{\leftarrow} \mathbb{Z}_q^\star$), où X est soit la solution du problème $\mathsf{DBDH}^1-\mathsf{M}$, soit aléatoirement choisi dans $\mathcal{G}_1$, est donnée à $\mathcal{B}$. $\mathcal{B}$ distingue ces deux cas de la façon suivante :

Implémentation : $\mathcal{B}$ choisit aléatoirement $\alpha_0, \beta_0, \beta_1 \overset{R}{\leftarrow} \mathbb{Z}_q^\star$ et calcule :

$$zP = \alpha_0 P + \beta_0 A \qquad \alpha_1 P = zP - \beta_1 A \qquad \pi_0 = \beta_0 P \qquad Z = e(P, zP).$$

$\mathcal{B}$ pose $Q = A$, $\mathsf{pk} = (g, Q, Z)$ et une sous-clef de déchiffrement équivalente (α_0, π_0), et l'information de traçage $(\alpha_0 P, \beta_0 P, \alpha_1 P, \beta_1 P)$. La sous-clef de déchiffrement équivalente (α_0, π_0) peut être considérée comme un élément aléatoire dans l'ensemble Π_{pk}. $\mathcal{B}$ transmet pk, la sous-clef de déchiffrement équivalente (α_0, π_0) et l'information de traçage $(\alpha_0 P, \beta_0 P, \alpha_1 P, \beta_1 P)$ à $\mathcal{A}$.

Challenge : $\mathcal{B}$ choisit aléatoirement $d \in \mathcal{M}$, construit un texte chiffré $(c_1 = B, c_2 = X, d)$ et le transmet à $\mathcal{A}$. Grâce aux choix aléatoires de (B, X, d) et de la fonction de hachage H dans la famille des fonctions de hachage universelles, le challenge (c_1, c_2, d) est un texte chiffré aléatoire.

Décision : Si l'attaquant $\mathcal{A}$ retourne $d \oplus H(e(\alpha_0 c_1, c_1) \cdot e(\pi_0, X))$, $\mathcal{B}$ retourne aléatoirement 1 ou 0. Sinon, $\mathcal{B}$ retourne 0 (X n'est probablement pas égale à ak^2P).

On considère le cas où $X = ak^2P$, le texte chiffré (kP, X, d) est valide. Sous l'hypothèse du théorème sur l'attaquant $\mathcal{A}$, il retourne correctement le texte clair $m = d \oplus H(e(\alpha_0 c_1, c_1) \cdot e(\pi_0, X))$. Dans ce cas, l'algorithme $\mathcal{B}$ retourne aléatoirement 1 ou 0 et la probabilité que $\mathcal{B}$ donne une décision correcte est donc égale à $1/2$.

Dans le cas où $X \neq ak^2P$, le texte chiffré (kP, X, d) est aléatoire et invalide. Sous l'hypothèse du théorème sur l'attaquant $\mathcal{A}$, il retourne une valeur différente du texte clair espéré $d \oplus H(g^{\alpha k^2 + a\beta k'^2})$ avec probabilité ε. Si $\mathcal{A}$ retourne un texte clair différent de ce texte clair espéré, $\mathcal{B}$ sait répondre correctement que X n'est pas égal à ak^2P. Si $\mathcal{A}$ retourne le texte clair espéré (la probabilité de cet événement est au plus $1 - \varepsilon$), $\mathcal{B}$ répond aléatoirement 1 ou 0. Alors, la probabilité que $\mathcal{B}$ retourne une décision correcte est $\varepsilon + (1 - \varepsilon) \cdot 1/2 = 1/2 + \varepsilon/2$.

En combinant ces deux cas, on voit facilement que $\mathcal{B}$ peut retourner une décision correcte au problème $\mathsf{DBDH}^1-\mathsf{M}$ avec probabilité $1/2 + \varepsilon/4$. □

8.3 Schéma à plusieurs usagers

8.3.1 Description

Considérons $C = \{\omega_1, \ldots, \omega_N\}$, un code $(N, c, \ell, \varepsilon)$ résistant aux coalitions. Ce code contient N mots de code sur l bits et sur l'alphabet $\{0, 1\}$. Il résiste aux coalitions de c

usagers, *i.e.* permet de construire un algorithme de traçage pour trouver un des traîtres avec probabilité de succès $1-\varepsilon$, à partir d'un mot de code « faisable » fabriqué par une coalition de c usagers. Les détails des codes résistants aux coalitions sont décrits dans [25]. Le cas multi-usagers est simplement une *ℓ-instantiation* de schémas de deux usagers. En effet, nous construisons un système multi-usagers en utilisant C-un code $(N, c, \ell, \varepsilon)$ résistant aux coalitions pour bien combiner ℓ systèmes à deux usagers S_1, S_2, ..., S_ℓ :

Implémentation : Étant donné les paramètres de sécurité k, c et ε :

Étape 1 : Générer un nombre premier q de k bits et deux groupes $\mathcal{G}_1$, $\mathcal{G}_2$ d'ordre q ainsi qu'une application bilinéaire admissible $e : \mathcal{G}_1 \times \mathcal{G}_1 \to \mathcal{G}_2$. Choisir aléatoirement un générateur $P \overset{R}{\leftarrow} \mathcal{G}_1$ et poser $g = e(P, P)$, qui est un générateur du groupe $\mathcal{G}_2$.

Étape 2 : Générer $C = \{\omega_1, .., \omega_N\}$, un code $(N, c, \ell, \varepsilon)$ résistant aux coalitions.

Étape 3 : Prendre aléatoirement des éléments $a, z_j \overset{R}{\leftarrow} \mathbb{Z}_q^\star$, et poser $Q = aP$, $Z_j = g^{z_j}$, pour $j = 1, \ldots, \ell$.

Étape 4 : Choisir une fonction $H : \mathcal{G}_1 \to \mathcal{M}$.

Les paramètres du système sont $\mathsf{params} = (q, \mathcal{G}_1, \mathcal{G}_2, e, P, H)$. Remarquons que ces paramètres sont communs à tous les systèmes à deux usagers S_1, S_2, ..., S_ℓ.

Clef secrète du centre : l'élément a, et l'ensemble $(z_j)_{j=1,\ldots,\ell}$

Clef de chiffrement : la combinaison des clefs de chiffrement des ℓ schémas à deux usagers : $\mathsf{pk} = (g, Q, \{Z_j = g^{z_j}\}_{j=1,\ldots,\ell})$

Clef d'usager : l'usager u_i (pour $i \in \mathbb{Z}_N$) est associé à un mot de code $\omega_i \in C$ et une « représentation » correspondante $(\alpha_{\omega_{i,j},j}, \beta_{\omega_{i,j},j})$ de g^{z_j} dans la base (g, g^a), où $\omega_{i,j}$ est le j-ième bit du mot de code ω_i. Rappelons que $(\alpha_{b,j}, \beta_{b,j})$ est une « représentation » de g^{z_j} dans la base (g, g^a). Remarquons aussi que l'autorité génère ces clefs pour les usagers mais ne les leur donne pas. Chaque usager reçoit une sous-clef de déchiffrement équivalente, décrite ci-dessous.

Sous-clef de déchiffrement équivalente : l'usager u_i (pour $i \in \mathbb{Z}_N$) reçoit une sous-clef de déchiffrement équivalente $\Pi_i = (\Pi_{\omega_{i,1},1}, \ldots, \Pi_{\omega_{i,l},\ell})$. Plus précisément, pour $j = 1, \ldots, \ell$,

$$\Pi_{\omega_{i,j},j} = (\alpha_{\omega_{i,j},j}, \pi_{\omega_{i,j},j} = \beta_{\omega_{i,j},j} P).$$

Algorithme de chiffrement : L'espace des textes clairs du système est $\mathcal{M}^\ell$. À partir de l'entrée $(m_1, \ldots, m_\ell)$, l'algorithme de chiffrement génère un aléa $k \in \mathbb{Z}_q$ et retourne un texte chiffré $(c_1, c_2, d_1, \ldots, d_\ell)$ dans $\mathcal{G}_1 \times \mathcal{G}_1 \times \mathcal{G}_2^\ell$, où : $c_1 = kP$, $c_2 = k^2 aP$ et $d_j = m_j \oplus H(Z_j^{k^2})$.

Algorithme de déchiffrement : Sur un texte chiffré $(c_1, c_2, d_1, \ldots, d_\ell)$, l'usager u_i calcule, grâce à sa sous-clef de déchiffrement équivalente, $Z_j^{k^2} = e(\alpha_{\omega_{i,j},j} c_1, c_1) \cdot e(\pi_{\omega_{i,j},j}, c_2)$ et $m_j = d_j \oplus H(Z_j^{k^2})$.

Pour l'analyse de sécurité, on peut utiliser l'hypothèse suivante, de [73] : l'*hypothèse à seuil* selon la quelle « un décodeur pirate qui retourne correctement une fraction p du texte clair de longueur λ, où $(1-p)$ est une fonction non-négligeable en λ, est inutile ». Cependant,

comme mentionné dans [73], en utilisant une transformation « *all-or-nothing* » [106, 28], cette hypothèse n'est plus nécessaire.

Proposition 61 *Etant donné ℓ systèmes de deux usagers. La coalition de deux usagers dans $(\ell - 1)$ systèmes n'affaiblit pas la sécurité du système restant.*

Preuve. Supposons qu'il existe un attaquant qui, disposant d'une information I du système S_1 et de toutes les informations des systèmes $S_2, \ldots, S_\ell$, peut avoir un avantage ε contre le système S_1 (pour un but G quelconque). Alors, on peut construire un algorithme $\mathcal{B}$ qui, disposant seulement d'une information I du système S_1, peut avoir un avantage ε contre le système S_1.

L'idée est que l'algorithme $\mathcal{B}$ peut simuler toutes les informations concernant cette coalition : étant donné les paramètres $\mathsf{params} = (q, \mathcal{G}_1, \mathcal{G}_2, e, P, H)$ et l'information publique (g, Q, Z_1), $\mathcal{B}$ peut facilement générer des informations pour les systèmes S_j, $j = 2, 3, \ldots, \ell$:
- il génère aléatoirement des éléments $\alpha_{0,j}$, $\beta_{0,j}$, $\alpha_{1,j}$;
- il calcule $Z_j = e(\alpha_{0,j}P, P) \cdot e(\beta_{0,j}P, Q)$;
- il crée les sous-clefs de déchiffrement équivalentes par $\pi_{0,j} = \beta_{0,j}P$ et $\pi_{1,j} = \alpha_{0,j}P + \beta_{0,j}Q - \alpha_{1,j}P$.

□

Cette proposition, combinée avec le fait que C est un code $(N, c, \ell, \varepsilon)$ résistant aux coalitions, conduit au corollaire suivant :

Corollaire 62 *Le schéma décrit ci-dessus est un schéma à N usagers qui supporte le traçage de traîtres.*

En ce qui concerne la traçabilité publique, avec les informations publiques, n'importe qui peut retrouver le mot de code associé au décodeur pirate. Cette phase est interactive et est donc la plus coûteuse. Cependant, l'utilisation des codes classiques résistant aux coalitions ne permet pas de faire publiquement tout le traçage car la phase de traçage effectif des traîtres à partir d'un mot de code pirate est secrète. Cette phase est une procédure *off-line* et doit être réalisée par une autorité de confiance.

8.3.2 Comparaison de l'efficacité avec le schéma de Kiayias-Yung

Dans le schéma KY, le taux de texte chiffré est 3. Alors que dans le nôtre, il est asymptotiquement égale à 1. On peut se demander pourquoi la construction ci-dessus n'est pas applicable au schéma de Kiayias et Yung ?

La raison est que dans notre schéma à deux usagers, même une coalition de deux usagers ne révèle pas d'information sur a, alors que dans le schéma de de Kiayias et Yung, une telle coalition révèle complètement a. Par conséquent, dans le cas multi-usagers de

leur schéma, si on utilise le même a pour les ℓ schémas à deux usagers, la coalition de deux usagers peut révéler cette valeur de a, puis toutes les z_i. Ceci permet évidemment une construction anonyme d'un décodeur pirate. C'est pourquoi ils ont dû utiliser une valeur différente de a dans chaque sous-schéma à deux usagers.

8.4 Conclusion

Nous avons proposé un schéma qui améliore le schéma de Kiayias et Yung selon plusieurs critères : premièrement, les taux de transmission sont réduits, « approchées » des valeurs optimales ; deuxièmement nous introduisons une fonctionnalité intéressante : la *traçabilité publique*. La possibilité d'obtenir la propriété complète de traçabilité publique dans le cas multi-usagers reste cependant un problème ouvert.