Partage de Secret

Tarik Kaced Judicaël Courant

VERIMAG, UMR 5401 CNRS - Université Grenoble

Séminaire d'équipe, 10 juin 2008

Qu'est ce que garder un secret ?

- Comment garder le secret le code d'activation de la force de frappe britannique ?
- Proposition : le brûler.
- Répond parfaitement à l'exigence de confidentialité.
- Mais : besoin de pouvoir recouvrer le secret (propriété d'intégrité).

Qu'est ce que garder un secret ?

- Comment garder le secret le code d'activation de la force de frappe britannique ?
- Proposition : le brûler.
- Répond parfaitement à l'exigence de confidentialité.
- Mais : besoin de pouvoir recouvrer le secret (propriété d'intégrité).

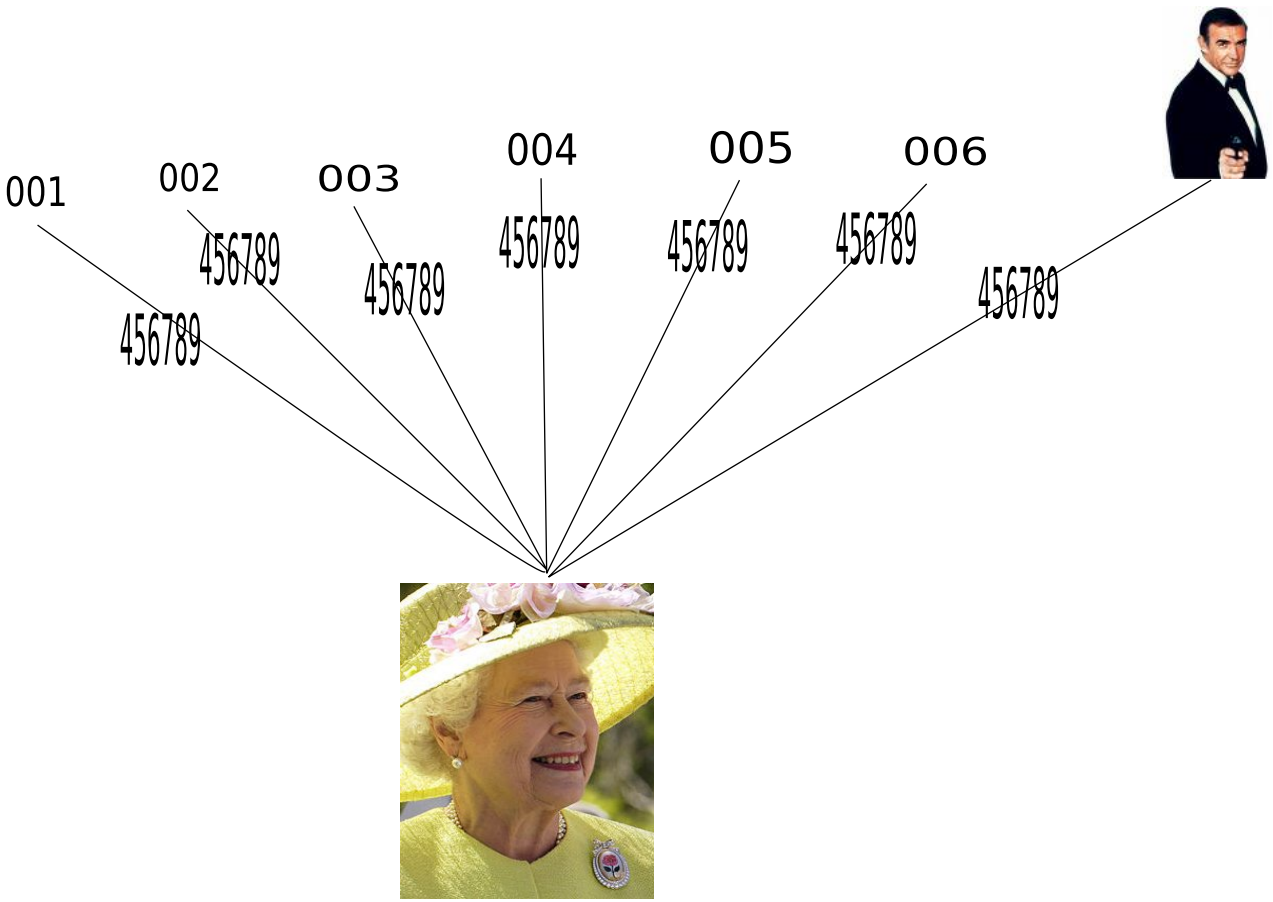
Qu'est ce que garder un secret ?

- Comment garder le secret le code d'activation de la force de frappe britannique ?
- Proposition : le brûler.
- Répond parfaitement à l'exigence de confidentialité.
- Mais : besoin de pouvoir recouvrer le secret (propriété d'intégrité).

Qu'est ce que garder un secret ?

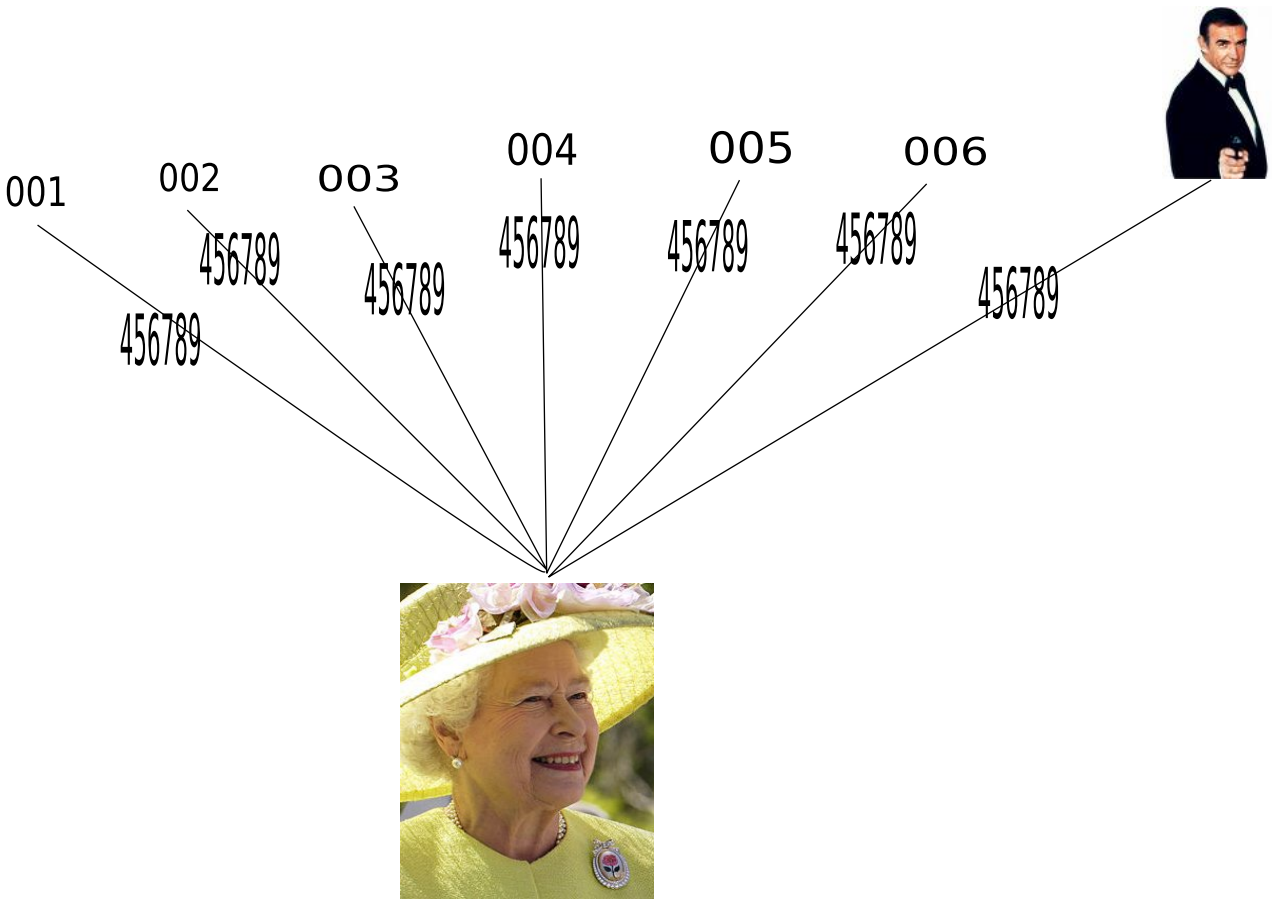
- Comment garder le secret le code d'activation de la force de frappe britannique ?
- Proposition : le brûler.
- Répond parfaitement à l'exigence de confidentialité.
- Mais : besoin de pouvoir recouvrer le secret (propriété d'intégrité).

Approche Naïve.



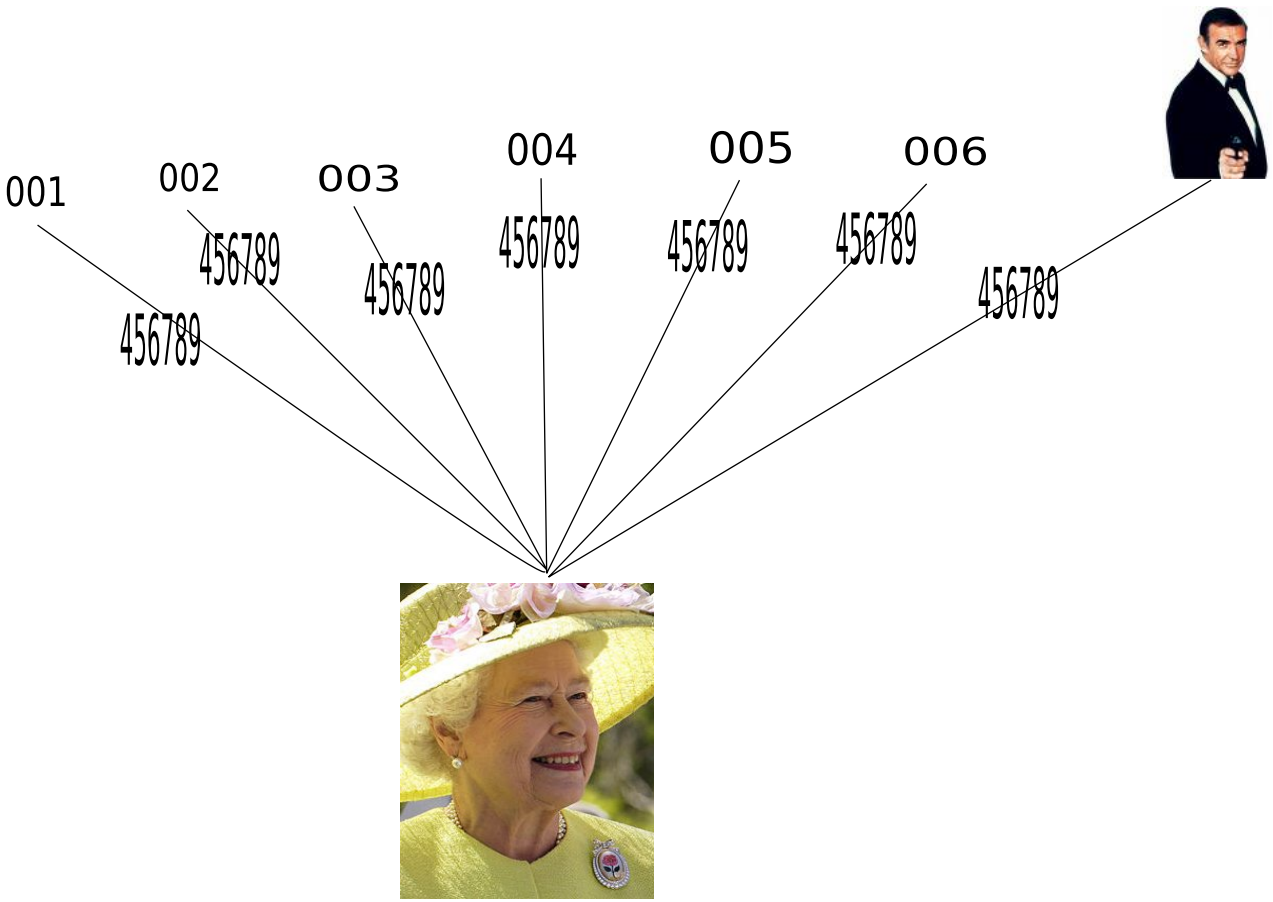
- Gros risque pour la confidentialité : un agent corrompu peut divulguer le secret.
- Risque pour l'intégrité ?

Approche Naïve.



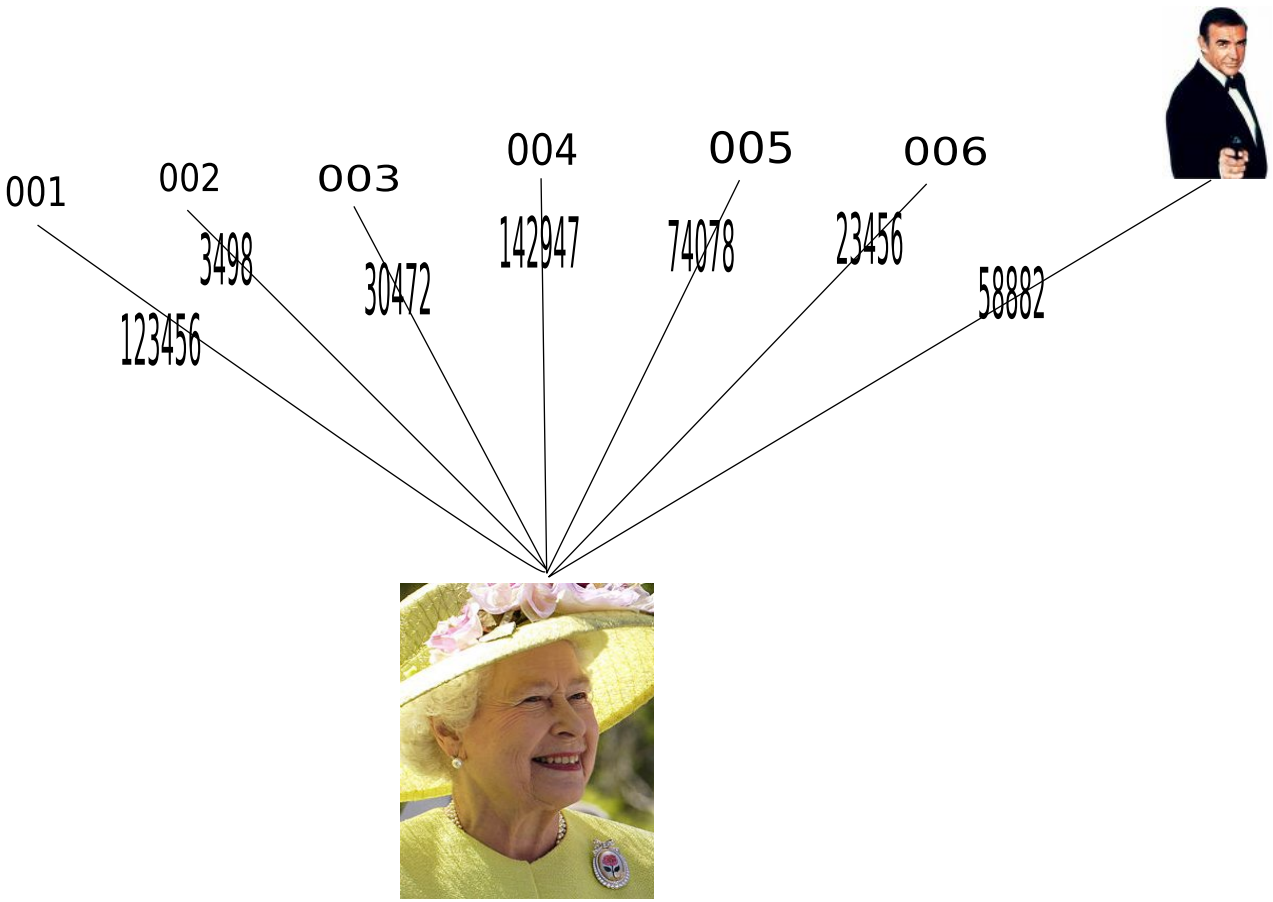
- Gros risque pour la confidentialité : un agent corrompu peut divulguer le secret.
- Risque pour l'intégrité ?

Approche Naïve.



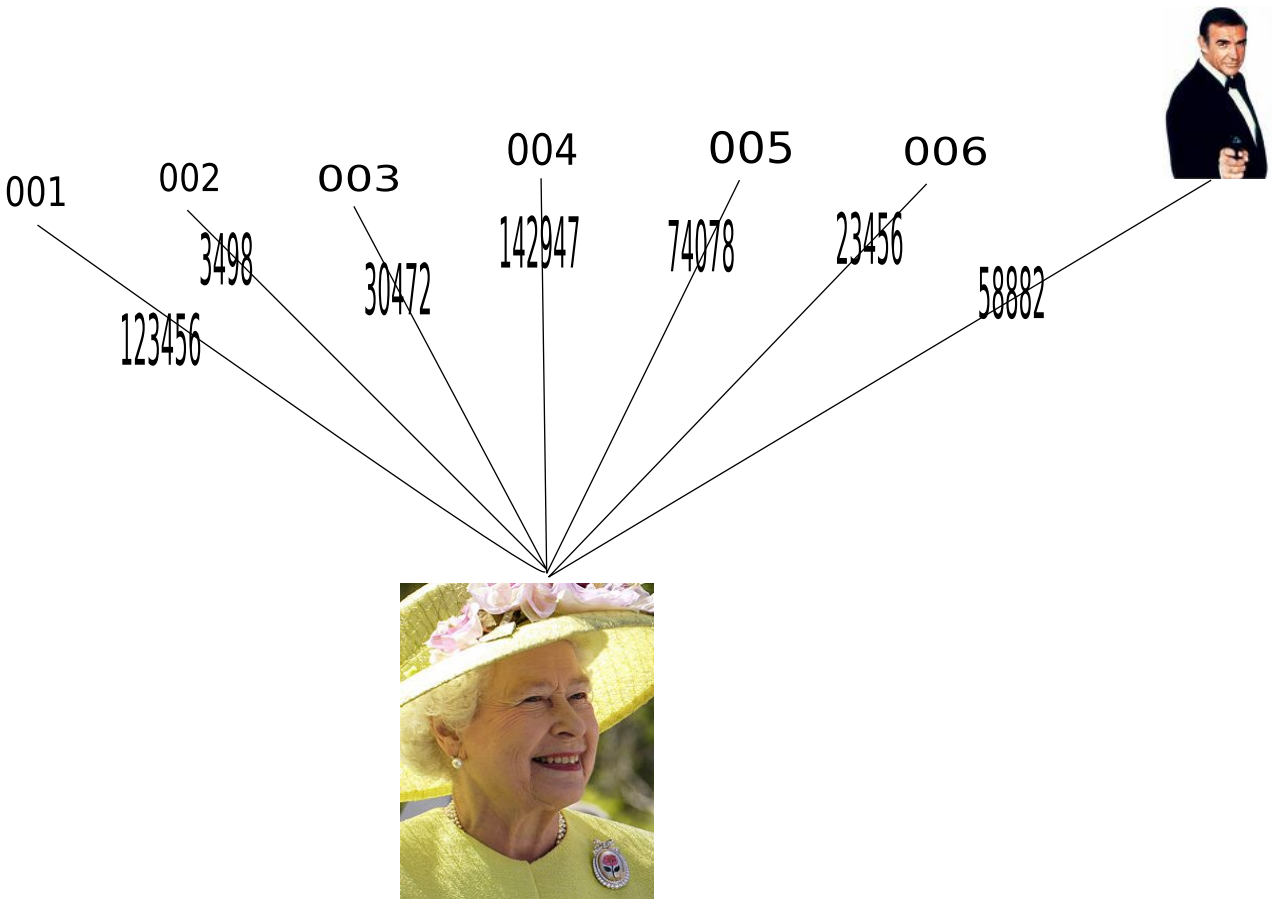
- Gros risque pour la confidentialité : un agent corrompu peut divulguer le secret.
- Risque pour l'intégrité ?

Approche Un peu moins naïve.



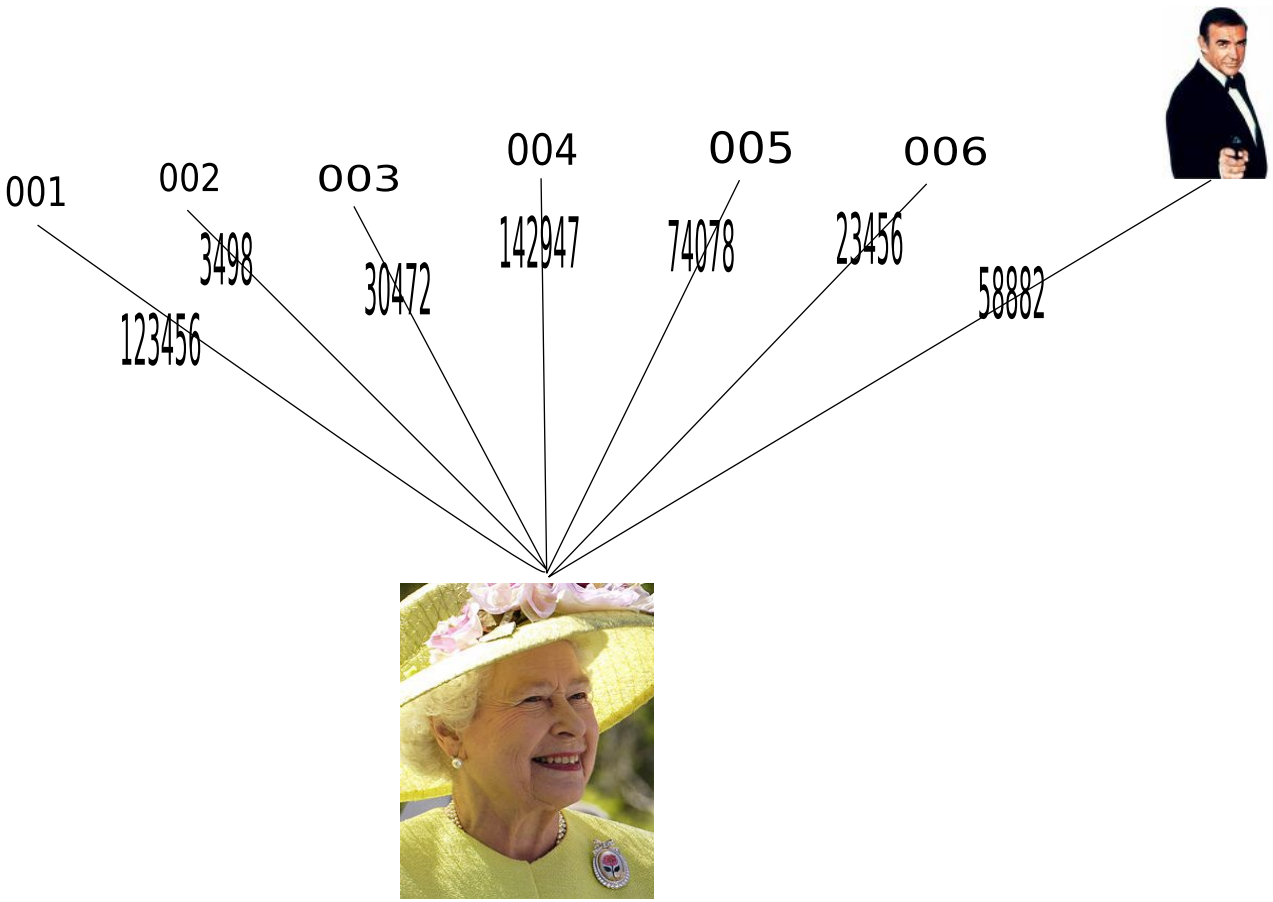
- Risque pour la confidentialité : très faible (corruption de tous les agents)
- Risque pour l'intégrité : énorme (un mort $\longrightarrow$ secret perdu).

Approche Un peu moins naïve.



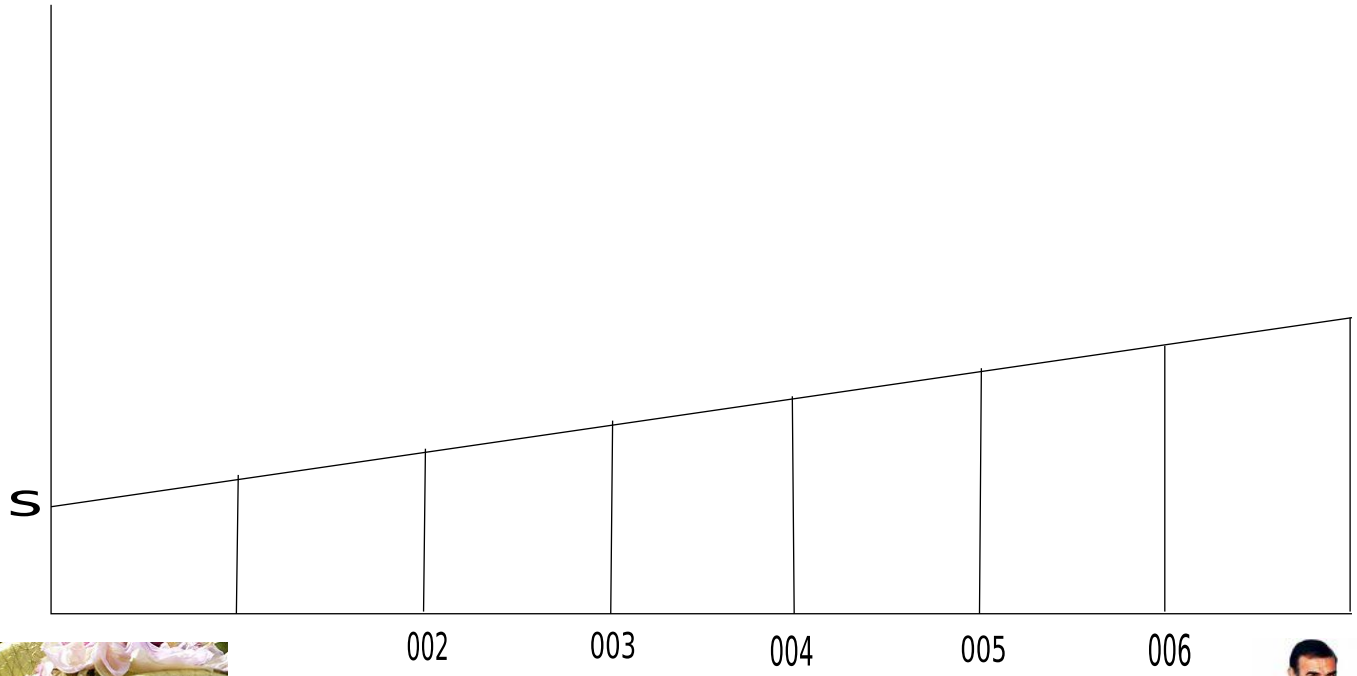
- Risque pour la confidentialité : très faible (corruption de tous les agents)
- Risque pour l'intégrité : énorme (un mort $\longrightarrow$ secret perdu).

Approche Un peu moins naïve.



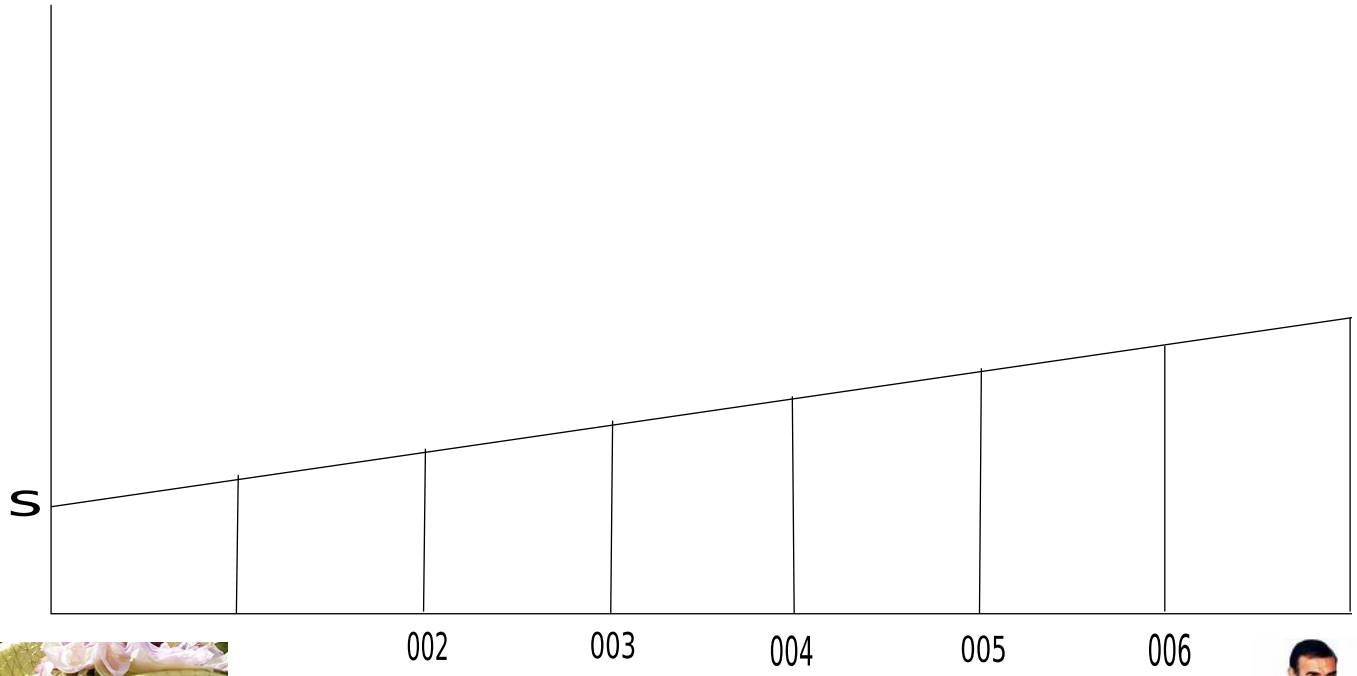
- Risque pour la confidentialité : très faible (corruption de tous les agents)
- Risque pour l'intégrité : énorme (un mort $\longrightarrow$ secret perdu).

Vers un partage à seuil.



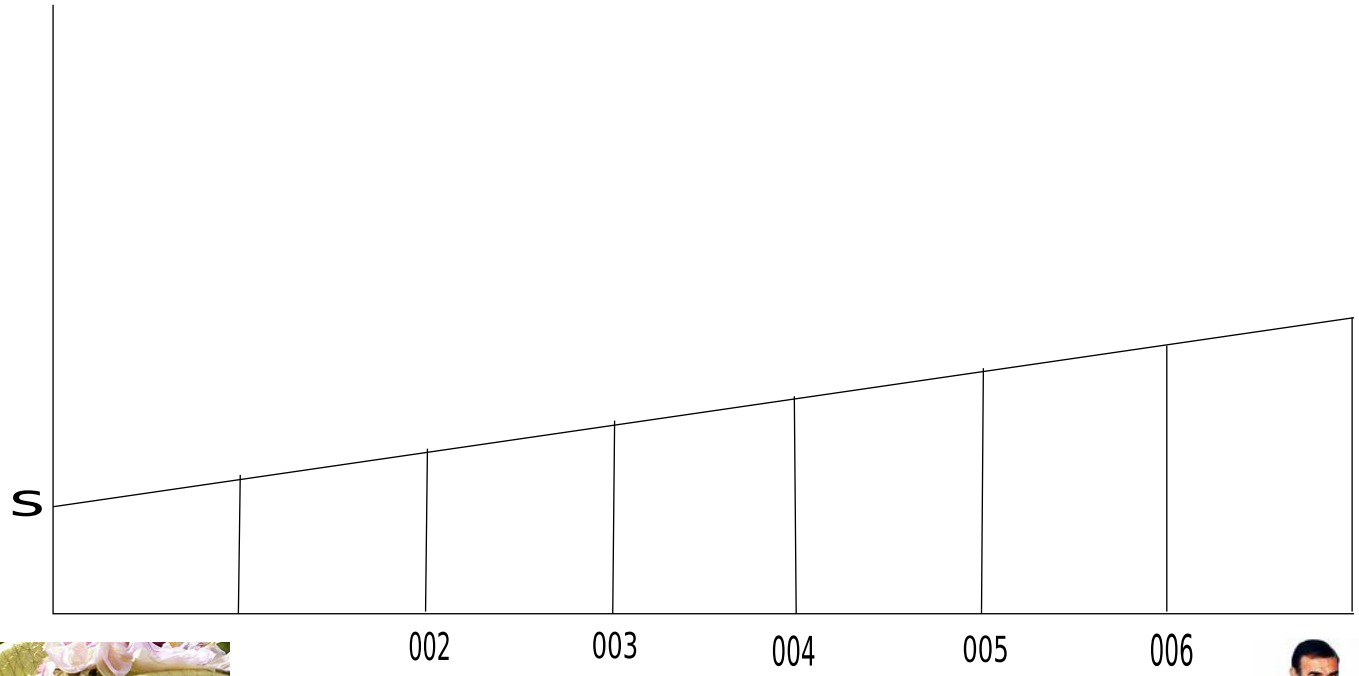
- Risque pour la confidentialité : un seul agent ne peut rien révéler.
- Risque pour l'intégrité : deux agents suffisent le reconstituer.

Vers un partage à seuil.



- Risque pour la confidentialité : un seul agent ne peut rien révéler.
- Risque pour l'intégrité : deux agents suffisent le reconstituer.

Vers un partage à seuil.



- Risque pour la confidentialité : un seul agent ne peut rien révéler.
- Risque pour l'intégrité : deux agents suffisent le reconstituer.

Partage à seuil de Shamir (1979)

- Nombre de participants n, avec un seuil de m
- Distribution du secret s : on choisit au hasard $a_1, \ldots, a_{m-1}$, et on distribue $s + \sum_{i=1}^{m-1} a_i j^i$ au participant j
- Recouvrement du secret : interpolation de Lagrange
- Confidentialité : m participants nécessaires pour obtenir une information.
- Intégrité : comment vérifier l'intégrité des m participants ?

Partage à seuil de Shamir (1979)

- Nombre de participants n, avec un seuil de m
- Distribution du secret s : on choisit au hasard $a_1, \ldots, a_{m-1}$, et on distribue $s + \sum_{i=1}^{m-1} a_i j^i$ au participant j
- Recouvrement du secret : interpolation de Lagrange
- Confidentialité : m participants nécessaires pour obtenir une information.
- Intégrité : comment vérifier l'intégrité des m participants ?

Partage à seuil de Shamir (1979)

- Nombre de participants n, avec un seuil de m
- Distribution du secret s : on choisit au hasard $a_1, \ldots, a_{m-1}$, et on distribue $s + \sum_{i=1}^{m-1} a_i j^i$ au participant j
- Recouvrement du secret : interpolation de Lagrange
- Confidentialité : m participants nécessaires pour obtenir une information.
- Intégrité : comment vérifier l'intégrité des m participants ?

Partage à seuil de Shamir (1979)

- Nombre de participants n, avec un seuil de m
- Distribution du secret s : on choisit au hasard $a_1, \ldots, a_{m-1}$, et on distribue $s + \sum_{i=1}^{m-1} a_i j^i$ au participant j
- Recouvrement du secret : interpolation de Lagrange
- Confidentialité : m participants nécessaires pour obtenir une information.
- Intégrité : comment vérifier l'intégrité des m participants ?

Partage à seuil de Shamir (1979)

- Nombre de participants n, avec un seuil de m
- Distribution du secret s : on choisit au hasard $a_1, \ldots, a_{m-1}$, et on distribue $s + \sum_{i=1}^{m-1} a_i j^i$ au participant j
- Recouvrement du secret : interpolation de Lagrange
- Confidentialité : m participants nécessaires pour obtenir une information.
- Intégrité : comment vérifier l'intégrité des m participants ?

Implantation

Flottants déconseillés :

- Problème de précision des calculs.
- Comment tirer un réel au hasard ? Uniformité impossible, risque de fuite d'information.
- Risque de fuites : erreurs de calcul $\longrightarrow$ informations sur le secret.
- Portabilité problématique (architecture, compilateur, etc.)

Utilisation de $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$:

- Corps : tout ce qui précède marche si $n \leq p$.
- Fini : facile de tirer des valeurs au hasard uniformément.
- Secret parfait si strictement moins de m fuites.

Implantation

Flottants déconseillés :

- Problème de précision des calculs.
- Comment tirer un réel au hasard ? Uniformité impossible, risque de fuite d'information.
- Risque de fuites : erreurs de calcul $\rightarrow$ informations sur le secret.
- Portabilité problématique (architecture, compilateur, etc.)

Utilisation de $\mathbb{Z}_p = \mathbb{Z}/p\mathbb{Z}$:

- Corps : tout ce qui précède marche si $n \leq p$.
- Fini : facile de tirer des valeurs au hasard uniformément.
- Secret parfait si strictement moins de m fuites.

Vérifier l'intégrité (VSS, Feldman 1987)

Idée :

- Prendre des « empreintes digitales » des morceaux de secret
- Les rendre publiques
- Vérifier chaque morceau lorsqu'on reconstitue le secret
- Impact sur la confidentialité : on peut vérifier si une valeur candidate est le secret ou non

Prise d'empreinte dans VSS

- Groupe $G = \{1, g, \ldots, g^{p-1}\}$
- Empreinte de $x \in \mathbb{Z}_p$: g^x
- Calcul d'empreinte facile : $2 \log_2 x$ multiplications dans G maxi.
- Utile : $g_1^x = g_2^x$ seulement si $g_1 = g_2 (x \mapsto g^x$est bijective).
- Calcul de la fonction réciproque $\log_g$: problème du log discret.
- Difficile avec certains G (courbes elliptiques, sous groupes de $\mathbb{Z}/q\mathbb{Z}$, ...) et p suffisamment grand ($\approx 2^{500}$).

Sécurité de VSS

- log discret facile $\Rightarrow$ VSS pas sûr
- Réciproque ?
- Qu'est-ce qu'un problème difficile ?

Sécurité de VSS

- log discret facile $\Rightarrow$ VSS pas sûr
- Réciproque ?
- Qu'est-ce qu'un problème difficile ?

Sécurité de VSS

- log discret facile $\Rightarrow$ VSS pas sûr
- Réciproque ?
- Qu'est-ce qu'un problème difficile ?

Modéliser la Sécurité

- Jeu entre un attaquant et un défenseur
- Jeu : programme séquentiel probabiliste (tirage de valeurs au hasard)
- Avantage de l'attaquant : probabilité de succès en un temps déterminé

Modélisation : exemples

- Problème du log discret. Avantage de $\mathscr{A}$ $Pr[\hat{x} = x]$ dans le jeu :

$$\mathbf{Jeu - DL} \left[\begin{array}{l} x \leftarrow \mathbb{Z}_p \\ y \leftarrow g^x \\ \hat{x} \leftarrow \mathscr{A}(y) \end{array} \right.$$

- Jeu VSS : Avantage de $\mathscr{A}$ $Pr[\hat{s} = s]$ dans le jeu :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ a_1 \leftarrow \mathbb{Z}_p, \ldots, a_{m-1} \leftarrow \mathbb{Z}_p \\ s_i \leftarrow s + \sum_{j=1}^{m-1} a_j . i^j \text{ pour } i \in \{1, \ldots, n\} \\ \hat{s} \leftarrow \mathscr{A}(g^{a_1}, \ldots, g^{a_{m-1}}, s_1, \ldots, s_{m-1}, g^s) \end{array} \right.$$

Comparaison de jeux

- DL facile $\Rightarrow$VSS facile : Existence d'un adversaire d'avantage q en temps t contre DL$\Rightarrow$ existence d'un adversaire d'avantage q en temps t contre VSS.
- Réciproque ?

Comparaison de jeux

- DL facile $\Rightarrow$VSS facile : Existence d'un adversaire d'avantage q en temps t contre DL$\Rightarrow$ existence d'un adversaire d'avantage q en temps t contre VSS.
- Réciproque ?

Réduction de DL à VSS (1)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ a_1 \leftarrow \mathbb{Z}_p, \ldots, a_{m-1} \leftarrow \mathbb{Z}_p \\ (s_0, \ldots, s_{m-1}) \leftarrow u(s, a_1, \ldots, a_{m-1}) \\ \hat{s} \leftarrow \mathscr{A}(g^{a_1}, \ldots, g^{a_{m-1}}, s_1, \ldots, s_{m-1}, g^s) \end{array}\right.$$

- u application linéaire
- u injective ($s_i = 0$ pour tout $i \in \{0, \ldots, m-1\}$ implique $s = 0$ et $a_i = 0$ pour tout $i \in \{1, \ldots, m-1\}$)
- u bijective ($u : \mathbb{Z}_p^m \to \mathbb{Z}_p^m$)

Réduction de DL à VSS (1)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ a_1 \leftarrow \mathbb{Z}_p, \ldots, a_{m-1} \leftarrow \mathbb{Z}_p \\ (s_0, \ldots, s_{m-1}) \leftarrow u(s, a_1, \ldots, a_{m-1}) \\ \hat{s} \leftarrow \mathscr{A}(g^{a_1}, \ldots, g^{a_{m-1}}, s_1, \ldots, s_{m-1}, g^s) \end{array}\right.$$

- u application linéaire
- u injective ($s_i = 0$ pour tout $i \in \{0, \ldots, m-1\}$ implique $s = 0$ et $a_i = 0$ pour tout $i \in \{1, \ldots, m-1\}$)
- u bijective ($u : \mathbb{Z}_p^m \to \mathbb{Z}_p^m$)

Réduction de DL à VSS (1)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ a_1 \leftarrow \mathbb{Z}_p, \ldots, a_{m-1} \leftarrow \mathbb{Z}_p \\ (s_0, \ldots, s_{m-1}) \leftarrow u(s, a_1, \ldots, a_{m-1}) \\ \hat{s} \leftarrow \mathscr{A}(g^{a_1}, \ldots, g^{a_{m-1}}, s_1, \ldots, s_{m-1}, g^s) \end{array}\right.$$

- u application linéaire
- u injective ($s_i = 0$ pour tout $i \in \{0, \ldots, m-1\}$ implique $s = 0$ et $a_i = 0$ pour tout $i \in \{1, \ldots, m-1\}$)
- u bijective ($u : \mathbb{Z}_p^m \to \mathbb{Z}_p^m$)

Réduction de DL à VSS (1)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ a_1 \leftarrow \mathbb{Z}_p, \ldots, a_{m-1} \leftarrow \mathbb{Z}_p \\ (s_0, \ldots, s_{m-1}) \leftarrow u(s, a_1, \ldots, a_{m-1}) \\ \hat{s} \leftarrow \mathscr{A}(g^{a_1}, \ldots, g^{a_{m-1}}, s_1, \ldots, s_{m-1}, g^s) \end{array}\right.$$

- u application linéaire
- u injective ($s_i = 0$ pour tout $i \in \{0, \ldots, m-1\}$ implique $s = 0$ et $a_i = 0$ pour tout $i \in \{1, \ldots, m-1\}$)
- u bijective ($u : \mathbb{Z}_p^m \to \mathbb{Z}_p^m$)

Réduction de DL à VSS (2)

Récrivons VSS :

$$\left[\begin{array}{l} s_0 \leftarrow \mathbb{Z}_p \\ s_1 \leftarrow \mathbb{Z}_p, \ldots, s_{m-1} \leftarrow \mathbb{Z}_p \\ (s, a_1, \ldots, a_{m-1}) \leftarrow u^{-1}(s_0, s_1, \ldots, s_{m-1}) \\ \hat{s} \leftarrow \mathscr{A}(g^{a_1}, \ldots, g^{a_{m-1}}, s_1, \ldots, s_{m-1}, g^s) \end{array}\right.$$

Réduction de DL à VSS (3)

Récrivons VSS :

$$\left[\begin{array}{l} s_0 \leftarrow \mathbb{Z}_p \\ s_1 \leftarrow \mathbb{Z}_p, \ldots, s_{m-1} \leftarrow \mathbb{Z}_p \\ (\alpha_0, \alpha_1, \ldots, \alpha_{m-1}) \leftarrow lift(u^{-1})(g^{s_0}, g^{s_1}, \ldots, g^{s_{m-1}}) \\ \hat{s} \leftarrow \mathscr{A}(\alpha_1, \ldots, \alpha_{m-1}, s_1, \ldots, s_{m-1}, \alpha_0) \end{array}\right.$$

Avec :

- $map_f : (x_0, \ldots, x_{m-1}) \mapsto (f(x_0), \ldots, f(x_{m-1}))$
- $lift(u^{-1})$:

$$\begin{array}{rcl} G^m & \longrightarrow & G^m \\ (g_0, \ldots, g_{m-1}) & \mapsto & map_{x \mapsto g^x}(u^{-1}(log_g(g_0), \ldots, \log_g(g_{m-1}))) \end{array}$$

Calcul de $lift(u^{-1})$

Soit $(c_{ij}) = u^{-1}$
Soit $(\alpha_0, \ldots, \alpha_{m-1}) = lift(u^{-1})(\beta_0, \ldots, \beta_j)$

$$\alpha_j = g^{\Sigma_{i=0}^{m-1} c_{ji} \log_g \beta_j} = \prod_{i=0}^{m-1} \beta_j^{c_{ji}}$$

Peut se faire sans calculer de logarithme discret

Calcul de $lift(u^{-1})$

Soit $(c_{ij}) = u^{-1}$
Soit $(\alpha_0, \ldots, \alpha_{m-1}) = lift(u^{-1})(\beta_0, \ldots, \beta_j)$

$$\alpha_j = g^{\sum_{i=0}^{m-1} c_{ji} \log_g \beta_j} = \prod_{i=0}^{m-1} \beta_j^{c_{ji}}$$

Peut se faire sans calculer de logarithme discret

Réduction de DL à VSS (4)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ y \leftarrow g^s \\ \hat{s} \leftarrow \mathscr{B}(y) \end{array}\right.$$

où $\mathscr{B}(\alpha)$ est l'algorithme suivant :

$$\left[\begin{array}{l} s_1 \leftarrow \mathbb{Z}_p, \ldots, s_{m-1} \leftarrow \mathbb{Z}_p \\ (\alpha_0, \alpha_1, \ldots, \alpha_{m-1}) \leftarrow \mathit{lift}(u^{-1})(\alpha, g^{s_1}, \ldots, g^{s_{m-1}}) \\ \hat{s} \leftarrow \mathscr{A}(\alpha_1, \ldots, \alpha_{m-1}, s_1, \ldots, s_{m-1}, \alpha) \end{array}\right.$$

- $\mathscr{B}$: attaquant contre le logarithme discret.
- Avantage de $\mathscr{B}$: même probabilité de succès que $\mathscr{A}$, temps de calcul légèrement plus long (quelques mises à la puissance et multiplications).
- $\rightarrow$On « garantit » la difficulté de VSS en la ramenant à celle de DL (problème réputé difficile depuis 40 ans)

Réduction de DL à VSS (4)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ y \leftarrow g^s \\ \hat{s} \leftarrow \mathscr{B}(y) \end{array}\right.$$

où $\mathscr{B}(\alpha)$ est l'algorithme suivant :

$$\left[\begin{array}{l} s_1 \leftarrow \mathbb{Z}_p, \ldots, s_{m-1} \leftarrow \mathbb{Z}_p \\ (\alpha_0, \alpha_1, \ldots, \alpha_{m-1}) \leftarrow \mathit{lift}(u^{-1})(\alpha, g^{s_1}, \ldots, g^{s_{m-1}}) \\ \hat{s} \leftarrow \mathscr{A}(\alpha_1, \ldots, \alpha_{m-1}, s_1, \ldots, s_{m-1}, \alpha) \end{array}\right.$$

- $\mathscr{B}$: attaquant contre le logarithme discret.
- Avantage de $\mathscr{B}$: même probabilité de succès que $\mathscr{A}$, temps de calcul légèrement plus long (quelques mises à la puissance et multiplications).
- $\rightarrow$On « garantit » la difficulté de VSS en la ramenant à celle de DL (problème réputé difficile depuis 40 ans)

Réduction de DL à VSS (4)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ y \leftarrow g^s \\ \hat{s} \leftarrow \mathscr{B}(y) \end{array}\right.$$

où $\mathscr{B}(\alpha)$ est l'algorithme suivant :

$$\left[\begin{array}{l} s_1 \leftarrow \mathbb{Z}_p, \ldots, s_{m-1} \leftarrow \mathbb{Z}_p \\ (\alpha_0, \alpha_1, \ldots, \alpha_{m-1}) \leftarrow \mathit{lift}(u^{-1})(\alpha, g^{s_1}, \ldots, g^{s_{m-1}}) \\ \hat{s} \leftarrow \mathscr{A}(\alpha_1, \ldots, \alpha_{m-1}, s_1, \ldots, s_{m-1}, \alpha) \end{array}\right.$$

- $\mathscr{B}$: attaquant contre le logarithme discret.
- Avantage de $\mathscr{B}$: même probabilité de succès que $\mathscr{A}$, temps de calcul légèrement plus long (quelques mises à la puissance et multiplications).
- $\rightarrow$On « garantit » la difficulté de VSS en la ramenant à celle de DL (problème réputé difficile depuis 40 ans)

Réduction de DL à VSS (4)

Récrivons VSS :

$$\left[\begin{array}{l} s \leftarrow \mathbb{Z}_p \\ y \leftarrow g^s \\ \hat{s} \leftarrow \mathscr{B}(y) \end{array}\right.$$

où $\mathscr{B}(\alpha)$ est l'algorithme suivant :

$$\left[\begin{array}{l} s_1 \leftarrow \mathbb{Z}_p, \ldots, s_{m-1} \leftarrow \mathbb{Z}_p \\ (\alpha_0, \alpha_1, \ldots, \alpha_{m-1}) \leftarrow \mathit{lift}(u^{-1})(\alpha, g^{s_1}, \ldots, g^{s_{m-1}}) \\ \hat{s} \leftarrow \mathscr{A}(\alpha_1, \ldots, \alpha_{m-1}, s_1, \ldots, s_{m-1}, \alpha) \end{array}\right.$$

- $\mathscr{B}$: attaquant contre le logarithme discret.
- Avantage de $\mathscr{B}$: même probabilité de succès que $\mathscr{A}$, temps de calcul légèrement plus long (quelques mises à la puissance et multiplications).
- $\rightarrow$On « garantit » la difficulté de VSS en la ramenant à celle de DL (problème réputé difficile depuis 40 ans)

Redistribution

On veut garder un secret sur du long terme :

- Les agents vont finir par mourir ou se vendre à l'ennemi
- Il faut prévoir une relève.
- On ne veut pas reconstituer le secret : trop dangereux.

Redistribution : idée

- Redistribuer chacune des parts des m anciens agents aux n' nouveaux agents
- Problème : le nombre de parts augmente ($m \times n'$ sous-parts). Risque de devenir ingérable.
- Recombiner les parts

Redistribution : idée

- Redistribuer chacune des parts des m anciens agents aux n' nouveaux agents
- Problème : le nombre de parts augmente ($m \times n'$ sous-parts). Risque de devenir ingérable.
- Recombiner les parts

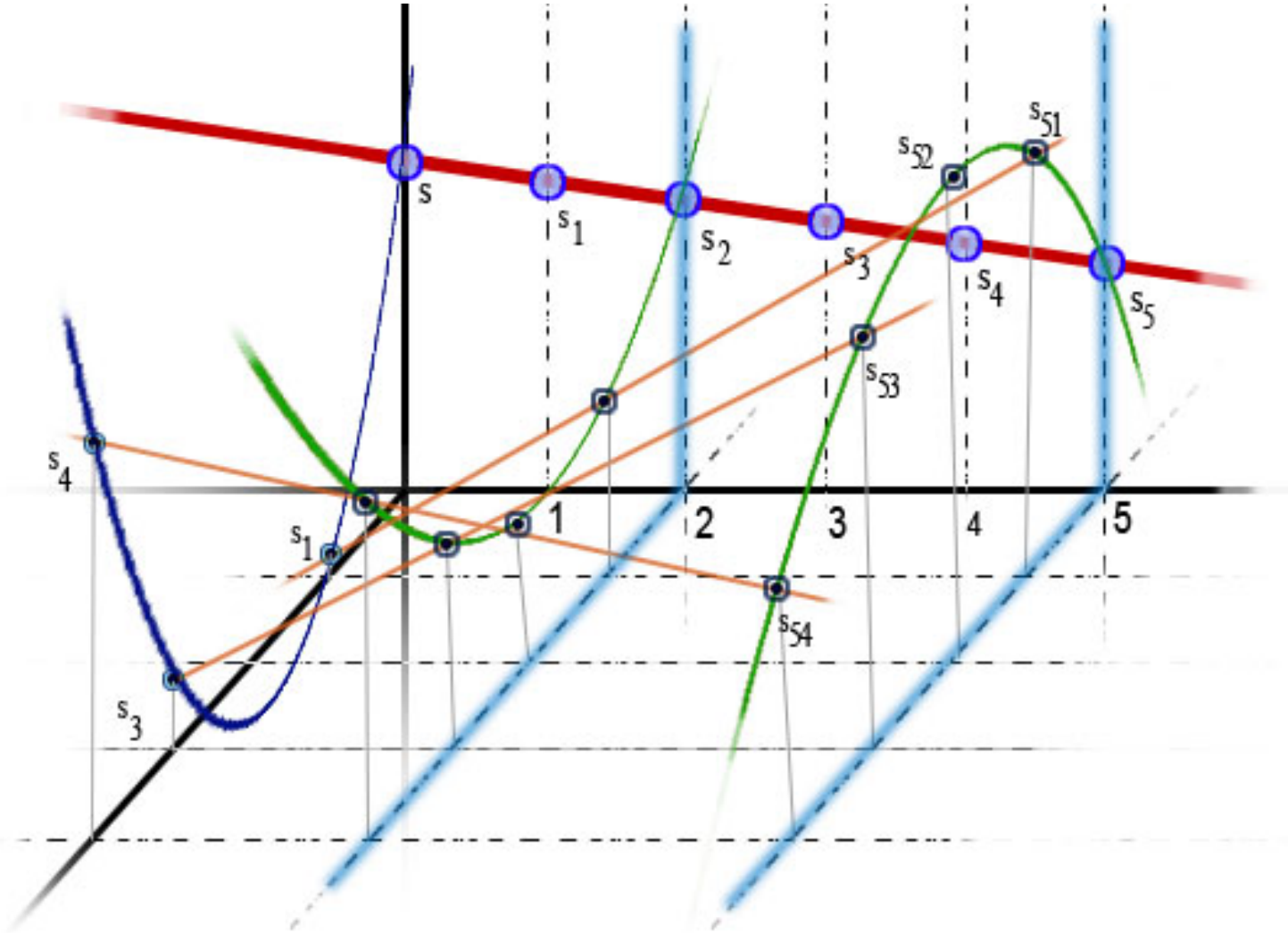
Redistribution : idée

- Redistribuer chacune des parts des m anciens agents aux n' nouveaux agents
- Problème : le nombre de parts augmente ($m \times n'$ sous-parts). Risque de devenir ingérable.
- Recombiner les parts

Redistribution : idée

- Redistribuer chacune des parts des m anciens agents aux n' nouveaux agents
- Problème : le nombre de parts augmente ($m \times n'$ sous-parts). Risque de devenir ingérable.
- Recombiner les parts

Redistribution : description géométrique



Nouveaux problèmes

- Risques supplémentaires : on fournit à l'adversaire les exponentielles de nouvelles valeurs.
- Est-ce que ça lui facilite le calcul ?
- Réponse : possible (multiplication du risque à chaque étape par un facteur important)
- $\longrightarrow$Compromis à trouver entre taille du groupe, valeur choisie pour m et n, et fréquence des redistributions.

Nouveaux problèmes

- Risques supplémentaires : on fournit à l'adversaire les exponentielles de nouvelles valeurs.
- Est-ce que ça lui facilite le calcul ?
- Réponse : possible (multiplication du risque à chaque étape par un facteur important)
- $\longrightarrow$Compromis à trouver entre taille du groupe, valeur choisie pour m et n, et fréquence des redistributions.

Nouveaux problèmes

- Risques supplémentaires : on fournit à l'adversaire les exponentielles de nouvelles valeurs.
- Est-ce que ça lui facilite le calcul ?
- Réponse : possible (multiplication du risque à chaque étape par un facteur important)
- $\longrightarrow$Compromis à trouver entre taille du groupe, valeur choisie pour m et n, et fréquence des redistributions.

Conclusion & Perspectives

- Étude théorique du partage de secret : sécurité basée sur un vrai modèle mathématique/informatique (modèle de la cryptographie computationnelle)
- Applications pratiques à terme : stockage à long terme de données sensibles (dossier médical), réparties sur plusieurs serveurs.
- Applicabilité industrielle ? (économiquement trop cher, peu de sécurité exigée)

Conclusion & Perspectives

- Étude théorique du partage de secret : sécurité basée sur un vrai modèle mathématique/informatique (modèle de la cryptographie computationnelle)
- Applications pratiques à terme : stockage à long terme de données sensibles (dossier médical), réparties sur plusieurs serveurs.
- Applicabilité industrielle ? (économiquement trop cher, peu de sécurité exigée)

Conclusion & Perspectives

- Étude théorique du partage de secret : sécurité basée sur un vrai modèle mathématique/informatique (modèle de la cryptographie computationnelle)
- Applications pratiques à terme : stockage à long terme de données sensibles (dossier médical), réparties sur plusieurs serveurs.
- Applicabilité industrielle ? (économiquement trop cher, peu de sécurité exigée)