Présentation des Travaux de Thèse: Délégation de Calculs Cryptographiques

Olivier Sanders

Orange Labs et École Normale Supérieure

05 Février 2014

Plan

- Thèse CIFRE
- Sujets de Recherche
- Délégation de preuves de connaissances

Thèse CIFRE

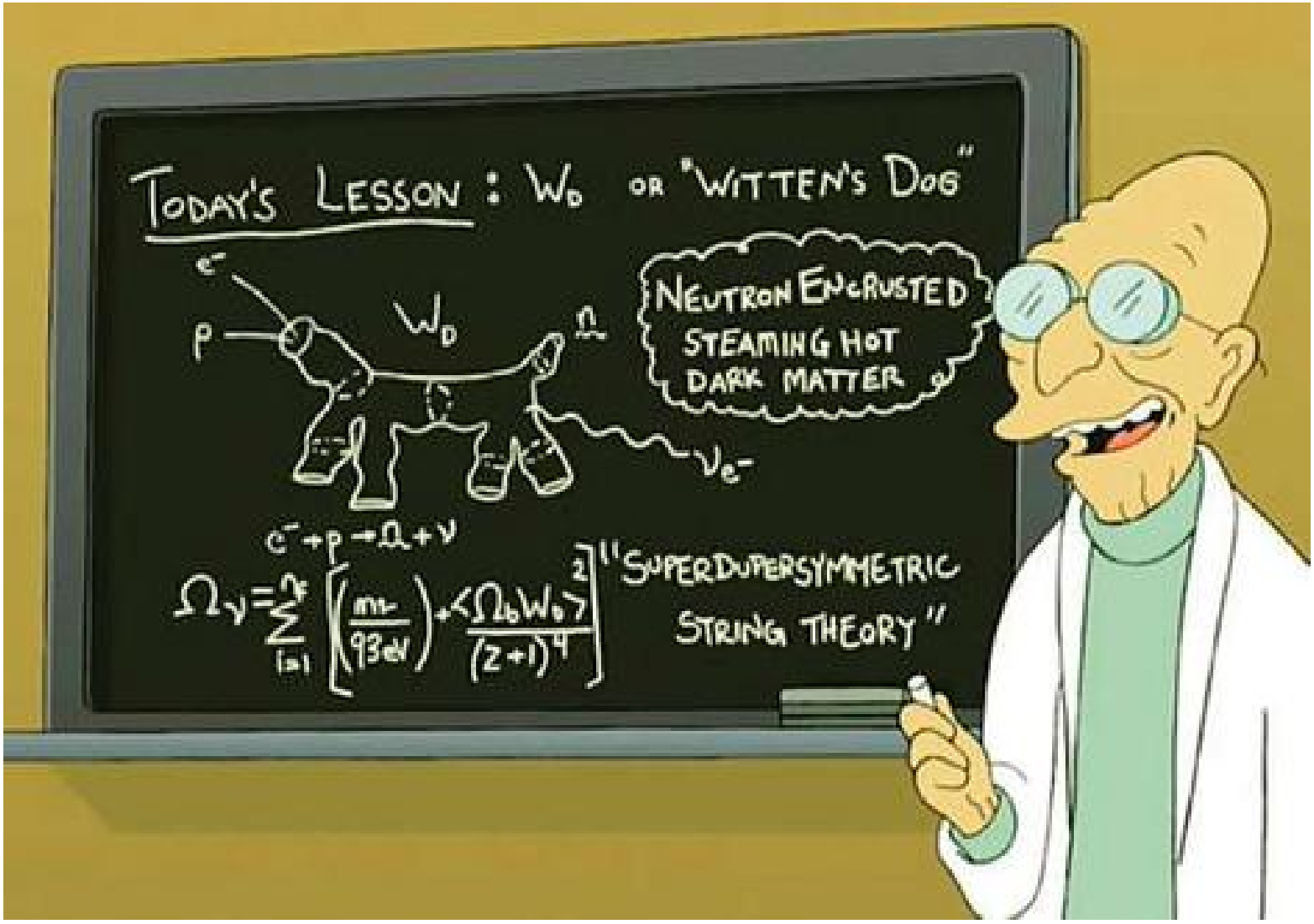
2 établissements

Entreprise

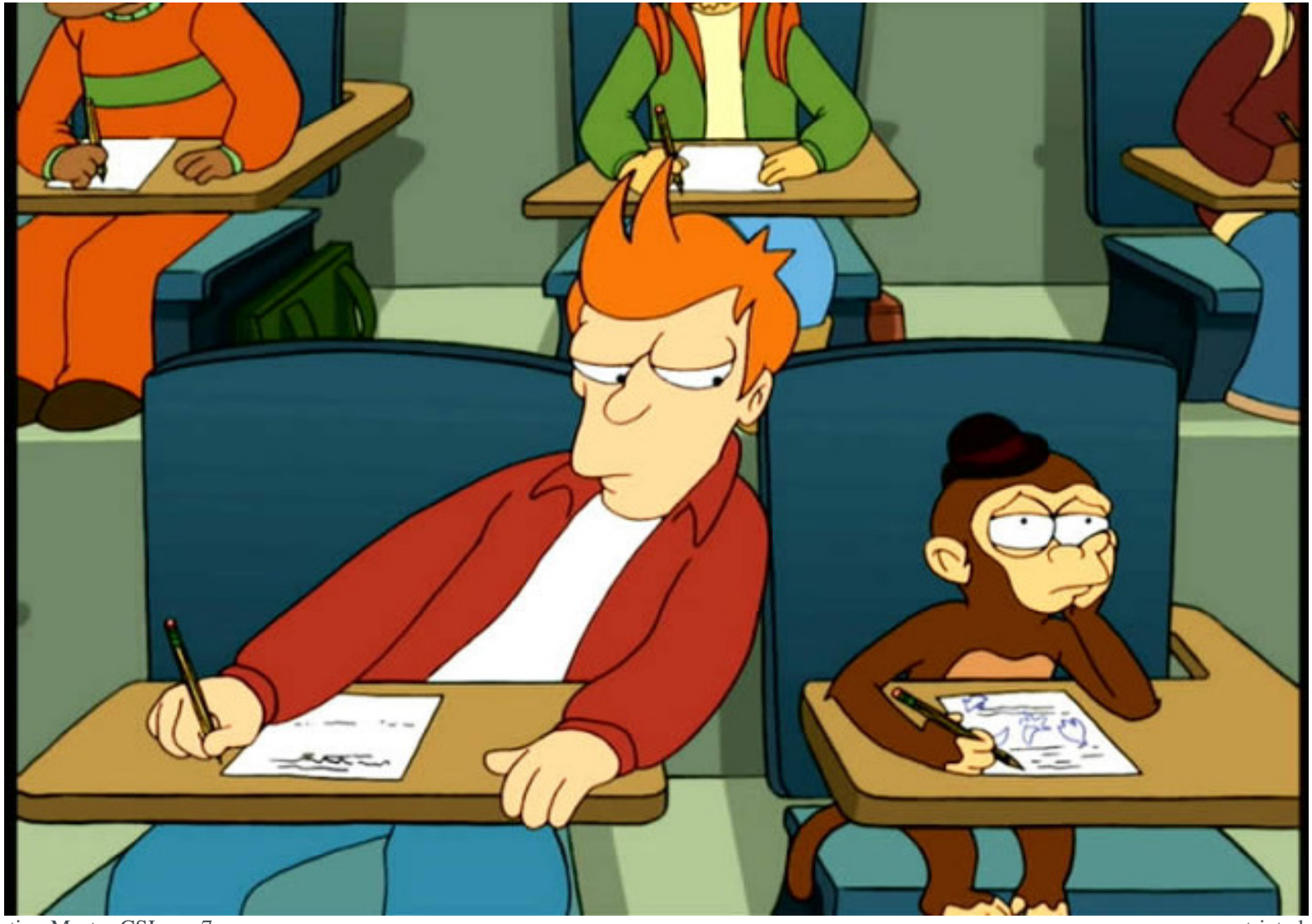
Université

2 encadrants

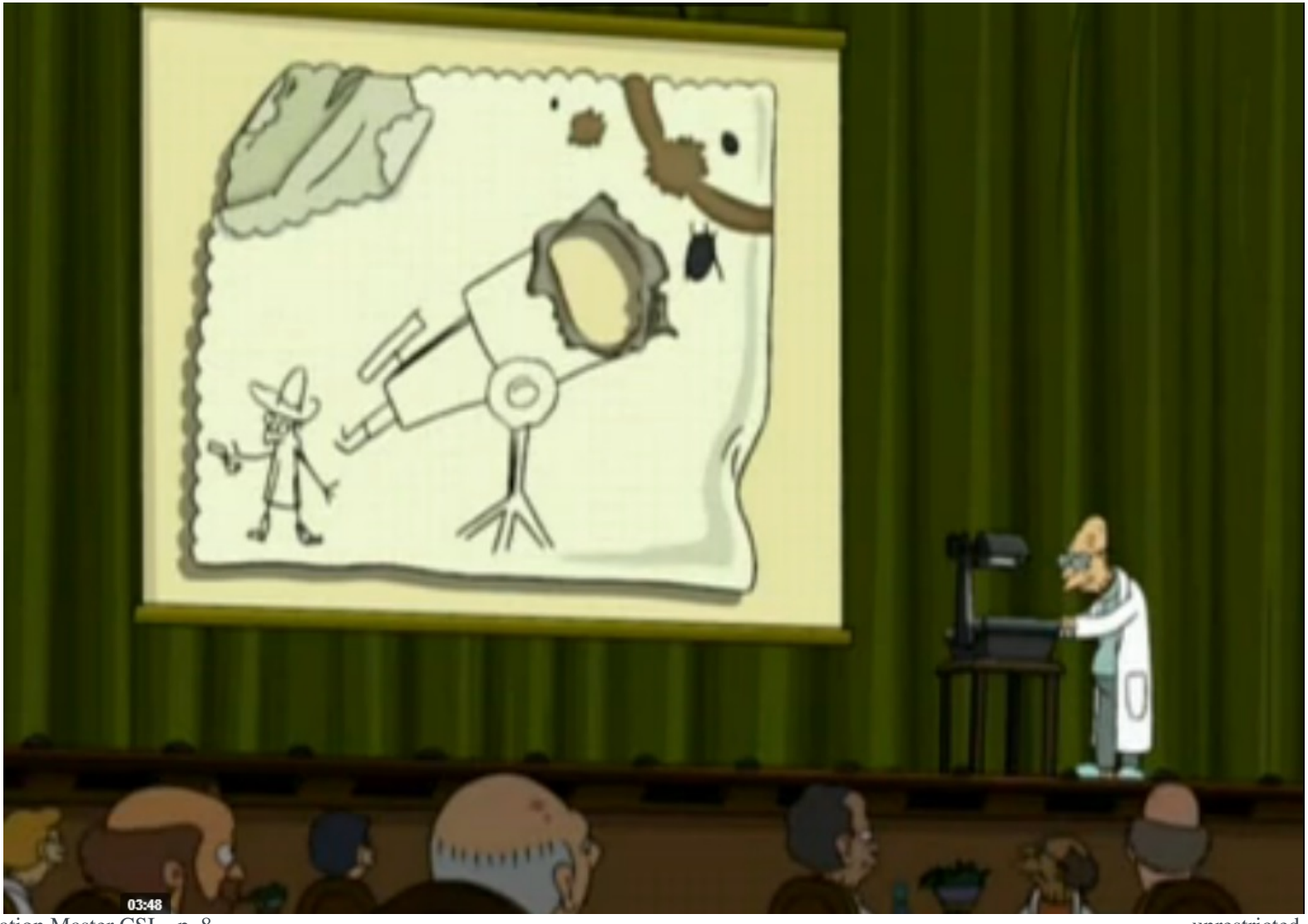
1 sujet



Proposer des réponses “originales”



Les présenter en conférence



Sujets de Recherche

Contexte

- La cryptographie moderne offre de nombreuses fonctionnalités, conciliant des intérêts à priori divergents (authentification anonyme, transfert inconscient,...).
- Les protocoles sous-jacents deviennent de plus en plus complexes, nécessitant notamment des opérations mathématiques coûteuses (couplages, exponentiations).
- Certains périphériques (carte SIM, TPM,...) sont alors incapables de supporter de tels algorithmes.

SIM/Téléphone

- La carte SIM a accès à un périphérique nettement plus puissant: le téléphone.
- Le niveau de sécurité n'est cependant pas le même: la SIM bénéficie d'un très haut niveau de sécurité contrairement au téléphone.

  $\Longrightarrow$ Aucun secret ne doit être révélé au téléphone!

LYRICS

- L'objectif du projet est d'implémenter un pass de transport anonyme sur smartphone NFC.
- Les contraintes sont très fortes: l'authentification doit se faire en moins de 300 ms.
- Certaines primitives cryptographiques (signatures de groupe, DAA) offrent les fonctionnalités requises mais sont particulièrement complexes.

DAA

- Les DAA (Direct Anonymous Attestations) permettent aux utilisateurs d'un groupe de signer au nom du groupe.
- Les signatures sont anonymes, excepté pour une entité: l'autorité d'ouverture.
- La traçabilité est contrôlée: seules les signatures ayant un élément commun (le basename) peuvent être reliées.

Un exemple de DAA

L'utilisateur dispose d'un secret z et d'un certificat A, B, C, D. Lors de la validation il obtient du lecteur un challenge m ainsi qu'un *basename* bsn.

Utilisateur(z, A, B, C, D, m, bsn)

$J \leftarrow \mathcal{H}(bsn); K \leftarrow [z]J$

Étape 1: Masquage du certificat

$I \xleftarrow{\$} \mathbb{Z}_p$

$(R, S, T, U) \leftarrow ([I]A, [I]B, [I]C, [I]D)$

Étape 2: Preuve de connaissance de z

$k \xleftarrow{\$} \mathbb{Z}_p; R_1 \leftarrow [k]J; R_2 \leftarrow [k]S$

$c \leftarrow H(R_1, R_2, R, S, T, U, m, bsn)$

$s = k + cz$

Un exemple de DAA

L'utilisateur dispose d'un secret z et d'un certificat A, B, C, D. Lors de la validation il obtient du lecteur un challenge m ainsi qu'un *basename* bsn.

SIM(s, m, bsn)	Téléphone(A, B, C, D)
$J \leftarrow \mathcal{H}(bsn); K \leftarrow [z]J$	
Étape 1: Masquage du certificat	
$I \xleftarrow{\$} \mathbb{Z}_p$	
$(R, S, T, U) \leftarrow ([I]A, [I]B, [I]C, [I]D)$	
Étape 2: Preuve de connaissance de z	
$k \xleftarrow{\$} \mathbb{Z}_p; R_1 \leftarrow [k]J; R_2 \leftarrow [k]S$	
$c \leftarrow H(R_1, R_2, R, S, T, U, m, bsn)$	
$s = k + cz$	

Un exemple de DAA

L'utilisateur dispose d'un secret z et d'un certificat A, B, C, D. Lors de la validation il obtient du lecteur un challenge m ainsi qu'un *basename* bsn.

SIM(s, m, bsn)	Téléphone(A, B, C, D)
$J \leftarrow \mathcal{H}(bsn); K \leftarrow [z]J$	
Étape 1: Masquage du certificat	
	$I \xleftarrow{\$} \mathbb{Z}_p$
	$(R, S, T, U) \leftarrow ([I]A, [I]B, [I]C, [I]D)$
Étape 2: Preuve de connaissance de z	
$k \xleftarrow{\$} \mathbb{Z}_p; R_1 \leftarrow [k]J; R_2 \leftarrow [k]S$	
$c \leftarrow H(R_1, R_2, R, S, T, U, m, bsn)$	?
$s = k + cz$	

Délégation de preuves de connaissance dans les groupes bilinéaires

Preuves de connaissance

- Une preuve de connaissance est un protocole entre deux entités $\mathcal{P}$ et $\mathcal{V}$ dans lequel $\mathcal{P}$ prouve à $\mathcal{V}$ qu'elle connaît des secrets sans rien révéler dessus.
- Un utilisateur dont la clé secrète est z et la clé publique $[z]G$ peut prouver connaissance de z en utilisant le protocole dû à Schnorr:

$$
\begin{array}{lcl}
\mathcal{P} & & \mathcal{V} \\
\hline
k \leftarrow \mathbb{Z}_p; R \leftarrow [k]G & \xrightarrow{R} & \\
 & \xleftarrow{c} & c \leftarrow \{0,1\}^t \\
s = k + c \cdot z & \xrightarrow{s} & [s]G \stackrel{?}{=} R \cdot [c]([z]G)
\end{array}
$$

Complexité de Schnorr

- La complexité de ce protocole est proportionnelle au nombre de preuves à fournir. Par exemple prouver connaissance de α_1 et α_2 tels que:

$$
\begin{array}{cc}
V_1 = [\alpha_1]A_1 & V_{n+1} = [\alpha_2]A_{n+1} \\
\ldots & \ldots \\
V_n = [\alpha_1]A_n & V_{n+m} = [\alpha_2]A_{n+m} \\
\multicolumn{2}{c}{V_{n+m+1} = [\alpha_1]A_{n+m+1} + [\alpha_2]A_{n+m+s+1}} \\
\multicolumn{2}{c}{\ldots} \\
\multicolumn{2}{c}{V_{n+m+s} = [\alpha_1]A_{n+m+s} + [\alpha_2]A_{n+m+2s}}
\end{array}
$$

demande à la SIM $n + m + 2s$ multiplications scalaires.

Complexité de Schnorr

- La complexité de ce protocole est proportionnelle au nombre de preuves à fournir. Par exemple prouver connaissance de α_1 et α_2 tels que:

$$
\begin{array}{cc}
V_1 = [\alpha_1]A_1 & V_{n+1} = [\alpha_2]A_{n+1} \\
\ldots & \ldots \\
V_n = [\alpha_1]A_n & V_{n+m} = [\alpha_2]A_{n+m} \\
\multicolumn{2}{c}{V_{n+m+1} = [\alpha_1]A_{n+m+1} + [\alpha_2]A_{n+m+s+1}} \\
\multicolumn{2}{c}{\ldots} \\
\multicolumn{2}{c}{V_{n+m+s} = [\alpha_1]A_{n+m+s} + [\alpha_2]A_{n+m+2s}}
\end{array}
$$

  demande à la SIM $n + m + 2s$ multiplications scalaires.

- Notre protocole n'en demande que 2, pré-calculables.

Groupes Bilinéaires

- On considère 3 groupes $\mathbb{G}_1, \mathbb{G}_2$ et $\mathbb{G}_T$ d'ordre p premier munis d'un "couplage" e:

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$$

  Tels que $\forall u, v \in \mathbb{Z}_p$ et $A, B \in \mathbb{G}_1,\ G_2 \in \mathbb{G}_2$:

$$e([u]A, [v]G_2) = e(A, G_2)^{u \cdot v}$$
$$e(A + B, G_2) = e(A, G_2) \cdot e(B, G_2)$$

- Ces groupes sont fréquemment utilisés en cryptographie car ils offrent plus de fonctionnalités que les groupes RSA tout en gardant des tailles inférieures.

Méthodologie

- On divise $\mathcal{P}$ en deux entités: la SIM qui gardera les secrets et le téléphone qui devra effectuer la majeure partie des calculs.
- Le téléphone pourra accéder à quelques informations supplémentaires mais pas aux secrets.
- La preuve doit rester "zero-knowledge" vis à vis de $\mathcal{V}$.

Une première idée

- Éviter de calculer un engagement par base:
  $\Rightarrow$ on les calcule dans $\mathbb{G}_2$:
  $[k_1]A_1, [k_1]A_2, ..., [k_1]A_m \quad \Longrightarrow \quad [k_1]G_2$
  $[k_2]A_{m+1}, [k_2]A_{m+2}, ..., [k_2]A_{m+n} \quad \Longrightarrow \quad [k_2]G_2$
  $[k_1]A_{m+n+1} + [k_2]A_{m+n+s+1}, \cdots$
  $[k_1]A_{m+n+s} + [k_2]A_{m+n+2s}$

- $[k_1]G_2$ et $[k_2]G_2$ permettent de retrouver $[\alpha_1]G_2$ et $[\alpha_2]G_2$
  $\Rightarrow$ on n'est plus zero-knowledge!

- Cependant $[\alpha_1]G_2$ et $[\alpha_2]G_2$ ne permettent pas de retrouver les secrets α_1 et α_2.
  $\Rightarrow$ on peut donner $[k_1]G_2$ et $[k_2]G_2$ au téléphone

Une deuxième idée

- Pour éviter la fuite d'information, le téléphone va lier ces nouveaux engagements aux bases concernées:

$$\text{Il choisit } b_i \xleftarrow{\$} \mathbb{Z}_p \text{ et envoie } [b_i^{-1}]A_i \text{ et } [b_i] \cdot [k_i]G_2$$

- Cela ne marche cependant que pour les relations n'impliquant qu'une base!
- Par exemple, pour $V = [\alpha_1]A_1 + [\alpha_2]A_2$, on révèlerait $e(A_1, G_2)^{\alpha_1}$ et $e(A_2, G_2)^{\alpha_2}$.

Une première version

- Soient $a_i \in \mathbb{Z}_p$ tels que $A_i = [a_i]G$. On suppose (provisoirement) que l'on connaît $\widetilde{A}_1 = [a_1]G_2$ et $\widetilde{A}_2 = [a_2]G_2$.
- Le téléphone choisit $t \xleftarrow{\$} \mathbb{Z}_p$ et envoie $Z_1 = [b_1^{-1}]A_1$, $Z_2 = [b_2^{-1}]A_2$, $B_1 = [b_1]([k_1]G_2 + [t]\widetilde{A}_1)$ et $B_2 = [b_2]([k_2]G_2 - [t]\widetilde{A}_2)$
- La preuve est alors zero-knowledge.
- On peut généraliser cette méthode à des relations plus compliquées.

Vérification

- La vérification par $\mathcal{V}$ se fait dans le groupe $\mathbb{G}_T$:

$$e([s_1]A_1 + [s_2]A_2 - [c]V, G_2) \stackrel{?}{=} e(Z_1, B_1) \cdot e(Z_2, B_2)$$

- Le protocole est complet (seule une entité connaissant les secrets peut produire la preuve).

- L'écart de puissance entre une SIM et un PC justifie ce compromis. Une multiplication scalaire sur SIM prend 50 ms tandis qu'un couplage sur PC peut prendre moins d'une ms.

Une deuxième version

- La nécessité de connaître les $\widetilde{A}_i$ peut être problématique mais on peut s'en débarrasser.
- Le téléphone choisit maintenant $t_1, t_2 \xleftarrow{\$} \mathbb{Z}_p$ et envoie Z_1, Z_2, $B'_1 = [b_1]([k_1]G_2 + [t_1]G_2)$ et $B'_2 = [b_2]([k_2]G_2 + [t_2]G_2)$ mais également $H = [t_1]A_1 + [t_2]A_2$.
- L'équation de vérification devient:

$$e(H + [s_1]A_1 + [s_2]A_2 - [c]V, G_2) \stackrel{?}{=} e(Z_1, B'_1) \cdot e(Z_2, B'_2)$$

Conclusion

- Le coût pour la carte SIM ne dépend que du nombre de secrets.
- Chacune de ces multiplications scalaires (1 par secret) est précalculable.
- Durant la phase *online* la SIM n'aura besoin d'effectuer qu'un hachage et des opérations dans $\mathbb{Z}_p$.
- Toutes les valeurs pré-calculées peuvent être envoyées au téléphone. La SIM peut ne garder qu'une graine (pour un générateur d'aléas) et quelques indices.

Merci

Protocole

TPM	Host	Verifier
$\forall j \in \{1,\ldots,m\}$, $k_j \xleftarrow{\$} \mathbb{Z}_p, \widetilde{Z}_j \leftarrow [k_j]\widetilde{G}$	$\forall i \in \{1,\ldots,r\}$, $(b_{i,j})_j \xleftarrow{\$} (\mathbb{Z}_p^*)^m, (t_{i,j})_j \xleftarrow{\$} (\mathbb{Z}_p)^m$ $H_i \leftarrow \sum_{j\in\mathcal{J}_i}[t_{i,j}]A_{v_{i,j}}$	
$\xrightarrow{\{\widetilde{Z}_j\}_j}$	$\forall i \in \{1,\ldots,r\}, \forall j \in \mathcal{J}_i$, $Z_{i,j} \leftarrow [b_{i,j}^{-1}]A_{v_{i,j}}$ $\widetilde{B}_{i,j} \leftarrow [b_{i,j}](\widetilde{Z}_j + [t_{i,j}]\widetilde{G})$	
	$\xrightarrow{\{H_i\}_i, \{Z_{i,j}, \widetilde{B}_{i,j}\}_{i,j}}$	
$\xleftarrow{c}$	$\xleftarrow{c}$	$c \xleftarrow{\$} \{0,1\}^\ell$
$\forall j \in \{1,\ldots,m\}$, $s_j \leftarrow k_j + c\alpha_j \bmod p$		
$\xrightarrow{\{s_j\}_j}$	$\xrightarrow{\{s_j\}_j}$	$\forall i \in \{1,\ldots,r\}$
	$e\left(H_i + \sum_{j\in\mathcal{J}_i}[s_j]A_{v_{i,j}} - [c]V_i, \widetilde{G}\right) \stackrel{?}{=} \prod_{j\in\mathcal{J}_i} e(Z_{i,j}, \widetilde{B}_{i,j})$	