Preuve de sécurité du schéma de signature de Courtois, Finiasz et Sendrier

Léonard Dallot

GREYC - UMR 6072
Univertité de Caen
France

18 Mars 2008

Motivations

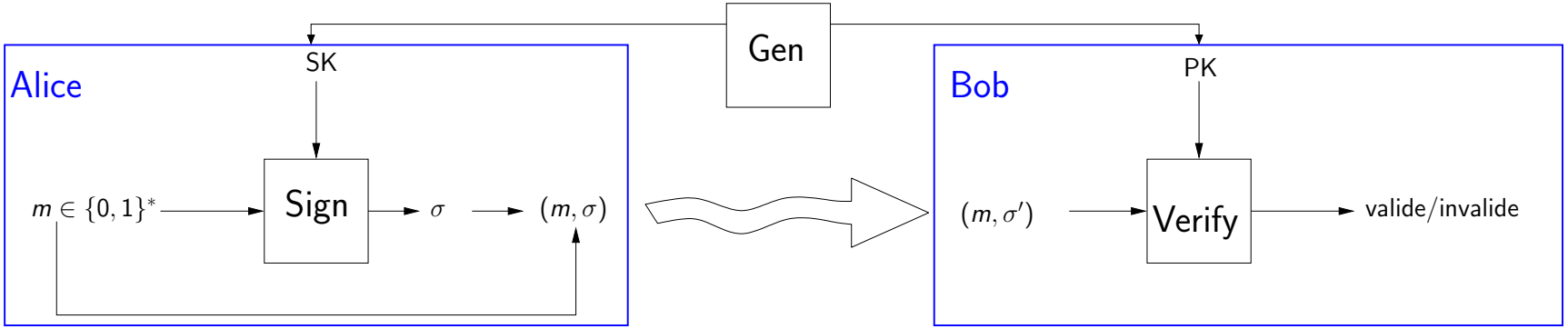
Cryptographie fondée sur la théorie des codes

- Chiffrement : McElliece [DSN Prog. Rep, 1978], Niederreiter [Prob. Contr. Inform. Theory, 1986]
- Identification à divulgation nulle de connaissance : Stern [EUR0CRYPT'89])
  - Heuristique de Fiat-Shamir (peu efficace)
- Signature : Courtois, Finiasz & Sendrier [ASIACRYPT 2001]
  - Seul schéma non cassé
  - Fournir une preuve de sécurité réductioniste
  - Implique une modification du schéma

Plan

Signature

Schémas de signature



Définition (Schéma de signature)

- *Gen*(1^κ) *: algorithme de génération*
  - *paramètre de sécurité* κ
  - *clef publique et clef secrète* (PK, SK)
- *Sign*(m, SK) *: algorithme de signature*
  - *message* m *et clef secrète* SK
  - *signature* σ *(peut être probabiliste)*
- *Verify*(m, σ', PK) *: algorithme de vérification*
  - *message* m*, signature candidate* σ' *et clef publique* PK
  - *valide/invalide*

Modèle de l'oracle aléatoire (ROM)

- Introduit par Bellare & Rogaway [ACM, 1993].
- Idéalise les fonctions de hachage, vues comme renvoyant une valeur aléatoire uniformément distribuée.

Modèle de sécurité EF-CMA dans le ROM

Forge existentielle lors d'une attaque à messages choisis (EF-CMA)

- Le plus fort modèle de sécurité pour les schémas de signature.

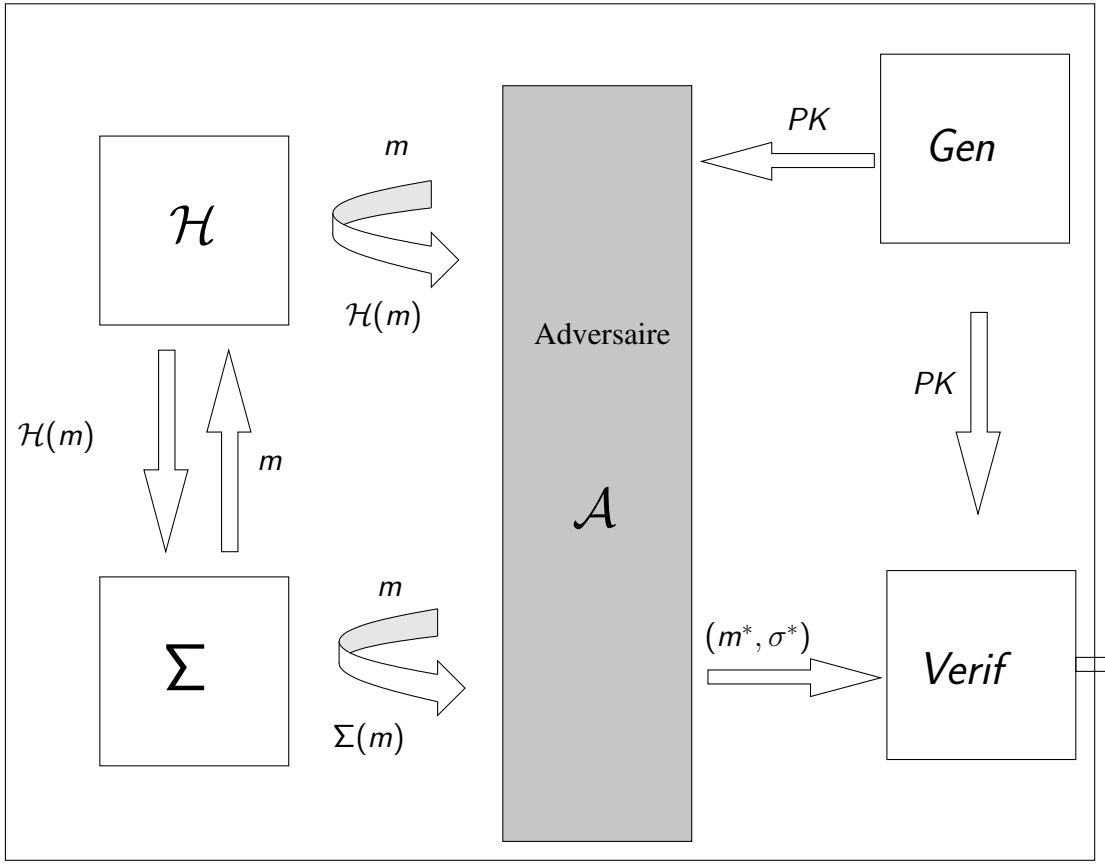
Dans le modèle de l'oracle aléatoire, un $(\tau, q_\Sigma, q_\mathcal{H})$-adversaire EF-CMA $\mathcal{A}$:

- a accès à un oracle de signature Σ (q_Σ requêtes),
- a accès à un oracle de hachage $\mathcal{H}$ ($q_\mathcal{H}$ requêtes),
- renvoie en temps au plus τ une forge (m^*, σ^*).

La forge (m^*, σ^*) est valide si :

- $Verify_S(m^*, \sigma^*, PK) = 1$
- σ^* n'a jamais été renvoyé par Σ.

Jeu de sécurité EF-CMA dans le ROM

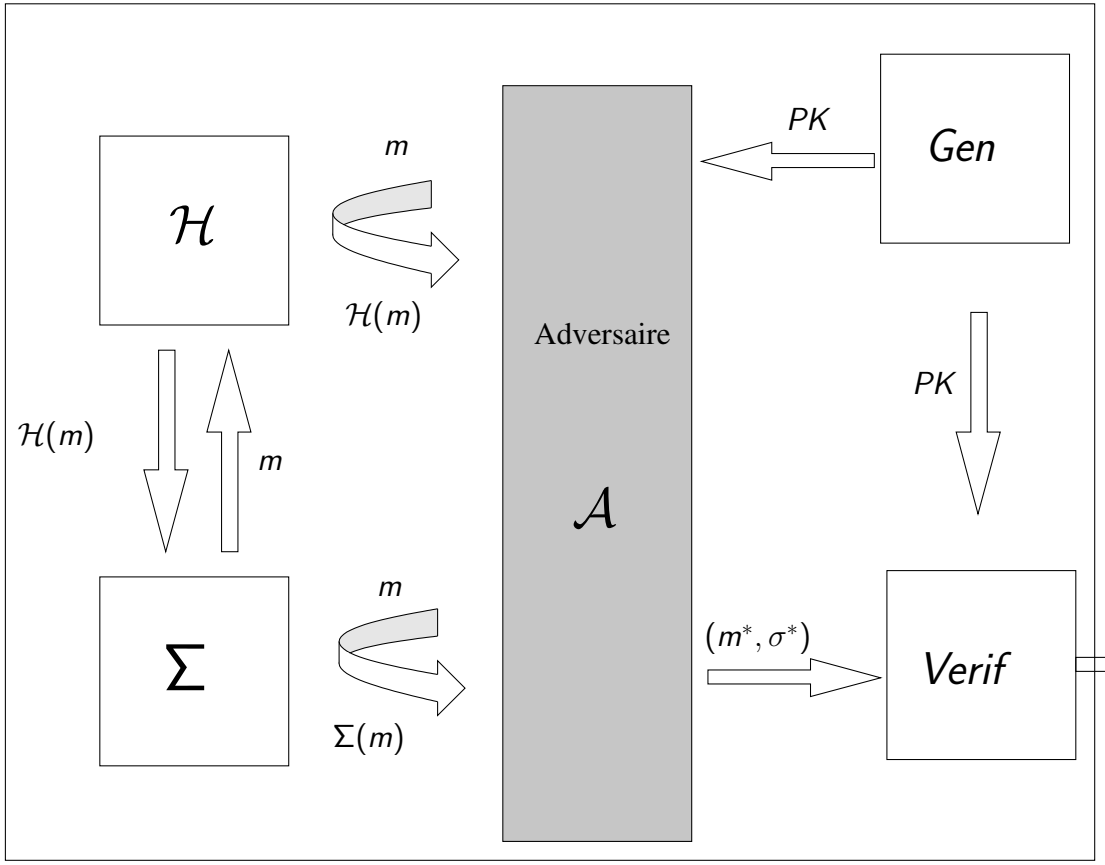


$$Succ_S^{EF-CMA}(\mathcal{A})$$

$$=$$

$$\Pr[Verif = 1]$$

Jeu de sécurité EF-CMA dans le ROM



$$Succ_S^{EF-CMA}(\mathcal{A}) = \Pr[Verif = 1]$$

Signature fondée sur la théorie des codes

Notations

- (n, k)-code binaire $\mathcal{C}$: sous-espace vectoriel de $\mathbb{F}_2^n$ de dimension k
- Décrit par une *matrice* $(n-k) \times k$ *dite de parité* H
- Le syndrome s d'un mot $x \in \mathbb{F}_2^n$ est $s = Hx^T \in \mathbb{F}_2^{n-k}$
- $\mathcal{C} = \{x \in \mathbb{F}_2^n | Hx^T = 0\}$
- Poids de Hamming d'un mot x : nombre de positions non nulles dans le mot. Noté $\mathrm{wt}(x)$.
- *Décoder* un syndrome $s \in \mathbb{F}_2^{n-k}$: retrouver $e \in \mathbb{F}_2^n$ de poids inférieur à t tel que $He^T = s$.

Codes de Goppa

- V. D. Goppa [Prob. Inf. Transm., 1970]
- Utilisés en cryptographie pour les chiffrements McEliece & Niederreiter.
- Longueur $n = 2^m$, Dimension $k = n - mt$, Décode t erreurs.
- Probabilité de décodage d'un syndrome aléatoire $\frac{1}{t!}$.
- Décodage efficace (Algorithme de Patterson, Algorithme de Berlekamp-Massey)

Problèmes difficiles

Goppa Code Distinguishing (GD) [Sendrier, 2002]

Entrée : Une matrice H binaire $(n-k) \times n$
Sortie : H est-elle la matrice de parité d'un code de Goppa permuté ou d'un code aléatoire ?

Problèmes difficiles

Goppa Code Distinguishing (GD) [Sendrier, 2002]

Entrée : Une matrice H binaire $(n-k) \times n$
Sortie : H est-elle la matrice de parité d'un code de Goppa permuté ou d'un code aléatoire ?

Définition

Un distingueur $\mathcal{D}$ est un algorithme prenant une matrice H en entrée et renvoyant un bit.

$$Adv^{GD}(\mathcal{D}) = \left| \Pr[H \xleftarrow{R} Gop(n,k) : \mathcal{D}(H) = 1] - \Pr[H \xleftarrow{R} Bin(n,k) : \mathcal{D}(H) = 1] \right|$$

Problèmes difficiles

Goppa Code Distinguishing (GD) [Sendrier, 2002]

Entrée : Une matrice H binaire $(n-k) \times n$
Sortie : H est-elle la matrice de parité d'un code de Goppa permuté ou d'un code aléatoire ?

Définition

Un distingueur $\mathcal{D}$ est un algorithme prenant une matrice H en entrée et renvoyant un bit.

$$Adv^{GD}(\mathcal{D}) = \left|\Pr[H \xleftarrow{R} Gop(n,k) : \mathcal{D}(H) = 1] - \Pr[H \xleftarrow{R} Bin(n,k) : \mathcal{D}(H) = 1]\right|$$

Définition

GD est dit $(\tau_{GD}, \epsilon_{GD})$-difficile si, pour tout distingueur $\mathcal{D}$ s'exécutant en temps au plus τ_{GD}, $Adv^{GD}(\mathcal{D}) \leq \epsilon_{GD}$

Problèmes difficiles

Goppa Parameterized Bounded Decoding problem (GPBD)

Entrées : Une matrice H binaire $(n-k) \times n$ et un syndrome $s \in {\mathbb{F}_2}^{n-k}$
Question : Existe-t-il un mot $x \in \mathbb{F}_2^n$ tel que $\mathrm{wt}(x) \leq \frac{n-k}{\log_2 n}$ et $Hx^T = s$?

- NP-difficile [Finiasz, 2004]

Problèmes difficiles

Goppa Parameterized Bounded Decoding problem (GPBD)

Entrées : Une matrice H binaire $(n-k) \times n$ et un syndrome $s \in {\mathbb{F}_2}^{n-k}$
Question : Existe-t-il un mot $x \in \mathbb{F}_2^n$ tel que $\mathrm{wt}(x) \leq \frac{n-k}{\log_2 n}$ et $Hx^T = s$?

Version Calculatoire (cGPBD)

Entrées : Une matrice H binaire $(n-k) \times k$ et un syndrome $s \in {\mathbb{F}_2}^{n-k}$
Sortie : Un mot $x \in \mathbb{F}_2^n$ tel que $\mathrm{wt}(x) \leq t = \frac{n-k}{\log_2 n}$ et $Hx^T = s$

Problèmes difficiles

Goppa Parameterized Bounded Decoding problem (GPBD)

Entrées : Une matrice H binaire $(n-k) \times n$ et un syndrome $s \in {\mathbb{F}_2}^{n-k}$
Question : Existe-t-il un mot $x \in \mathbb{F}_2^n$ tel que $\mathrm{wt}(x) \leq \frac{n-k}{\log_2 n}$ et $Hx^T = s$?

Version Calculatoire (cGPBD)

Entrées : Une matrice H binaire $(n-k) \times k$ et un syndrome $s \in {\mathbb{F}_2}^{n-k}$
Sortie : Un mot $x \in \mathbb{F}_2^n$ tel que $\mathrm{wt}(x) \leq t = \frac{n-k}{\log_2 n}$ et $Hx^T = s$

$$Succ^{GPBD}(\mathcal{D}) = \Pr\left[Hx^T = s \text{ et } \mathrm{wt}(x) \leq t \mid x \leftarrow \mathcal{D}(H, s)\right]$$

Définition

cGPBD est dit $(\tau_{GPBD}, \epsilon_{GPBD})$-difficile si pour tout décodeur $\mathcal{D}$ s'exécutant en temps au plus τ_{GPBD}, $Succ^{GPBD}(\mathcal{D}) \leq \epsilon_{GPBD}$.

Schéma de Courtois, Finiasz & Sendrier : Génération

$\mathrm{Gen}_{CFS}(1^\kappa)$:

- Choisir les paramètres n et k en fonction de 1^κ
- $\mathcal{C}_0$: Code de Goppa binaire de matrice de parité H_0.
- U : matrice non singulière $(n-k) \times (n-k)$ aléatoire.
- P : matrice de permutation $n \times n$ aléatoire.
- $t = \frac{n-k}{\log_2 n}$.
- h : fonction de hachage $\{0,1\}^* \longrightarrow \mathbb{F}_2^{n-k}$.
- Renvoyer $(H = UH_0P, t, h)$ comme clef publique et (H_0, U, P) comme clef secrète

Schéma de Courtois, Finiasz & Sendrier : Signature

$\mathrm{Sign}_{CFS}(m, H_0, U, P)$:

0. $i \leftarrow 0$
1. $i \leftarrow i + 1$
2. $s \leftarrow h(m \| i)$
3. Décoder $U^{-1}s$ en utilisant H_0 en un mot x_0 tel que $\mathrm{wt}(x_0) \leq t$
4. Si le décodage échoue, retourner à l'étape 1
5. $x \leftarrow x_0 P^{-1}$
6. Renvoyer (x, i) comme signature

Schéma de Courtois, Finiasz & Sendrier : Vérification

$\text{Verify}_{CFS}(m, x', i', H)$:

1. $s' \leftarrow H{x'}^T$
2. $s \leftarrow h(m \| i')$
3. La signature est valide si $\begin{cases} s' = s \\ \text{wt}(x') \leq t \end{cases}$

Modification du schéma

Schéma général de preuve

Objectif

Utiliser un $(\tau, q_\Sigma, q_\mathcal{H})$-adversaire EF-CMA $\mathcal{A}^{\Sigma,\mathcal{H}}$ pour résoudre un problème difficile et ainsi majorer sa probabilité de succès.

Moyen

- Simuler les oracles Σ et $\mathcal{H}$
- Tenter de deviner la requête à $\mathcal{H}$ destinée à la forge
- Substituer à la réponse le “challenge” du problème difficile

Cas de CFS

- Simuler Σ
- Simuler $\mathcal{H}$

Cas de CFS

- Simuler Σ

$$\Sigma(m) = (x, i) \quad \text{tels que} \begin{cases} Hx^T = \mathcal{H}(m\|i) \\ \mathrm{wt}(x) \leq t \end{cases}$$

- Simuler $\mathcal{H}$

Cas de CFS

- Simuler Σ

$$\Sigma(m) = (x, i) \quad \text{tels que} \begin{cases} Hx^T = \mathcal{H}(m\|i) \\ \mathrm{wt}(x) \leq t \end{cases}$$

- Simuler $\mathcal{H}$
  - $\mathcal{H}(m, 1), \ldots, \mathcal{H}(m, i-1)$ doivent être non décodables
  - $\mathcal{H}(m, i)$ doit être décodable
  - $\mathcal{H}(m, i+1), \mathcal{H}(m, i+2), \ldots$ peuvent être choisis aléatoirements

Cas de CFS

- Simuler Σ

$$\Sigma(m) = (x, i) \quad \text{tels que} \begin{cases} Hx^T = \mathcal{H}(m\|i) \\ \mathrm{wt}(x) \leq t \end{cases}$$

- Simuler $\mathcal{H}$
  - $\mathcal{H}(m,1), \ldots, \mathcal{H}(m, i-1)$ doivent être non décodables
  - $\mathcal{H}(m,i)$ doit être décodable
  - $\mathcal{H}(m,i+1), \mathcal{H}(m,i+2), \ldots$ peuvent être choisis aléatoirements

Cas de CFS

- Simuler Σ

$$\Sigma(m) = (x, i) \quad \text{tels que} \begin{cases} Hx^T = \mathcal{H}(m \| i) \\ \mathrm{wt}(x) \leq t \end{cases}$$

- Simuler $\mathcal{H}$
  - $\mathcal{H}(m, 1), \ldots, \mathcal{H}(m, i-1)$ doivent être non décodables
  - $\mathcal{H}(m, i)$ doit être décodable
  - $\mathcal{H}(m, i+1), \mathcal{H}(m, i+2), \ldots$ peuvent être choisis aléatoirements

Puisqu'une simulation ne peut produire de tels syndromes, cette information pourrait-elle être exploitée par un adversaire ?

Schéma de Courtois, Finiasz & Sendrier modifié

Les algorithmes de génération des clefs et de vérification restent inchangés.

$\mathrm{Sign}_{mCFS}(m, H_0, U, P)$:

1. $i \xleftarrow{R} \{1, \ldots, 2^{n-k}\}$
2. $s \leftarrow h(m \| i)$
3. Décoder $U^{-1}s$ en utilisant H_0 en un mot x_0 tel que $\mathrm{wt}(x_0) \leq t$
4. Si le décodage échoue, retourner à l'étape 1
5. $x \leftarrow x_0 P$
6. Renvoyer (x, i) comme signature

Remarque

Modifier ainsi le schéma augmente la taille des signatures qui passent de $\log_2 t!$ *bits (environ 19 bits avec les paramètres originaux) à* $n - k = mt$ *bits (144 bits).*

Preuve de sécurité

Synopsis

Hypothèses

GD et *cGPBD* sont respectivement $(\tau_{GD}, \epsilon_{GD})$ et $(\tau_{GPBD}, \epsilon_{GPBD})$-difficiles.

On veut résoudre une instance $< H^*, s^* >$ de cGPBD en utilisant un $(\tau, q_\Sigma, q_\mathcal{H})$-adversaire $\mathcal{A}$

Synopsis

Hypothèses

GD et *cGPBD* sont respectivement $(\tau_{GD}, \epsilon_{GD})$ et $(\tau_{GPBD}, \epsilon_{GPBD})$-difficiles.

On veut résoudre une instance $< H^*, s^* >$ de cGPBD en utilisant un $(\tau, q_\Sigma, q_\mathcal{H})$-adversaire $\mathcal{A}$

- Donner H^* comme clef publique à $\mathcal{A}$
  - Altère la probabilité de succès de $\mathcal{A}$ d'au plus ϵ_{GD}
- Initialiser les simulations des oracles Σ et $\mathcal{H}$
- Exécuter $\mathcal{A}^{\Sigma,\mathcal{H}}(H)$
- Si la simulation réussi, $\mathcal{A}$ renvoie (m^*, i^*, x^*) tels que $H^*(x^*)^T = s^*$ et $\mathrm{wt}(x^*) \leq t$.

Simulation des oracles

- La simulation fixe un indice de signature $\Lambda(m)$ pour chaque message sur lequel porte une requête (ou à chaque fois qu'une nouvelle signature est demandée pour m)
- $\mathcal{H}$ interrogé sur m et $i \neq \Lambda(m)$:
  - Tirer un syndrome aléatoire $s \xleftarrow{R} \mathbb{F}_2^{n-k}$
  - Identique à l'oracle aléatoire
- $\mathcal{H}$ interrogé sur m et $\Lambda(m)$:
  - Tirer $x \xleftarrow{R} \{x \in \mathbb{F}_2^n \mid \mathrm{wt}(x) \leq t\}$
  - Calculer $s = Hx^T$
  - Stocker x et s
  - $\mathcal{H}(m, i)$ renvoie s
- L'oracle de signature est alors capable de "décoder" le résultat de $\mathcal{H}(m, \Lambda(m))$
- La c-ième requête de hachage renvoie le challenge s^*.

Probabilité de succès

- La simulation augmente légèrement la probabilité qu'un haché soit décodable
- Risque d'incohérence lorsque $\mathcal{A}$ fait une requête à $\mathcal{H}$ déjà effectuée par Σ

$$\Pr[\mathit{echec}] \leq 2 - \left(1 - \frac{1}{2^{n-k}}\right)^{q_{\mathcal{H}}+q_\Sigma} - \left(1 - \frac{q_\Sigma}{2^{n-k}}\right)^{q_{\mathcal{H}}}$$

Probabilité de succès

- $\mathcal{A}$ effectue sa forge sur la c-ième requête de hachage avec une probabilité $\frac{1}{q_{\mathcal{H}}+q_{\Sigma}+1}$
- $\mathcal{A}$ détecte le changement de clef avec une probabilité au plus ϵ_{GD}
- La simulation réussi avec une probabilité au plus ϵ_{GPBD}

$$Succ_{mCFS}^{EF-CMA}(\mathcal{A}) \leq (q_{\mathcal{H}} + q_{\Sigma} + 1)\epsilon_{GPBD} + \epsilon_{GD} + Pr[echec]$$

Conclusion

Conclusion

La modification apportée au schéma nous a permis :

- d'éxhiber une preuve de sécurité du schéma.
- de quantifier sa sécurité.

La réduction n'est pas fine :

- Existe-t-il une meilleur réduction ?
- Une étude de l'optimalité est nécessaire.

Le schéma peut servir de "brique de base" pour d'autres constructions prouvées sûrs :

- Identification et Signature fondées sur l'identité [Cayrel, Gaborit, Galindo, Girault, 2008]