Chapitre IV

Performances des Schémas

“Rien ne marque tant le jugement solide d’un homme, que de savoir choisir entre les grands inconvénients.”

J.-F. de Gondi

Dans ce chapitre, les performances des différents schémas d’identification, décrits dans le chapitre précédent, sont évalués. Une étude comparative entre ces derniers et les schémas SD de M. Girault, S. Harari et J. Stern, est effectuée. Cette étude se termine par une comparaison avec d’autres schémas ne dépendant pas du problème SD.

DANS un cadre pratique, le prouveur peut être assimilé à un support portable de type carte à puces. Sa puissance de calcul, et sa capacité de stockage sont donc limitées. De ce fait, nous nous sommes particulièrement intéressés à la valeur de ces deux quantités. De plus, un dernier paramètre, non négligeable, est le nombre de bits échangés lors d’un processus d’identification ; c’est le *débit de transaction* du schéma. En particulier, plus ce débit est important, et plus le coût de la communication entre le prouveur et le vérifieur sera important.

Les deux premières contraintes, mentionnées ci-dessus, ne s’appliquent pas au vérifieur, qui, dans certains cas, peut être assimilé à un serveur possédant d’importants moyens de calculs.

Pour tous les schémas qui ont été proposés, les calculs s’effectuent sur le corps $\mathbb{F}_2$ et n’utilisent, uniquement, que les opérateurs logiques `xor` et `and`. Si cela parait évident pour le schéma G-SD, il en est de même pour les schémas MR G-SD, CR G-SD et T-SD, puisque, d’après les propositions 5.1.1 et 5.2.5, du chapitre i, la multiplication de deux éléments du corps $\mathbb{F}_{2^m}$, consiste en une série d’additions binaires.

§1. Fonction de hachage et Permutation

Tous les schémas, proposés, utilisent une fonction de hachage. D'après le chapitre ii, §4, cette fonction doit simuler au mieux la sortie d'un oracle aléatoire. D'après M. Bellare et P. Rogaway, un bon candidat est, par exemple, la fonction $< x >= MD5(xx)$. Seuls les principes généraux de cette fonction sont rappelés, ainsi qu'une estimation du nombre d'opérations nécessaires au calcul de $MD5(x)$.

$MD5$ manipule en entrée un bloc de 512 bits et renvoie un bloc de 128 bits.

1. Tout d'abord, le bloc x est complété afin qu'il ne manque que 64 bits pour que sa longueur soit un multiple de 512,
2. Ensuite, une représentation sur 64 bits, de la longueur du message, est jointe au résultat. On dispose donc d'une chaîne y composée de m blocs x de longueur 512,
3. Quatre variables A, B, C et D de 32 bits sont initialisées,
4. Pour chaque bloc m de 512 bits, 4 rondes sont effectuées. Chacune d'entre elles consiste en 16 exécutions d'une opération calculant une fonction non linéaire de trois des variables A, B, C et D. Puis d'autres opérations sont effectuées afin de changer les valeurs de ces variables. Il y a quatre fonctions non linéaires, une pour chaque ronde :

$$\begin{array}{lcl} F(X,Y,Z) & = & (X \wedge Y) \vee (\neg X \wedge Z) \\ G(X,Y,Z) & = & (X \wedge Z) \vee (Y \wedge \neg Z) \\ H(X,Y,Z) & = & X \oplus Y \oplus Z \\ I(X,Y,Z) & = & Y \oplus (X \vee \neg Z) \end{array}$$

Une étude de la fonction $MD5$, montre que l'exécution de l'étape 4 nécessite en moyenne $2^{13.7}$ opérations binaires.

Pour chacun des schémas proposés, le coût du calcul des mises en gage c_1, c_2 et c_3, sera séparé de celui des autres opérations effectuées par le prouveur. En effet, on peut très bien utiliser d'autres types de fonction de hachage, ce qui changera la valeur de ce paramètre.

D'après les travaux de M. Girault et de J. Stern [GiSt], la longueur des valeurs hachées, doit être au moins de 128 bits afin d'éviter une attaque basée sur le paradoxe des anniversaires. Les schémas SD de S. Harari et de M. Girault, n'utilisent pas de fonction de hachage. Nous verrons cependant que leur complexité de calcul est nettement plus élevée que celle des autres schémas, même si on tient compte du coût de calcul des mises en gage.

Tous les schémas SD, ainsi que ceux du chapitre précédent, utilisent une permutation. Afin de réduire le débit de transaction, lors de l'envoi de celle-ci, on suppose qu'un générateur de permutation aléatoire est utilisé.

En fonction d'un germe de ℓ_σ bits, le générateur construit une permutation σ. La mise au point de permutations parfaitement aléatoires est un problème difficile. Le lecteur pourra consulter les travaux de M. Luby et C. Rackoff [LuRa] et de J. Patarin [Pat$_1$, Pat$_2$], concernant l'élaboration de générateurs basés sur le schéma du DES.

La complexité de calcul nécessaire à la génération d'une permutation n'a pas été pris en compte, lors de l'évaluation des performances des différents schémas, puisque ces derniers utilisent tous un tel générateur. Pour les paramètres minimaux, que nous étudierons, le germe sera composé de 120 bits quel que soit le schéma considéré. Ainsi le nombre d'opérations, nécessaires à l'élaboration d'une permutation, sera le même pour tous les schémas.

§2. Notations, Conventions

- $C_{<>}$ représentera le nombre d'opérations effectuées pour le calcul des mises en gage, lors d'un processus d'identification,
- h représentera le nombre de bits renvoyé par la fonction de hachage,
- Soit σ une permutation, on notera ℓ_σ le nombre de bits du germe représentant la permutation σ,
- Il sera supposé que le corps $\mathbb{F}_{2^m}$ est engendré par un trinôme irréductible, $p(x)$, de degré m (cf. Annexe A),
- Soit α une racine de $p(x)$, on utilisera comme base de $\mathbb{F}_{2^m}$, la base $\beta = \{1, \alpha, \dots, \alpha^{m-1}\}$.
- Le nombre d'étapes garantissant la fiabilité d'un protocole d'identification, sera notée r,
- Ω représentera le plus petit facteur de travail nécessaire à la cryptanalyse du schéma étudié,

Pour chacun des schémas décrits précédemment, nous avons donc étudié :

- La complexité des calculs effectués par le prouveur,
- Le débit de transaction du schéma,
- La capacité de stockage nécessaire au prouveur. Cette dernière est déterminée par les paramètres que doit constamment stocker le prouveur afin de pouvoir exécuter un processus d'identification. Si ce dernier est une carte à puces, ceci représente la quantité de ROM à intégrer sur la carte.

Il ne sera pas donné d'expression de $C_{<>}$, en fonction de n ou de k, puisque cette quantité ne peut se calculer que lorsque l'on connaît réellement la taille des arguments de la fonction de hachage. La valeur de $C_{<>}$ ne sera donc détaillée que lorsque les paramètres à utiliser, pour un schéma donné, seront fixés.

Tous les schémas seront comparés en considérant la taille réelle de la matrice utilisée. Ceci permettra de mettre en évidence ceux pour lesquels l'utilisation d'un générateur aléatoire est indispensable, pour stocker la matrice, si le support envisagé est une carte à puces.

§3. Le Schéma G-SD

• ROM

Les quantités secrètes du prouveur sont :

- m, un vecteur binaire de longueur k,
- e, un vecteur binaire de longueur n,
- mG, un vecteur binaire de longueur n.

De plus, le prouveur doit stocker la matrice $G(k, n)$, afin de pouvoir calculer la quantité uG, u étant un vecteur quelconque. Sans affecter la sécurité du système, on peut toujours supposer que G est sous forme systématique, i.e $G = (I_k \mid M)$, donc seule la partie M est stockée.

Ainsi, le nombre de bits devant être stockés par le prouveur est :

$$k + 2n + (n - k)k.$$

• Complexité des calculs

Pour chaque étape du protocole, voici les quantités calculées par le prouveur ainsi que le nombre d'opérations binaires qu'elles nécessitent. Tout d'abord, afin de pouvoir transmettre les mises en gage c_1, c_2 et c_3, le prouveur calcule :

- uG : $(2k - 1)(n - k)$ opérations binaires,
- $uG + mG$: n additions binaires,
- $uG + x$: n additions binaires,

Puis

- Si $b = 0$, A calcule : $u + m$ $\rightarrow$ k additions binaires,
- Si $b = 1$, A calcule : $uG + mG$ $\rightarrow$ $(2k - 1)(n - k) + n$ opérations binaires,

Un processus d'identification nécessite donc en moyenne

$$r(\frac{8}{3}k(n - k) + n + \frac{5}{3}k)$$

opérations binaires.

Remarque : Après l'envoi des mises en gage, il est supposé que le prouveur ne garde pas en mémoire, la quantité uG, puisque dans deux cas sur trois, il n'aura plus à utiliser ce résultat. Dans le cas $b = 1$, il doit donc recalculer cette quantité. Ceci permet de minimiser la quantité de mémoire vive nécessaire sur la carte.

• Débit de transaction

A chaque étape, les quantités transmises sont :

- c_1, c_2, c_3 : $3h$ bits
- Si $b = 0$, : $u + m$ et σ $\to k + \ell_\sigma$ bits,
- Si $b = 1$, : $(u + m)G\sigma$ et $e\sigma$ $\to 2n$ bits,
- Si $b = 2$, : u et σ $\to k + \ell_\sigma$ bits,

Le débit moyen de transaction du schéma est donc

$$r(3h + \frac{2}{3}(k + \ell_\sigma + n)).$$

En résumé on a

	Schéma G-SD
ROM	$k + 2n + (n - k)k$ bits
Complexité de calcul	$r(\frac{8}{3}k(n - k) + n + \frac{5}{3}k)$
Débit de transaction	$r(3h + \frac{2}{3}(k + \ell_\sigma + n))$ bits

Tab. 1. Complexité de G-SD

Tout d'abord afin de démontrer l'efficacité du schéma par rapport aux autres schémas SD (cf. §7), voici les performances de G-SD, pour la contrainte $n = 2k$. Les paramètres minimaux utilisés sont ceux suggérés par J. Stern pour son schéma. Pour Ω de l'ordre de 2^{70}, ces paramètres sont :

$$n = 512, k = 256, \ell_\sigma = 120, h = 128, r = 35.$$

	Schéma G-SD
Etapes	35
ROM	66816 bits
Complexité de calcul	$2^{22.5}$
Débit de transaction	34160 bits
$C_{<>}$	$2^{20.4}$

Tab. 2. Performances de G-SD

Cependant, nous avons montré que pour le schéma G-SD, il était possible de réduire la dimension de la matrice G, afin d'améliorer les performances générales. Ainsi, pour $\Omega > 2^{60}$, les paramètres minimaux (optimaux), à utiliser, sont

$$n = 512, k = 120, \ell_\sigma = 120, h = 128, r = 35.$$

	Schéma G-SD (optimal)
Etapes	35
ROM	48184 bits
Complexité de calcul	2^{22}
Débit de transaction	30986 bits
$C_{<>}$	$2^{20.4}$

Tab. 3. Performances de G-SD optimal

Pour ces paramètres on obtient par rapport au schéma G-SD initial :

- un gain de 30% en ROM,
- un gain de 10% au niveau du débit de transaction,
- une complexité de calcul 3 fois moins importante.

§4. Le Schéma MR G-SD

Ce schéma est le même que le schéma G-SD. La seule différence provient du fait que la matrice G utilisée est représentée par deux éléments du corps fini $\mathbb{F}_{2^k}$. Cette optimisation permet de diminuer la ROM et la complexité de calcul du prouveur. Le débit de transaction théorique reste inchangé.

Nous rappelons que ce schéma n'est valable que pour $n = 2k$. Tous les résultats seront donc donnés en fonction du paramètre k.

• ROM

Les quantités secrètes du prouveur sont :

- m, un vecteur binaire de longueur k,
- e, un vecteur binaire de longueur $2k$,
- (my_1, my_2), un vecteur binaire de longueur $2k$.

Remarque : nous rappelons que le prouveur ne stocke pas les éléments y_1 et y_2, car les produits entre y_i et u, un élément quelconque de $\mathbb{F}_{2^k}$, sont effectués en utilisant la méthode de Berlekamp.

Le prouveur doit donc stocker $5k$ bits.

• Complexité des calculs

Pour chaque étape du protocole, voici les quantités calculées par le prouveur ainsi que le nombre d'opérations binaires qu'elles nécessitent. Tout d'abord, afin de pouvoir transmettre les mises en gage c_1, c_2 et c_3, le prouveur calcule :

- (uy_1, uy_2) : $k\omega_\beta(y_1)$+ $k\omega_\beta(y_2)$ additions binaires,
- $(uy_1, uy_2) + (my_1, my_2)$: $2k$ additions binaires,
- $(uy_1, uy_2) + x$: $2k$ additions binaires,

Puis

- Si $b = 0$, A calcule : $u + m$ → k additions binaires,
- Si $b = 1$, A calcule : $(uy_1, uy_2) + (my_1, my_2)$ → $k\omega_\beta(y_1) + k\omega_\beta(y_2)$ $+2k$ additions binaires,

En posant, $t = \omega_\beta(y_1) = \omega_\beta(y_2)$, un processus d'identification nécessite en moyenne

$$rk(\frac{8}{3}t + 5)$$

opérations binaires.

Remarque : Comme pour le schéma G-SD, il est supposé qu'après l'envoi des mises en gage, le prouveur ne garde pas en mémoire, la quantité (uy_1, uy_2).

A k fixé, la complexité de calcul du prouveur dépend essentiellement de t. C'est donc ce paramètre qu'il faudra minimiser pour augmenter les performances du schéma tout en tenant compte des remarques faites au paragraphe 8 du chapitre iii.

En résumé on a

	Schéma MR G-SD
ROM	$5k$ bits
Complexité de calcul	$rk(\frac{8}{3}t+5)$
Débit de transaction	$r(3h+\frac{2}{3}(k+\ell_\sigma+n))$ bits

Tab. 4. Complexité de MR G-SD

Pour $k=255$, il existe des trinômes irréductibles de degré k. Pour un seuil de sécurité équivalent à celui du schéma G-SD, les paramètres minimaux à utiliser sont

$$n=510, k=255, \ell_\sigma=120, h=128, r=35, t=80.$$

	Schéma MR G-SD
Etapes	35
ROM	1275 bits
Complexité de calcul	$2^{20.9}$
Débit de transaction	34090 bits
$C_{<>}$	$2^{20.4}$

Tab. 5. Performances de MR G-SD

§5. Le Schéma CR G-SD

Ce schéma n'est valide que lorsque $n=2k$. Tous les résultats seront donc donnés en fonction de k. De plus, étant donné qu'un processus d'identification fait intervenir des calculs dans les corps $\mathbb{F}_{2^k}$ et $\mathbb{F}_{2^{2k}}$, la base de $\mathbb{F}_{2^k}$ *(resp. de* $\mathbb{F}_{2^{2k}}$*)* sera notée β_1 *(resp. de* β_2*)*

• ROM

Les quantités secrètes du prouveur sont :

- π_2^{-1}, une permutation de l'ensemble $\{1,\ldots,2k\}$,
- α^k, un élément de $\mathbb{F}_{2^k}$,
- m, e, u et mG, des vecteurs binaires de longueur $2k$.

Remarque : l'élément α^k est stockée car le prouveur effectue une multiplication classique dans le corps $\mathbb{F}_{2^k}$. Ce dernier effectue aussi une multiplication, dans le corps $\mathbb{F}_{2^{2k}}$, entre ρ, une quantité secrète, et un élément quelconque de $\mathbb{F}_{2^{2k}}$. La méthode de Berlekamp permet d'effectuer ce calcul sans stocker ρ.

Le nombre de bits devant être stockés par le prouveur est :

$$\ell_{\pi_2^{-1}} + 9k.$$

• Complexité des calculs

Pour chaque étape du protocole, voici les quantités calculées par le prouveur ainsi que le nombre d'opérations binaires qu'elles nécessitent. Tout d'abord, afin de pouvoir transmettre les mises en gage c_1, c_2 et c_3, le prouveur calcule :

- $u + v$: $2k$ additions binaires,
- $(\eta v_1, \eta v_2)$: $2(k(\omega_{\beta_1}(\eta) - 1) + I_{\beta_1}(\eta))$ additions binaires,
- $y\rho^{-1}$: $2k\omega_{\beta_2}(\rho^{-1})$ additions binaires,
- $mG + v$: $2k$ additions binaires,
- $x + v$: $2k$ additions binaires.

Puis

- Si $b = 0$, A calcule : $(\eta w_1, \eta w_2) + m \rightarrow 2(k(\omega_{\beta_1}(\eta) - 1) + I_{\beta_1}(\eta)) + 2k$ additions binaires,
- Si $b = 1$, A calcule : $v + mG \rightarrow 2k$ additions binaires,
- Si $b = 2$, A calcule : $(\eta w_1, \eta w_2) \rightarrow 2(k(\omega_{\beta_1}(\eta) - 1) + I_{\beta_1}(\eta))$ additions binaires.

Ainsi en posant $t_1 = \omega_{\beta_1}(\eta)$, $t_2 = \omega_{\beta_2}(\rho^{-1})$ et en majorant $I_{\beta_1}(\eta)$ *(resp.* $I_{\beta_2}(\eta)$ *)* par $k-1$ *(resp.* $2k-1$*)*, un processus d'identification nécessite en moyenne

$$r(2k(\frac{11}{3} + \frac{5}{3}t_1 + t_2) - \frac{10}{3})$$

opérations binaires.

Remarque : nous rappelons que, dans un premier temps, la quantité v $(= \tau G)$ est calculée par le vérifieur, puis envoyée au prouveur. Elle intervient, lors de l'envoi des mises en gage et dans la questions $b = 1$. Il est donc supposé que cette valeur est stockée par le prouveur à chaque étape du protocole.

• Débit de transaction

A chaque étape, les quantités transmises sont :

- $y\rho^{-1}$: $2k$ bits
- z : $2k$ bits
- c_1, c_2, c_3 : $3h$ bits

- Si $b = 0$, : $\tau + m$ et σ $\rightarrow 2k + \ell_\sigma$ bits,
- Si $b = 1$, : $(\tau + m)G\sigma$ et $e\sigma$ $\rightarrow 4k$ bits,
- Si $b = 2$, : τ et σ $\rightarrow 2k + \ell_\sigma$ bits,

Le débit moyen de transaction du schéma est donc

$$r(4k + 3h + \frac{2}{3}(4k + \ell_\sigma)).$$

En résumé on a

	Schéma CR G-SD
ROM	$9k + \ell_{\pi_2^{-1}}$ bits
Complexité de calcul	$r(2k(\frac{11}{3} + \frac{5}{3}t_1 + t_2) - \frac{10}{3})$
Débit de transaction	$r(4k + 3h + \frac{2}{3}(4k + \ell_\sigma))$

Tab. 6. Complexité de CR G-SD

Pour $k = 255$, il existe des trinômes irréductibles de degré 255 et 510. Pour Ω de l'ordre de 2^{70} les paramètres minimaux à utiliser sont

$$n = 510, k = 255, \ell_\sigma = \ell_{\pi_2^{-1}} = 120, h = 128, r = 35, t_1 = 20, t_2 = 20.$$

Voici les performances du schéma

	Schéma CR G-SD
Etapes	35
ROM	2415 bits
Complexité de calcul	$2^{19.9}$
Débit de transaction	75740 bits
$C_{<>}$	$2^{20.4}$

Tab. 7. Performances de CR G-SD

§6. Le Schéma T-SD

Ce schéma n'est valide que lorsque $n = 2k$. Tous les résultats seront donc donnés en fonction de k.

• ROM

Les quantités secrètes du prouveur sont :

- η, η^{-1} deux éléments de $\mathbb{F}_{2^{2k}}$,
- s un vecteur binaire de longueur $2k$,
- u un vecteur binaire de longueur $2k$,
- α^{2k} un élément de $\mathbb{F}_{2^{2k}}$,

Le prouveur doit donc stocker $10k$ bits.

• Complexité des calculs

Pour chaque étape du protocole, voici les quantités calculées par le prouveur ainsi que le nombre d'opérations binaires qu'elles nécessitent. Tout d'abord, afin de pouvoir transmettre les mises en gage c_1, c_2 et c_3, le prouveur calcule :

- $\varepsilon\eta$: $I_\beta(\varepsilon) + 2k(\omega_\beta(\varepsilon) - 1)$+2k additions binaires,
- $\eta^{-2^k} z + \varepsilon$: $I_\beta(\eta^{-2^k}) + 2k(\omega_\beta(\eta^{-2^k}) - 1)$ additions binaires,
- $u + u_1$: $2k$ additions binaires,
- $\rho_2\rho_1$: $I_\beta(\rho_2) + 2k(\omega_\beta(\rho_2) - 1)$ additions binaires,
- $\gamma\rho_1$: $I_\beta(\rho_1) + 2k(\omega_\beta(\rho_1) - 1)$ additions binaires,
- $\rho_2 u_2$: $I_\beta(\rho_2) + 2k(\omega_\beta(\rho_2) - 1)$ additions binaires,
- γu_1 : $I_\beta(u_1) + 2k(\omega_\beta(u_1) - 1)$ additions binaires,
- $\gamma u_1 + s$: $2k$ additions binaires.

Puis

- Si $b = 0$, A calcule : $\gamma u_1 + s \to 2k$ additions binaires,

Ainsi en posant

$$t = \omega_\beta(\eta^{-2^k}) = \omega_\beta(\rho_1) = \omega_\beta(\rho_2) = \omega_\beta(u_1) = \omega_\beta(\varepsilon),$$

et en majorant, pour tout $\xi \in \mathbb{F}_{2^{2k}}$, $I_\beta(\xi)$ par $2k - 1$, un processus d'identification nécessite en moyenne

$$r(12kt + \frac{2}{3}k - 6)$$

opérations binaires.

Remarque : Dans ce protocole, on a intérêt à ce que le prouveur stocke la quantité γu_1, car cette dernière intervient dans toutes les questions posées par le vérifieur. Sous cette hypothèse, le prouveur n'effectue aucun calcul lorsque $b = 1$ ou 2. Pour $b = 0$, il n'effectue qu'une addition binaire entre deux vecteurs de longueur $2k$.

• Débit de transaction

A chaque étape, les quantités transmises sont :

- $\varepsilon\eta$: $2k$ bits,
- z : $2k$ bits,
- $\rho_2\rho_1, \gamma\rho_1, \rho_2 u_2$: $6k$ bits,
- c_1, c_2, c_3 : $3h$ bits
- Si $b = 0$, : $\gamma u_1 + s$ et σ $\rightarrow 2k + \ell_\sigma$ bits,
- Si $b = 1$, : $(\gamma u_1)\sigma$ et $s\sigma$ $\rightarrow 4k$ bits,
- Si $b = 2$, : γu_1 et σ $\rightarrow 2k + \ell_\sigma$ bits,

Le débit moyen de transaction du schéma est donc

$$r(10k + 3h + \frac{2}{3}(4k + \ell_\sigma)).$$

En résumé on a

	Schéma T-SD
ROM	$10k$ bits
Complexité de calcul	$r(12kt + \frac{2}{3}k - 6)$
Débit de transaction	$r(10k + 3h + \frac{2}{3}(4k + \ell_\sigma))$

Tab. 8. Complexité de T-SD

Pour $k = 255$, il existe des trinômes irréductibles de degré 255. Pour Ω, de l'ordre de 2^{70}, nous suggérons d'utiliser comme paramètres minimaux

$$n = 510, k = 255, \ell_\sigma = 120, h = 128, r = 35, t = 20.$$

Voici les performances du schéma

	Schéma T-SD
Etapes	35
ROM	2550 bits
Complexité de calcul	2^{21}
Débit de transaction	129290 bits
$C_{<>}$	$2^{20.4}$

Tab. 9. Performances de T G-SD

Voici pour résumer cette étude, un tableau récapitulant les performances des divers schémas. Pour chaque paramètre étudié, le meilleur résultat obtenu apparaît en gras dans le tableau ci-dessous :

	G-SD	G-SD opt.	MR G-SD	CR G-SD	T-SD
Etapes	35	35	35	35	35
ROM	66816	48184	**1275**	2415	2550
Complexité de calcul	$2^{22.5}$	2^{22}	$2^{20.9}$	$\mathbf{2^{19.9}}$	2^{21}
Débit de transaction	34160	**30986**	34090	75740	129290
$C_{<>}$	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$

Tab. 10. Performances des différents schémas

Les schémas MR G-SD, CR G-SD et T-SD, sont des alternatives qui ont été proposées afin d'améliorer, à paramètres et seuils de sécurité équivalents, les performances du schéma G-SD (notamment en ce qui concerne la ROM et la complexité de calcul).

Il résulte de cette étude que le schéma MR G-SD réalise le meilleur compromis pour les trois paramètres étudiés. Par rapport au schéma G-SD, nous obtenons ainsi :

- une diminution de 98% de la taille de la ROM,
- une réduction de 53% de la complexité de calcul,
- un débit de transaction équivalent.

L'inconvénient du schéma provient de la forte structure de la matrice utilisée.

La variante CR G-SD propose une alternative en utilisant une matrice beaucoup moins structurée. Pour obtenir des performances proches de MR G-SD, une méthode a été proposée afin que le prouveur n'effectue plus aucun calcul faisant intervenir la matrice G. On obtient alors un schéma qui

. Par rapport à G-SD :

- réduit de 96% la taille de la ROM,
- diminue de 83% la complexité de calcul,
- possède un débit de transaction deux fois plus important.

. Par rapport à MR G-SD :

- double la taille de la ROM,
- diminue de 50% la complexité de calcul.
- possède un débit de transaction deux fois plus important,

Ainsi, par rapport à MR G-SD, l'introduction de l'aléatoire, dans la matrice G, se traduit par une augmentation de la taille de la ROM et du débit de transaction. De plus, le schéma possède un inconvénient dû au fait que chaque prouveur A est représenté, entre autres, par une matrice $\hat{G}_A$, de taille $(2k, 4k)$. Cette dernière doit être stockée par le vérifieur, ce qui représente environ $64Ko$ pour les paramètres suggérés (pour chaque prouveur).

Le schéma T-SD constitue une sorte d'intermédiaire entre les deux précédents schémas. Il utilise une matrice appartenant à une classe beaucoup plus large que celle du schéma MR G-SD, mais plus restreinte que celle du schéma CR G-SD. En contrepartie, par rapport à ce dernier schéma, il n'est plus nécessaire que le vérifieur dispose d'une importante quantité de mémoire. Cependant le schéma possède un débit de transaction élevé.

Par rapport au schéma G-SD, on obtient

- une réduction de 96% de la taille de la ROM,
- une diminution de 64% du nombre d'opérations effectuées par le prouveur,
- un débit de transaction $3,8$ fois plus important.

Par rapport au schéma MR G-SD

- la taille de la ROM est 2 fois plus importante,
- le nombre d'opérations est pratiquement identique,
- le débit de transaction est $3,8$ fois plus important.

Par rapport au schéma CR G-SD

- la taille de la ROM est augmenté de 4%,
- 2 fois plus de calculs sont effectués,
- le débit de transaction est $1,7$ fois plus élevé.

Les schémas ne sont pas “figés”, c'est-à-dire que, s'il existe un algorithme efficace pouvant résoudre le problème G-SD (ou SD), pour les paramètres suggérés, ce dernier peut évoluer afin d'utiliser, par exemple, les paramètres $n = 1024, k = 512, p = 120$. Cependant l'augmentation de ces paramètres rend de moins en moins praticable l'implémentation sur une carte à puces.

A titre d'exemple voici l'évolution du débit de transaction et de la quantité de ROM nécessaire, lorsque k varie et $n = 2k$. Nous n'avons étudié que ces deux paramètres puisque, mis à part le schéma G-SD, la complexité de calcul des autres schémas varie aussi en fonction du poids d'un certain élément. On note Θ, la valeur des deux paramètres étudiés. Voici l'évolution de cette quantité, en fonction de k.

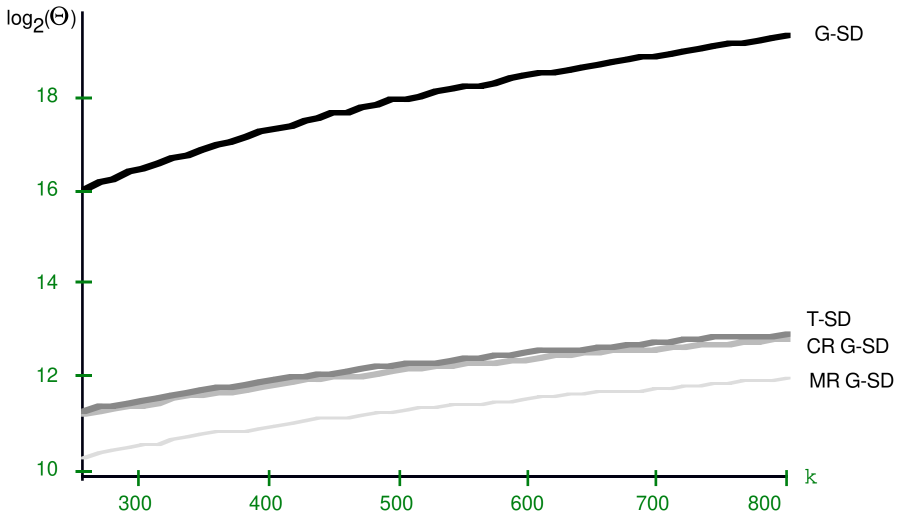

Fig. 11. Evolution de la taille de la ROM

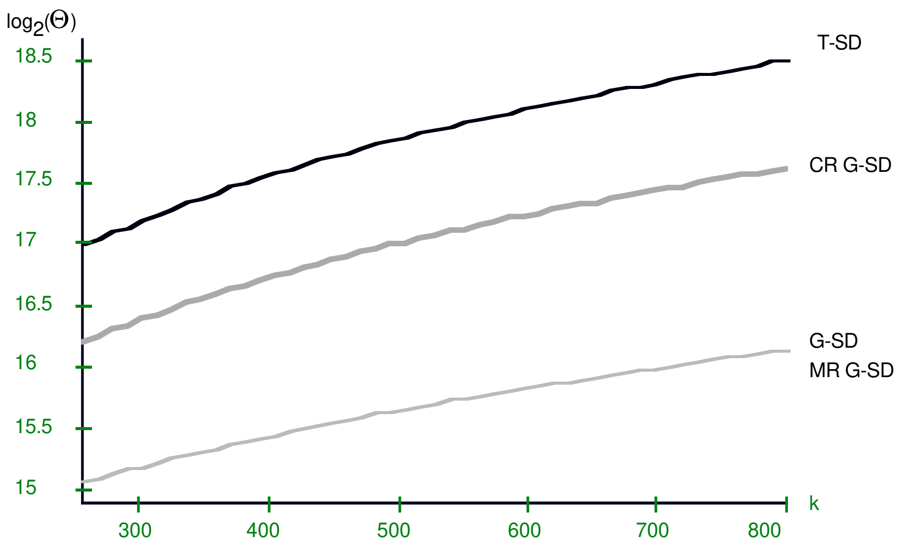

Fig. 12. Evolution du débit de transaction

§7. Comparaison avec les schémas SD

Afin de vérifier l'efficacité des schémas proposés, une comparaison a été effectuée avec les schémas SD de M. Girault, S. Harari et J. Stern. En premier lieu, voici dans l'ordre chronologique, une brève description de ces schémas, le lecteur trouvera de plus amples informations dans [Gir], [Har] et [Ste3].

7.1. Le schéma de S. Harari

Ce schéma a été élaboré en 1988. C'est le premier protocole d'identification utilisant le problème SD.

Quantités publiques de chaque prouveur	: $H(k,n)$ une matrice de contrôle d'un code de poids minimum impair μ,
Quantités secrètes de chaque prouveur	: s un mot du code de poids μ,

Le schéma se compose d'une seule étape, qui se déroule de la façon suivante :

- A choisit un entier quelconque $\ell \in [\ell_{\min}, \ell_{\max}]$,
- A engendre aléatoirement ℓ vecteurs binaires r_i, de longueur n, de poids impairs, à supports disjoints,
- A envoie à B
$$t_i = H{r_i}^t \quad \text{et} \quad w_i = \omega(r_i),$$
- B choisit un vecteur binaire e de longueur ℓ, de poids impair w, tel que $\frac{\ell}{3} \leqslant w \leqslant \frac{2\ell}{3}$, et le transmet à A,
- A choisit une permutation π de l'ensemble $\{1,\dots,\ell\}$ et l'envoie à B,
- A et B calculent $t = e\pi$,
- B envoie un élément $b \in \{t, \bar{t}\}$, où $\bar{t}$ est le complément à 1 de t,
- A calcule $r = \sum_{i=1}^{\ell} b_i r_i$, et transmet à B le vecteur $y = r + s$,
- B vérifie que :
  - Première condition sur les poids : $\omega(y)$ est pair et distinct de $\sum_{i=1}^{\ell} b_i w_i$ (impair),
  - Condition sur le syndrome : $Hy^t = \sum_{i=1}^{\ell} b_i t_i$,
  - Deuxième condition sur les poids :
$$\sum_{i=1}^{\ell} b_i w_i - \mu \leqslant \omega(y) \leqslant \sum_{i=1}^{\ell} b_i w_i + \mu.$$

Les caractéristiques du schéma sont les suivantes :

- ROM : $k(n-k)+n$ bits,
- Complexité des calculs : $k\ell(2(n-k)+1)+n(\ell+1)$ opérations binaires,
- Débit de transaction : $\ell(k+\log_2(n)+1)+\ell_\pi+n$ bits,

Les paramètres minimaux suggérés par l'auteur sont

$$n = 2000, k = 1000, \ell_{\min} = 100, \ell_{\max} = 200, \ell_\pi = 120, \mu \simeq 75.$$

7.2. Le schéma de M. Girault

Si l'auteur reconnaît lui-même que son schéma ne peut être utilisé dans un cadre pratique, son but était de définir un schéma où
- la probabilité de succès de fraude soit bornée par $(\frac{1}{2})^r$,
- aucune fonction de hachage ne soit nécessaire.

Quantités publiques : $H(k,n)$ une matrice binaire quelconque de rang k,

Quantités secrètes de chaque prouveur : s un vecteur binaire de longueur n,

Quantités publiques de chaque prouveur : $i = Hs^t$ et $p = \omega(s)$.

Le schéma se compose de r étapes, chacune d'entre elles se déroulant de la façon suivante :

- A engendre aléatoirement S une matrice binaire de rang k et σ une permutation de l'ensemble $\{1,\dots,n\}$, et envoie

$$M = SH\sigma \text{ et } T = Si.$$

- B choisit un élément quelconque b de $\{0,1\}$,
- Si b vaut 0,
  - A dévoile S et σ,
  - B vérifie que $SH\sigma = M$ et $Si = T$,
- Si b vaut 1,
  - A dévoile $s\sigma$,
  - B vérifie que $M(s\sigma)^t = T$ et que $\omega(s\sigma) = p$,

Les caractéristiques du schéma sont les suivantes :

- ROM : $k(n-k)+n$ bits
- Complexité des calculs : $rk(2k-1)(n-k+1)$ opérations binaires,
- Débit de transaction : $r(k(n+1)+\frac{1}{2}(n+k^2+\ell_\sigma))$ bits,

Les paramètres minimaux suggérés par l'auteur sont

$$n = 512, k = 256, p = 56, r = 20, \ell_\sigma = 120.$$

7.3. Le schéma de J. Stern

Ce schéma est le plus performant des schémas SD.

Quantités publiques : $H(k, n)$ une matrice binaire quelconque de rang k, une fonction de hachage $< . >$,

Quantités secrètes de chaque prouveur : s un vecteur binaire de longueur n,

Quantités publiques de chaque prouveur : $i = Hs^t$ et $p = \omega(s)$.

Le schéma se compose de r étapes, chacune d'entre elles se déroulant de la façon suivante :

- A choisit aléatoirement un vecteur y de longueur n et une permutation σ de $\{1, \ldots, n\}$, et envoie

$$c_1 =< \sigma, Hy^t >, c_2 =< y\sigma >, c_3 =< (y + s)\sigma >$$

- B envoie un élément quelconque, b, de $\{0, 1, 2\}$,
- Si b vaut 0,
  - A dévoile $y + s$ et σ,
  - B vérifie la validité de c_1 et c_3,
- Si b vaut 1,
  - A dévoile y et σ,
  - B vérifie la validité de c_1 et c_2,
- Si b vaut 2,
  - A dévoile $y\sigma$ et $s\sigma$,
  - B vérifie que $\omega(e\sigma) = p$, et contrôle la validité de c_2 et c_3.

Les caractéristiques du schéma sont les suivantes :

- ROM : $k(n - k) + n$ bits
- Complexité des calculs : $r(k(2(n - k) + 1) + n + \frac{n}{3})$ opérations binaires,
- Débit de transaction : $r(3h + \frac{2}{3}(2n + \ell_\sigma))$ bits,

Les paramètres minimaux suggérés par l'auteur sont

$$n = 512, k = 256, p = 56, r = 35, \ell_\sigma = 120, h = 128.$$

7.4. Comparaisons générales

La comparaison de tous les schémas SD a été faite en prenant en compte les paramètres minimaux garantissant la sécurité des différents schémas. Pour chaque paramètre étudié, la meilleure performance obtenue apparaît en gras dans le tableau ci-dessous.

	Harari	Girault	Stern	G-SD	G-SD opt.	MR G-SD	CR G-SD	T-SD
Etapes	**1**	20	35	35	35	35	35	35
Rom	1002000	66048	66048	66816	48184	**1275**	2415	2550
Calculs	$2^{28.2}$	$2^{29.3}$	$2^{22.13}$	$2^{22.5}$	2^{22}	$2^{20.9}$	$\mathbf{2^{19.9}}$	2^{21}
Débit	153900	3288240	40133	34160	**30986**	34090	75740	129290
$C_{<>}$	/	/	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$	$2^{20.4}$

Tab. 13. Performances générales

Comparativement au schéma de J. Stern, qui était, pour l'instant, le plus performant des schémas SD, il ressort de cette étude, qu'à paramètres égaux et sous la contrainte $n = 2k$, le schéma G-SD améliore le débit de transaction, pour une complexité de calcul pratiquement semblable et un seuil de sécurité identique.

Le schéma G-SD opt. est à ce jour le schéma SD, le plus performant, dont la sécurité repose sur une instance quelconque du problème SD. Par rapport au schéma de J. Stern, nous obtenons

- une diminution de 27% de la taille de la ROM,
- une complexité de calcul équivalente,
- une réduction de 23% du débit de transaction.

Pour terminer, voici une liste concernant les avantages et les inconvénients des différents schémas SD :

- S. Harari
  - Avantages : Le schéma se déroule en une seule étape et n'utilise aucune fonction de hachage,
  - Inconvénients : Chaque prouveur est caractérisé par une matrice H de taille $(1000, 2000)$. Ceci suppose que le vérifieur possède une capacité de stockage importante (environ 250 Ko, pour chaque prouveur). Les paramètres sont trop élevés pour envisager une implémentation sur carte à puces. Le schéma n'est plus fiable (cf.chapitre suivant).

- M. Girault
  - Avantages : La probabilité de succès d'un fraudeur est réduit à 2^{-r} (au lieu de $(\frac{2}{3})^r$ pour les schémas proposés et celui de J. Stern). Aucune fonction de hachage n'est nécessaire,
  - Inconvénients : La complexité des calculs et surtout le débit de transaction sont très élevés, pour un seuil de sécurité équivalent à celui des autres schémas.
- J. Stern
  - Avantages : Le débit de transaction et la complexité des calculs sont suffisamment faibles pour envisager une implémentation sur carte à puces,
  - Inconvénients : La quantité de ROM nécessaire au prouveur est importante. Le schéma utilise une fonction de hachage.
- G-SD, G-SD opt.
  - Avantages : Parmi les schémas utilisant une matrice publique quelconque, c'est celui qui obtient les meilleures performances pour les paramètres étudiés,
  - Inconvénients : La taille de la matrice est encore assez importante. Le schéma utilise une fonction de hachage.
- MR G-SD
  - Avantages : Ce schéma possède le meilleur résultat en ce qui concerne la taille de la ROM. De plus sa complexité de calcul est pratiquement la plus faible. Globalement, il obtient les meilleures performances en ce qui concerne les critères étudiés,
  - Inconvénients : La matrice utilisée n'est plus quelconque, ce qui ouvre la voie à de nouvelles attaques. Le schéma utilise une fonction de hachage.
- CR G-SD
  - Avantages : La matrice utilisée possède beaucoup moins de structures que celle du schéma MR G-SD. Il possède, de plus, la plus petite complexité de calculs,
  - Inconvénients : Le vérifieur doit posséder une importante capacité de stockage. Le débit de transaction est élevé. Le schéma utilise une fonction de hachage.
- T-SD
  - Avantages : Pour des résultats semblables à ceux de CR G-SD, en ce qui concerne la taille de la ROM et la complexité de calcul, il n'est pas nécessaire que le vérifieur possède une grande capacité de stockage,

- Inconvénients : La matrice utilisée n'est pas quelconque. Elle possède moins de structures que la matrice du schéma MR G-SD (car une quantité aléatoire intervient) mais la classe, à laquelle elle appartient, est beaucoup moins riche que celle du schéma CR G-SD. Le schéma utilise une fonction de hachage. Son débit de transaction est élevé.

§8. Un autre point de vue

Dans un cadre pratique, la carte à puces n'est pas le seul support dédié à l'implémentation d'un schéma d'identification. Un autre exemple d'application est l'identification par le serveur d'une entreprise d'une personne désirant se connecter, via un modem connecté à un ordinateur portable, au réseau interne de l'entreprise.

Ainsi, selon le cadre d'utilisation, certains paramètres seront plus ou moins prioritaires. Si le processus de déroule entre deux machines puissantes, alors la complexité des calculs, et la ROM, ne sont plus des contraintes importantes. De même, si les deux machines sont reliés physiquement par un câble, alors le débit de transaction peut être considéré comme secondaire. Par contre, si la liaison s'effectue par lignes téléphoniques, alors ce paramètre est essentiel (coût de la communication).

Pour terminer, nous comparons donc tous ces schémas d'identification en négligeant un des paramètres. Pour cela, une note de 10 a été attribué à tous les paramètres du schéma de J. Stern. A partir de quoi, une note pour chacun des paramètres des autres schémas a été calculée. Les résultats sont donnés sous forme de graphiques :

• ROM négligée

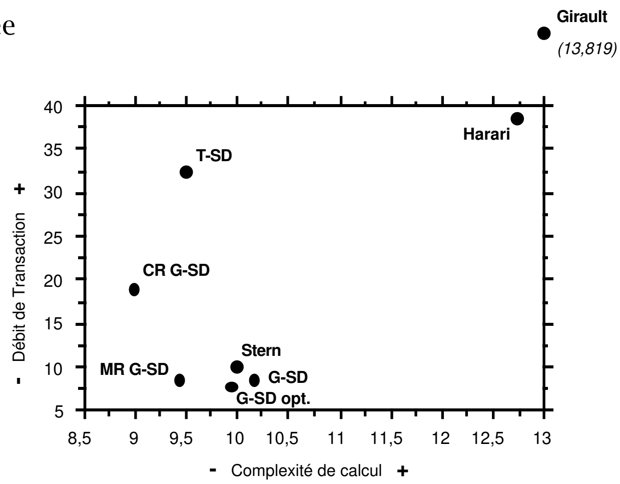

Fig. 14. Complexité de Calcul et Débit de transaction

• Complexité de calcul négligée

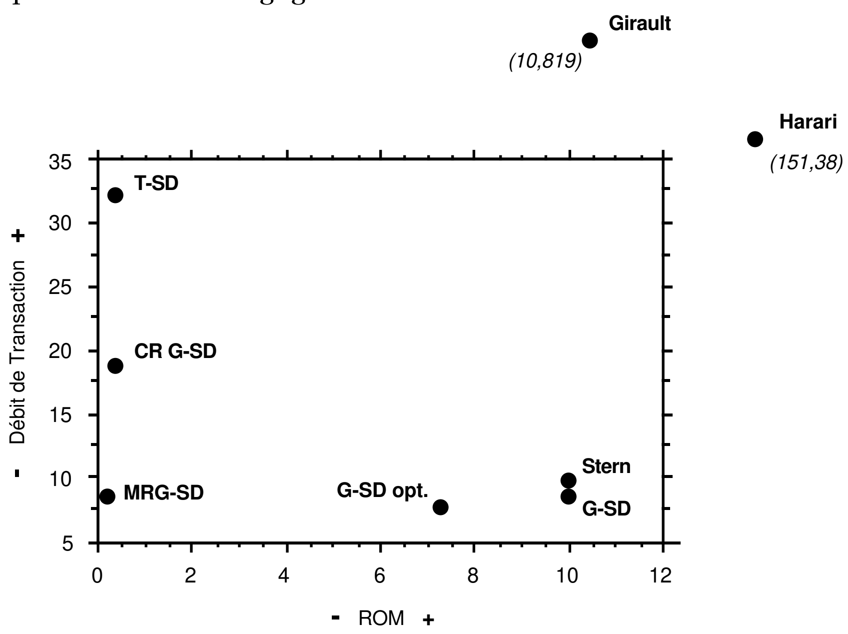

Fig. 15. ROM et Débit de transaction

• Débit de transaction négligée

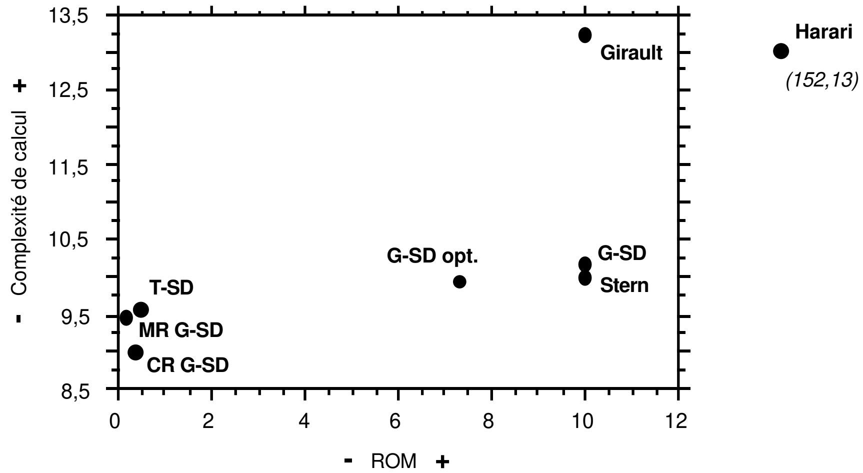

Fig. 16. ROM et Complexité de Calcul

Remarque : Il apparaît clairement, dans ce dernier graphique, que le but que nous nous étions fixé en ce qui concerne la mise au point d'alternatives au schéma G-SD, afin de réduire la taille de la ROM et la complexité de calcul, est atteint.

§9. Le Schéma de Fiat-Shamir

Cet algorithme a été proposé en 1986 par A. Fiat et A. Shamir. Sa sécurité dépend de la difficulté de calculer des racines carrées modulo un nombre composite, n, lorsque les facteurs de ce premier sont inconnus. Dans la suite, I représentera une chaîne binaire contenant des informations concernant l'identité du prouveur.

Quantités publiques	: $n = pq$ et k deux entiers, une fonction pseudo-aléatoire f à valeur dans $]0, n]$, une fonction de hachage $< . >$.
Quantités publiques de chaque prouveur	: $v_j = f(I, j), j = 1, \ldots, k$ et I,
Quantités secrètes de chaque prouveur	: $s_j, j = 1, \ldots, k$ vérifiants $s_j^2 = v_j^{-1} \pmod n$,

Le schéma se compose de t étapes, chacune d'entre elles se déroulant de la façon suivante :

- A choisit un élément quelconque $r \in \{0, \ldots, n\}$ et calcule $x = r^2 \pmod n$. A envoie à B $< f(x) >$,
- B envoie un élément quelconque $b = (b_1, \ldots, b_k) \in \{0, 1\}^k$,
- A calcule $y = r \prod_{b_j=1} s_j \pmod n$ et le transmet à B,
- B vérifie que $< f(y^2 \prod_{b_j=1} v_j \pmod n) > = < f(x) >$.

Les performances du schéma sont les suivantes (en posant $m = \lceil \log_2 n \rceil$) :

- ROM	: km bits
- Complexité des calculs	: $t\frac{k+2}{2}$ multiplications modulaires sur des entiers de m bits,
- Débit de transaction	: $t(h + k + m)$ bits,

Les paramètres minimaux suggérés par l'auteur sont

$$m = 512, k = 5, t = 4, h = 128.$$

Remarque : Dans ce schéma, la fonction de hachage utilisée consiste simplement à renvoyer les 128 premiers bits de la quantité passée en argument.

Le problème de la multiplication modulaire a été largement étudié en cryptographie ([BGV], [Mon], [ShVu], [Wal]) car l'efficacité de nombreux systèmes dépend de la rapidité d'exécution de cette opération.

On peut estimer le coût d'une multiplication entre deux entiers de m bits, modulo un nombre de m bits, à $3(m^2 + m)$ opérations binaires. Ainsi, le nombre d'opérations binaires, pour un processus d'identification est

$$3t(\frac{k+2}{2})(m^2 + m).$$

Le schéma G-SD opt. est le schéma le plus performant, dont la sécurité est prouvée. Si d'autres types de cryptanalyse ne sont pas mises à jour, le schéma MR G-SD, réalise les meilleures performances pour tous les paramètres étudiées. Nous les avons donc comparés au schéma de Fiat-Shamir. Voici les résultats.

	Fiat-Shamir	G-SD opt	MR G-SD
Etapes	4	35	35
ROM	2560	48184	1275
Complexité de calcul	$2^{23.4}$	2^{22}	$2^{20.9}$
Débit de transaction	2580	30986	34090

Tab. 17. Comparaison avec le schéma de Fiat-Shamir

Le schéma de Fiat-Shamir reste très performant en ce qui concerne le débit de transaction et la taille de la ROM. Nous tenons à souligner que bien qu'il possède une complexité de calcul proche de celle du schéma MR G-SD, l'implémentation pratique de ce dernier est beaucoup plus simple. En effet, dans le protocole MR G-SD, le prouveur effectue au plus trois additions binaires (xor) sur des mots de longueur 510 et trois multiplications dans le corps $\mathbb{F}_{2^{255}}$. La multiplication s'effectue à l'aide de la méthode de Berlekamp et peut être implémentée à l'aide d'un simple registre à décalage de longueur 255, dont un certain nombre de cellules sont connectées par des portes xor (80 pour les paramètres suggérés).

L'implémentation d'une multiplication modulaire nécessite un circuit plus compliqué. Ce dernier doit comporter un multiplicateur qui se compose généralement de m^2 portes and et $m^2 - m$ additionneurs [CLR]. De plus, il faut implémenter l'algorithme ci-dessous réalisant la réduction, d'un nombre x de

ℓ bits, modulo un nombre n de m bits ($\ell > m$) [Knu, 5, Algorithm 4.3.1 A]. Ce dernier nécessite l'utilisation d'un additionneur.

entrée : - $x = \sum_{i=0}^{\ell-1} x_i 2^i$ un entier de ℓ bits,
- $n = \sum_{i=0}^{m-1} n_i 2^i$ un entier de m bits

début
  Si $x > 2^{\ell-m}n$ **alors** $x \leftarrow x - 2^{\ell-m}n$;
  Pour i variant de $\ell - 1$ **à** m **faire**
    début
      Si $x_i = 1$ **alors** $q \leftarrow 1$
      sinon $q \leftarrow x_{i-1}$;
      Tant que $q(2 + n_{m-2}) > 4x_i + 2x_{i-1} + x_{i-2}$ **faire** $q \leftarrow q - 1$;
      $x \leftarrow x - 2^{i-m}qn$;
      Si $x < 0$ **alors** $x \leftarrow x + 2^{i-m}n$;
    fin
fin.

sortie : - $x \pmod n$

Fig. 18. Réduction modulaire

§10. Les autres alternatives

Depuis 1989, d'autres schémas d'identification, dépendant d'un problème $\mathcal{NP}$-complet, ont été élaborés. Nous ne les décrirons pas, mais nous donnerons le problème de décision associé à chacun d'entre eux, ainsi que les paramètres minimaux suggérés par les auteurs. Nous avons évalué les performances de ces différents schémas. A la différence des schémas SD, les opérations s'effectuent sur des octets et non plus sur des bits. Nous avons cependant calculé leur complexité en termes d'opérations binaires, ceci afin de pouvoir comparer leurs performances avec celles des schémas proposés.

Le premier schéma a été proposé à Crypto'89 par A. Shamir [Sha] et dépend du problème des "noyaux permutés" :

Nom : PKP (Permuted Kernel Problem)
Entrée : A une matrice $m \times n$ sur $\mathbb{F}_p$,
V un vecteur de longueur n sur $\mathbb{F}_p$.
Question : Existe-t-il π une permutation de $\{1, \ldots, n\}$ tel que $V\pi \in \ker(A)$?

Les paramètres suggérés par A. Shamir sont

$$n = 32, k = 16, p = 251.$$

Cependant une cryptanalyse, due à J. Patarin et P. Chauvaud [PaCh], montrent que les paramètres à utiliser sont

$$n = 64, k = 32, p = 251.$$

Le deuxième schéma a été proposé à Crypto'94 par J. Stern [Ste4].

Nom : CLE (Constrained Linear Equation)
Entrée : Un entier q, un système S de r équations linéaires à k inconnues, à coefficients dans $\mathbb{Z}_q$, $X \subset \mathbb{Z}_q$.
Question : Existe-t-il une solution de S, dont les k éléments appartiennent à X ?

Les paramètres suggérés par l'auteur sont

$$q = 257, k = 40, r = 20, \mathrm{card}(X) = 16.$$

Le dernier schéma a été proposé à Eurocode'94 par D. Pointcheval [Poi].

Nom : PPP (Permuted Perceptron Problem)
Entrée : A une matrice $m \times n$ constituée uniquement de $+1$ et de -1, un multi-ensemble S, de m éléments de $\mathbb{F}_p$.
Question : Existe-t-il un vecteur Y de longueur n, composé uniquement de $+1$ et de -1, tel que $\{(AY)_i, i = 1, \ldots m\} = S \pmod p$?

Les paramètres suggérés sont

$$n = 117, m = 101, p = 251.$$

Ces trois schémas utilisent une fonction de hachage et le principe des mises en gage. Ce sont des systèmes de preuve interactive à divulgation nulle. Seule leur version à trois passes (i.e, à chaque étape les échanges entre Alice et Bob sont au nombre de trois) a été étudiée, excepté pour le schéma PKP qui ne propose qu'une version à 5 passes. Voici leurs performances :

	PKP	CLE	PPP
Etapes	20	35	48
ROM	17016 bits	3520 bits	11934 bits
Complexité de calcul	$2^{22.3}$	$2^{21.8}$	$2^{23.5}$
Débit de transaction	17920 bits	29400 bits	88284 bits

Fig. 19. Les autres alternatives

Remarque : Pour PPP, la valeur donnée pour le débit de transaction est différente de celle annoncée par l'auteur. En fait, ce dernier utilise un générateur aléatoire pour toutes les quantités transmises, et n'envoie donc qu'un germe de 64 bits. Dans les autres schémas, et dans ceux que nous avons proposés, cette technique n'est utilisée que lors de l'envoi de la permutation. C'est pourquoi, afin de pouvoir effectuer une comparaison, nous donnons ici, le débit de transaction réel de PPP.