Chapitre V

Cryptanalyse du schéma de S. Harari

"Ceux qui se vantent de lire les lettres chiffrées sont de plus grands charlatans que ceux qui se vanteraient d'entendre une langue qu'ils n'ont point apprise."

Voltaire

Ce chapitre présente trois cryptanalyses efficaces du schéma d'identification de S. Harari. L'une d'entre elles utilise les algorithmes présentés au chapitre ii.

Le schéma d'identification de S. Harari a été le premier protocole dont la sécurité dépendait de la difficulté de résolution du problème SD. Trois attaques possibles sont proposées. Les deux premières montrent, comment un fraudeur peut, dans 50% des cas, se faire passer pour Alice auprès de Bob. De nouveaux paramètres sont proposés afin de rétablir la sécurité du schéma. Cette modification augmente, d'un facteur 20, le débit de transaction et la complexité de calcul du schéma initial. Il était alors intéressant d'étudier s'il était possible d'utiliser une matrice de taille équivalente à celle des autres schémas SD, afin de diminuer la valeur de ces deux quantités.

La troisième attaque montre que le schéma n'est plus fiable si on réduit la taille de la matrice publique. Pour les valeurs initiales, suggérées par l'auteur, cette attaque permet, dans certains cas, à un vérifieur malhonnête d'obtenir, lors des transactions avec le prouveur, le secret de ce dernier. De plus, il apparaît clairement qu'une diminution de la taille de la matrice utilisée, augmente l'efficacité de cette attaque.

§1. Généralités

L'efficacité des deux premières attaques provient du fait que les quantités r_i engendrés par le prouveur ne sont jamais dévoilées. Ainsi un fraudeur peut tricher sur les quantités w_i et t_i qu'il envoie à un vérifieur honnête.

Les attaques sont efficaces lorsque le vecteur question b choisi par B correspond au vecteur $e\pi$, où π est une permutation de l'ensemble $\{1, \ldots, \ell\}$, choisie par $\tilde{A}$. Le vecteur e est de longueur ℓ, et vérifie

$$\frac{\ell}{3} \leqslant \omega(e) \leqslant \frac{2\ell}{3}.$$

Pour simplifier la description des algorithmes qui vont suivre, il sera supposé que la permutation π, choisie par $\tilde{A}$, transforme le mot e de poids $\frac{\ell}{3} + j$ $(0 \leqslant j \leqslant \frac{\ell}{3})$, en un mot e', de longueur ℓ, tel que

$$e' = (\underbrace{111\ldots11}_{\frac{\ell}{3}+j}0\ldots0).$$

Cette hypothèse n'est faite que pour clarifier les résultats.

Avec cette convention, un éventuel fraudeur sait qu'une question de B, ne fera intervenir que les quantités $w_1, \ldots, w_{\frac{2\ell}{3}}$ et $t_1, \ldots, t_{\frac{2\ell}{3}}$.

Nous rappelons que le schéma de S. Harari dépend essentiellement des trois paramètres suivants :

$$n = 2000, k = 1000, \ell \in [100, 200].$$

Le secret d'un prouveur honnête consiste en un mot s, de poids impair μ, appartenant à un code linéaire C, dont une matrice de contrôle est publique.

Pour plus de simplicité, il sera supposé que ℓ est un multiple de 3. Dans le cas général, il faudra remplacer, dans ce qui suit, $\frac{\ell}{3}$ par $\lceil \frac{\ell}{3} \rceil$.

§2. Première cryptanalyse

Soient $w_1, \ldots, w_\ell$ et $t_1, \ldots t_\ell$ les quantités dévoilées par un fraudeur $\tilde{A}$. Soit e un vecteur question posé par B, de poids $\frac{\ell}{3} + j$, et soit $e' = e\pi$ le vecteur défini dans le paragraphe précédent, où π est choisi par $\tilde{A}$. Alors, le vecteur réponse y envoyé par $\tilde{A}$ est valide si

. $\omega(y) \neq \sum_{i=1}^{\frac{\ell}{3}+j} w_i$, (Première condition sur les poids)

. $H(y) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i$, (Condition sur les syndromes)

. $\omega(y) - \mu \leqslant \sum_{i=1}^{\frac{\ell}{3}+j} w_i \leqslant \omega(y) + \mu$, (Deuxième condition sur les poids).

Nous nous proposons de montrer comment construire :

- une suite c_j, de mots du code C, tel que $\omega(c_j) = \sum_{i=1}^{\frac{\ell}{3}+j} w_i$,

- une suite de vecteurs r_j de poids μ, ou $\mu - 1$, tel que $H(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i$ et $\omega(c_j + r_j)$ soit pair.

Pour une telle construction alors le vecteur $y = c_j + r_j$, satisfait, pour la question e', les trois conditions du schéma.

2.1. Construction des c_j et de $w_1, \dots, w_\ell$

• $\tilde{A}$ choisit $\frac{\ell}{3} + 1$ mots du code C, $c_0, \dots, c_{\frac{\ell}{3}}$, de poids distincts compris entre 900 et 1100. Notons $p_j = \omega(c_j)$, on peut toujours supposer, à une renumérotation près que

$$p_0 < p_1 < \cdots < p_{\frac{\ell}{3}}.$$

Les c_j sont choisis tel que :
- $p_0 < 1000$,
- $p_{\frac{\ell}{3}} > 1000$,
- Si $\frac{\ell}{3}$ est pair, alors
  . p_j est pair si j est pair, impair sinon,
- Si $\frac{\ell}{3}$ est impair, alors
  . p_j est impair si j est pair, pair sinon,

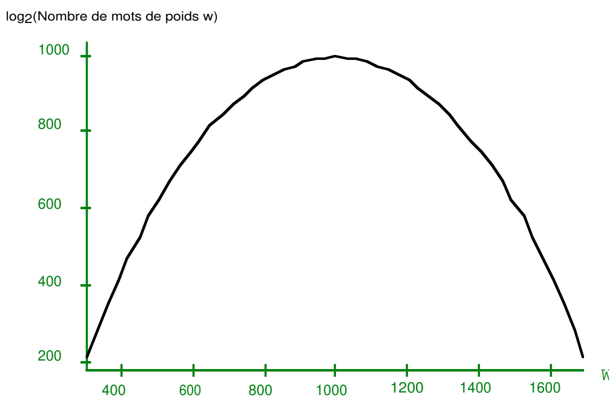

Fig. 1. Distribution des poids d'un code (2000, 1000)

- De telles contraintes sont faciles à satisfaire puisque, si c est un mot du code C, engendré aléatoirement, alors

$$Pr(900 \leqslant \omega(c) \leqslant 1100) \simeq 0,9999933\dots$$

et

$$Pr(900 \leqslant \omega(c) \leqslant 1000) = Pr(1000 \leqslant \omega(c) \leqslant 1100) \simeq 0.5.$$

• $\tilde{A}$ calcule aléatoirement des entiers impairs, $w_1, \dots w_{\frac{\ell}{3}}$ tel que $p_0 = \sum_{i=1}^{\frac{\ell}{3}} w_i$,

• Pour j variant de 1 à $\frac{\ell}{3}$, $\tilde{A}$ calcule un entier impair $w_{\frac{\ell}{3}+j}$ tel que

$$\sum_{i=1}^{\frac{\ell}{3}+j} w_i = p_j.$$

• Puis $\tilde{A}$ choisit aléatoirement $\frac{\ell}{3}$ entiers impairs $w_{\frac{2\ell}{3}+1}, \dots, w_\ell$ et envoie la suite $w_1, \dots, w_\ell$ à B.

2.2. Construction des r_j et de $t_1, \dots, t_\ell$

Posons $\varepsilon_j = \begin{cases} 1 & \text{si } p_j \text{ est pair,} \\ 0 & \text{sinon.} \end{cases}$

• $\tilde{A}$ choisit un mot quelconque, r_0, de poids $\mu - \varepsilon_0$ et calcule $t = Hr_0$.

• Il construit alors aléatoirement $\frac{\ell}{3}$ vecteurs de longueur k, $t_1, \dots t_{\frac{\ell}{3}}$, tel que

$$\sum_{i=1}^{\frac{\ell}{3}} t_i = t.$$

• Pour j variant de 1 à $\frac{\ell}{3}$, $\tilde{A}$ calcule un vecteur r_j de poids $\mu - \varepsilon_j$. Posons

$$t_{\frac{\ell}{3}+j} = H(r_j) + \sum_{i=1}^{\frac{\ell}{3}+j-1} t_i.$$

Alors, on a $H(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i$.

• Les vecteurs $t_{\frac{2\ell}{3}+1}, \dots, t_\ell$ sont choisis aléatoirement. $\tilde{A}$ envoie les éléments $t_1, \dots t_\ell$ à B.

A cette étape du protocole :

• Le vérifieur B choisit un vecteur e, de poids impair compris entre $\frac{\ell}{3}$ et $\frac{2\ell}{3}$ (i.e $\omega(e) = \frac{\ell}{3} + j$) et l'envoie à $\tilde{A}$.

• $\tilde{A}$ choisit une permutation π tel que $e' = e\pi = (\underbrace{111\dots11}_{\frac{\ell}{3}+j}0\dots0)$ et envoie π à B.

• Les deux intervenants calculent $e' = e\pi$.

- B pose alors la question $b = e'$ ou $b = \overline{e'}$.
- Si la question posée est e', $\tilde{A}$ renvoie $y = c_j + r_j$. Alors,
  - $\omega(y) \neq \sum_{i=1}^{\ell} b_i w_i$, puisque cette somme est un entier impair et, par construction, le poids de $c_j + r_j$ est pair,
  - $H(y) = H(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i = \sum_{i=1}^{\ell} b_i t_i$,
  - $\omega(c_j) - \mu \leqslant \omega(y) \leqslant \omega(c_j) + \mu$,

  Or, par définition de c_j,

$$\omega(c_j) = \sum_{i=1}^{\frac{\ell}{3}+j} w_i = \sum_{i=1}^{\ell} b_i w_i.$$

Le vecteur y satisfait donc les trois conditions du schéma. Ainsi, un fraudeur a, en moyenne, une chance sur 2 de satisfaire le protocole d'identification.

Remarquons que la permutation π peut être différente à chaque étape. En effet, si l'attaque décrite n'était valide que pour la permutation qui consiste à ramener tout le support de e dans les $\frac{\ell}{3} + j$ premières composantes de e', alors un vérifieur B peut stopper le protocole, en refusant d'utiliser cette permutation.

Pour utiliser une permutation π quelconque, il suffit que le fraudeur choisisse, en premier lieu, une permutation quelconque, σ, de l'ensemble $\{1, \ldots, \ell\}$. Au lieu d'envoyer les séquences $w_1, \ldots, w_\ell$ et $t_1, \ldots, t_\ell$, le fraudeur envoie $w'_1, \ldots, w'_\ell$ et $t'_1, \ldots, t'_\ell$, où $w'_i = w_{\sigma(i)}$ et $t'_i = t_{\sigma(i)}$. Lorsque B pose une question e de poids $\frac{\ell}{3} + j$, $\tilde{A}$ choisit la permutation π tel que le $i^{\text{ième}}$ bit du support de e se retrouve en position $\sigma(i)$ ($i = 1, \ldots, \frac{\ell}{3} + j$).

§3. Deuxième cryptanalyse

Dans l'attaque précédente, la construction des éléments c_j dépend du choix de certains mots du code de poids compris entre 900 et 1100. Bien qu'un mot du code, engendré aléatoirement, vérifiera presque toujours ces contraintes, ce choix reste probabiliste. Nous proposons, dans ce paragraphe, une méthode déterministe permettant de construire une suite d'éléments r_j vérifiant

- $\omega(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} w_i$,
- $H(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i + x$,

où $x = Hz$, z étant un vecteur quelconque de poids μ. Dans ce cas, le vecteur $y = r_j + z$ répond à la question e de poids $\frac{\ell}{3} + j$.

• $\tilde{A}$ choisit aléatoirement un mot z de poids μ. Posons $x = Hz$,

• $\tilde{A}$ choisit aléatoirement ℓ entiers impairs $w_1, \dots, w_\ell$, et construit un vecteur r_0 de poids $\sum_{i=1}^{\frac{\ell}{3}} w_i$. Posons $t = H(r_0) + x$.

• $\tilde{A}$ calcule $\frac{\ell}{3}$ vecteurs de longueur k, $t_1, \dots, t_{\frac{\ell}{3}}$, tel que $t = \sum_{i=1}^{\frac{\ell}{3}} t_i$,

• Pour j variant de 1 à $\frac{\ell}{3}$, $\tilde{A}$ construit :
  - un vecteur r_j de poids $\sum_{i=1}^{\frac{\ell}{3}+j} w_i$,
  - le vecteur $t_{\frac{\ell}{3}+j} = H(r_j) + \sum_{i=1}^{\frac{\ell}{3}+j-1} t_i + x$ (d'où $H(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i + x$),

• $\tilde{A}$ engendre aléatoirement $t_{\frac{2\ell}{3}+1}, \dots, t_\ell$, et envoie à B, les éléments $w_1, \dots w_\ell$ et $t_1, \dots t_\ell$.

A cette étape du protocole :

• Le vérifieur B choisit un vecteur e, de poids impair compris entre $\frac{\ell}{3}$ et $\frac{2\ell}{3}$ (i.e $\omega(e) = \frac{\ell}{3} + j$) et l'envoie à $\tilde{A}$.

• $\tilde{A}$ choisit une permutation π tel que $e' = e\pi = (\underbrace{111\dots11}_{\frac{\ell}{3}+j}0\dots0)$ et envoie π à B.

• Les deux intervenants calculent $e' = e\pi$.

• B pose alors la question $b = e'$ ou $b = \overline{e'}$.

• Si la question posée est e', $\tilde{A}$ renvoie $y = r_j + z$. Alors,
  - $\omega(y) \neq \sum_{i=1}^{\ell} b_i w_i$. En effet cette somme est un nombre impair, or μ et $\omega(r_j)$ sont impairs, donc $\omega(y)$ est pair,
  - $H(y) = H(r_j) + x = \sum_{i=1}^{\frac{\ell}{3}+j} t_i = \sum_{i=1}^{\ell} b_i t_i$,
  - $\omega(r_j) - \mu \leqslant \omega(y) \leqslant \omega(r_j) + \mu$,

  Or, par définition de r_j,

$$\omega(r_j) = \sum_{i=1}^{\frac{\ell}{3}+j} w_i = \sum_{i=1}^{\ell} b_i w_i.$$

Le vecteur y satisfait donc les trois conditions du schéma. Ainsi, un fraudeur a, en moyenne, une chance sur 2 de satisfaire le protocole d'identification.

Remarques :

- Pour des raisons identiques à celle du paragraphe précédent, la permutation π peut être quelconque,

- L'attaque se généralise, de façon évidente, pour toute question e de poids compris entre deux quantités quelconques ℓ_1 et ℓ_2. De plus, elle est totalement indépendante du poids du vecteur secret.

§4. Nouveaux Paramètres

Afin de renforcer la sécurité du schéma, il suffit d'exécuter ce dernier plusieurs fois de suite. Un processus d'identification correspondra à r exécutions du protocole initial proposé par S. Harari. La probabilité de succès des attaques décrits précédemment sera alors bornée par 2^{-r}. Pour $r = 20$, un fraudeur n'aura qu'une chance sur un million de satisfaire le protocole.

Avec cette modification, les performances du schéma sont les suivantes :

Etapes	20
ROM	1002000
Calculs	$2^{32.5}$
Débit	3078000

Tab. 2. Schéma Modifié

Les nouveaux paramètres obtenus sont trop élevés pour envisager une implémentation pratique du schéma, à moins de diminuer la taille de la matrice utilisée. Le prochain paragraphe montre que cette optimisation n'est pas possible.

Les deux attaques, décrites précédemment, permettent à un fraudeur, d'usurper l'identité d'un prouveur honnête. La dernière attaque, que nous proposons, montre, comment un vérifieur malhonnête, peut extraire des informations, sur le secret d'un prouveur, à partir des quantités échangées avec ce dernier.

§5. Troisième cryptanalyse

Lors de l'élaboration du schéma de S. Harari, aucun travail de synthèse n'avait été encore réalisé en ce qui concerne les réelles performances des algorithmes décrits au chapitre ii.

De nos jours, grâce aux travaux de A. Canteaut et F. Chabaud, leur efficacité pratique a été prouvée. Au vu des résultats obtenus dans [CaCh2], nous nous proposons, dans ce paragraphe, de montrer :

- qu'il est possible de retrouver, dans certains cas, le secret s, d'un prouveur honnête
- qu'il est impossible de diminuer les paramètres du système de S. Harari.

L'attaque est basée sur le fait qu'un prouveur honnête envoie ℓ quantités, $w_1, \ldots, w_\ell$, qui sont les poids de vecteurs secrets $r_1, \ldots r_\ell$, à support disjoints. Ces valeurs satisfont donc

$$\sum_{i=1}^{\ell} w_i \leqslant 2000 \quad \text{(longueur du code).}$$

Remarque : Dans les deux attaques précédentes, on peut choisir les w_i de façon à ce qu'ils satisfassent cette contrainte.

Pour $\ell = 100$, un prouveur honnête peut par exemple choisir 100 vecteurs de poids au plus 20, et la contrainte sera satisfaite. Cependant si,

$$\forall i, \omega(r_i) \leqslant 20,$$

alors à l'aide des algorithmes décrits au chapitre ii, étant donné que $t_i = H(r_i)$ est connu, il est possible à un vérifieur malhonnête de retrouver tous les vecteurs r_i. En effet, ceci revient à chercher un mot de poids 21, dans un code $(2001, 1001)$. La complexité de cette recherche est de l'ordre de 2^{40}.

L'efficacité de cette attaque croit en fonction de ℓ, puisque plus ce paramètre augmente et plus le poids des r_i est contraint.

Afin d'éviter cette attaque, il faut trouver une méthode permettant de construire le plus de vecteurs r_i possibles, tel que la seule connaissance de w_i et de t_i ne permette pas de retrouver r_i. La recherche d'un mot de poids 36 dans un code $(2001, 1001)$ nécessite environ 2^{52} opérations.

Ainsi, la configuration optimale pour le choix des r_i, consiste à construire le plus grand nombre de mots de poids 35, tel que $\sum_{i=1}^{\ell} w_i \leqslant 2000$. Il est facile de montrer que ce nombre est donné par

$$\mathcal{N} = \lfloor \frac{2000 - \ell}{34} \rfloor.$$

Pour $\ell = 100$, on peut construire 55 vecteurs r_i de poids 35. Les 45 vecteurs restants seront de poids 1.

Pour $\ell = 200$, on peut construire 52 vecteurs r_i de poids 35. Les 148 vecteurs restants seront de poids 1.

Pour ces deux cas, la recherche de tous les r_i, de poids 35, aboutit à une complexité de l'ordre de 2^{58}.

Cependant, à ℓ fixé, il est clair qu'un vérifieur peut retrouver $\ell - \mathcal{N}$ des vecteurs r_i. En effet ces derniers étant de poids 1, les t_i associés correspondent à une colonne de la matrice publique H. Plus on s'éloigne de la configuration optimale et plus ce nombre va augmenter.

L'idée est alors la suivante : lorsque B choisit un vecteur e de poids $\frac{\ell}{3} + j$, alors la probabilité pour que le support de la question e corresponde, après l'action de la permutation π, aux $\ell - \mathcal{N}$ éléments connus de B, est

$$P_\ell = \frac{\binom{\ell-\mathcal{N}}{\frac{\ell}{3}+j}}{\binom{\ell}{\frac{\ell}{3}+j}}.$$

Pour ℓ fixé, cette probabilité atteint son minimum pour $j = \frac{\ell}{3}$. Voici donc l'évolution de cette quantité, lorsque ℓ varie dans l'intervalle $[100, 200]$ et $j = \frac{\ell}{3}$.

Fig. 3. Probabilité de succès de l'attaque

Il apparaît donc que lorsque $\ell \geqslant 130$, il suffit de répéter, en moyenne, au plus 2^{45} fois le protocole pour que le support de e corresponde aux $\ell - \mathcal{N}$ éléments connus de B. Ce dernier peut en déduire alors, à partir de

$$y = \sum_{i=1}^{\ell} e_i r_i + s,$$

le secret s.

Dans le protocole, ℓ est un entier quelconque choisi dans l'intervalle $[100, 200]$. Donc

$$Pr(\ell \geqslant 130) = \frac{7}{10}.$$

Ainsi la complexité d'une telle attaque est comprise entre 2^{37} et 2^{46}.

Afin d'éviter cette dernière, il faut donc restreindre ℓ à l'intervalle $[100, 120]$. Ceci conduit à une complexité d'attaque comprise entre 2^{48} et 2^{55}. Cependant, en cryptographie, cette intervalle est considéré comme une zone de sécurité "critique".

D'autre part, la méthode proposée montre qu'il est impossible d'utiliser des paramètres équivalents à ceux des autres schémas SD. En effet, tout d'abord le paramètre ℓ doit toujours être supérieur à 60, afin d'éviter à un prouveur malhonnête de préparer des quantités pouvant répondre à toutes les questions de B. Pour

$$n = 512, k = 256$$

un prouveur honnête devra donc construire, au moins, 60 vecteurs, à supports disjoints, de poids w_i, tel que

$$\sum_{i=1}^{\ell} w_i \leqslant 512.$$

D'après le chapitre ii, la configuration optimale revient à construire, le plus grand nombre de vecteurs de poids 56.

Pour $\ell = 60$, ce nombre vaut 8. Les 52 autres vecteurs restants seront donc de poids 1. Notons C_ℓ le nombre moyen de fois où le protocole doit être répété, afin que le support de e corresponde, après permutation, aux r_i connus du vérifieur. Voici l'évolution de cette quantité lorsque $\ell \geqslant 60$

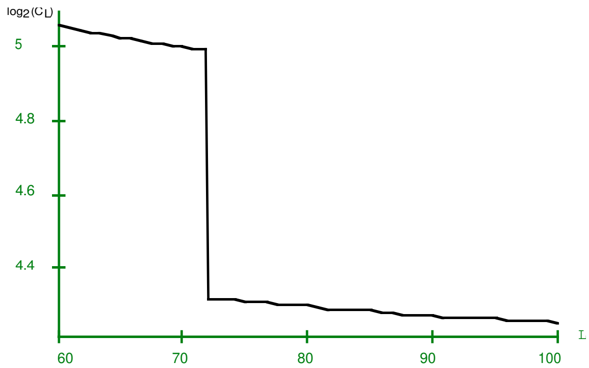

Fig. 4. Complexité de l'attaque pour un code (512, 256)

Ainsi, pour $n = 512, k = 256$, le système n'est pas fiable. Voici l'évolution du paramètre C_ℓ, pour $n = 1024$ et $k = 512$.

Fig. 5. Complexité de l'attaque pour un code (1024, 512)

L'attaque reste donc efficace pour ces paramètres. Ainsi, il est impossible d'utiliser pour ce schéma les paramètres qui sont utilisés dans les autres schémas SD.

Il est possible d'augmenter la complexité de l'attaque en diminuant le poids des $\mathcal{N}$ vecteurs r_i, tout en s'assurant que la complexité de recherche reste supérieure à 2^{60}. Ceci permet alors de choisir certains des $\ell - \mathcal{N}$ autres vecteurs de poids égal à 2, ce qui nécessite une recherche exhaustive pour retrouver leur valeur (combinaison linéaire des colonnes de H prises 2 à 2). On aboutit alors à une quantité C_ℓ bornée par 2^{45}, ce qui ne suffit pas à garantir la sécurité du schéma [VER4].

Tous ces résultats ont été obtenus en supposant que le prouveur utilisait une configuration optimale pour les r_i. Si ces derniers sont choisis tel que les poids w_i vérifient $w_i \leqslant \frac{n}{\ell}$, la figure 6 donne, pour les paramètres initiaux suggérés par S. Harari, le nombre moyen d'opérations nécessaires, C_ℓ, pour retrouver tous les r_i, en utilisant l'un des algorithmes décrits dans le chapitre ii.

Pour conclure, les résultats obtenus montrent que les paramètres, proposés par S. Harari, sont trop faibles. En effet, l'efficacité des deux précédentes attaques est dû au fait que, lors du protocole, le prouveur peut choisir une certaine permutation, ceci afin de s'assurer que le vérifieur ne puisse pas poser des questions de son choix. Il suffit donc de changer cette partie du protocole, pour rétablir la sécurité du schéma. Cependant les résultats obtenus par la dernière attaque, montrent que, pour les dimensions suggérées, le

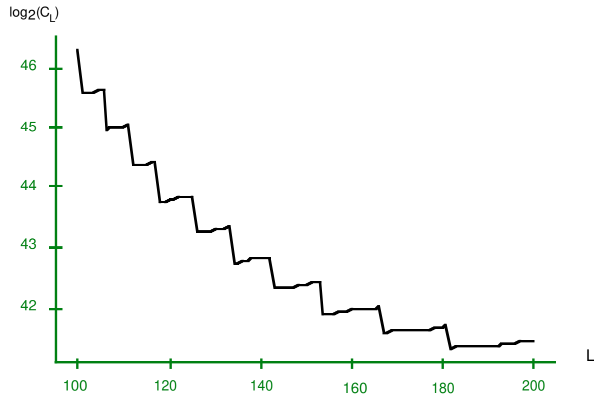

Fig. 6. Complexité de l'attaque pour un code (2000, 1000)

poids des vecteurs r_i est trop contraint, et que la seule connaissance du couple (t_i, w_i) permet de retrouver un certain nombre des vecteurs r_i. Il faut donc augmenter la taille de la matrice, afin de diminuer la contrainte sur les vecteurs r_i. Ceci affectera bien sûr les performances du schéma.