Cryptanalysis of Harari's Identification Scheme

Pascal Véron

Cryptanalysis of Harari's Identification Scheme

Pascal Véron*

G.E.C.T., Université de Toulon et du Var, B.P. 132,
83957 La Garde Cedex, FRANCE

Abstract. In this paper, it is shown that the first identification scheme based on a problem coming from coding theory, proposed in 1988 by S. Harari, is not secure.

1 Introduction

An identification scheme is a cryptographic protocol which enables party A (called the "prover" or Alice) to prove his identity polynomially many times to party B (called the "verifier" or Bob) without enabling B to misrepresent himself as A to someone else.

As it is often the case in cryptography, the first identification schemes were based on hard problems from number theory [5], [8], [12]. In the last few years, some new schemes, whose security depends on an NP-complete problem, have been proposed. The first identification scheme based on an NP-complete problem coming from error correcting codes (SD problem) have been proposed by S. Harari in 1988 [9]. The Syndrome Decoding problem can be stated as follows [1]:

Name : SD
Input : $H(k,n)$ a parity check matrix of a binary $[n,k]$ code, i a syndrome, p an integer.
Question : Is there a vector e of length n such that $He^{\mathrm{t}} = i$ and $\omega(e) \leq p$?

Remark. e^{t} denotes the transpose of the vector e.

The purpose of this paper is to show that Harari's scheme is not secured. First, we give a deterministic method which allows a false prover to pass the protocol every second time. We give then new parameters so as to restore the security of the scheme, and we show that these latter cannot be reduced (so as to improve the practical performances of the scheme). Indeed, if the parameters are shortened, it is possible, for a dishonest verifier, to find prover's secret by using some probabilistic decoding algorithms.

* veron@marie.polytechnique.fr

2 Harari's Identification Scheme

This scheme is interactive. The principle is the following: the prover knows a secret quantity s which satisfies a public predicate. To identify himself, the prover must convince, with overwhelming probability, the verifier that he knows s without revealing any information about it.

Prover's Public Data : $H(1000, 2000)$ a parity check matrix of a binary code whose minimum weight, μ ($\in [50, 100]$), is odd,

Prover's secret : s a codeword of weight μ.

Remark. see [9] for how to construct a $[2000, 1000]$ code, whose minimum weight is, with high probability, μ.

The prover shows that he knows the secret s as follows:

1. A chooses an integer $\ell \in [100, 200]$,
2. A randomly computes ℓ binary vectors r_i, of length n and odd weights, and whose supports are disjoint,
3. A sends to B: $t_i = Hr_i^{\mathrm{t}}$ and $w_i = \omega(r_i)$,
4. B randomly computes a binary vector e of length ℓ, whose weight w is odd and satisfies, $\frac{\ell}{3} \leq w \leq \frac{2\ell}{3}$, and sends it to A,
5. A randomly computes a permutation π of $\{1, \ldots, \ell\}$ and sends it to B,
6. A and B compute $t = e\pi$,
7. B randomly chooses $b \in \{t, \bar{t}\}$, where $\bar{t}$ is the binary complement of t,
8. A computes $r = \sum_{i=1}^{\ell} b_i r_i$, and sends to B the vector $y = r + s$,
9. B checks that:
   - First condition on the weights: $\omega(y) \neq \sum_{i=1}^{\ell} b_i w_i$ ($\omega(y)$ is even, $\sum_{i=1}^{\ell} b_i w_i$ is odd),
   - Condition on the syndrome : $Hy^{\mathrm{t}} = \sum_{i=1}^{\ell} b_i t_i$,
   - Second condition on the weights :

$$\sum_{i=1}^{\ell} b_i w_i - \mu \leq \omega(y) \leq \sum_{i=1}^{\ell} b_i w_i + \mu$$

The practical performances of the scheme are shown in table 1.

Remark. For a smart card application, these three parameters are very important. In this context, the prover is a portable device with few memory and limited computing power. The Rom gives the size of the memory needed on the card. The prover's workfactor is the average number of binary operations computed by the prover during an identification process. The transmission rate is the number of bits exchanged between Alice and Bob.

Rom	1002000 bits
Prover's Workfactor	$2^{28.2}$
Transmission rate	153900 bits

Table 1. Harari's Scheme

3 Cryptanalysis

The attack is efficient if, at step 7, B chooses b equal to $e\pi$. We recall that e is a vector of odd weight $\frac{\ell}{3}+q$, $0 \leq q \leq \frac{\ell}{3}$. So as to simplify the description of this attack, we will assume that ℓ is divisible by 3, otherwise $\frac{\ell}{3}$ must be replaced by $\lceil \frac{\ell}{3} \rceil$. At step 5, the prover chooses a permutation π of $\{1,\ldots,\ell\}$. For more clearness, we will assume that the permutation chosen by the dishonest prover is such that

$$e\pi = (\underbrace{1\cdots1}_{\frac{\ell}{3}+q}0\cdots0)$$

(It is easy to see that the method we are going to describe can be changed so as to work with any permutation π). It is possible, for a dishonest prover, to build three sequences $(w_i)_{1\leq i\leq \ell}, (t_i)_{1\leq i\leq \ell}, (x_j)_{0\leq j\leq \frac{\ell}{3}}$ (w_i are odd integers, t_i are vectors of length k and x_j are vectors of length n) such that

$$\begin{array}{rcl} \omega(x_j) & = & \sum_{i=1}^{\frac{\ell}{3}+j} w_i \\ \omega(x_j) & \equiv & \frac{\ell}{3}+j \bmod 2 \\ Hx_j^{\mathrm{t}} & = & \sum_{i=1}^{\frac{\ell}{3}+j} t_i + x \end{array} \tag{3.1}$$

where $x = Hz^{\mathrm{t}}$, z being a random vector of odd weight μ. Now, suppose that at step 8, the dishonest prover sends to B, $y = x_q + z$. This vector satisfies:

- $\omega(y) \neq \sum_{i=1}^{\frac{\ell}{3}+q} w_i = \sum_{i=1}^{\ell} b_i w_i$ (since $\frac{\ell}{3}+q$ is odd),
- $Hy^{\mathrm{t}} = H(x_q^{\mathrm{t}}) + x = \sum_{i=1}^{\frac{\ell}{3}+q} t_i = \sum_{i=1}^{\ell} b_i t_i$,
- $\omega(x_q) - \mu \leq \omega(y) \leq \omega(x_q) + \mu$, now $\omega(x_q) = \sum_{i=1}^{\frac{\ell}{3}+q} w_i = \sum_{i=1}^{\ell} b_i w_i$.

Thus, this vector satisfies the three conditions of the protocol. Hence, an intruder can misrepresent himself as Alice every second time (when $b = e\pi$) without knowing the secret s.

Here is how to construct the three sequences $(w_i)_{1\leq i\leq \ell}, (t_i)_{1\leq i\leq \ell}$ and $(x_j)_{0\leq j\leq \frac{\ell}{3}}$ (let $\tilde{A}$ be a dishonest prover):

- $\tilde{A}$ randomly computes a word z of weight μ. Let $x = Hz^{\mathrm{t}}$,
- $\tilde{A}$ randomly chooses ℓ odd integers $w_1,\ldots,w_\ell$, and computes a vector x_0 of weight $\sum_{i=1}^{\frac{\ell}{3}} w_i$. Let $t = H(x_0^{\mathrm{t}}) + x$.
- $\tilde{A}$ computes $\frac{\ell}{3}$ vectors of length k, $t_1,\ldots,t_{\frac{\ell}{3}}$, such that $t = \sum_{i=1}^{\frac{\ell}{3}} t_i$,
- For j from 1 to $\frac{\ell}{3}$, $\tilde{A}$ computes:

- x_j a vector of weight $\sum_{i=1}^{\frac{\ell}{3}+j} w_i$,
- the vector $t_{\frac{\ell}{3}+j} = H(x_j^t) + \sum_{i=1}^{\frac{\ell}{3}+j-1} t_i + x$ (hence $H(x_j^t) = \sum_{i=1}^{\frac{\ell}{3}+j} t_i + x$),

• $\tilde{A}$ randomly chooses $t_{\frac{2\ell}{3}+1}, \ldots, t_\ell$, and sends to B, the quantities $w_1, \ldots, w_\ell$ and $t_1, \ldots, t_\ell$. It is easy to check that these sequences satisfy (3.1).

4 New Parameters

If the original protocol is repeated r times, then the probability of success of the previous attack is bounded by 2^{-r}. For $r = 20$, a dishonest prover has only one chance over one million to satisfy the protocol and the performances of the scheme are the following:

Rom	1002000 bits
Prover's Workfactor	$2^{32.5}$
Transmission rate	3078000 bits

Table 2. New Parameters

So as to reduce these parameters, it is enough to lower the size of the matrix H. The other schemes based on SD problem ([6], [14]) use a $(256, 512)$ parity check matrix. We are going to show that such a matrix cannot be used in Harari's scheme.

First, notice that if someone is able to find the vectors r_i from the pair (w_i, t_i) then he can find the secret s from the vector y sent by the prover. This is exactly the SD problem. To find r_i is equivalent to find a word c_i of weight $w_i + 1$ (whose last bit is 1) in the binary code whose parity check matrix is $(H \mid t_i)$.

Probabilistic algorithms defined by J.S. Leon [10] and J. Stern [13] are able, for some parameters, to solve this problem. Their efficiency depends essentially on the weight of the word to find. When Harari's scheme was proposed, there were very few results concerning the validity of these algorithms. Since the recent work of A. Canteaut and H. Chabanne [2], and F. Chabaud [3], their practical efficiency has been proved.

The success of the proposed attack is essentially due to the fact that the vectors r_i (computed at step 2) have disjoint supports and so they must verify

$$\sum_{i=1}^{\ell} \omega(r_i) \leq n \tag{4.1}$$

Thus the integers w_i are constrained.

Let $\tilde{B}$ be a dishonest verifier and let $n = 512$ and $k = 256$:

• Suppose first that the weight of the r_i satisfy $w_i \simeq \frac{n}{\ell}$. Notice that it is enough to choose $\ell = 60$ so as the number of vectors e that can be chosen by B be sufficiently high in order to prevent any imposture from a dishonest prover (by pre-computing for each possible query an answer which satisfies the protocol). Thus each vector r_i as a weight less or equal than 9. The search for a word of weight 10 in a $(513, 256)$ code needs about 2^{24} operations. Thus $\tilde{B}$ can compute all the vectors r_i in less than 2^{30} operations. It is clear that the efficiency of this attack grows with ℓ because of relation (4.1).

• To avoid this attack, taking into account the results given in [2], [3], the prover can try to choose some r_i with weight much more greater than 9 so that the knowledge of w_i and t_i be insufficient to find back r_i. But clearly this implies that all the other vectors will have a weight strictly less than 9. As an example, we will consider that the prover computes as much r_i as possible that cannot be found by Leon's or Stern's algorithm.

The search for a word of weight 45 in a $(513, 256)$ code needs about 2^{60} operations. Thus, for $\ell = 60$, the maximum number of vectors r_i (with disjoint supports) that can have weight 45 is 10, and in this case all the other vectors have weight no more greater than 2.

For any ℓ, this maximum number is given by $N = \lfloor \frac{512-\ell}{44} \rfloor$, and a simple computation shows that for ℓ being between 60 and 100, all the other vectors could not have their weight greater than 2. Thus, it is easy for $\tilde{B}$ to find back $\ell - N$ of the r_i in less than $(\ell - N)2^{20}$ operations (some r_i are columns from H, the others can be found by exhaustive search). Let $r_{i_1}, \ldots, r_{i_{\ell-N}}$ denote the vectors known by $\tilde{B}$. Now $\tilde{B}$ chooses at step 7 of the protocol $b = e\pi$. Notice that for any q $(0 \leq q \leq \frac{\ell}{3})$, $\frac{\ell}{3} + q < \ell - N$. Indeed, since $N \leq \frac{512-\ell}{44}$, then $N < \frac{\ell}{3}$, as soon as $\ell > 33$. Thus the support of b can be included in $\{i_1, \ldots, i_{\ell-N}\}$. In this case, $\tilde{B}$ can find back s from the answer y. The probability of this event is (for any ℓ and $\omega(e) = \frac{\ell}{3} + q$):

$$P_{\ell,q} = \frac{\binom{\ell-N}{\frac{\ell}{3}+q}}{\binom{\ell}{\frac{\ell}{3}+q}}$$

When ℓ is fixed, $P_{\ell,q}$ is minimum when $q = \frac{\ell}{3}$. Thus the complexity C_ℓ of the attack is bounded by $\frac{\Omega_\ell}{P_{\ell,\frac{\ell}{3}}}$ where Ω_ℓ is the workfactor for finding back the $\ell - N$ vectors r_i. Table 3 shows the evolution of this value for $60 \leq \ell \leq 100$.

ℓ	$\log_2(C_\ell)$
60	44.5
70	44
80	42
90	41.9
100	41.9

Table 3. Workfactor of the attack

Thus, it is clear that a $(256, 512)$ matrix cannot be used in Harari's scheme. Similar results can be found for a $(512, 1024)$ matrix.

References

1. E.R. Berlekamp, R.J. Mc Eliece & H.C.A. Van Tilborg: On the inherent intractability of certain coding problems, IEEE Trans. Inform. Theory, (1978) 384–386
2. A. Canteaut & H. Chabanne: A further improvement of the work factor in an attempt at breaking Mc Eliece's cryptosystem, Proceedings of Eurocode'94, (1994) 163–167
3. F. Chabaud: On the Security Of Some Cryptosystems Based On Error-Correcting Codes, Pre-proceedings of Eurocrypt'94, (1994) 127–135
4. U. Feige, A. Fiat & A. Shamir: Zero-knowledge proofs of identity, Proc. 19th ACM Symp. Theory of Computing, (1987), 210–217
5. A. Fiat, A. Shamir: How To Prove Yourself: Practical Solutions to Identification and Signatures Problems, Advances in Cryptology, Crypto'86, Lecture Notes in Computer Science **263**, (1986) 186–194
6. M. Girault: A (non-practical) three-pass identification protocol using coding theory, Advances in Cryptology, Auscrypt'90, Lecture Notes in Computer Science **453**, (1990) 265–272
7. S. Goldwasser, S. Micali and C. Rackoff: The knowledge complexity of interactive proof systems, Proc. 17th ACM Symp. Theory of Computing, (1985), 291–304
8. L.C. Guillou and J.-J. Quisquater: A practical zero-knowledge protocol fitted to security microprocessor minimizing both transmission and memory, Advances in Cryptology, Eurocrypt'88, **330**, (1988) 123–128
9. S. Harari: A New Authentication Algorithm, Proceedings of Coding Theory and Applications, Lecture Notes in Computer Science **388**, (1988) 91–105
10. J.S. Leon: A probabilistic algorithm for computing minimum weights of large error-correcting codes, IEEE Trans. Inform. Theory, **IT-34(5)**: 1354–1359
11. F.J. MacWilliams & N.J.A. Sloane: The Theory of error-correcting codes, North-Holland, Amsterdam-New-York-Oxford, 1977
12. C.P. Schnorr: Efficient signature generation by smart cards, Journal of Cryptology, **4**, (1991) 161–174
13. J. Stern: A method for finding codewords of small weight, Coding Theory and Applications, Lecture Notes in Computer Science **434**, 173–180
14. J. Stern: A new identification scheme based on syndrome decoding, Crypto'93, Lecture Notes in Computer Science **773**, Springer-Verlag (1994) 13–21