Improved Identification Schemes Based on Error-Correcting Codes

Pascal Véron

AAECC 8, 57–69 (1997)

AAECC
Applicable Algebra in Engineering, Communication and Computing

Improved Identification Schemes Based on Error-Correcting Codes

Pascal Véron

G.E.C.T., Université de Toulon et du Var, B.P. 132, F-83957 La Garde Cedex, France
veron@marie.polytechnique.fr

Received March 10, 1995; revised version December 1, 1995

Abstract. As it is often the case in public-key cryptography, the first practical identification schemes were based on hard problems from number theory (factoring, discrete logarithms). The security of the proposed scheme depends on an NP-complete problem from the theory of error correcting codes: the syndrome decoding problem which relies on the hardness of decoding a binary word of given weight and given syndrome. Starting from Stern's scheme [18], we define a dual version which, unlike the other schemes based on the SD problem, uses a generator matrix of a random linear binary code. This allows, among other things, an improvement of the transmission rate with regards to the other schemes. Finally, by using techniques of computation in a finite field, we show how it is possible to considerably reduce:

- the complexity of the computations done by the prover (which is usually a portable device with a limited computing power),
- the size of the data stored by the latter.

Keywords: Identification scheme, NP-complete problem, SD problem, Zero-knowledge.

1. Introduction

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

From a theoretical point of view, this can be done by using a zero-knowledge interactive proof system. This type of interactive proofs was introduced in 1985 by S. Goldwasser, S. Micali and C. Rackoff [11]. In 1986, A. Fiat and A. Shamir have demonstrated the practical significance of such proofs so as to establish user identities and to digitally sign messages [7].

From a practical point of view (for example a smart card application), the prover can be identified to a smart card and the verifier to the organization receiving this card. Thus, it is supposed that the prover has a reduced computational power and a little amount of memory. These constraints do not apply to the verifier.

The first (practical) zero-knowledge identification schemes were based on hard problems from number theory (factoring, discrete logarithm) and relied on arithmetic operations on large numbers.

Since 1988, some new schemes, which use simple operations and whose security depends on an NP-complete problem, have been proposed (see [9, 12, 15, 16, 18, 19]).

Among all these schemes, three of them [9, 12, 18] rely on an NP-complete problem from error correcting codes: the one of Syndrome Decoding ("SD" problem [3]).

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 $He^t = i$ and $\omega(e) \leq p$?

Remark. e^t denotes the transpose of the vector e and $\omega(e)$ its Hamming weight.

As mentioned in [3], the SD problem, stated in terms of generator matrix is also NP-complete since one can go from the parity-check to the generator matrix (or vice-versa) in polynomial time. Thus the decision problem of the SD problem becomes:

Name: G-SD
Input: $G(k, n)$ a generator matrix of a binary $[n, k]$ code $\mathscr{C}$, x a binary vector of length n and p an integer.

Question: Is there a vector e of length n and weight p such that $x + e \in \mathscr{C}$?

Thus the problem is to know if there exists (m, e) such that $x = mG + e$ where e is a word of weight p. The identification schemes proposed by M. Girault, S. Harari and J. Stern use a parity check matrix of a random linear binary code, which is common to all users. Like Niederreiter did with McEliece's cryptosystem, we define, in this paper, the dual version of Stern's scheme by using the generator matrix of a random linear binary code. In this way, we obtain a zero-knowledge scheme which, among all the schemes based on the SD problem, has the smallest transmission rate.

In the last two sections, we study optimal parameters and we propose a variant of the scheme so as to:

- reduce the complexity of the computation done by the prover,
- lower the size of the matrix stored by the prover.

These two constraints are to be taken into account if the protocol is to be implemented in a smart card (or at least the part of the scheme concerning the prover).

2. The Identification Scheme

The principle of the protocol is the following: Alice (the prover) knows the solution of an NP-complete problem for a given data s (its secret). Bob (the verifier) asks Alice a series of questions. If Alice really knows s, she can answer all the questions correctly. If she does not, she has a probability q of answering correctly. After r successful iterations of the protocol, Bob will be convinced that Alice knows s with

probability $1 - q^r$. Furthermore, none of the questions or answers gives Bob any information about Alice's secret (only about her knowledge of it).

In what follows small letters will be used for vectors and capital letters for matrices. Sometimes the term "word" will be used instead of "vector". All operations are computed over $\mathbb{F}_2$.

Notations.
- Let σ be a permutation over $\{1, \ldots, n\}$ and y be a vector of length n, then $y\sigma$ is defined as the vector z such that $z_j = y_{\sigma(j)}$ for $1 \leq j \leq n$,
- Let x be a binary vector of length n, $\omega(x)$ is the weight of x, i.e. the number of bits of x whose value is 1,
- $\langle x \rangle$ denotes the action of a collision-free hash function on the string x,
- A vector is considered as a matrix with one row,
- M^t (*resp.* y^t) denotes the transpose of the matrix M (*resp. of the word* y). We also introduce some notations given in [8]:
- $\bar{A}$ represents the real prover who follows its designated protocol,
- $\tilde{A}$ represents a probabilistic polynomial time cheater,
- A represents either $\bar{A}$ or $\tilde{A}$,
- $\bar{B}$ represents the real verifier who follows its designated protocol,
- $\tilde{B}$ represents an arbitrary polynomial time program which tries to extract additional information from $\bar{A}$,
- (X, Y) represents the execution of the two party protocols in which X is the prover and Y is the verifier,
- $(X, Y)[I]$ denotes the state of the polynomial-time probabilistic Turing machine Y at the end of the execution of the protocol when X and Y share a public data I. This state is equal to:
  - "success" if A satisfies the protocol,
  - "failure" otherwise.

The scheme uses a random binary matrix $G(k, n)$ common to all users. It can be considered as the generator matrix of a random linear binary code. Without loss of generality, we can assume that G is given under the form $G = (I_k | M)$ where M is a random $k \times (n - k)$ matrix, since it is well known that a Gaussian elimination doesn't change the code generated by G. The context is as follows:

Common Public Data : $G(k, n)$ a binary matrix of rank k, a hash function denoted by $\langle \cdot \rangle$,
Prover's Secret Data : m a binary vector of length k, e a binary vector of length n,
Prover's Public Data : $x = mG + e$ and $p = \omega(e)$.

Remarks.
- The pair (x, p) is the public identification of the prover,
- The prover's data can be computed by a certification center having the confidence of all users or the prover can choose his secret keys and the center certifies the corresponding public keys.

Suppose that A wants to prove his identity to B. The protocol includes r rounds, each of these being performed as follows:

- A randomly computes:
  - a binary vector u of length k,
  - a random permutation σ of the set $\{1, \ldots, n\}$,

and sends to B three commitments:

$$c_1 = \langle \sigma \rangle, c_2 = \langle (u+m)G\sigma \rangle, c_3 = \langle (uG+x)\sigma \rangle$$

- B sends a random element b of $\{0,1,2\}$.
- If b is 0,
  - A discloses $u+m$ and σ,
  - B checks the validity of c_1 and c_2 (since G and $\langle \cdot \rangle$ are public).
- If b is 1,
  - A discloses $(u+m)G\sigma$ and $e\sigma$,
  - B checks the validity of c_2 and c_3 and that $\omega(e\sigma) = p$($\langle \cdot \rangle$ is public and since $x = mG + e$, then $(u+m)G\sigma + e\sigma = (uG+x)\sigma$).
- If b is 2,
  - A discloses σ and u,
  - B checks the validity of c_1 and c_3 (since $\langle \cdot \rangle$, G and x are public).

3. Properties of the Scheme

Let $I = \{G, x, p\}$ be the public data shared by A and B and let $P(I, s)$ be the following predicate:

$P(I, s)$ = "s is a pair (m, e) which satisfies $x = mG + e, \omega(e) = p$; $G, x, p \in I$" then

Proposition 3.1. *The protocol is an interactive proof of knowledge for $P(I, s)$.*

Proof.

Completeness: It is obvious that each prover which knows a valid pair (m, e) for the public data I can answer correctly any of B's queries. Thus

$$Pr((\bar{A}, \bar{B})[I] = \text{"success"}) = 1.$$

Soundness:

Lemma 3.2. *If $\bar{B}$ accepts $\tilde{A}$ proof with probability $\geq (\frac{2}{3})^r + \varepsilon$, then there exists a polynomial time probabilistic machine M which, with overwhelming probability, either computes a valid secret pair (m, e) or finds a collision for the hash function.*

Proof. Let T be the execution tree of $(\tilde{A}, \bar{B})$ corresponding to all possible questions of the verifier when the adversary has a random tape RA. $\bar{B}$ may ask 3 possible questions at each stage. First we are going to show that, unless a hash collision has been found, a secret key (m, e) can be computed from a vertex with 3 sons. Then we will show that a polynomial time M can find such a vertex in T with overwhelming probability.

Let V be a vertex with 3 sons. This corresponds to a situation where 3 commitments c_1, c_2, c_3 have been made and where the three queries were properly answered. Let $u' + m'$ and σ' be the answers to the query $b = 0$; y'' and e'' be the answers to the query $b = 1$, and u''', σ''' be the answers to the query $b = 2$. We have

$$\begin{aligned} \langle \sigma' \rangle = c_1 = \langle \sigma''' \rangle \\ \omega(e'') = p \\ \langle (u'+m')G\sigma' \rangle = c_2 = \langle y'' \rangle \\ \langle y'' + e'' \rangle = c_3 = \langle (u'''G + x)\sigma''' \rangle \end{aligned}$$

Thus, either a collision for the hash function has been found, or else

$$x = (u' + m' + u''')G + e''\sigma'^{-1},$$

where $e''\sigma'^{-1}$ is a word of length n and weight p. Therefore, the pair $(u' + m' + u''', e''\sigma'^{-1})$ is a valid secret key and can be used in order to impersonate $\tilde{A}$.

Now, the assumption implies that the probability for T to have a vertex with 3 sons is at least ε. Indeed, let us consider RA as a set of μ elements, where $\tilde{A}$ randomly picks its values, and let Q be the set $\{0,1,2\}$. These two sets are considered as probability spaces both of them with the uniform distribution.

A pair $(c,b) \in (RA \times Q)^r$ represents the commitments, queries and answers exchanged between $\tilde{A}$ and $\bar{B}$ during an identification process. We will say that (c,b) is a "valid" pair, if the execution of $(\tilde{A}, \bar{B})$ leads to the success state.

Let V be the subset of $(RA \times Q)^r$ composed of all the valid pairs. The hypothesis of the lemma means that:

$$\frac{\text{card}(V)}{\text{card}((RA \times Q)^r)} \geq \left(\frac{2}{3}\right)^r + \varepsilon.$$

Let Ω_r be a subset of RA^r such that:

- If $c \in \Omega_r$, then $2^r + 1 \leq \text{card}\{b, (c,b) \text{ be valid}\} \leq 3^r$,
- If $c \in RA^r \setminus \Omega_r$, then $0 \leq \text{card}\{b, (c,b) \text{ be valid}\} \leq 2^r$.

Then, $V = \{\text{valid } (c,b), c \in \Omega_r\} \cup \{\text{valid } (c,b), c \in RA^r \setminus \Omega_r\}$, therefore:

$$\text{card}(V) \leq \text{card}(\Omega_r)3^r + (\mu^r - \text{card}(\Omega_r))2^r.$$

Thus

$$\frac{\text{card}(V)}{\text{card}((RA \times Q)^r)} \leq \left(\frac{\text{card}(\Omega_r)}{\text{card}(RA^r)} + 2^r\left(3^{-r} - \frac{\text{card}(\Omega_r)}{\text{card}((RA \times Q)^r)}\right)\right)$$

$$\leq \frac{\text{card}(\Omega_r)}{\text{card}(RA^r)} + \left(\frac{2}{3}\right)^r.$$

It follows that:

$$\frac{\text{card}(\Omega_r)}{\text{card}(RA^r)} \geq \varepsilon.$$

This shows that the probability that an intruder might answer to (at least) $2^r + 1$ of the verifier's queries, by choosing random values, is greater than ε. Now, if more than $2^r + 1$ queries are bypassed by an intruder then $T(RA)$ has at least $2^r + 1$ leaves, i.e. $T(RA)$ has at least a vertex with 3 sons.

So, by resetting $\tilde{A}\frac{1}{\varepsilon}$ times, and by repeating again, it is possible to find an execution tree with a vertex with 3 sons with probability arbitrary close to one. □

The first conclusion of this lemma implies that $\langle \cdot \rangle$ is not collision free and the second conclusion contradicts the intractability, in polynomial time, of the G-SD problem. It follows that:

$$Pr((\tilde{A}, \bar{B})[I] = \text{"success"}) \leq \left(\frac{2}{3}\right)^r. \qquad \square$$

Let us denote by $R_{A,B}$ the concatenation of all the bits exchanged between A and B during an identification process. We will say that $R_{A,B}$ is the *communication tape* of

(A, B). Because of the probabilistic nature of interactive protocols, a probability distribution is defined on $R_{A,B}$. We can state the following result:

Proposition 3.3. *The protocol is a zero-knowledge interactive proof for $P(I, s)$ in the random oracle model.*

Proof. Let us denote by $x \square y$ the concatenation of the binary strings x and y. So as to mimic a dishonest verifier, we have to assume that he will devise a peculiar strategy with regards to the commitments sent by the prover. Let $St(c_1, c_2, c_3)$ be such a strategy. We have $St(c_1, c_2, c_3) \in \{0, 1, 2\}$. Let us consider:

$$\phi_m : \mathbb{F}_2^k \to \mathbb{F}_2^k$$
$$u \mapsto u + m$$
$$\psi : \mathbb{F}_2^k \to \mathbb{F}_2^n$$
$$u \mapsto uG$$

ϕ_m is an automorphism of $\mathbb{F}_2^k$ and if $\mathscr{C}$ is the code generated by G, then ψ is an isomorphism of $\mathbb{F}_2^k$ onto $\mathscr{C}$.

Here is the polynomial-time probabilistic Turing machine M which produces a communication tape whose probability distribution is indistinguishable from the probability distribution of a communication tape coming from a fair identification process.

1. M randomly picks a query, b of $\{0, 1, 2\}$,
   - If $b = 0$, M chooses:
     · y a random element of $\mathbb{F}_2^k$,
     · σ any permutation of $\{1, \ldots, n\}$,

   and computes $c_1 = \langle \sigma \rangle$ and $c_2 = \langle yG\sigma \rangle$ and substitutes c_3 by a random string. Let $H = c_1 \square c_2 \square c_3$ and $Ans = y \square \sigma$. It is obvious that y and $u + m$ have the same probability distribution. Indeed, let z be any element of $\mathbb{F}_2^k$, since u is a random element of $\mathbb{F}_2^k$, then:

$$Pr(u + m = z) = Pr(u = \phi_m^{-1}(z))$$
$$= \frac{1}{2^k}$$
$$= Pr(y = z).$$

   - If $b = 1$, M chooses:
     · y any element of the code $\mathscr{C}$,
     · e' any element of $\mathbb{F}_2^n$ of weight p,
     · σ any permutation of $\{1, \ldots, n\}$,

   and computes $c_2 = \langle y\sigma \rangle$ and $c_3 = \langle (y + e')\sigma \rangle$ and substitutes c_1 by a random string. Let $H = c_1 \square c_2 \square c_3$ and $Ans = y\sigma \square e'\sigma \cdot e'\sigma$ has the same probability distribution than $e\sigma$, moreover let z be any codeword of $\mathscr{C}$, then

$$Pr((u + m)G = z) = Pr(u = \phi_m^{-1}(\psi^{-1}(z)))$$
$$= 2^{k-n}$$
$$= Pr(y = z),$$

   thus $(u + m)G$ is uniformly distributed among codewords of $\mathscr{C}$ and can be replaced by y.

– If $b = 2$, M chooses:
  · y any element of $\mathbb{F}_2^k$,
  · σ any permutation of $\{1, \ldots, n\}$,
  and computes $c_1 = \langle \sigma \rangle$ and $c_3 = \langle (yG + x)\sigma \rangle$ and substitutes c_2 by a random string. Let $H = c_1 \square c_2 \square c_3$ and $Ans = y \square \sigma$.
2. M computes $b' = St(H)$,
3. If $b' = b$, then M writes on the tape $\mathscr{R}$, the quantities H, b and Ans, otherwise M goes back to step 1.

Thus, in $3r$ rounds on average, M produces a communication tape $\mathscr{R}$, indistinguishable from a communication tape $\mathscr{R}_{A,B}$ coming from a fair identification process executed in r rounds. □

Remark. The Random Oracle model [1] allows to assume that the hash function has specific statistical independence properties.

4. Security of the Scheme

Security of the scheme is linked to the parameters n, k, p and r. Two kinds of attack are to be considered:

- Let H be a parity check matrix of the code defined by G. In order to impersonate $\bar{A}$, an intruder has to be able to compute a word e' of weight p whose image under H is Hx^t (this is the SD problem). Indeed, in this case $H(x + e')^t = 0$, thus $x + e'$ is a codeword of the code defined by G, i.e. $x + e' = m'G$. Then the new pair (m', e') is a valid secret key. The search for e' is an NP-hard problem and is equivalent to the search for a word of weight $p + 1$ (whose last bit is 1) in the code defined by the following parity check matrix:

$$(H | Hx^t)$$

- Let G' be the matrix defined by

$$G' = \begin{pmatrix} G \\ x \end{pmatrix}$$

Then e belongs to the code $\mathscr{C}'$ defined by G'. Thus, the search for e is equivalent to the search for a codeword of weight p in the code $\mathscr{C}'$.

Therefore, these two attacks boil down to the search for a word of given weight in a particular linear binary code. For some parameters, probabilistic algorithms of A. Canteaut & H. Chabanne [5], J. S. Leon [13] and of J. Stern [17] can find a solution to this problem. These algorithms can easily find words of small weight even in very large code. Their efficiency is largely detailed in [6].

In the general case, the search for a word of weight p in a random binary code becomes very complex when p is close to the Varshamov Gilbert bound. This bound gives a theoretical estimate of the minimal distance d of a random linear binary code. For

$$n = 512, \quad k = 256, \quad p = 56$$

the workfactor of the different algorithms is about 2^{70}. This size guarantees the security of the scheme.

However, according to the previous section, the probability of success of an intruder is bounded by $(\frac{2}{3})^r$. Indeed, without knowing the secret pair (m, e), various strategies can be used in order to impersonate $\bar{A}$:

- Randomly choose u and σ and replace m by some arbitrary vector m'. In this case, the false prover hopes that b is 0 or 2,
- Randomly choose u and σ and replace x by $x' = m'G + e'$ where m' is a random vector of length k and e' is a random vector of length n and weight p. In this case, the false prover hopes that b is 0 or 1,
- Randomly choose u and σ and compute $(uG + x)$. Then, compute y' and e' such that $(uG + x) = y' + e'$ and $\omega(e') = p$. Replace $(u + m)G$ by y', and e by e'. In this case, the false prover hopes that b is 1 or 2.

Thus minimal parameters which guarantee the security of the scheme are

$$n = 512, \quad k = 256, \quad p = 56 \quad \text{and} \quad r = 35.$$

The complexity of the various attacks is then, at least 2^{70} and the probability of success of the different frauds is about 10^{-6}.

5. Performances of the Scheme

Let us recall that all operations are done over the two-element field $\{0, 1\}$. That is to say that all operations are performed over bits. An addition between two bits is a logical "xor" and a multiplication is a logical "and".

If $n = \mathcal{O}(k)$, the complexity of the computations done by the verifier and the prover is quadratic ($\mathcal{O}(rk^2)$) as in Stern's scheme. The complexity of Harari's scheme and Girault's scheme is cubic ($\mathcal{O}(rk^3)$).

The scheme is "flexible", that is, if there was a successful approach to solve the SD problem for the suggested parameters, the algorithm could be altered to use, say, $n = 1024$, $k = 512$ and $p = 110$.

Operations to perform are very simple and can be implemented in hardware in a quite efficient way. Moreover all the tricks proposed for the implementation of Stern's scheme (see [18]) can be applied to our scheme.

The only "complex" operation, that has to be done by the prover, is the computation of uG. In the next section, new parameters are given so as to lower the complexity of this computation and reduce the transmission rate of the scheme.

To show the efficiency of the scheme, we have compared it with the schemes of M. Girault, S. Harari and of J. Stern. Comparisons have been made without using none of the tricks suggested by the authors. For the length and the kind of hash function which are used in the different schemes, the reader can refer to [10]. Conventions taken are the following:

- Chosen parameters are minimal parameters which guarantee the security of the scheme ($n = 2000$, $k = 1000$ for Harari's scheme, $n = 512$, $k = 256$, $p = 56$ for Girault's scheme, Stern's scheme and for our scheme),
- the hash values are 128 bits long,
- a permutation comes from a seed of 120 bits via a pseudo random generator.

Results are given in Table 1.

Remark. "ZK" means that it has been proved formally that the scheme is an interactive zero-knowledge proof system while "IP" means that it has only been proved that the scheme is an interactive proof system.

The proposed scheme has the smallest transmission rate. In the next section we propose new parameters so as to reduce the latter more and so as to lower the size of the data stored by the prover.

Table 1. Comparison of the SD schemes

	Stern's scheme	Harari's scheme	Girault's scheme	Our scheme
Rounds	35	1	20	35
No. of bits sent by round	$\simeq 1146$	–	$\simeq 164412$	$\simeq 976$
Global transmission rate	$\simeq 40133$ bits	204200 bits	$\simeq 3288240$ bits	$\simeq 34160$bits
Properties	ZK	–	IP	ZK

6. Reduction of the Size of the Data and of the Transmission Rate (Optimal Parameters)

Contrary to the scheme proposed in [9, 12, 18], it is enough to lower the dimension of the matrix G in order to reduce the transmission rate of the scheme. Thus, for $n = 512$, we have searched the minimal value of k for which

$$\Omega(n, k, p) > 2^{60},$$

where ρ is the theoretical minimal distance of a binary linear $[n, k]$ code and $\Omega(n, k, \rho)$ denotes the workfactor of the probabilistic algorithms mentioned in section 4. Notice that this workfactor may not be considered as unacceptable, if the protocol is to be implemented (for example) on a bounded-life smart card. We have found the following (minimal) parameters:

$$n = 512, \quad k = 120, \quad p = 114.$$

For such parameters, the first attack (*cf. section 4*) consists in finding a word of weight 115 in a (513, 121)-code. The second attack consists in finding a word of weight 114 in a (512, 121)-code. The workfactor for these attacks is greater than 2^{60} [6].

Notice, that the reduction of the dimension of the matrix implies a slight decrease in the computation complexity of uG. Moreover, the non-systematic part of the matrix G (which must be stored by the prover) is made up now of 47040 bits instead of 65536 bits.

Table 2 sums up the performances of the new scheme. We have compared it with Stern's scheme since, among the schemes based on SD problem, the latter has the smallest computation complexity and the smallest transmission rate. Convention taken are the same as the one taken for table 1. We have studied:

- the complexity of the computations (additions, multiplications over bits) done by the prover,
- the transmission rate between the prover and the verifier,
- the size of the matrix stored by the prover.

Of course, the matrices being of the form $(I_k | M)$, only M is stored.

Notice that it is useless in Stern's scheme to choose a (120, 512) matrix since the transmission rate depends on the length of the code (i.e. the number of columns of the public matrix) and not on its dimension.

As compared to Stern's scheme, the improved scheme saves about 25% of the transmitted bits. One way to obtain such a gain in Stern's scheme is to use a public matrix which has at most 320 columns and more than 60 rows. For such parameters, the probabilistic algorithms mentioned in section 4 are quite efficient. As an

Table 2. Improved scheme

	Stern's scheme	Our scheme	Improved scheme
Rounds	35	35	35
No. of bits sent by round	$\simeq 1146$	$\simeq 976$	$\simeq 885$
Global transmission rate	$\simeq 40133$ bits	$\simeq 34160$ bits	$\simeq 30986$ bits
Size of the matrix	65536 bits	65536 bits	47040 bits
Prover's Workfactor	$\simeq 2^{22.13}$	$\simeq 2^{22.5}$	$\simeq 2^{22}$
Properties	ZK	ZK	ZK

Table 3. Transmission rate of other schemes

PKP	$\simeq 17760$ bits
CLE	$\simeq 29400$ bits
PPP	$\simeq 58448$ bits

example, for $n = 320$ and $k = 160$, p is about 35 and the workfactor of an attack is about 2^{46}.

For information, table 3 gives the transmission rate of the three-pass version of the schemes based on another NP-complete problem (except for PKP scheme which has only a five-pass version):

- the Permuted Kernel Problem [16],
- the Constraint Linear Equation problem [19],
- the Permuted Perceptron Problem [15].

Remark. For these three schemes computations are done over bytes, not over bits.

We now propose a variant which considerably reduces:

- the size of the data stored by the prover,
- the computation complexity of the latter.

7. A Variant Using Computations in $\mathbb{F}_{2^k}$

As previously mentioned, the most "complex" operation, which must be done by the prover, is the computation of uG. For the initial parameters, this operation needs about 2^{17} logical operations over bits. The proposed variant is only valid when $k|n$ and will not work for all k [20]. As an example, it is supposed that n is equal to $2k$.

Let $\beta = \{\beta_1, \ldots, \beta_k\}$ be a basis of $\mathbb{F}_{2^k}$ and let $x = \sum_{i=1}^{k} x_i \beta_i$ be an arbitrary element of $\mathbb{F}_{2^k}$. In what follows, we will often identify the element x with the vector $(x_1, \ldots, x_k)$ which belongs to $\mathbb{F}_2^k$. From now on, it is supposed that $\mathbb{F}_{2^k}$ is generated by the polynomial $z^k + z^j + 1$, when, of course, such a polynomial is irreducible [2, 20].

Proposition 7.1. *Let $\beta = \{1, \alpha, \ldots, \alpha^{k-1}\}$ be a basis of $\mathbb{F}_{2^k}$, α being a root of an irreducible polynomial of degree k. Let $\gamma = \sum_{i=1}^{k} \gamma_i \beta_i$ be a fixed element of $\mathbb{F}_{2^k}$ and $\rho = \sum_{i=1}^{k} \rho_i \beta_i$ be an arbitrary element of $\mathbb{F}_{2^k}$. The product $\rho\gamma$ needs at most $k(\omega_\beta(\gamma) - 1) + k - 1$ additions between bits* [2, 14], *where $\omega_\beta(\gamma)$ denotes the Hamming weight of $(\gamma_1, \ldots, \gamma_k)$.*

Notice that since $\mathbb{F}_{2^k}$ is isomorphic to $\mathbb{F}_2^k$, we can write that

$$\rho\gamma = (\rho_1, \ldots, \rho_k)[\gamma]_\beta$$

where

$$[\gamma]_\beta = \begin{bmatrix} \overline{\gamma\beta_1} \\ \vdots \\ \overline{\gamma\beta_k} \end{bmatrix}$$

is a $k \times k$ binary matrix, $\overline{\gamma\beta_i}$ being the binary image of $\gamma\beta_i$. We will say that $[\gamma]_\beta$ is the *β-product matrix* of γ.

Now, if k is fixed, the computation of $\rho\gamma$ depends essentially on $\omega_\beta(\gamma)$. Thus, the main idea is to replace the matrix G by the β-product matrix of two arbitrary elements γ_1 and γ_2 such that $\omega_\beta(\gamma_1)$ and $\omega_\beta(\gamma_2)$ be small. Thus the public matrix G is replaced by the following $(k, 2k)$ matrix:

$$([\gamma_1]_\beta, [\gamma_2]_\beta).$$

The protocol is still the same, but the computation of uG (*resp.* of $(u+m)G\sigma$) is replaced by the computation of $(u\gamma_1, u\gamma_2)$ (*resp.* of $((u+m)\gamma_1, (u+m)\gamma_2)\sigma$) where the binary vectors u and m are considered as elements of $\mathbb{F}_{2^k}$. This can be easily done in parallel by four shift registers of size k. Moreover in order to compute these quantities the prover does not need to store the whole matrix G, he only needs to store γ_1 and γ_2 or, better still, since these values are fixed it is possible to compute the two products with Berlekamp's algorithm [4] by using dual trace basis. Last but not least, the computation of uG (*resp.* of $(u+m)G\sigma$) needs at most $k(\omega_\beta(\gamma_1) + \omega_\beta(\gamma_2)) - 2$ additions.

Now, for k equal to 255 there exists irreducible polynomials of the form $z^{255} + z^j + 1$. Thus, for

$$n = 510, \quad k = 255, \quad p = 56, \quad r = 35, \quad \omega_\beta(\gamma_1) = \omega_\beta(\gamma_2) = 80$$

the computation of uG needs about $2^{15.3}$ logical operations and we get, for an identification process, the following results:

- Size of the matrix stored by the prover: 510 bits,
- Complexity computation for the prover: $\leq 2^{20.8}$,
- Global transmission rate: $\simeq 34090$ bits.

We obtain, this way, a computation complexity which is, at least, 4 times smaller than the one of the other schemes based on the SD problem and a storage capacity which is at least 128 times smaller.

Thus, if the intractability assumption of the SD problem stays for this kind of matrix G, then we have an identification scheme which, among all the other schemes based on this problem, has:

- the smallest transmission rate,
- the smallest computation complexity,
- the smallest storage capacity needed by the prover.

Of course, the structure of the matrix G opens up new strategies of cryptanalysis. We point out here a possible attack.

We have $x = mG + e = (m\gamma_1 + e_1, m\gamma_2 + e_2)$, where e_1 and e_2 are two vectors of length k, and $e = (e_1, e_2)$. Let $\mathbb{F}_{2^{k'}}$ be the biggest subfield of $\mathbb{F}_{2^k}$ then, if $m \in \mathbb{F}_{2^{k'}}$ and $\mathrm{Tr}_{\mathbb{F}_{2^k}:\mathbb{F}_{2^{k'}}}(e_1) = 0$, the secret key (m, e) can be found. Indeed, let us recall that γ_1, γ_2 and

x are public data, thus it is possible to compute

$$\mathrm{Tr}_{\mathbb{F}_{2^k}:\mathbb{F}_{2^\ell}}(m\gamma_1 + e_1) = m\mathrm{Tr}_{\mathbb{F}_{2^k}:\mathbb{F}_{2^\ell}}(\gamma_1).$$

From this, we can find back m, and then we compute $e = x + mG$. The same attack can be done if $m \in \mathbb{F}_{2^\ell}$ and $\mathrm{Tr}_{\mathbb{F}_{2^k}:\mathbb{F}_{2^\ell}}(e_2) = 0$.

For $k = 255$, we have $\ell = 85$. So as to avoid this attack it is enough to choose $m \in \mathbb{F}_{2^k} \setminus \mathbb{F}_{2^\ell}$ which gives $2^{255} - 2^{85}$ possible choices for m.

7.1. Open Problem In fact, as pointed out by J. Stern, the underlying problem to solve for this kind of matrix G is the following:

Let γ_1, γ_2 and z be three public elements of $\mathbb{F}_{2^k}$. Find e_1 and e_2 two elements of $\mathbb{F}_{2^k}$ such that

$$\begin{cases} \gamma_1 e_2 + \gamma_2 e_1 & = z \\ \omega_\beta(e_1) + \omega_\beta(e_2) & = p. \end{cases}$$

Indeed, since x is public and is equal to $(m\gamma_1 + e_1, m\gamma_2 + e_2)$, we obtain that:

$$\underbrace{\gamma_2(m\gamma_1 + e_1) + \gamma_1(m\gamma_2 + e_2)}_{z} = \gamma_1 e_2 + \gamma_2 e_1.$$

We welcome attacks and suggestions from readers in order to solve such a system.

8. Conclusion

We have defined an identification scheme which among all the schemes based on the SD problem has, in its initial form, the smallest transmission rate. Moreover, we have proposed a variant so as to reduce the complexity computation of the prover and the size of the data stored by the latter.

9. References

1. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. Proceedings of the 1st ACM Conference Comput. Commun. Security, 62–73 (1993)
2. Berlekamp, E. R.: Algebraic Coding Theory, McGraw-Hill Book Company, 1968
3. Berlekamp, E. R., Mc Eliece, R. J., Van Tilborg, H. C. A.: On the inherent intractability of certain coding problems. IEEE Trans. Inform. Theory, 384–386 (1978)
4. Berlekamp, E. R.: Bit-Serial Reed-Solomon Encoders. IEEE Trans. Inform. Theory, vol IT-**28**, **6**, 869–874 (1982)
5. Canteaut, A., Chabanne, H.: A further improvement of the workfactor in an attempt at breaking Mc Eliece's cryptosystem. Proceedings of Eurocode'94, 163–167
6. Chabaud, F.: On the Security of Some Cryptosystems Based On Error-Correcting Codes, Eurocrypt'94. Lecture Notes in Computer Science Vol. 950, pp. 131–139. Berlin, Heidelberg, New York: Springer 1995
7. Fiat, A., Shamir, A.: How To Prove Yourself: Practical Solutions to Identification and Signatures Problems. Advances in Cryptology, Crypto'86, Lecture Notes in Computer Science Vol. 263, pp. 186–194. Berlin, Hiedelberg, New York: Springer
8. Feige, U., Fiat, A., Shamir, A.: Zero-knowledge proofs of identify. Proc. 19th ACM Symp. Theory of Computing, 210–217 (1987)
9. Girault, M.: A (non-practical) three-pass identification protocol using coding theory, Advances in Cryptology, Auscrypt'90, Lecture Notes in Computer Science Vol. 453, pp. 265–272. Berlin, Heidelberg, New York: Springer

10. Girault, M., Stern, J.: On the length of cryptographic hash-values used in identification schemes. Crypto'94, Lecture Notes in Computer Science Vol. 839, pp. 202–215 Berlin, Heidelberg, New York: Springer 1994
11. Goldwasser, S., Micali S., Rackoff, C.: The knowledge complexity of interactive proof systems. Proc. 17th ACM Symp. Theory Computing, 291–304 (1985)
12. Harari, S. A New Authentication Algorithm, Proceedings of Coding Theory and Applications, Lecture Notes in Computer Science Vol. 388, pp. 91–105, Berlin, Heidelberg, New York: Springer 1988
13. Leon, J. S.: A probabilistic algorithm for computing minimum weights of large error-correcting codes, IEEE Trans. Inform. Theory, IT-34(5): 1354–1359
14. MacWilliams, F. J., Sloane, N. J. A.: The Theory of error-correcting codes, North-Holland, Amsterdam-New-York-Oxford, 1977
15. Pointcheval, D.: Neural Networks and their cryptographic applications. Proc. Eurocode'94, 183–193
16. Shamir, A.: An efficient identification scheme based on permuted kernels. Proc. Crypto'89, Lecture Notes in Computer Science Vol. 435, pp. 606–609, Berlin, Heidelberg, New York: Springer
17. Stern, J.: A method for finding codewords of small weight. Coding Theory and Applications. Lecture Notes in Computer Science Vol. 434, pp. 173–180. Berlin, Heidelberg, New York: Springer
18. Stern, J.: A new identification scheme based on syndrome decoding, Crypto'93, Lecture Notes in Computer Science Vol. 773, pp. 13–21, Berlin, Heidelberg, New York: Springer 1994
19. Stern, J.: Designing identification schemes with keys of short size. Crypto'94, Lecture Notes in Computer Science Vol. 839, pp. 164–173, Berlin, Heidelberg, New York: Springer 1994
20. Zierler, N.: On the Theorem of Gleason and Marsh. Proc. Am. Math. Soc., 9: 236–237, Math. Rev., 20: 851, 1958