Authentication and Signature based on error-correcting codes

Guillaume GOY

April 6, 2022

Abstract

In this paper, we talk about the authentication based on error correcting code. We will start by presenting NP-hard problem, on which security stands, called syndrome decoding problem, and some derived problems. Secondly we present the Stern scheme presented in 1993, and its improvements. For each scheme, we show the three mains proprieties : zero-knowledge, soundness and completeness. Improvements stand mainly on two new features : the rank metric and the use of double circulant codes. We will present these two features. Then we will talk about the Veron protocol and improvements as well. We talk a little bit about the Chen scheme, which has been fully cryptanalized. Finally we show how to create a signature from a zero-knowledge authentication scheme thanks to the Fiat-Shamir heuristic.

Contents

1 Prerequisites **3**
- 1.1 Global Notations . . . 3
- 1.2 Error-correcting codes . . . 3
- 1.3 Rank metric . . . 4
- 1.4 Double Circulant Codes . . . 5

2 NP-hard problems based on error correcting codes **5**
- 2.1 Syndrome Decoding [SD] problem . . . 5
- 2.2 Generator Matrix Syndrome Decoding [G-SD] problem . . . 6
- 2.3 q-ary Syndrome Decoding [q-SD] problem . . . 6
- 2.4 Rank Syndrome Decoding [R-SD] problem . . . 6

3 Stern Scheme **6**
- 3.1 3-passe protocol . . . 6
- 3.2 5-passe improvement . . . 9

4 Veron Scheme **12**
- 4.1 Original protocol . . . 12
- 4.2 5-passe improvement . . . 14
- 4.3 Second Improvement . . . 16

5 Chen scheme **19**
- 5.1 Original protocol . . . 19
- 5.2 Chen attacks . . . 21

6 Signatures **22**
- 6.1 Fiat-Shamir heuristic . . . 22
- 6.2 Signature based on Stern Scheme . . . 23

1 Prerequisites

1.1 Global Notations

- e^T denotes the transpose of the vector e
- $x||y$ means the concatenation of x and y
- $< x >$ denotes the subspace generated by x

1.2 Error-correcting codes

Let n and k be two integer such that $k < n$. A $[n, k]$-linear error-correcting code is a k-dimensional subspace of a n-dimensional vector space over a finite field $\mathbb{F}_q$.

Hamming weight : The Hamming weight of a vector $x = (x_1, \cdots, x_m) \in GF(q)^m)$ is the number of non-zero coordinates of x. We denote by $w_H(x)$ the Hamming weight of x.

$$w_H(x) = \#\{x_i \text{ such that } x_i \neq 0\}$$

Hamming distance : The Hamming distance between two vectors is the weight of the difference of the two vectors. We denote by $d_H(x, y)$ the Hamming distance between x and y.

$$d_H(x, y) = w_H(x - y)$$

Minimal distance : The minimal distance d of a code C is the minimal distance between two distinct points of this code.

$$d = min\{d_H(x, y) \text{ such that } x \neq y \in C\}$$

Error correcting capability : The error correcting capability t of a code is the maximum errors that the code can decode. usually, $t = \lfloor \frac{d-1}{2} \rfloor$

Generator matrix Let C be a $[n, k, t]$-linear code over $\mathbb{F}_q$. A generator matrix G of C is a $(k \times n)$ matrix such that the rows of G form a basis of C :

$$C = \{xG : x \in \mathbb{F}_q^k\}$$

parity-check matrix : A parity-check matrix H of C is a $((n - k) \times n)$ matrix such that the rows of H form a basis of the orthogonal subspace of C.

$$C^T = \{xH : x \in \mathbb{F}_q^k\}$$

Furthermore we have $G \times H = 0_{k\times(n-k)}$ which means :

$$\forall x \in C : Hx^T = 0$$

Hamming bound : We denote by $A_q(n,d)$ the maximum number of words in a $[n,d]$-code C over $\mathbb{F}_q$. Then we have :

$$A_q(n,d) \leqslant \frac{q^n}{\sum_{i=0}^{t} \binom{n}{i}(q-1)^i}$$

Gilbert Varshamov (GV) bound : With the same notations :

$$\frac{q^n}{\sum_{i=0}^{d-1} \binom{n}{i}(q-1)^i} \leqslant A_q(n,d)$$

1.3 Rank metric

Start with q a power of a prime number p and m an integer. We look at the finite field $GF(q^m)$. Let $\beta = (\beta_1, \cdots, \beta_m)$ be a basis of $GF(q^m)$ over $GF(q)$. Let V_n be a vector space of dimension n over $GF(q^m)$ $(n \leqslant m)$. For every $v = (v_1, \cdots, v_n) \in V_n$ we associate the matrix $\Phi(v) = (v_{i,j}) \in M_{n\times m}$ where $v_{i,j}$ is the j^{th} coordinate of v_i in the basis β. We denote by Φ^{-1} the inverse map. We define the rank of v as the rank of the matrix V. We note this value $rank(v)$

Rank distance : Let x and y two vectors of V_n. We can define the distance between x and y on the rank metric by : $d_R(x,y) = rank(x-y)$.

Proof. Let x, y and z three vectors of $GF(q^m)$

1. Splitting : Let's show that $d_R(x,y) = 0 \iff x = y$
   Let's suppose $d_R(x,y) = 0$, which is equivalent to $rank(x-y) = 0 \iff \Phi(x-y) = 0_{n\times m}$ then without loss of generality, we get $x - y = 0_n \iff x = y$.

2. Symmetry : Let's show that $d_R(x,y) = d_R(y,x)$. It's clear that we have $\Phi(x-y) = -\Phi(x-y)$ and the rank of these two matrices is the same.

3. Triangle inequality : Let's show that $d_R(x,z) \leqslant d_R(x,y) + d_R(y,z)$
   We have :

   $$\begin{aligned} & d_R(x,y) + d_R(y,z) \\ = \; & \dim < x-y > + \dim < y-z > \\ = \; & \dim(< x-y > \cup < y-z >) + \dim(< x-y > \cap < y-z >) \end{aligned}$$

   On the other hand, we also know that $< x-z > \subset (< x-y > \cup < y-z >)$, which means that $\dim < x-z > \leq \dim(< x-y > \cup < y-z >)$. Hence, we can conclude that:

   $$d_R(x,z) \leq d_R(x,y) + d_R(y,z).$$

□

We showed that d_R is indeed a distance over $GF(q^m)$, then the rank metric is well defined.

By extension, we note $w_R(x)$ the weight of x in the rank metric corresponding to the dimension of the vector space generated by x over $GF(q^m)$.

1.4 Double Circulant Codes

Let $n = 2k$ for any integer k. Consider a vector $x = (x_1, x_2) \in GF(q)^n$ as a pair of two vector $x_1, x_2 \in GF(q)^k$. An $[n, k]$ linear code C is double circulant if $\forall c = (c_1, c_2) \in C$ then $c' = (rot_1(c_1), rot_1(c_2)) \in C$. More formally, by considering each block c_1, c_2 as a polynom in $GF(q)[X]/(X^n - 1)$, the code C is double circulant if $\forall c = (c_1, c_2) \in C$ then $(X \cdot c_1, X \cdot c_2) \in C$. Transgressively, $\forall x = (x_1, x_2) \in GF(q)^n$ we denotes by $drot_r(x) := (rot_r(x_1), rot_r(x_2))$.

A systematic double circulant $[n, k]$ code is a double circulant code with a parity-check matrix of the form $H = [I_k|A]$ where I_k is the identity matrix of size k and A is a $k \times k$ circulant matrix

2 NP-hard problems based on error correcting codes

2.1 Syndrome Decoding [SD] problem

Let H be a parity-check matrix of a binary [n,k] code C, s a syndrome, and w an integer. Can we recover a vector e with a length of n such that $He^T = s$ and $w_H(e) = w$?

This problem has been proved NP-complete by Berlekamp, McEliece and Tilborg in 1978 [3]. We are basing our proof on a more recent paper by Matthieu Finiasz published in 2009 [6]. The proof is based on a reduction to a well-known NP-complete problem named the Three dimensional matching.

Three Dimensional Matching problem : Let's take a subset

$$U \subseteq T \times T \times T$$

where $T = (t_1, \cdots, t_n)$ is a finite set. The problem is to find another subset $W \subseteq U$ such that $\forall ((w_1, w_2, w_3), (w'_1, w'_2, w'_3)) \in W^2, w_1 \neq w'_1, w_2 \neq w'_2, w_3 \neq w'_3$. We will show that if we are able to solve the [SD] problem, we are also able to solve the three dimensional matching problem.

Proof. Let's suppose that an attacker is able to solve the [SD] problem. For all subsets U defined as above, we denote by $U_i = ((U_i)_1, (U_i)_2, (U_i)_3)$ the elements of U. We also define the matrix $M \in M_{|U| \times 3|T|}$ by :

$$M = (m_{i,j}) \text{ where } m_{i,j} = 1 \text{ if } (U_i)_{j \bmod |T|} = t_{j \bmod |T|}$$

If we use this matrix as a parity-check matrix, take $s = (1, \cdots, 1) \in GF(2)^{|U|}$ and $w = |T|$, then by solving the [SD] problem on these parameters, we also give a solution for the three dimensional matching. Indeed we have the solution x we found such that $s = Hx^T$ and $w_H(x) = w = |T|$ can't have two elements agree in one of the $|T|$ coordinates otherwise one coordinate of s should be more than 1. □

2.2 Generator Matrix Syndrome Decoding [G-SD] problem

Let G be a generator matrix of a binary [n,k] code C, x a vector with a length of n and w a small integer. Can we recover a vector e with a length of n such that $x + e \in C$ and $w_H(e) = w$?

The same problem in term of generator matrix is also NP-complete. We can show that if we can resolve the [G-SD] problem, we are able to solve the [SD] problem, which shows that [G-SD] is also NP-complete

Proof. Let's assume that we can solve the [G-SD] problem in a polynomial time and that we have H a parity-check matrix of a binary $[n, k]$ code C, s a syndrome and w an integer. Then by systematic form, we are able to find G a generator matrix of the code C. Moreover, it's easy to find $x\prime$ such that $Hx\prime^T = s$ (we don't consider the Hamming weight of $x\prime$). Now we are able to find $e\prime$ with [G-SD] such that $x\prime + e\prime \in C$ and $w_H(e\prime) = w$. Now by multiplying each side by H : $H(x\prime + e\prime)^T = 0 \iff Hx\prime^T + He\prime^T = 0 \iff s = Hx\prime^T = He\prime^T$ and $w_H(e\prime) = w$, $e\prime$ a solution to the [SD] problem. □

2.3 q-ary Syndrome Decoding [q-SD] problem

This problem is just a generalisation of the [SD] problem over an arbitrary finite field. Let's take a matrix $H \in M_{k,n}(GF(q^m))$ and s a vector of $GF(q^m)^k$ and an integer w. Can we recover an element $x \in GF(q^m)^n$ such that $Hx^T = s$ and $w_H(x) = w$?
According to [4], this problem stays NP-complete.

2.4 Rank Syndrome Decoding [R-SD] problem

Let H be a parity-check matrix of a C code over $GF(q^m)$, s a syndrome and w an integer. Can we recover an element x of $GF(q^m)^n$ such that $Hx^T = y^T$ and $w_R(x) = w$?

3 Stern Scheme

In this section, I will present the Stern scheme, proposed by Stern in 1993 [8] as well as some improvements.

3.1 3-passe protocol

This scheme is a 3 passes scheme with a probability of cheating of $\frac{2}{3}$ and stands on the SD problem. We will show that if an attacker can cheat more than 2 times out of 3, he is able to solve the SD problem.

In this scheme, we introduce the notion of permutation of a vector. Let $x = (x_1, \cdots, x_n)$ be a vector with a length of n, we define $\sigma(x) := (x_{\sigma(1)}, \cdots, x_{\sigma(n)}$

where σ is a permutation of $1, \cdots, n$. I will write σ_i is there are several distinct permutations.

Let C be a binary $[n, k]$ code, H a parity-check matrix of c and h a hash function.

Private key : x a word of C with a length of w

Public key : $s = Hx^T$ the syndrome of x and w

3-passe protocol

Prover $\mathsf{P}(x, w, H)$		Verifier $\mathsf{V}(s = Hx^T, H, w)$
$\sigma \leftarrow_\$ S_n$, $y \leftarrow_\$ GF(q)^n$		
$c_1 = h(\sigma \| Hy^T)$		
$c_2 = h(\sigma(y))$		
$c_3 = h(\sigma(x + y))$	$\xrightarrow{c_1, c_2, c_3}$	$b \leftarrow_\$ \{0, 1, 2\}$
	$\xleftarrow{b}$	
If $b = 0$	$\xrightarrow{\sigma, y}$	**Check** $h(\sigma \| Hy^T) = c_1, h(\sigma(y)) = c_2$
If $b = 1$	$\xrightarrow{\sigma, y + x}$	**Check** $h(\sigma \| H(y + x)^T - s) = c_1$
		$h(\sigma(x + y)) = c_3$
If $b = 2$	$\xrightarrow{\sigma(x), \sigma(y)}$	**Check** $h(\sigma(y)) = c_2, h(\sigma(x + y)) = c_3$
		$w_H(\sigma(x)) = w$

Figure 1: The identification scheme proposed by J. Stern

Completeness : For the case $b = 1$, in the commitment c_1, we have :

$$H(x + y)^T - s = Hx^T + Hy^T - s = Hy^T$$

which is exactly the expected value. Now it's clear that if the prover follows the scheme, the verifier can't refuse the authentication.

Zero-knowledge : The first thing we can see is that the hamming weight of x is a public value. I first guessed it was a really important information about x, But with the parameters $n = 1024$ and $w = 50$ which is a really small weight, there are already more than 2^{85} possibilities for x, which is more than the capability of a brute-force attack. Therefore, the knowledge of this information is not an advantage for any attacker.

Now we have to show that any other element exchanged during the scheme behaves like random values.

Proof. Let's assume that a dishonest verifier has 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's build a polynomial-time probabilistic Turing machine M which produces a random communication indistinguishable from a communication coming from a fair authentication process. We denote by Ans the answer sent by M after the challenge.

1. M randomly picks a query $b \in \{0, 1, 2\}$
   - If $b = 0$, M chooses $\sigma \leftarrow_{\$} S_n$ and $y \leftarrow_{\$} GF(q)^n$ and computes $c_1 = h(\sigma||Hy^T)$ and $c_2 = h(\sigma(y))$. It substitutes c_3 by a random value. $Ans = (\sigma, y)$, it's clear that all elements have the same probability distribution as in the fair authentication.
   - If $b = 1$, M chooses $\sigma \leftarrow_{\$} S_n$ and $z \leftarrow_{\$} GF(q)^n$ and computes $c_1 = h(\sigma||Hz^T)$ and $c_2 = h(\sigma(z))$. It substitutes c_2 by a random value. $Ans = (\sigma, z)$. Let r be any element of $GF(q)^n$. Since y is a random element of $GF(q)^n$, we have :
     $$\mathbb{P}(x + y = r) = \frac{1}{q^n} = \mathbb{P}(z = r)$$
     then it's clear that all elements have the same probability distribution as in the fair authentication.
   - If $b = 2$, M chooses $\sigma \leftarrow_{\$} S_n$ and $z, x' \leftarrow_{\$} GF(q)^n$ such that $w_H(x') = w$. M computes $c_1 = h(\sigma||Hz^T)$ and $c_2 = h(\sigma(z))$. It substitutes c_2 by a random value. $Ans = (\sigma(z), \sigma(x'))$, it's clear that all elements have the same probability distribution as in the fair authentication.
2. M computes $b' = St(c_1, c_2, c_3)$
3. If $b' = b$, then M saves the quantities (c_1, c_2, c_3, b, Ans) otherwise M goes back to step 1.

Finally, in $3r$ executions on average, M produces a communication indistinguishable from a fair authentication protocol with r rounds. □

Soundness : We said that the probability of cheating is $\frac{2}{3}$, I will prove that if an attacker can cheat more than 2 times out of 3 then he can solve the [q-SD] problem.

Proof. Let suppose that an attacker can give for each value of the challenge b a good answer, then this attacker is able to give $Y_1, \sigma_1, Y_2, \sigma_2, Y_3$ and Y_4 such that :

$$\begin{cases} h(\sigma_1||HY_1^T) &= c_1 \\ h(\sigma_1(Y_1)) &= c_2 \\ h(\sigma_2||HY_2^T + s) &= c_1 \\ h(\sigma_2(Y_2)) &= c_3 \\ h(Y_3) &= c_2 \\ h(Y_3 + Y_4) &= c_3 \\ w_H(Y_4) &= w \end{cases}$$

which means :

$$\iff \begin{cases} h(\sigma_1||HY_1^T) &= h(\sigma_2||HY_2^T + s) \\ h(\sigma_1(Y_1)) &= h(Y_3) \\ h(\sigma_2(Y_2)) &= h(Y_3 + Y_4) \\ w_H(Y_4) &= w \end{cases}$$

Either the attacker is able to find a collision on a hash function, or we have :

$$\iff \begin{cases} \sigma_1||HY_1^T &= \sigma_2||HY_2^T + s \\ \sigma_1(Y_1) &= Y_3 \\ \sigma_2(Y_2) &= Y_3 + Y_4 \\ w_H(Y_4) &= w \end{cases} \iff \begin{cases} \sigma_1 &= \sigma_2 \\ HY_1^T &= HY_2^T + s \\ \sigma_1(Y_1) &= Y_3 \\ \sigma_2(Y_2) &= Y_3 + Y_4 \\ w_H(Y_4) &= w \end{cases}$$

It's clear we have the equivalence between these two systems. I continue by noting $\sigma := \sigma_1 = \sigma_2$

$$\iff \begin{cases} HY_1^T &= HY_2^T + s \\ \sigma(Y_1) &= Y_3 \\ \sigma(Y_2) &= Y_3 + Y_4 \\ w_H(Y_4) &= w \end{cases}$$

Finally, by adding the second and third lines and transforming the first one :

$$\iff \begin{cases} H(Y_1 + Y_2)^T &= s \\ \sigma(Y_1 + Y_2) &= Y_4 \\ w_H(Y_4) &= w \end{cases} \iff \begin{cases} H(Y_1 + Y_2)^T &= s \\ \\ w_H(Y_1 + Y_2) &= w \end{cases}$$

So the attacker is able to find a solution for the [q-SD] problem. We showed that if an attacker is able to cheat more than 2 times out of 3 then he can either solve a NP-complete problem or find a collision on a hash function. □

3.2 5-passe improvement

P.L. Cayrel1, P. Véron2, and S.M. El Yousfi Alaoui1 [4] presented in 2011 an improvement of this scheme that permits to reduce the cheating probability from $\frac{2}{3}$ to $\frac{q}{2(q-1)}$, which can be as close as we want from $\frac{1}{2}$ because q is a parameter of the protocol. by using 5 passes instead of 3. The protocol is the following :

Let σ be a permutation of $\{1, \cdots, n\}$ and $\gamma \in GF(q)^n$ such that $\forall i, \gamma_i \neq 0$. We define the transformation $\Sigma_{\gamma,\sigma}$ as :

$$\begin{array}{rccc} \Sigma_{\gamma,\sigma} & : & GF(q)^n & \to & GF(q)^n \\ & & v & \to & (\gamma_{\sigma(1)} v_{\sigma(1)}, \cdots, \gamma_{\sigma(n)} v_{\sigma(n)}) \end{array}$$

We have immediately two properties :

- $\forall \alpha \in GF(q), \forall v \in GF(q)^n, \Sigma_{\gamma,\sigma}(\alpha v) = \alpha \Sigma_{\gamma,\sigma}(v)$
- $\forall v \in GF(q)^n, w_H(\Sigma_{\gamma,\sigma}(v)) = w_H(v)$

Let C be a $[n,k]$ code over $GF(q)$ with q a power of a prime number, H a parity-check matrix of c and h a hash function.
Secret key : x a word of C of weight w
Public key : $s = Hx^T$ the syndrome of x and w

5-passe improvement

Prover $\mathsf{P}(x,w)$		Verifier $\mathsf{V}(s = Hx^T)$
$\sigma \leftarrow_\$ S_n,\ \gamma, u \leftarrow_\$ GF(q)^n$		
$c_1 = h(\sigma \| \gamma \| Hu^T)$		
$c_2 = h(\Sigma_{\gamma,\sigma}(u) \| \Sigma_{\gamma,\sigma}(x))$	$\xrightarrow{c_1, c_2}$	$\alpha \leftarrow_\$ GF(q)$
$\beta = \Sigma_{\gamma,\sigma}(u + \alpha x)$	$\xleftarrow{\alpha}$	
	$\xrightarrow{\beta}$	$b \leftarrow_\$ \{0,1\}$
	$\xleftarrow{b}$	
If $b = 0$	$\xrightarrow{\sigma, \gamma}$	**Check** $c_1 = h(\sigma \| \gamma \| H(\Sigma_{\gamma,\sigma}^{-1}(\beta)^T) - \alpha s)$
If $b = 1$	$\xrightarrow{\sigma(x)}$	**Check** $c_2 = h(\beta - \alpha \Sigma_{\gamma,\sigma}(x) \| \Sigma_{\gamma,\sigma}(x))$
		$w_H(\Sigma_{\gamma,\sigma}(x)) = w$

Figure 2: The Stern 5-passe protocol

Completeness : For the first commitment, we have

$$H(\Sigma_{\gamma,\sigma}^{-1}(\beta)^T) - \alpha s = H(\Sigma_{\gamma,\sigma}^{-1}(\Sigma_{\gamma,\sigma})^T) - \alpha s = Hu^T + \alpha Hx^T - \alpha Hx^T = Hu^T$$

which is exactly the expected value, and for the second commitment :

$$\beta - \alpha \Sigma_{\gamma,\sigma}(x) = \Sigma_{\gamma,\sigma}(u + \alpha x) - \alpha \Sigma_{\gamma,\sigma}(x) = \Sigma_{\gamma,\sigma}(u)$$

which is the expected value as well. After clarification of these two points, it's clear that if the prover follows the scheme and knows the secret key x, the verifier can't refuse the authentication.

Zero-knowledge : We use the same idea of building a polynomial-time probabilistic Turing machine M which create random communication indistinguishable from a fair authentication protocol.

Proof. Since the protocol needs two challenges, let's assume that the dishonest verfier have two peculiar strategies depends on the value of the commitments. $St_1(c_1, c_2)$ generate a value $\alpha \in GF(q)$ and $St_2(c_1, c_2, \beta) \in \{0, 1\}$. The machine M is constructed as follow :

1. M picks a query $b \in \{0, 1\}$
   - If $b = 0$, M chooses randomly $u, \gamma \leftarrow_{\$} GF(q)^n$ and $\sigma \in S_n$ and solve the equation $Hx'^T = y$ for some x' not necessarily satisfying the condition $w_H(x') = w$. M computes the commitment $c_1 = h(\sigma||\gamma||Hu^T)$ and substitutes c_2 by a random value. M applies $St_1(c_1, c_2)$ to get α and then computes $\beta = \Sigma_{\gamma,\sigma}(u + \alpha x')$. M have now all the value needed. then the communication set is (c_1, c_2) and $Ans = (\beta, \gamma, \sigma)$. Since elements are chosen randomly by the same method, it's clear that their follow the same probability distribution.
   - If $b = 1$, M chooses randomly $\sigma \in S_n$ and $u, \gamma, x' \leftarrow_{\$} GF(q)^n$ such that $w_H(x') = w$. M computes $c_2 = h(\Sigma_{\gamma,\sigma}(u), \Sigma_{\gamma,\sigma}(x'))$ and substitutes c_1 by a random value. From $\alpha = St_1(c_1, c_2)$ M computes $\beta = \Sigma_{\gamma,\sigma}(u + \alpha x')$. Then the communication set is (c_1, c_2) and $Ans = (\Sigma_{\gamma,\sigma(x')})$. Again, it's clear that any value exchanged follow the same probability distribution as in a fair authentication protocol.
2. M computes $b' = St_2(c_1, c_2, \beta)$
3. If $b' = b$, then M saves the quantities $(c_1, c_2, \beta, b, Ans)$ otherwise M goes back to step 1

So in $2r$ execution on average, M produces a communication indistinguishable from a fair authentication protocol with r rounds. □

Soundness : We said that the cheating probability is equal to $\frac{q}{2(q-1)}$. We will show that if an attacker is able to cheat more than one time out of two, then he is able to find a collision in a hash function or to solve the [SD] problem which is a NP-complete problem.

Proof. Let's suppose an attacker is able to cheat more than q times out of $2(q-1)$, then, by the pigeonhole principle, for at least two values of the first challenge (that we will note α_1 and α_2), the attacker is able to cheat regardless of the value of the second challenge ($b \in \{0, 1\}$). Then for the challenges $(\alpha_1, 0)$ resp.$((\alpha_1, 1),(\alpha_2, 0),(\alpha_2, 1))$ the attacker is able to give $(\beta_1, \sigma_1, \gamma_1)$ resp.$((\beta_1, z_1),(\beta_2, \sigma_2, \gamma_2),(\beta_2, z_2))$ such that :

$$\begin{cases} h\left(\sigma_1||\gamma_1||H(\Sigma^{-1}_{\gamma,\sigma_1}(\beta_1)^T) - \alpha_1 s\right) &= c_1 \\ h(\beta_1 - \alpha_1 z_1||z_1) &= c_2 \\ w_H(z_1) &= w \\ h\left(\sigma_2||\gamma_2||H(\Sigma^{-1}_{\gamma,\sigma_2}(\beta_2)^T) - \alpha_2 s\right) &= c_1 \\ h(\beta_2 - \alpha_2 z_2||z_2) &= c_2 \\ w_H(z_2) &= w \end{cases}$$

Which means :

$$\iff \begin{cases} h\left(\sigma_1||\gamma_1||H(\Sigma^{-1}_{\gamma,\sigma_1}(\beta_1)^T) - \alpha_1 s\right) &= h\left(\sigma_2||\gamma_2||H(\Sigma^{-1}_{\gamma,\sigma_2}(\beta_2)^T) - \alpha_2 s\right) \\ h(\beta_1 - \alpha_1 z_1||z_1) &= h(\beta_2 - \alpha_2 z_2||z_2) \\ w_H(z_1) &= w \\ w_H(z_2) &= w \end{cases}$$

Here we have two possibilities : Either the attacker is able to find a collision on the hash function (what we consider as a difficult problem), or we automatically have :

$$\iff \begin{cases} \sigma_1 &= \sigma_2 \\ \gamma_1 &= \gamma_2 \\ H(\sigma_1^{-1}(\beta_1)^T) - \alpha_1 s &= H(\sigma_2^{-1}(\beta_2)^T) - \alpha_2 s \\ \beta_1 - \alpha_1 z_1 &= \beta_2 - \alpha_2 z_2 \\ z_1 &= z_2 \\ w_H(z_1) &= w \\ w_H(z_2) &= w \end{cases}$$

I continue by noting $\sigma := \sigma_1 = \sigma_2$ and $\gamma := \gamma_1 = \gamma_2$ and $z := z_1 = z_2$

$$\iff \begin{cases} H(\Sigma^{-1}_{\gamma,\sigma}(\beta_1)^T) - \alpha_1 s &= H(\Sigma^{-1}_{\gamma,\sigma}(\beta_2)^T) - \alpha_2 s \\ \beta_1 - \alpha_1 z &= \beta_2 - \alpha_2 z \\ w_H(z) &= w \end{cases}$$

Then by rewriting the equations we have :

$$\iff \begin{cases} H\left(\Sigma^{-1}_{\gamma,\sigma}(\beta_1 - \beta_2)^T(\alpha_1 - \alpha_2)^{-1}\right) &= s \\ (\beta_1 - \beta_2)^T(\alpha_1 - \alpha_2)^{-1} &= z \\ w_H(z) &= w \end{cases} \iff \begin{cases} H\Sigma^{-1}_{\gamma,\sigma}(z) &= s \\ w_H(z) &= w \end{cases}$$

So the attacker is able to find a solution for the [q-SD] problem. Finally, we showed that the cheating probability can't be more than $\frac{q}{2(q-1)}$ □

4 Veron Scheme

4.1 Original protocol

The Veron scheme is an improvement of the Stern Scheme, It was proposed in 1997 by Veron [9]. The scheme is based on a different formulation of the secret. Instead of using a parity-check matrix of a C code, Veron uses a generator matrix of a code. The cheating probability for this scheme is $\frac{2}{3}$. This protocol stands on the [G-SD] problem.
Let C be a binary $[n, k]$ code over $GF(2)$, G a generator matrix of C and h a hash function.

Private key : m a vector with a length of k and e a binary vector with a length of n

Veron scheme

Prover $\mathsf{P}(m,e)$		Verifier $\mathsf{V}(x,w)$
$\sigma \leftarrow_\$ S_n,\ u \leftarrow_\$ GF(2)^k$		
$c_1 = h(\sigma)$		
$c_2 = h(\sigma((u+m)G))$		
$c_3 = h(\sigma(uG+x))$	$\xrightarrow{c_1, c_2, c_3}$	$b \leftarrow_\$ \{0,1,2\}$
	$\xleftarrow{b}$	
If $b=0$	$\xrightarrow{\sigma,\ u+m}$	**Check** $h(\sigma) = c_1$
		$h(\sigma((u+m)G)) = c_2$
If $b=1$	$\xrightarrow{\sigma((u+m)G)), \sigma(e)}$	**Check** $h(\sigma((u+m)G)) = c_2$
		$h(\sigma((u+m)G) + \sigma(e)) = c_3$
		$w_H(\sigma(e)) = w$
If $b=2$	$\xrightarrow{\sigma,\ u}$	**Check** $h(\sigma) = c_1$
		$h(\sigma(uG+x)) = c_3$

Figure 3: The identification scheme proposed by P. Veron

Public key : $x = mG + e$ and $w = w_H(e)$

Completeness : The only point we have to clarify is the verification of the third commitment in the case $b = 1$:

$$\sigma((u+m)G) + \sigma(e) = \sigma\left(uG + mG + e\right) = \sigma\left(uG + x\right)$$

which is exactly the expected value.

Soundness : This proof is rightly the same as in the 3-passe Stern protocol, except that this scheme stands on the [G-SD] problem instead of the [SD] problem.

Zero-knowledge : We will show that any element exchanged during the scheme behaves like random values.

Proof. Let's assume that a dishonest verifier has 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's build a polynomial-time probabilistic Turing machine M which produces a random communication indistinguishable from a communication coming from a fair authentication process. We denote by Ans the answer sent by M after the challenge.

1. M randomly picks a query $b \in \{0, 1, 2\}$

- If $b = 0$, M chooses $\sigma \leftarrow_{\$} S_n$ and $y \leftarrow_{\$} GF(2)^k$ and computes $c_1 = h(\sigma)$ and $c_2 = h(\sigma(yG))$. It substitutes c_3 by a random value. $Ans = (\sigma, y)$, it's clear that all elements have the same probability distribution as in the fair authentication. Indeed, let r be a random element of $GF(2)^k$, since u is a random element of $GF(2)^k$ we have :

$$\mathbb{P}(u + m = r) = \frac{1}{2^k} = \mathbb{P}(y = r)$$

- If $b = 1$, M chooses $\sigma \leftarrow_{\$} S_n$ and y any element of the code C and $e' \in GF(2)^n$ such that $w_H(e') = w$ and computes $c_2 = h(\sigma(y))$ and $c_3 = h(\sigma(y + e'))$. It substitutes c_1 by a random value. $Ans = (\sigma(y), \sigma(e'))$. It's clear that e has the same probability distribution than e', furthermore, let r be any element of $GF(q)^n$ then we have :

$$\mathbb{P}((u + m)G = r) = \frac{1}{2^{n-k}} = \mathbb{P}(y = r)$$

  then it's clear that all elements have the same probability distribution as in the fair authentication.

- If $b = 2$, M chooses $\sigma \leftarrow_{\$} S_n$ and $y \leftarrow_{\$} GF(q)^k$. M computes $c_1 = h(\sigma)$ and $c_3 = h(\sigma(yG + x))$. It substitutes c_2 by a random value. $Ans = (\sigma, y)$, it's clear that all elements have the same probability distribution as in the fair authentication.

2. M computes $b' = St(c_1, c_2, c_3)$
3. If $b' = b$, then M saves the quantities (c_1, c_2, c_3, b, Ans) otherwise M goes back to step 1.

Finally, in $3r$ executions on average, M produces a communication indistinguishable from a fair authentication protocol with r rounds. □

4.2 5-passe improvement

In 2011, C. Aguilar and P. Gaborit and J. Schrek [1] proposed a new zero-knowledge authentication scheme based on the Veron protocol with 5-passe permit to reduce the cheating probability from $\frac{2}{3}$ to an asymptotic cheating probability of $\frac{1}{2}$. Furthermore, they also used double circulant code to reduce the size of the keys.
Let consider C a $[n, k]$ double circulant code over $GF(2)$, G a generator matrix of C of size $k \times n$ and h a hash function.

Private key : $m \in GF(2)^k$ and $e \in GF(2)^n$ such that $w_H(e) = w$
Public key : $x = mG + e$ and $w_H(e) = w$

5-passe Veron scheme

Prover $\mathsf{P}(m, e)$		Verifier $\mathsf{V}(x, w)$
$\sigma \leftarrow_\$ S_n$, $u \leftarrow_\$ GF(2)^k$		
$c_1 = h(\sigma)$		
$c_2 = h(\sigma(uG))$	$\xrightarrow{c_1, c_2}$	$r \leftarrow_\$ \{0, \cdots, k-1\}$
$m_r = Rot_r(m)$	$\xleftarrow{r}$	
$e_r = Rot_r(e)$		
$c_3 = h(\sigma(uG + e_r))$	$\xrightarrow{c_3}$	$b \leftarrow_\$ \{0, 1\}$
	$\xleftarrow{b}$	
If $b = 0$	$\xrightarrow{\sigma, u + m_r}$	**Check** $h(\sigma) = c_1$
		$h(\sigma((u + m_r)G + x_r)) = c_3$
If $b = 1$	$\xrightarrow{\sigma(uG), \sigma(e_r)}$	**Check** $h(\sigma(uG)) = c_2$
		$h(\sigma(uG) + \sigma(e_r)) = c_3$
		$w_H(\sigma(e_r)) = w$

Figure 4: Improvement of the Veron Scheme

Completeness : The only point we have to clarify is the verification of the third commitment in the case $b = 0$, we have :

$$\sigma\left((u + m_r)G + x_r\right) = \sigma(uG + m_rG + x_r) = \sigma(uG + e_r)$$

which is exactly the expected value. Now it's clear that if the prover and the verifier follow the protocol and if the prover knows the secret x, the verifier cant refuse the authentication.

Zero-knowledge :

Proof. Let suppose that a dishonest verifier have a peculiar strategy with regards to the commitments sent by the prover. Let $St_1(c_1, c_2)$ be such a strategy for the first challenge such that $St_1(c_1, c_2) \in \{0, \cdots, k-1\}$ and $St_2(c_1, c_2, c_3) \in \{0, 1\}$ be a strategy for the second answer. Let's construct a polynomial-time probabilistic Turing machine M which produces a random communication indistinguishable from a fair communication coming from an authentication process. We denotes by Ans the answer sent by M after the challenges. M is constructed as follow :

1. M randomly picks a query $b \in \{0, 1\}$

- If $b = 0$, M chooses $\sigma \leftarrow_{\$} S_n$ and $v \leftarrow_{\$} GF(2)^k$. M computes $c_1 = h(\sigma)$ and substitutes c_2 by a random value. M applies $St_1(c_1, c_2)$ to get r and computes $c_3 = h(\sigma(vG + x_r)$. Then the communication set is (c_1, c_2, C_3) and $Ans = (\sigma, v)$. It's clear that all elements have the same probability distribution as in the fair authentication process.
- If $b = 1$, M chooses and $z \leftarrow_{\$} GF(2)^n$, $v \leftarrow_{\$} GF(2)^k$ and $\sigma \leftarrow_{\$} S_n$ such that $w_H(z) = w$. M computes $c_2 = h(\sigma(uG))$. M applies $St_1(c_1, c_2)$ to get r and computes $c_3 = h(z)$. Then the communication set is (c_1, c_2, C_3) and $Ans = (\sigma(uG), z)$. It's clear that all elements have the same probability distribution as in the fair authentication process.

2. M computes $b' = St_2(c_1, c_2, c_3)$
3. If $b' = b$, then M saves the values $(c_1, c_2, c_3, r, b, Ans)$, otherwise M goes back to step 1.

Finally, in $2r$ executions on average, M produces a communication indistinguishable from a fair authentication protocol with r rounds. □

Soundness : We will prove that if an attacker is able to cheat much more than one time out of two, then he is also able to recover the secret key x. Under the zero-knowledge hypothesis, recover x is equivalent to solve the [G-SD] problem which is a NP-hard problem.

Proof. Let suppose that an attacker is able to answer $k + i$ challenges out of the $2k$. Then, by the pigeonhole principle, for at least i value of the first challenge (that we will note $\{r_1, \cdots, r_i\}$, the attacker is able to cheat regardless of the value of the second challenge ($b \in \{0, 1\}$). By rewriting the third commitment, we show that this attacker is able to construct $(c, z_1, \cdots, z_i)$ such that :

$$s_{r_j} = c + H \times z_j^T$$

By well choosing parameters, we are able to have, with a good probability, that one of the sifted value of the syndrome of x is the real secret we want to protect. Then If the attacker can cheat more than the predicted value, we show that he can retrieve the secret key. □

4.3 Second Improvement

This second improvement has been presented in 2019 by Emanuele Bellini, Florian Caullery, Philippe Gaborit, Marc Manzano and Víctor Mateu [2]. This improvement is based on the rank metric and on the use of double circulant codes. The cheating probability of this scheme is as near as we want from $\frac{1}{2}$. To present this scheme, we first need to present two new operations : Let $P \in M_{n,n}(GF(q))$, $Q \in M_{m,m}(GF(q))$ and $v \in GF(q^m)^n$ we define the function $\Pi_{P,Q}(v) = \Phi^{-1}(Q \times \Phi(v) \times P)$ which means :

$$\begin{array}{rccc} \Pi_{P,Q} : & GF(q^m)^n & \rightarrow & GF(q^m)n \\ & (v_1, \cdots, v_n) & \rightarrow & (\pi_1, \cdots, \pi_n) \end{array}$$

where for each $i \in \{1, \cdots, n\}$

$$\pi_i = \sum_{k=1}^{m} \beta_k \sum_{i=1}^{m} \sum_{j=1}^{n} Q_{1,i} v_{i,j} P_{j,h}$$

We can show three properties about this function :
$\forall x, y \in GF(q^m)^n, P \in M_{n,n}(GF(q)), Q \in M_{m,m}(GF(q)), a, b \in GF(q)$

1. (rank preservation) $w_R(x) = w_R\left(\Pi_{P,Q}(x)\right)$
2. (linearity) $a\Pi_{P,Q}(x) + b\Pi_{P,Q}(y) = \Pi_{P,Q}(ax + by)$
3. (reversing) If $w_R(x) = w_R(y)$, it's possible to find $P\prime \in M_{n,n}(GF(q)), Q\prime \in M_{m,m}(GF(q))$ such that $x = \Pi_{P\prime,Q\prime}(y)$

These three properties will be useful for the following protocol.
Let $n = 2k$, we already introduce the rot_i and $drot_i$ notations in section 1.4. For any $x \in GF(q^m)^k$ and for any $y \in GF(q^m)^n$, given $\alpha = (\alpha_1, \cdots, \alpha_k) \in GF(q)^k$, we define by $\Gamma\prime_\alpha(x)$ the linear combination of all possible k rotations of $k - i$ positions of x and by $\Gamma_\alpha(y)$ the linear combination of all possible k double rotations of i positions of y :

$$\Gamma\prime_\alpha(x) = \sum_{i=1}^{k} \alpha_i \times rot_{k-i}(x) \in GF(q^m)^k$$

$$\Gamma_\alpha(y) = \sum_{i=1}^{k} \alpha_i \times drot_i(y) \in GF(q^m)^n$$

For any generator matrix G of a double circulant code and $x \in GF(q^m)^k$, we have the following property : $\Gamma_\alpha(x \times G) = \Gamma\prime_\alpha(x) \times G$. This property will be useful for the following as well.

Let G be a random vector of $GF(q^m)^n$. We extend G to $G\prime \in GF(q^m)^{k \times n}$ a double circulant matrix. $G\prime$ describes a code.

Private key : x a vector with a length of k and e a vector with a length of n. We choose e such that $w_R(e) = r$

Public key : $y = x \times G\prime + e$ and r.

Second improvement of Veron scheme

Prover $\mathsf{P}(x, e)$		Verifier $\mathsf{V}(y, r, G)$
$u \leftarrow_{\$} GF(q^m)^k$		
$P, Q \leftarrow_{\$} M_{m,m}(GF(q))$		
$c_1 = h(P, Q)$		
$c_2 = h(\Pi_{P,Q}(u \times G)$	$\xrightarrow{c_1, c_2}$	$\alpha = (\alpha_1, \cdots, \alpha_k) \leftarrow_{\$} GF(q)^k$
		α_i not all the same
$c_3 =$	$\xleftarrow{\alpha}$	
$h(\Pi_{P,Q}(u \times G + \Gamma_\alpha(e))$	$\xrightarrow{c_3}$	
	$\xleftarrow{b}$	$b \leftarrow_{\$} \{0, 1\}$
If $b = 0$		
$r_1 = (P, Q)$		
$r_2 = u + \Gamma\prime_\alpha(x)$	$\xrightarrow{r_1, r_2}$	**Check** $h(r_1) = c_1$
		$h(\Pi_{r_1}(r_2 \times G + \Gamma_\alpha(y))) = c_3$
If $b = 1$		
$r_1 = \Pi_{P,Q}(u \times G)$		
$r_2 = \Pi_{P,Q}(\Gamma_\alpha(e))$	$\xrightarrow{r_1, r_2}$	**Check** $h(r_1) = c_2$
		$h(r_1 + r_2) = c_3$
		$w_H(r_2) = r$

Figure 5: Second improvement of the Veron Scheme

Completeness : In the case $b = 0$, the verfication of c_3 is valid because :

$$u \times G + \Gamma_\alpha(e) = u \times G + \Gamma_\alpha(x \times G) + \Gamma_\alpha(y) = (u + \Gamma\prime_\alpha(x)) \times G + \Gamma_\alpha(y)$$

thank to the $\Gamma_\alpha(x \times G) = \Gamma\prime_\alpha(x) \times G$ property. Secondly, we have α_i all different, without that $w_R(\Gamma_\alpha(e)) = 0$ or 1 or 2 depending of α_i being equal to 0 or not, and the check in the case $b = 1$ would fail. After these clarifications, it's clear that if the prover respects the protocol and knows the secrets x and e, the verifier cannot refuse the authentication.

Zero-knowledge :

Proof. Let suppose that a dishonest verifier have a peculiar strategy with regards to the commitments sent by the prover. Let $St_1(c_1, c_2)$ be such a strategy for the first challenge such that $St_1(c_1, c_2) \in GF(q)^k$ with α_i not all the same and $St_2(c_1, c_2, c_3) \in \{0, 1\}$ be a strategy for the third commitment. Let's construct

a polynomial-time probabilistic Turing machine M which produces a random communication indistinguishable from a fair communication coming from an authentication process. We denotes by Ans the answer sent by M after the challenges. M is constructed as follow :

1. M randomly picks a query $b \in \{0, 1\}$
   - If $b = 0$, M chooses $P' \leftarrow_{\$} M_{n,n}(GF(q))$, $Q' \leftarrow_{\$} M_{m,m}(GF(q))$ and $v \leftarrow_{\$} GF(q^m)^n$. M computes $c_1 = h(P', Q')$ and substitutes c_2 by a random value. M applies $St_1(c_1, c_2)$ to get α and computes $c_3 = h(\Pi_{P',Q'}(v \times G + \Gamma_\alpha(x)))$. Then the communication set is (c_1, c_2, C_3) and $Ans = ((P', Q'), v)$. It's clear that all elements have the same probability distribution as in the fair authentication process.
   - If $b = 1$, M chooses and $P' \leftarrow_{\$} M_{n,n}(GF(q))$, $Q' \leftarrow_{\$} M_{m,m}(GF(q))$ and $v, z \leftarrow_{\$} GF(q^m)^n$ such that $w_R(z) = r$. M computes $c_2 = h(\Pi_{P',Q'}(v))$ and substitutes c_1 by a random value.
     M applies $St_1(c_1, c_2)$ to get α and computes $c_3 = h(\Pi_{P',Q'}(v) + z)$. Then the communication set is (c_1, c_2, C_3) and $Ans = (\Pi_{P',Q'}(v), z)$. It's clear that all elements have the same probability distribution as in the fair authentication process.
2. M computes $b' = St_2(c_1, c_2, c_3)$
3. If $b' = b$, then M saves the values $(c_1, c_2, c_3, r, b, Ans)$, otherwise M goes back to step 1.

Finally, in $2s$ executions on average, M produces a communication indistinguishable from a fair authentication protocol with s rounds. □

5 Chen scheme

5.1 Original protocol

The Chen protocol is a 5-passe Zero-knowledge authentication protocol based on rank metric proposed by Chen in 1995 [5]. The cheating probability for this protocol is $\frac{q}{2(q-1)}$ which can be as near as we want from $\frac{1}{2}$. It stands on the difficulty to solve the [R-SD] problem. This protocol has the interesting property of avoiding the hash functions, this feature is really interesting for low-cost cryptography, nevertheless this protocol has been cryptanalized in 2011 by P. Gaborit, J. Schrek and G. Zémor [7]. It would appear that the only way to prevent these attacks is to use hash function.

Let C be a binary $[n, k]$ code over $GF(q^m)$ with q a power of a prime number and H a parity-check matrix of c.

Secret key : x a word of C of low rank weight w

Public key : $s = Hx^T$ the syndrome of x

Chen Protocol

Prover $\mathsf{P}(x, w)$		Verifier $\mathsf{V}(s = Hx^T, w)$
$r \leftarrow_\$ V_n$, $P \leftarrow_\$ GL_n(GF(q^m))$		
$c_1 = HP^T r^T$, $c_2 = Hr^T$		
	$\xrightarrow{c_1, c_2}$	$\lambda \leftarrow_\$ GF(q^m)$
$y = r + \lambda x P^{-1}$	$\xleftarrow{\lambda}$	
	$\xrightarrow{y}$	$b \leftarrow_\$ \{0, 1\}$
	$\xleftarrow{b}$	
If $b = 0$	$\xrightarrow{P}$	**Check** $HP^T y^T = c_1 \lambda s$
If $b = 1$	$\xrightarrow{r}$	**Check** $c_2 = Hr^T$
		If $\lambda = 0$, check $w_R(w - r) = 0$
		If $\lambda \neq 0$, check $w_R(w - r) = w$

Figure 6: The identification scheme proposed by K. Chen

Zero-knowledge : Chen first proposed a proof of zero-knowledge in 1995. However, we will show that the zero-knowledge is not totally respected. The proof didn't take care about half of the cases during the protocol.

Completeness : In the case $b = 0$, we have :

$$HP^T y^T = HP^T (r + \lambda x P^{-1})^T = HP^T r^T + \lambda H x^T = c_1 + \lambda s$$

After this clarification, it's clear that if the prover respects the protocol and knows the secret x, the verifier cannot refuse the authentication.

Soundness : We will show that if an attacker is able to cheat to many times, then he is also able to solve an NP-hard problem.

Proof. Let's suppose that an attacker is able to cheat more than q time out of $2(q-1)$. Then there are at least two values of the first challenge (which we will note λ_1 and λ_2 such that the attacker is able to cheat regardless of the value of the second challenge $b \in \{0, 1\}$. Then for the challenges $(\lambda_1, 0)$, resp.$((\lambda_1, 1), (\lambda_2, 0), (\lambda_2, 1))$, the attacker is able to give (y_1, P)
resp.$((y_1, r), (y_2, P), (y_2, r))$ such that :

$$\left\{\begin{array}{rcl} HP^T y_1^T &=& c_1 + \lambda_1 s \\ Hr^T &=& c_2 \\ w_R(y_1 - r) &=& w \\ HP^T y_2^T &=& c_1 + \lambda_2 s \\ Hr^T &=& c_2 \\ w_R(y_2 - r) &=& w \end{array}\right. \iff \left\{\begin{array}{rcl} HP^T(y_1^T - y_2^T) &=& (\lambda_1 - \lambda_2)s \\ w_R(y_1 - r) &=& w \\ w_R(y_2 - r) &=& w \end{array}\right.$$

Then we note $x' = (\lambda_1 - \lambda_2)^{-1} P(y_1 - y_2)$. Thanks to the first line, it's clear that $Hx'^T = s$. We will show that $x = x'$. We have that $H(x - x')^T = s - s = 0$. Now look at $w_R(x - x')$ which is equal to $w_R(x - y_1 + y_2) = w_R(x - y_1 + y_2 + r - r)$ Thanks to the Belgian theorem. Now by triangle inequality : $w_R(x - x') \leqslant w_R(x) + w_R(y_1 - r) + w_R(y_2 - r) = 3w < d$. By hypothesis we have $x = x'$. Then the attacker is able to recover the secret key. So he is able to solve the [R-SD] problem, which conclude the proof. □

If the secret key is unknown, the cheating probability is $\frac{q}{2(q-1)}$.

5.2 Chen attacks

In this section, we will show that the Chen Protocol actually presents some leaks of data and then present how can we exploit the leakage of the protocol in order to recover the secret key.

Support attack : In the cases where $b = 1$, an attacker who just observe the exchanges between the prover and the verifier has, for each part of the authentication, some information : $(c_1, c_2, \lambda, y, b, r)$. Let's assume that the prover is a honest prover, then the attacker has an extra piece of information : $y = r + \lambda x P^{-1}$. So the attacker is able to find the value of xP^{-1} by computing $\lambda^{-1}(y - r)$. The problem is that xP^{-1} generates the same vector space E as x. So if the attacker wait long enough, he is able to re-build E of dimension w. He can construct a basis of E over $GF(q)$. We denote $\gamma = (\gamma_1, \cdots, \gamma_r)$ this basis. Furthermore, he constructs a basis of $GF(q^m)$ over $GF(q)$ such that the first r coordinates are γ. We call this basis γ'. We know that $x = (x_1, \cdots, x_r) \in E$ where $(x_1, \cdots, x_r)$ are the unknowns of the problem. We can express each x_i in the basis γ : $x_i = \sum_{k=1}^{r} a_{k,j}\gamma_k$ with $a_{k,j} \in GF(q)$. Secondly, we know the value of $s = Hx^T$ that we can express in the basis γ'. then we have $(n - k) \times m$ equations of $GF(q)$ and $n \times r$ unknowns. If $n \times r \leqslant (n - k) \times m$, then we are able to retrieve the secret x straight away.

Linear attack : In the cases which $b = 0$, an attacker who just observe one exchange between the prover and the verifier have some information : $(c_1, c_2, \lambda, y, b, P)$. Particularly, he have access to $H(xP^{-1})^T$ by using $y = r + \lambda x P^{-1}$ and computing $H(xP^{-1})^T = (Hy^T - c_2)\lambda^{-1}$. The knowledge of $s = Hx^T$ and $H(xP^{-1})^T$ gives enough linearly independent equations about the coordinates of x to recover the secret.

6 Signatures

6.1 Fiat-Shamir heuristic

The Fiat-Shamir heuristic is a method to create a digital signature from a Zero-knowledge proof. In their paper in 1986, Fiat and Shamir explain their idea with a scheme based on Discrete logarithm [DL] problem.

Let G be a finite group of order q, and g a generator of g. private key : α an element of $\mathbb{Z}_q^*$
public key : $h = g^\alpha$

The prover wants to prove to the verifier that he knows α

Scheme

Prover $\mathsf{P}(\alpha)$		Verifier $\mathsf{V}(h = g^\alpha)$
$r \leftarrow_\$ \mathbb{Z}_q^*$		
$u = g^r$	$\xrightarrow{u}$	$c \leftarrow_\$ \mathbb{Z}_q^*$
	$\xleftarrow{c}$	
$t = r - c\alpha$	$\xrightarrow{t}$	**Check** $t = uh^c$

Figure 7: Identification scheme proposed by Fiat and Shamir

This authentication protocol stands on the representation problem in a cyclic group :

Let's take g a generator and $h = g^\alpha$ a random element. In real life, g and h are linked, but under the [DL] hypothesis, they are two distinct elements. They can be considered as a basis of G. I mean, each element u of G can be written as $u := g^a h^b$ (where a and b aren't unique). But if we can find (a_1, b_1) and (a_2, b_2) such that $u = g^{a_1} h^{b_1} = g^{a_2} h^{b_2}$ then we have $g^{a_1 + b_1 \times \alpha} = g^{a_2 + b_2 \times \alpha}$ that means we are able to find α as $\alpha \equiv \frac{a_2 - a_1}{b_1 - b_2} \mod q$ contradicting the hypothesis.

So we have that the representation of each $u \in G$ is unique in the basis g, h. The protocol said that if you are able to solve it, you are also able to find a correct representation with random parameters, so you know the secret α.

Now we want to transform this Zero-knowledge authentication into a signature. To do this, we need to remove the interaction, signing being a process that is done alone. The idea of the Fiat-Shamir transformation is just to avoid the verifier's work, the prover create his challenge himself. But if he can choose it, he is able to select r and t satisfying the verification. So the challenge must be something behaves like a random value. To perform that, we use a hash function dependent on the first calculated value :

Signature Scheme based on Zero-knowledge authentication

Prover $\mathsf{P}(\alpha)$		Verifier $\mathsf{V}(h = g^{\alpha})$
$r \leftarrow_{\$} \mathbb{Z}_q^*$		
$u = g^r$	$\xrightarrow{\quad u \quad}$	
$c = h(m\|\|u)$		$c = h(m\|\|u)$
$t = r - c\alpha$	$\xrightarrow{\quad t \quad}$	**Check** $t = uh^c$

Figure 8: Identification scheme proposed by Fiat and Shamir

This scheme must be repeated enough times to ensure that the probability of forging a signature be as small as we want. The signature of the message m is $((u_1, t_1), \cdots, (u_K, t_K))$ where each couple (c_i, t_i) satisfies the $t_i = u_i hh(u||m)$. The security of the signature depends on the security of the authentication scheme. If the probability of cheating is $p \in [0, 1]$, and the security level S (for example, $S = 2^{80}$ we have to proceed K times the protocol where K is such that $p^K \leqslant \frac{1}{S}$.

The Fiat-Shamir transformation is an heuristic method, which means that for any Zero-knowledge authentication we can build, we are also able to find a signature scheme.

6.2 Signature based on Stern Scheme

Based on the 5-passe stern protocol, we are able to create a signature scheme :

A signature based on zero-knowledge authentication

Prover $\mathsf{P}(x, w)$		Verifier $\mathsf{V}(s = Hx^T)$
$\sigma \leftarrow_\$ S_n,\ \gamma, u \leftarrow_\$ GF(q)^n$		
$c_1 = h(\sigma \| \gamma \| Hu^T)$		
$c_2 = h(\sigma(u) \| \sigma(x))$	$\xrightarrow{c_1, c_2}$	
$\alpha = h(c_1 \| c_2) \mod q$		
$\beta = \sigma(u + \alpha x)$	$\xrightarrow{\beta}$	
$b = h(c_1 \| c_2 \| \beta) \mod 2$	$\xrightarrow{b}$	
If $b = 0$	$\xrightarrow{\sigma, \gamma}$	**Check** $c_1 = h(\sigma \| \gamma \| H(\sigma^{-1}(\beta)^T) - \alpha s)$
If $b = 1$	$\xrightarrow{\sigma(x)}$	**Check** $c_2 = h(\beta - \alpha\sigma(x) \| \sigma(x))$
		$w_H(\sigma(x)) = w$

Figure 9: Signature scheme based on the 5-passe Stern protocol

For this scheme, at each round, the signature is $(c_1, c_2, \beta, b, \Delta)$ where $\Delta = (\sigma, \gamma)$ or $\Delta = (\sigma(x))$ depends on the value of b.

References

[1] C. Aguilar, P. Gaborit, and J. Schrek. A new zero-knowledge code based identification scheme with reduced communication. *NC*, pages 648–652, Oct 2011.

[2] Emanuele Bellini, Florian Caullery, Philippe Gaborit, Marc Manzano, and Víctor Mateu. Improved veron identification and signature schemes in the rank metric. *CoRR*, abs/1903.10212, 2019.

[3] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems (corresp.). *IEEE Transactions on Information Theory*, 24(3):384–386, May 1978.

[4] Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero-knowledge identification scheme based on the q-ary syndrome decoding problem. In Alex Biryukov, Guang Gong, and Douglas R. Stinson, editors, *Selected Areas in Cryptography*, pages 171–186, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[5] Kefei Chen. A new identification algorithm. In Ed Dawson and Jovan Golić, editors, *Cryptography: Policy and Algorithms*, pages 244–249, Berlin, Heidelberg, 1996. Springer Berlin Heidelberg.

[6] Matthieu Finiasz. Np-completeness of certain sub-classes of the syndrome decoding problem. *CoRR*, abs/0912.0453, 2009.

[7] Philippe Gaborit, Julien Schrek, and Gilles Zémor. Full cryptanalysis of the chen identification protocol. In Bo-Yin Yang, editor, *Post-Quantum Cryptography*, pages 35–50, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[8] Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology — CRYPTO' 93*, pages 13–21, Berlin, Heidelberg, 1994. Springer Berlin Heidelberg.

[9] Pascal Véron. Improved identification schemes based on error-correcting codes. *Applicable Algebra in Engineering, Communication and Computing*, 8(1):57–69, 1997.