Code based cryptography and steganography

Pascal Véron

Code based cryptography and steganography

Pascal Véron[1]

IMATH, Université du Sud Toulon-Var,
B.P. 20132, F-83957 La Garde Cedex, France
veron@univ-tln.fr

Abstract. For a long time, coding theory was only concerned by message integrity (how to protect against errors a message sent via some noisely channel). Nowadays, coding theory plays an important role in the area of cryptography and steganography. The aim of this paper is to show how algebraic coding theory offers ways to define secure cryptographic primitives and efficient steganographic schemes.

Cryptography

1 Introduction

Cryptography addresses the following problem : how to scramble a message before sending it in order to make it unintelligible to any outsider. In symmetric cryptography (or private key cryptography), the message is enciphered with a function e and deciphered using a function d. These two functions depend on a parameter k called the secret-key such that for all messages m, $d(e(m,k),k) = m$. As a consequence, this key must be shared by the sender and the recipient. In practice, this may be very difficult to achieve, especially if the key has to be sent via some channel. In 1976, W. Diffie and M.E. Hellman [37] laid the foundation for public key cryptography (or asymmetric cryptography) asking the following question: is it possible to use a pair of keys (k,ℓ) such that only k be necessary for encryption, while ℓ would be necessary for decryption ? For such a protocol, d and e must satisfy for all messages m, $d(e(m,k),\ell) = m$. A cryptosystem devised in this way is called a *public key cryptosystem* since k can be made public to all users. Obviously, it should be computationally infeasible to determine ℓ from k.

The security of all conventional public key cryptosystems actually deployed in practice depends on the hardness of two mathematical problems coming from number theory : integer factoring and discrete logarithm. At this time no one knows an efficient algorithm in order to solve them in a reasonable time although numerous researchers make good progress in this area. If the security of the schemes based on this two problems is well defined, one drawback is that they rely on arithmetic operations over large numbers. Moerover, Shor's quantum algorithm [97] published in 1994 poses a serious threat to the security of these conventional cryptosystems. Indeed, quantum computers (of an appropriate size) can potentially break them in polynomial time. Although such quantum computers still do not exist, there is a strong need to develop and study alternative public key cryptosystems that would be secured in a post quantum world.

Algebraic coding theory offers an alternative supposed to resist to quantum attackers. Remember that the aim of algebraic coding theory is to restore a message m sent via a channel disrupted by some natural perturbation and that the goal of cryptography is to intentionally scramble a message m before sending it, so that it becomes unintelligible except for its recipient. Obviously there are some links between these two fields. Security of code based cryptographic primitives depends on a problem which in its general form is a well known NP-complete problem : *the syndrome decoding problem*. Generally these protocols are easier to implement, use only basic operations over the two element field and provides fast encryption and decryption algorithms.

2 Minimal background in coding theory

In this section, we recall few notions on coding theory in order to understand the sequel of this paper. For a more complete overview on this topic, the reader is addressed to [74].

Definition 1 (Linear code). *A linear code $\mathcal{C}$ is a k-dimensional subspace of an n-dimensional vector space over a finite field $\mathbb{F}_q$, where k and n are positive integers with $k \leqslant n$, and q a prime power. The error-correcting capability of such a code is the maximum number t of errors that the code is able to decode.*

Definition 2 (Hamming weight). *The (Hamming) weight of a vector x is the number of non-zero entries. We use $\omega(x)$ to represent the Hamming weight of x.*

Definition 3 (Generator and Parity Check Matrix). *Let $\mathcal{C}$ be a linear code over $\mathbb{F}_q$. A generator matrix G of $\mathcal{C}$ is a matrix whose rows form a basis of $\mathcal{C}$:*

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

A parity check matrix H of $\mathcal{C}$ is is an $(n-k)\times n$ matrix whose rows form a basis of the orthogonal complement of the vector subspace $\mathcal{C}$, i.e. it holds that,

$$\mathcal{C} = \{x \in \mathbb{F}_q^n : H^t x = 0\}\,.$$

For the sequel, we will focus our attention on the decoding problem for binary linear codes (i.e. $q = 2$). First we recall two important results.

First result. A binary linear code $\mathcal{C}$ of length n can correct t errors if for any $x, y \in \mathcal{C}$ $(x \neq y)$, $B(x,t) \cap B(y,t) = \emptyset$ where $B(x,t) = \{y \in \{0,1\}^n \mid d(x,y) \leqslant t\}$ and $d(x,y)$ denotes the Hamming distance.

Second result. A binary linear code $\mathcal{C}(n,k)$ whose minimal distance is d can correct $\lfloor (d-1)/2 \rfloor$ errors.

Let $\mathcal{C}$ be a binary $[n,k,d]$ code. Let us consider a word c' such that $c' = c_0 + e$ where $c_0 \in \mathcal{C}$ and e is what is called an error vector. Let H be a parity check matrix of $\mathcal{C}$ and let s be the syndrome of c', i.e. $s = H^t c'$. Notice that the 2^k solutions x which satisfy the equation

$$H^t x = s\,, \tag{1}$$

are given by the set $\{u + e, u \in \mathcal{C}\}$ (remember that $\forall u \in \mathcal{C}, H^t u = 0$). If the Hamming weight of e (i.e. the number of non-zero bits of e) satisfies

$$\forall u \in \mathcal{C} \setminus \{0\},\ w(e) < w(u+e)\,, \tag{2}$$

then the error e is the minimum weight solution of (1).

Remark 1. If $w(e) \leqslant \lfloor (d-1)/2 \rfloor$, then e satisfies eq. (2).

Hence, without any extra information on the code, to decode c' one has to solve an optimization problem. Notice that searching for the minimum weight word which satisfies eq. (1) is equivalent to search for the closest codeword from c'. Indeed, it is easy to see that eq. (2) is equivalent to :

$$\forall u \in \mathcal{C} \setminus \{c_0\}, \quad d(c_0, c') < d(u, c')\,. \tag{3}$$

One goal of coding theory is to find codes for which the minimum weight solution of (1) can be computed in polynomial time without constraints on the size of H. Such a problem can be stated in a more general setting as it will be developped in the next section.

3 The Syndrome Decoding Problem

Except for the Mc Eliece's cryptosystem and the CFS signature scheme, the security of all the code based cryptographic schemes that we are going to detail is based on the difficulty of the Syndrome Decoding Problem. The SD problem is a decision problem which can be stated as follows :

Name : SD
Input : $H(r, n)$ a binary matrix , s a binary column vector with r coordinates, p an integer.
Question : Is there a binary vector e of length n such that $H^t e = s$ and $w(e) \leq p$?

In the context of coding theory, if H is a parity check matrix, this means that the problem to decide wether there exists or not a word of given weight and syndrome is NP-complete.

This decision problem is linked to the optimization problem induced by maximum likelihood decoding. Indeed, searching for the closest codeword of a received word x is equivalent to find the minimum weight solution e of the equation $H^t e = H^t x$. Now, let (H, s, p) be an instance of the SD problem, the vector e exists if and only if the minimum weight solution of $H^t x = s$ is less or equal than p. On the other hand, if one knows a polynomial time algorithm to solve SD, then it can be turned into a polynomial time algorithm to compute the minimal weight of a solution of the system $H^t x = s$. In 1978, E.R. Berlekamp, R.J. McEliece and H.C.A. Van Tilborg [13] proved that this problem is NP-complete reducing it to the Three-Dimensional Matching problem [56].

Remark 2. The problem still remains NP-complete if :

- the matrix H is full rank (as it is the case for a parity check matrix),
- we ask for an s with exactly p 1's.

The SD problem can be stated in terms of the generator matrix since one can go from the parity-check matrix to the generator matrix (or vice versa) in polynomial time:

Name : G-SD
Input : $G(k, n)$ a generator matrix of a binary (n, k) code $\mathcal{C}$, $x \in \{0, 1\}^n$ and $p > 0$ an integer.
Question : Is there a vector e of length n and weight p such that $x + e \in \mathcal{C}$?

While the SD problem is NP-complete, there exists weak matrices for which an efficient algorithm can be developed. Hence, one can alternatively define algebraic coding theory as the science whose one goal is to build easy instances of the SD problem, in order to set up polynomial time algorithms for decoding. However for a random matrix H, it is necessary to know for which parameters (n, r, p) the problem seems to be difficult to solve.

4 Algorithms for the SD problem

Nowadays, there exists eight probabilistic algorithms to compute a solution to the SD problem : Lee and Brickell's algorithm [70], Leon's algorithm [71], Stern's algorithm [100], the toolbox of A. Canteaut and F. Chabaud [25], Johansson and Jönsonn's algorithm [69], the "ball-collision" decoding algorithm [18], the MMT algorithm [75] and the "1+1=0" decoding algorithm [10]. All these algorithms are devoted to search a word of small weight in a random code.

Proposition 1. *SD problem is equivalent to the following problem :*
Input *: $H(k, n)$ a binary matrix of rank k, $p > 0$ an integer.*
Question *: Is there a vector $x \in \{0, 1\}^n$ such that $H^t x = 0$, $w(x) \leq p$ and $x_n = 1$?*

All these algorithms are based on the notion of information set decoding (ISD) introduced by Prange [88].

Definition 4. *Let G be a generator matrix of an $[n,k]$ code and $c = mG$ be a codeword. Let us denote by G_i the i^{th} column of G and let $I = \{i_1, \ldots, i_k\}$ such that $G_I = (G_{i_1}, \ldots, G_{i_k})$ be a $k \times k$ invertible submatrix. Then these k coordinates uniquely determine the vector m, since $m = (c_{i_1}, \ldots, c_{i_k})G_I^{-1}$. The set I is called an information set.*

Now suppose that a received word $x = (c + e)$ is such that no errors occur in the information set I. The error pattern e can be recovered by computing $(x_{i_1}, \ldots, x_{i_k})G_I^{-1} + x$. Hence, the main idea used in all the algorithms is to select random information sets from the generator matrix (or the parity check matrix for Stern's scheme) until the support of the error does not meet the selected set which leads to a probability of success of :

$$\frac{\binom{n-p}{k}}{\binom{n}{k}} \tag{4}$$

Using the usual binomial approximation this gives the following probability of success :

$$P_{\text{succ}} = \mathcal{O}(1).2^{-nH_2(p/n)-(1-k)H_2(p/(n-k))} \tag{5}$$

where $H_2(x)$ is the classical entropy function. Hence, the work factor (number of operations) needed to compute a solution for the SD problem can be roughly estimated by :

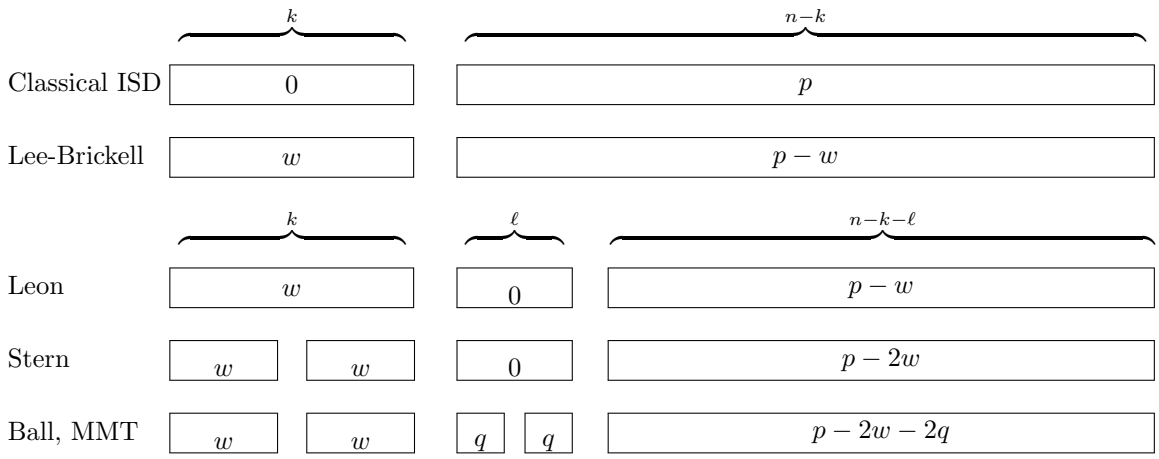
$$\frac{\text{Inv}(k)}{P_{\text{succ}}} \tag{6}$$

where $\text{Inv}(k)$ is the cost for inverting a $k \times k$ matrix. Usually this operation needs k^3 binary operations (notice that in order to be more precised, we should have take into account the probability for a random $k \times k$ matrix to be invertible). The algorithms of Lee and Brickell, Leon and Stern use some heuristic in order to minimize the call to the inverse procedure by :

1. taking into account information set which contains a small part (say w bits) of the support of the error pattern,
2. using a size-ℓ window of zeroes outside of I in order to constrain the possible locations for the error.

Canteaut and Chabaud combine these heuristics with a trick (proposed by J. Van Tilburg [105] and latter by H. Chabanne and B. Courteau [31]) in order to reduce the cost of the inverse procedure. Let I be the current information set for which the algorithm did not succeed, instead of randomly select k new columns, they exchange one column whose index is in I with a column whose index is in $\{1, \ldots, n\} \setminus I$ which decreases the cost of the Gaussian elimination. Interested readers can find a complete description and analysis of the first four algorithms in [24,25]. It follows from the study of [25] that the modified version of Stern's algorithm is the best one to solve the SD problem.

"Ball-collision", MMT and "1+1=0" algorithms are improvement of modified Stern's scheme where the major contribution comes in that some positions of the error vector are also fixed in two subsets Z_1 and Z_2 outside I. Moreover, the "1+1=0" algorithm adds a further improvement in the initial search step. Here is a graphical representation (from [19]) which illustrates how the word e is searched for a given information set I.



Nowadays, the "1+1=0" algorithm is the best one to solve the SD problem.

Another important result is that hard instances of the SD problem are obtained when the weight of the vector e is near from the theoretical minimal distance d of the code which is given by the Gilbert-Varshamov bound :

$$H_2(d/n) \simeq 1 - k/n. \tag{7}$$

Since random binary linear codes attain with overwhelming probability a rate $R(= k/n)$ (which is close to the Gilbert-Varshamov bound) the running time of the decoding algorithms (for random binary linear codes) can be expressed as a function of n and R only, namely $T(n, R)$. Let $T(n, R) = \mathcal{O}(2^{\theta n})$, where $\Theta = \lim_{n\to\infty} \frac{\log(T(n,R))}{n}$, table 1 gives the value of θ when R is close to the Gilbert Varshamov bound. In this table, half decoding means that we are searching for a word of weight $\lfloor (d-1)/2 \rfloor$ where d is the theoritical minimum distance of the code, while full decoding means that we are searching for the closest codeword from an arbitrary vector $x \in \mathbb{F}_2^n$ (see eq. 3). The

Table 1. Complexity of the decoding algorithms for SD for random codes.

	θ(half dec.)	θ(full dec.)
Lee − Brickell(1988)	0.05751	0.1208
Stern(1989)	0.05563	0.1167
Ball − collision(2011)	0.05558	0.1164
MMT(2011)	0.05364	0.1116
1 + 1 = 0(2012)	0.0497	0.1019

algorithm of Johansson and Jönsson is slightly different from the other one. The input is a list of received words and the goal is to try to decode one of them. Since the algorithm works with information set, all the tricks used in the other algorithms can be used in order to optimize it. The probability of success grows with the size of the initial list. When this list is to small, the performances are not better than those of the other algorithms (see table 2).

4.1 The q-SD problem

The SD problem can be considered over an arbitrary finite field.

Definition 5 (q-ary Syndrome Decoding (qSD) problem).
Input *: $H(r, n)$ a matrix over $\mathbb{F}_q$, s a vector with r coordinates over $\mathbb{F}_q$, an integer $p > 0$.*
Question *: Is there a q-ary vector e of length n such that $H^t e = s$ and $w(e) \leq p$?*

Table 2. Workfactor of Johansson and Jönsson algorithm

size of list	$n = 1024, k = 524, p = 50$	$n = 512, k = 256, p = 56$
1	$2^{68.1}$	$2^{72.2}$
2^5	$2^{63.7}$	$2^{68.9}$
2^{10}	$2^{59.5}$	$2^{65.9}$
2^{15}	$2^{56.2}$	$2^{64.1}$
2^{30}	$2^{50.2}$	2^{60}

In 1994, A. Barg proved that this last problem remains NP-complete [8, in russian]. In [87], C. Peters generalizes all the ISD algorithms to the case of codes over $\mathbb{F}_q$ with $q > 2$. As an example, to reach a complexity of 2^{128}, it is enough to choose a $[961, 771]$ code over $\mathbb{F}_{31}$ and a word of weight 48. For the same complexity, in the binary case, we have to choose a $[2960, 2988]$ code and a word of weight 57. If the matrix is a public key, the matrix over $\mathbb{F}_{31}$ can be stored usion $90Kb$ while the one over $\mathbb{F}_2$ needs $188Kb$.

4.2 Quantum computers and the SD problem

The SD problem cannot be polynomially solved using quantum computers. However, the Grover's quantum algorithm [63,64] for computing roots of a function can be used in order to speedup the probabilistic algorithms against SD. In [15], the author shows that the quantum version of the information set decoding algorithms takes time only $c^{(1/2+o(1))n/\log_2 n}$ to break a length n and rate R code (with $c = 1/(1-R)^{1-R}$) where as the non quantum version takes time $c^{(1+o(1))n/\log_2 n}$. As a consequence, protecting against these quantum attacks requires essentially quadrupling the key size.

5 The SD Identification Scheme

5.1 Introduction

The SD Identification scheme is the first cryptographic protocol whose security relies on the difficulty of the SD problem. An identification scheme is a cryptographic protocol which enables party A (called the "prover") to prove his identity (by means of an on-line communication) polynomially many times to party B (called the "verifier") without enabling B to misrepresent himself as A to someone else. In 1985, S. Goldwasser, S. Micali and C. Rackoff described a very nice solution to this problem with zero-knowledge proofs [61], where a user convinces with a non-negligible probability an entity that he knows the solution s of a public instance of a "difficult" problem without giving any information on s (see [89] for a nice introduction to zero-knowledge). In 1986, A. Fiat and A. Shamir proved the practical significance of zero-knowledge proofs for public-key identification [42]. Their scheme relies on the difficulty of factoring. Notice that, from a practical point of view, the prover may be identified to a smart card, hence it is supposed that he has reduced computational power and a small amount of memory. Since 1988, there were several attempts to build identification schemes which did not rely on number theory and use only very simple operations so as to minimize computing load. The idea to use error-correcting codes for identification is due to S. Harari [65], unfortunately his scheme was not zero-knowledge and not really practical due to its heavy communication load. Moreover, the scheme has been proved to be insecure in [106]. Another scheme proposed by M. Girault [59] has been cryptanalysed in [94].

5.2 Stern's scheme

The first truly practical scheme using error-correcting codes is due to J. Stern [101]. The scheme uses a fixed binary (k, n) parity check matrix H which is common to all users. In 1995, a dual

version of Stern's scheme has been defined : the G-SD identification scheme [107]. This version improves the communication complexity (number of bits exchanged during the protocol) for exactly the same level of security as those of Stern's scheme.
Table 3 lists the secret and public data used in the SD protocol. The pair (i, p) is the public identification of the prover. His 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. The principle of the protocol is the following: the prover (Alice) knows the secret vector s

Table 3. Public and secret data in the G-SD identification scheme

Common public data	: $H(k, n)$ a full rank binary matrix , a hash function denoted by $\langle . \rangle$.
Prover's secret data	: $s \in \{0, 1\}^n$.
Prover's public data	: $i = Hs$ and $p = \omega(s)$.

which satisfies $Hs = i$ and $p = \omega(s)$. 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 $1 - q^r$.
The identification scheme relies on the notion of commitment. Commitment is a protocol between Alice and Bob which operates in 3 stages:

- Stage 1: Alice hides a sequence u of bits and sends it to Bob. The hidden function is public and hard to invert.
- Stage 2: Alice and Bob execute some protocol,
- Stage 3: Alice reveals u, Bob checks the validity of the hidden value received during stage 1.

From a practical point of view, u is hidden via a cryptographic public hash function. Hence Alice sends to Bob the image $\langle u \rangle$ of u. The hash function must be collision-free (i.e. it should be "infeasible" to compute $u' \neq u$ such that $\langle u' \rangle = \langle u \rangle$). Discussion on the length of the hash value $\langle u \rangle$ can be found in [60]. Let us denote by $x.y$ the concatenation of the binary strings x and y and by $y\sigma$ the image of $y \in \{0, 1\}^n$ under the permutation σ of $\{1, \ldots, n\}$, the SD scheme includes r rounds each of these being performed as described in table 4.

5.3 Security and performances

It can be proved that:

- the scheme is zero-knowledge i.e., informally speaking, during the protocol the transactions contain no information on s (more formally one can construct a polynomial time machine S which outputs a communication tape having the same probability distribution as a real communication).
- a cheater can bypass the protocol with a probability bounded by $(2/3)^r$, otherwise one can construct a polynomial-time probabilistic machine which either outputs a valid secret s or finds collision for the public hash function.

Practical security of the scheme is linked to the parameters n, k, p and r. Let H be the parity check matrix used in the scheme. In order to impersonate A, an intruder has to be able to compute a word s of weight p whose image under H is i (this is the SD problem). If p is chosen slightly below

Table 4. A round of the SD scheme

- A randomly computes :
  - $y \in \{0,1\}^n$,
  - σ a permutation of $\{1,\ldots,n\}$.

  and send to B three commitments :

$$c_1 = \langle \sigma, Hy \rangle,\ c_2 = \langle (y+s)\sigma \rangle, c_3 = \langle y\sigma \rangle$$

- B sends a random element $b \in \{0,1,2\}$ (challenge).
- if $b = 0$,
  - A reveals y and σ,
  - B checks the value of c_1 and c_3.
- if $b = 1$,
  - A reveals $y + s$ and σ,
  - B checks the value of c_1 and c_2.
- if $b = 2$,
  - A reveals $y\sigma$ and $s\sigma$,
  - B checks the value of c_2 and c_3 and verifies that $w(s\sigma) = p$.

the value of the theoretical minimum distance of $\mathcal{C}$ then the probability that there exists a word $s' \neq s$ of weight p such that $Hs = Hs'$ is very low. Hence by choosing

$$n = 700, k = 350, p = 75,$$

searching the vector e with the probabilistic algorithms described in section 4 needs around 2^{70} operations. Moreover taking $r = 35$, the probability of success of a cheater is bounded by 10^{-6}.

If we envisage the prover as a smart card, essentially three parameters are to be taken into account: the communication complexity (number of bits exchanged during the protocol), the complexity of the computations done by the prover and the storage capacity needed by the prover. The SD identification scheme uses only very simple operations over the two element field (i.e. over bits) and can be implemented in hardware in a quite efficient way. One drawback is the size of the matrix H which must be stored by the prover. Another one is the communication complexity since at least 35 rounds are needed in order to achieve a reasonable level of security while for the same level (from a dishonest prover point of view) identification schemes based on number theory can be performed in only few rounds (4 rounds for Fiat-Shamir's scheme). Table 5 sums up the performances of Stern's scheme, G-SD scheme and Fiat-Shamir's scheme (1024 bits version) giving for each one : the number of rounds needed to achieve a probability of success of 10^{-6} for a dishonest prover, the total communication complexity, the size of the ROM (number of bits stored by the prover), the total prover's computation complexity (number of binary operations performed by the prover during the whole protocol).

Table 5. SD schemes versus Fiat-Shamir scheme

	SD	G-SD	Fiat-Shamir
Rounds	35	35	4
ROM	123550	124250	5120
Computation complexity	$2^{23.04}$	$2^{23.04}$	$2^{25.4}$
Communication complexity	52523	44357	4628

6 The McEliece's public-key cryptosystem

Despite Mc Eliece's cryptosystem be the first code based cryptosystem, we decide to not describe it first because its security does not directly rely on the SD problem.

Soon after Diffie-Helmman's paper on public key cryptography , R.L. Rivest, A. Shamir and L. Adleman exhibited such a system: the well known RSA cryptosystem based on the factorization of integers [90]. Merkle and Hellman [78] proposed another cryptosystem based on the difficulty of the integer packing "knapsack" problem. There were several variants around this latter but the development of the LLL algorithm made most of them insecure.
In 1978, R.J. McEliece defined the first public key cryptosystem using algebraic coding theory [76]. The basic idea is quite simple: use as a secret key a code $\mathcal{C}$ which belongs to a family of codes for which a polynomial time decoding algorithm exists and give as a public key an equivalent code $\mathcal{C}'$ which masks the algebraic structure of $\mathcal{C}$, so that $\mathcal{C}'$ looks like a random binary linear code. Table 6 describes the general protocol. Of course, one important parameter of this protocol is the

Table 6. A code based public key cryptosystem

Secret Key:
- G a generator matrix of a binary linear $[n, k, d]$ code $\mathcal{C}$ for which a polynomial time decoding algorithm $\mathcal{A}$ is known,
- S a non-singular random $k \times k$ binary matrix,
- P a random binary $n \times n$ permutation matrix.

Public Key: $G' = SGP$ and $t = \lfloor (d-1)/2 \rfloor$.

Encryption :

. Message : $m \in \{0,1\}^k$,
. Cryptogram : $c = mG' + e$ where $e \in \{0,1\}^n$ satisfies $w(e) = t$.

Decryption : Since $w(eP^{-1}) = w(e)$, successively compute :

. $mS = \mathcal{A}(cP^{-1}) = \mathcal{A}((mS)G + eP^{-1})$,
. $m = (mS)S^{-1}$.

code $\mathcal{C}$ to use:

- For n, k and d fixed, $\mathcal{C}$ must belong to a large family of codes so that it is impossible to find it via an exhaustive search. Notice that is is enough to find an equivalent code to the public one using an algorithm due to N. Sendrier [94] which can determine if two generator matrices define equivalent codes and can find back the permutation,
- a polynomial-time decoding algorithm must exist for $\mathcal{C}$,
- no information about the code $\mathcal{C}$ can be obtained from the generator matrix G'.

The third condition eliminates some classes of well known "decodable" codes such as generalized Reed-Solomon codes (as shown by V.M. Sidelnikov and S.O. Shestakov [99]), and concatenated codes (as shown by N. Sendrier [93]). The class of binary Goppa codes [62] as suggested by McEliece seems to satisfy these 3 conditions.

Definition 6. *Let $g(z) \in \mathbb{F}_{2^m}[z]$, $L = \{\alpha_1, \ldots, \alpha_n\} \subset \mathbb{F}_{2^m}$ such that $\forall i, g(\alpha_i) \neq 0$. The Goppa code $\Gamma(L, g)$, of length n over $\mathbb{F}_2$, is the set of codewords, i.e. $n-$tuples $(c_1, \ldots, c_n) \in \mathbb{F}_2^n$, satisfying*

$$\sum_{i=1}^{n} \frac{c_i}{z - \alpha_i} \equiv 0 \pmod{g(z)}.$$

Proposition 2. *The dimension k of $\Gamma(L, g)$ and its minimal distance d satisfy*

$$k \geqslant n - m \deg g(z)$$
$$d \geqslant \deg \bar{g}(z) + 1.$$

where $\bar{g}(z)$ is the lowest degree perfect square which is divisible by $g(z)$.

Remark 3. For irreducible Goppa codes (i.e. codes for which $g(z)$ is irreducible), we deduce that the minimum distance satisfies $d \geqslant 2 \deg g(z) + 1$.

6.1 Cryptanalysis

McEliece recommended using an irreducible binary Goppa code of length 1024 with $L = \mathbb{F}_{2^{10}}$ and $g(z)$ an irreducible polynomial of degree 50. Since the number of monic irreducible polynomials of degree 50 over $\mathbb{F}_{2^{10}}$ is given by $(\sum_{d|50} \mu(d) 2^{500/d})/50$ (where μ is the Möbius function), this gives about 2^{500} candidates which clearly prevents any exhaustive search. However, two other kind of attacks can be envisaged against McEliece's cryptosystem :

- a structural attack,
- a generic attack.

A structural attack A structural attack against McEliece's cryptosystem consists in studying the algebraic structure of the public code $\mathcal{C}$ in order to build a decoder (or at least to find some parameters of the hidden code). Remember that L and $g(z)$ are the two essential parameters for the decoding algorithm. Until know, there does not exist any algorithm which takes as input a generator matrix of a Goppa code and which outputs these two data. However, as pointed out by J.K. Gibson, if a generator matrix G of a binary Goppa code and L are known, it is then possible to find back the polynomial $g(z)$ [58] . Hence one can devise a cryptanalysis in three steps :

1. fix a permutation of $\mathbb{F}_{2^m}$ say $\bar{L} = \{\beta_1, \ldots, \beta_{2^m}\}$,
2. search for a permutation π of the columns of G' which transforms the public matrix into the generator matrix $\bar{G}$ of a $\Gamma(\bar{L}, \bar{g})$ Goppa code,
3. compute $\bar{g}$ from $\bar{G}$ and $\bar{L}$ and use the decoder of $\Gamma(\bar{L}, \bar{g})$ to decode the public code $\mathcal{C}$.

In [1], C.M. Adams and H. Meijer claim that there is no more than one permutation which satisfies step 2 of the cryptanalysis. This is not true, as proved by J.K. Gibson [58], who showed that there exists at least $m2^m(2^m - 1)$ such permutations. Unfortunately for $m = 10$, this represents less than $2^{-8713}\%$ of all the permutations !
Nevertheless, P. Loidreau and N. Sendrier developed a nice attack when the polynomial $g(z)$ has only binary coefficients [73]. They use the support splitting algorithm (SSA) [94] which is able to decide if two linear codes are equivalent and outputs the permutation. Their structural attack uses the fact that Goppa codes defined from a binary polynomial have a non-trivial automorphism group (and so the automorphism group of the corresponding public code is also non-trivial). This cryptanalysis brings out weak keys in McEliece's cryptosystem even if their number is negligible as compared to the number of possible keys. A "real" structural attack to date necessitates a proper classification of Goppa codes.

A generic attack Without the knowledge of L and g, it seems that it is computationally hard to make the difference between a random matrix and the generator matrix of a Goppa code. This is the Goppa code distinguishing problem (see section 6.4) :

Name : GD
Input : $G(k,n)$ a binary matrix,
Question : Does there exists $m \in \mathbb{N}$, $L \subset \mathbb{F}_{q^m}$ and $g(z) \in \mathbb{F}_{q^m}[z]$ such that G be a generator matrix of the $\Gamma(L,g)$ code ?

Since there does not exist any suitable algorithm which uses the underlying Goppa code structure of McEliece's cryptosystem, cryptanalysis of the system boils down to the general problem of the decoding of a random binary linear code (the G-SD problem). In fact, cryptanalysis of McEliece's cryptosystem relies on a variant of the G-SD problem. Indeed, the weight t of the error is linked to the parameters of the code. Let $n = 2^m$, it seems that for irreducible Goppa codes the dimension k always satisfies $k = n - mt$, hence $t = (n-k)/\log_2(n)$. The underlying problem to solve is then the following :

Name : GPBD (Goppa Parametrized Bounded Decoding)
Input : G a fullrank binary matrix $k \times n$, $y \in \{0,1\}^n$
Question : Does there exists $e \in \{0,1\}^n$ such that $y+e$ be a linear combination of rows from G and $w(e) \leqslant (n-k)/\log_2 n$?
This problem is NP-complete [43].

McEliece's cryptosystem with its original parameters can be cryptanalysed in $2^{64.2}$ binary operations using the algorithms to solve the SD problem [26]. Johansonn and Jönsson algorithm can output a cleartext from a list of 1024 cryptogram in $2^{59.5}$ operations. In order to obtain a security level of 2^{80} the parameters to use are [17] :

$$m = 11, n = 2048, k = 1685, t = 33\,.$$

For a security level of 2^{128}, a set of possible parameters is [46] :

$$m = 12, n = 4096, k = 3604, t = 41\,.$$

In [19], the authors proposed a bound on "future improvements" in attacks against the McEliece's cryptosystem, and suggested that designers use this bound to "choose durable parameters".

Remark 4. In its original form, the cryptosystem is vulnerable to active attacks where an intruder modifies the cryptogram and uses as an oracle a deciphering machine. The protocol is also vulnerable to message replay. That is to say that an intruder is able to distinguish the fact that two cryptogram come from the same plaintext and in this context he can devise an attack which can recover the message in less than 8 iterations for the original parameters.

6.2 Niederreiter's variant

In 1986, Niederreiter [84] defined the dual version of McEliece's cryptosystem using the parity check matrix of the code instead of the generator matrix (see table 7). From a security point of view Niederreiter's cryptosystem and McEliece's cryptosystem are equivalent (if used with exactly the same parameters [72]). However they differ from a practical point of view. Unlike McEliece's cryptosystem, it is not necessary to use a pseudo-random generator for encryption process. Notice, however that the plaintext is a n-binary word of weight t, hence we need a practical algorithm which maps the integers between 1 and $\binom{n}{t}$ to the set of words of weight t and length n and vice-versa. Such algorithms can be found in [47,92].

Niederreiter's cryptosystem allows to reduce by a factor of 2 the size of the public key. Indeed, the matrix H can be expressed as $H = (I_{n-k} \mid M)$, hence it is enough to store the $(n-k) \times n$ matrix M. Such a trick is impossible in McEliece's cryptosystem since if $G' = (I_k \mid M)$ and the original message is not random, the cryptogram $c = mG' + e$ would reveal a part of the plaintext.

Since the public key in Niederreiter's cryptosystem is smaller and the plaintext is a word of small weight, this implies that the number of operations involved during the encryption process is less than what is done in McEliece's cryptosystem. Finally, depending on the parameters, the

Table 7. Niederreiter's cryptosystem

Secret key :
- A binary linear code $\mathcal{C}[n, k, d]$ for which there exists a polynomial algorithm $\mathcal{A}$ able to correct $t \leqslant \lfloor (d-1)/2 \rfloor$ errors,
- $S(n-k, n-k)$ an invertible matrix,
- $P(n, n)$ a permutation matrix.

Public key : $(H' = SHP,\ t)$ where H is a parity check matrix of $\mathcal{C}$.

Encryption :

. Message : $m \in \{0,1\}^n$ of weight t,
. Cryptogram : $c = H'^t m$.

Decryption :

. Compute $S^{-1}c = HP^t m$,
. Since $w(P^t m) \leqslant t$, apply $\mathcal{A}$ to find back $P^t m$,
. Compute $m = {}^t(P^{-1}P^t m)$.

transmission rate (number of information symbols/ number of transmitted symbols) which is equal to $\log_2 \binom{n}{t}/(n-k)$ can be better or worst that those of McEliece (k/n).

Table 8 sums up these differences and makes a comparison with the RSA cryptosystem when used with a 2048 modulus and a public exponent e equal to $2^{16}+1$ as in `openssl` toolbox (the complexity is given as the number of binary operations to perform per information bit):

Table 8. A comparison between McEliece, Niederreiter and RSA cryptosystems

	McEliece $(2048, 1718, t=30)$	Niederreiter $(2048, 1718, t=30)$	RSA-2048 $e = 2^{16}+1$
Public-key size (Kbytes)	429.5	69.2	0.5
Transmission rate	83.9%	67.3%	100%
Encryption complexity	1025	46.63	40555
Decryption complexity	2311	8450	$6557176,5$

Remark 5. Notice that in his original paper, Niederreiter suggested using either a binary $[104, 24, 32]$ code (obtained by concatenation of other binary codes) or a $[30, 12, 19]$ Reed-Solomon code over $\mathbb{F}_{31}$. These two codes were verified as insecure by Brickell and Odlyzko [23] using the LLL algorithm.

6.3 Hardware and software implementations

To assess the performances of Mc Eliece's cryptosystem, several implementation have been realized. In [22], on a 32 bits processor, for a security level of 2^{128}, the software implementation of Mc Eliece's cryptosystem gains an order of magnitude for both encryption and decryption compared to RSA-2048 (CPU cycles are divided by 5 per byte to encrypt and by 100 per byte to decrypt).
An 8-bit version for AVR microprocessors and for FPGA is described in [39]. Once again, results show that Mc Eliece's cryptosystem gives better results compared to RSA but not compared to

elliptic cryptosystems.. A smart card implementation (16 bits processor) is described in [102], ciphering and deciphering is done in less than 2 seconds for a 2048 code length.
Hardware implementations of Mc Eliece's cryptosystem gave rise to several side channel attacks [103,98,66,27,80].

6.4 The Goppa Distinguishing problem

This problem has been stated in [32] by N. Courtois, M. Finiasz and N. Sendrier. It has been widely believed for ten years that this problem was computationally hard. As a consequence, this hardness assumption has been used in numerous proofs of security of code based cryptosystems [22,85,38,29,35]. However, notice that even if a security proof cannot be stated for a cryptosystem, it does not mean that there exists an efficient cryptanalysis against this scheme. Unformally speaking, it just means that it cannot be formally proved that an efficient algorithm breaking the scheme can be turned into an efficient algorithm being able to solve a well known difficult problem.

In 2010, J.C Faugère, A. Otmani, L. Perret and J.-P. Tillich proposed the first algorithm which can decide if a binay (k, n) matrix is a random one or generates a Goppa code [41]. The distinguisher is highly discriminant for high rate code (i.e. when k is near from n).

The main idea is to compute the rank of a linear system deduced from the generator matrix G. Goppa codes are a subset of alternant codes whose parity check matrix is :

$$V_r(x,y) = \begin{pmatrix} y_1 & y_n \\ y_1x_1 & y_nx_n \\ \vdots & \vdots \\ y_1x_1^{r-1} & y_nx_n^{r-1} \end{pmatrix}$$

where $x_i, y_i \in \mathbb{F}_{q^m}$. The corresponding alternant code (whose dimension is greater or equal than $n - mr$) is Ker $V_r(x,y) \cap \mathbb{F}_q^n$. Using this matrix, one can build a polynomial decoder which can correct up to $\lfloor r/2 \rfloor$ errors.

By defintion of the public encryption matrix G, we have $V_r(x,y)G^t = 0$, where the elements x_i and y_i are the solution of the system :

$$\{g_{i,1}y_1x_1^j + \cdots + g_{i,n}y_nx_n^j = 0 \mid i \in \{1,\ldots,k\},\ j \in \{0,\ldots,r-1\}\}\,. \tag{8}$$

For the parameters used in Mc Eliece's cryptosystem, such a system cannot be solved. Moreover, if we recover the x_i's and the y_i's only $r/2$ errors can be decoded instead of r. However, a distinguisher can be designed from this system. Using a linearization process, this system can be transformed in another one with k equations and $\binom{mr}{2}$ unknowns. For high-rate Goppa codes, the rank of this system is (with high probability) :

$$mr((2\ell+1)r - 2^\ell - 1)\,,$$

where $\ell = \lfloor \log_2 r \rfloor + 1$. It holds that the rank of the same system, obtained from a random binary matrix G, will be 0 or $\binom{mr}{2} - k$ depending wether $k \geqslant \binom{mr}{2}$ or not. Table 9 gives, for codes of length 2^m, the smallest r for which the distinguisher does not work. This result has to be seriously taken into account for the parameters to use in a code based cryptosystem whose security proof relies on the hardness of GD assumption.

7 Some other code based cryptosystems

During the past 20 years, a lot of code based cryptosystems have been designed. Here are few comments about these schemes and a list of bibliographical notes for further reading. Some of the

Table 9. Smallest order r of a binary Goppa code of length $n = 2^m$ for which the distinguisher does not work.

m	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
r_{min}	5	8	8	11	16	20	26	34	47	62	85	114	157	213	290	400

protocols listed below are described in [108].

A pseudo random generator In 1996, B. Fischer and J. Stern [47] defined a set of strongly one-way functions related to the SD problem. Using this set, they described an efficient pseudo random generator which can output 3500 bits/sec as compared to an RSA based generator (512 bits modulus) which outputs 1800 bits/sec. Their scheme has been improved in 2007 [54] using regular words and circulants codes (see table 10 section 8.3).

A signature scheme A signature scheme is a protocol where the recipient of a message M can check its integrity, the sender's identity and such that the sender cannot refute that he sent M. Usually the sender does not sign the message M itself but a hash value $h(M)$ of M. Every public key cryptosystem can be used to sign a message, using the deciphering algorithm with $h(M)$ as input, and the output of this algorithm as the signature s. The recipient uses s and the public key of the sender as inputs of the enciphering algorithm and checks that the output is equal to $h(M)$. Using Niederreiter's cryptosystem, N. Courtois, M. Finiasz and N. Sendrier proposed in 2001 [32] a signature scheme which ouputs very short signatures. The main problem is that hash values lie in the set of syndromes and must match the syndrome of an error of weight t in order to apply the deciphering function. For Goppa codes, the probability for a syndrome to be a "decodable" syndrome is roughly $1/t!$. Hence instead of directly compute the value $h(m)$ the idea is to successively compute $h(m||i)$ where i is a counter which is increased by 1 until a decodable syndrome be obtained ($||$ is the concatenating operator). The proof of security of the scheme [34] relies on the hardness of GPBD problem (see sec. 6.1) and GD problem (see sec. 6.4). Notice however that, since t must be small (at mots $t!$ attempts are needed to find a decodable syndrome), the scheme must use very large Goppa codes to resist against the various ISD algorithms. Since the dimension of a Goppa code is $n - mt$, it means that CFS uses high rate Goppa code and thus the GD problem falls in the area of parameters where it can be easily solved ! To reach a security level of 2^{80}, the scheme uses a code of length 2^{21} and codimension 210, and produces 211 bits signature. The size of the public matrix H is $52.5Mb$. This size can be reduced using parallel CFS [44]. CFS has been implemented in hardware on a FPGA (Field Programmable Gate Array) giving a signature time of 0.86 second [21] for a security level of 2^{63}.

A hash function At Mcrypt 2005, a provably collision resistant family of hash functions have been proposed by D. Augot, N. Finiasz and N. Sendrier [6]. The Fast Syndrome Based Hash function is based on the Merkle-Damgård design [36] which consists in iterating a compression function $\mathcal{F}$. This function takes as input a word of s bits, maps it to a word of length n and weight t and computes its syndrome from a given $r \times n$ parity check matrix (with $r < s$). The mapping is done using regular words in order to speed up the process.

Definition 7. *Let consider a binary word of size n as n/t consecutive blocks of size t. A (n, t) regular word is a word which has exactly one non-zero coordinate in each block.*

From an algorithmic point of view, the generation of (n, t) regular worlds is obviously easiest than the one of constant weight words. The security of the hash function relies on two new NP-complete problems [43] linked to the original SD problem : RSD and 2-RNSD.

Name : RSD (Regular Syndrome Decoding);
Input : H a fullrank $r \times n$ binary matrix , an integer t and a syndrome y,
question : Does there exists a (n, t) regular word $e \in \{0,1\}^n$ such that $H^t e = y$?
Name : 2-RNSD (2-Regular Null Syndrome Decoding)
Input : H a full rank $r \times n$ binary matrix r, p an integer,
Question : Does there exists a 2-regular (n, p) word e such that $H^t e = 0$?

Remark 6. A 2-regular (n, p) word is a word of length n such that each of the p consecutive blocks of size n/p contains either zero or two one.

Depending on the value of n, r and t, the hash function can be cryptanalysed using ISD algorithms or Wagner's generalized birthday technique [109]. Taking into account this two kind of attacks, the size of the output functions must be of at least 5ℓ bits for a security level of 2^ℓ. The proposed scheme has two main drawbacks :

- the size of the matrix H is large (around 1Mbytes for the parameters suggested in [6]). Paradoxically, the speed of the compression function can be improved with larger n while keeping a constant security level of 2^{80},
- usually the security of a hash function must be half its output size.

In 2007, an improvement of this scheme has been proposed by N. Finiasz, P. Gaborit and N. Sendrier [45]. Unfortunately the proposed parameters lead to two kind of cryptanalysis [91,48]. Taking into account these two attacks, a new version has been proposed for the SHA-3 challenge [5], but the function was quite slow and was not selected for the second round of competition. Later, an optimization (RFSB) has been proposed in [14]. The RFSB hash function runs at 13.62 cycles/byte while SHA-256 runs at 15 cycles/byte.

An identity based identification scheme The main problem in "real life" public key cryptography is to establish a link between a public key and its owner's identity. In 1984, Shamir introduced the notion of identity based public key cryptography [96]. The concept make use of a trusted third party : the KGC (Key Generation Center). This one has a master public key and a master secret key. From an identity i and the master public key, any one can derive the public key linked to i. In 2004, Bellare, Neven and Namprempre described a generic method to derive an identity base identification scheme from a standard authentication scheme [11]. As usual this concept has only been applied to number theory schemes. In 2007 [29], P.-L. Cayrel, P. Gaborit and M. Girault considered the combination of two code based schemes (CFS signature scheme and Stern's identification scheme) in order to produce the first identity based identification scheme using error correcting codes. The generation of Alice's parameters is obtained from an execution of the CFS signature's scheme. Hence in order to prevent an intruder to be able to compute Alice's secret key from her identity, one has to consider the parameters that guarantee the security of the CFS scheme. The drawback is that the CFS scheme uses very long Goppa codes while Stern's scheme uses shorter ones. Since the same matrix has to be used by the KGC and by the identification process, this will overload the communication complexity.

A ring signature scheme A t-out-of-N threshold ring signature scheme is a protocol which enable any t participating users belonging to a set of N users to produce a signature in such a way that the verifier cannot determine the identity of the t actual signers. Classical t-out-of-N threshold ring signature schemes based on number theory have complexity $\mathcal{O}(tN)$. Using Stern's three-pass identification scheme, Aguilar et al. [2] defined the first t-out-of-N threshold ring signature scheme whose complexity is $\mathcal{O}(N)$. Performances of the scheme has been improved in [28] and a security proof is given in [40].

8 Improving code based cryptosystems

There are essentially two drawbacks in code based cryptography. First, some protocols needs the generation of constant weight. This is a problem which involves computation which slow down the

whole process. Next, all the schemes depend on a public matrix whose size is greater than the usual public data used in number theory based cryptography. An issue to the first problem is to used regular words (see preceeding section) instead of constant weight words. For the second problem, numerous research have been done in order to find codes with a "compact" representation. At this stage, it is important to distinguish protocols which use Goppa codes (like Mc Eliece or CFS) from those which use random codes.

8.1 List decoding algorithms, Specific polynomials

In Mc Eliece's cryptosystem, for a given keysize, the security level will be increased by adding extra errors. Symmetrically, adding extra errors makes it possible to use shorter keys while keeping a similar security level, but it also requires the receiver to decode the additional errors. Let t be the error capacity of the code, in [4] authors described a "list decoding algorithm" which can correct up to $(n-\sqrt{n(n-4t-2)})/2) \geqslant t+1$ errors which is an improvement of a first algorithm decribed in [16]. Since we add extra errors, encrypting distinct codewords can lead to the same cryptogram. A list decoding algorithm outputs a list of candidates. Hence, if the initial message m has been first formatted, before computing mG, it should be easy to find back the correct codeword. Adding only one extra error, a security level of 2^{80} can be obtained using a $(1632, 1269, 33)$ Goppa code instead of $(2048, 1751, 27)$ Goppa code. The size of the public matrix will be 460647 bits instead of 520047 bits, i.e. 12% smaller [16]. Using a list decoding algorithm leads to shorter keys at the expense of a moderately increased decryption time.

Another idea to reduce the size of the public key, is to use Goppa codes over $\mathbb{F}_q$ built on polynomials of the form g^{q-1} where $g \in \mathbb{F}_q^m$ is an irreducible polynomial of degree t over $\mathbb{F}_q^m$. From [104], these codes have a better error-correction capacity: they can correct up to $\lfloor qt/2 \rfloor$ errors. Combining this trick with the preceeding one, a $[1633, 1297, 49]$ code over $\mathbb{F}_7$ with 2 extra errors achieves a security level of 2^{128} and leads to a public matrix of 1223423 bits [20]. For the same security level, over $\mathbb{F}_2$, the size of the matrix will be 1537536 bits using a $[2960, 2288, 57]$ code with one extra error.

8.2 Quasi cyclic and dyadic codes

Another way to reduce the size of the public key in Mc Eliece's cryptosystem is to used some structured codes which admit a "compact" representation. This issue has been first addressed in 2005 by P. Gaborit [52] by using set of s quasi-cyclic subcodes of a given BCH code. The particularity of quasi-cyclic codes is that the whole generator matrix can be derived from the knowledge of few rows. Hence it is enough to publish these few rows (a kind of compressed version of the public matrix) instead of the whole matrix. In 2007, M. Baldi and F. Chiaraluce proposed to use quasi-cyclic LDPC codes [7]. LDPC codes are defined by a very sparse parity-check matrix and can be represented in a compact form. These two propositions have been cryptanalyzed in [86].

In 2009, two new modifications have been proposed using alternant quasi-cyclic codes and quasi dyadic codes [12,79] and cryptanalyzed in [41,57]. The generator matrix of these two families can be derived from the knowledge of one row. Only the binary version of quasi dyadic codes has not been cryptanalyzed. With these codes, the size of the public key of Mc Eliece's cryptosystem and CFS signature scheme can be highly reduced [9] (see table 10).

8.3 Circulant codes

For protocols using random codes, a particular class of quasi-cyclic codes can be used, those whose generator matrix is obtained by concatenation of circulant matrix.

Definition 8. *A $r \times r$ circulant matrix is such that the $r-1$ latest rows are obtained by cyclic shifts of the first row.*

It was shown in [55] that, if one admits a small constraint on the size n of the code then such codes behave like purely random codes (in particular they satisfy the Gilbert-Varshamov bound). Hence they are well suited to be used in code base schemes for which a random matrix is needed. Although all classical algorithms used to find a word of given weight in a code do not give better results when applied to quasi cyclic codes, nowadays it is not known if the decoding of a random quasi cyclic code is an NP-complete problem.

In 2007, a modification of Stern's identification scheme has been proposed using as public matrix H, the concatenation of two $k \times k$ circulant matrices (the identity matrix and a random one) [53]. This way, the public matrix can only be described from the first line of the random matrix which in particular decreases the size of the data which must be stored by the prover. The underlying difficult problem upon which the security of the scheme is linked can be stated as follows :

Name : Syndrome Decoding of Double Circulant Linear Codes
Input : $H(k, 2k)$ a double binary circulant matrix , s a binary column vector with r coordinates, p an integer.
Question : Is there a binary vector e of length n such that $H^t e = s$ and $w(e) \leq p$?
Nowadays, it is not known if this problem is NP-complete.

Using this same trick and the regular words C. Laudauroux, P Gaborit, and N. Sendrier have defined in 2007 a modified version of Fischer-Stern's algorithm in order to speed the output of the generator : the SYND pseudo random generator [54]. They obtain this way a pseudo random generator as fast as AES in counter mode[67] with few memory requirement (around 1Kbytes). Moreover, the scheme has a formal proof of security.

8.4 Codes over $\mathbb{F}_q$

Stern's identification scheme has two major drawbacks :

1. since the probability of a successful impersonation is 2/3 for Stern's construction instead of 1/2 as in the case of Fiat-Shamirs protocol based on integer factorization, Stern's scheme uses more rounds to achieve the same security, typically 28 rounds for an impersonation resistance of 2^{16},
2. there is a common data shared by all users (from which the public identication is derived) which is very large, typically 66 Kbits. In Fiat Shamir's scheme, this common data is 1024 bits long.

In [30], using the q-SD problem, the authors proposed a 5-pass identication scheme for which the success probability of a cheater is 1/2, reducing this way the number of rounds needed for an identification process. Using quasi dyadic codes, they also reduce the size of the public data.

We sum up in table 10 the characteristics of this different improvements when applied to various code based schemes.

Remark 7. For the modified version of Stern's identification scheme there exists a variant in which the secret key is embedded in the public one. This allows to reduce again the size of the public and private data but increases the complexity computation and the global transmission rate (see [53] for more details).

9 Secret Sharing Schemes

A (k, n) secret sharing scheme is a protocol where a secret S is split into n pieces, each one being distributed to n users. If strictly fewer than k users meet together, they must not be able to compute S. Any assembly of k (or more) users can retrieve S. This problem was first considered by A. Shamir and he gives a solution using interpolation of polynomials over $\mathbb{Z}_p$, the secret being the

Table 10. Some characteristics of the improved schemes

McEliece[79]	
(n,k)	$(4096,2048)$
t	128
Pub.key	4 Ko
Security level	2^{128}
QD version	

Pseudo Random Generator[54]	
(n,k)	$(8192,256)$
t	32
Data	1.03 Ko
Trans. rate	1Gbits/sec
Security level	2^{152}
QC version + Regular words	

CFS[9]	
(m,t)	(21,10)
Signature cost	$2^{22.8}$
Pub. key	24.49 Mo
Security level	$2^{81.5}$
QD Version	

SD identification scheme[53]	
(n,k)	$(634,317)$
t	69
Public data	634 bits
Private data	951 bits
Trans. rate	40096 bits
Security level	2^{85}
QC Version	

q-SD identification scheme ($\mathbb{F}_{2^8}$)[30]	
(n,k)	$(134,67)$
t	49
Public data	1072 bits
Private data	1072 bits
Trans. rate	33040 bits
Security level	2^{87}
QD Version	

constant term of a polynomial f of degree $k-1$. Each participant owns a pair $(i, f(i))$ $(i \in \mathbb{Z}_p^{\times})$ and using Lagrange's formulas, any k users can compute f and deduce its constant term [95]. R.J. McEliece and D.V. Sarwate show that this scheme can be generalized using Reed-Solomon codes [77] for which a polynomial time decoding algorithm is known. Let $\{\alpha_1, \ldots, \alpha_n\}$ be the non-zero elements of the field $\mathbb{F}$ and $\mathcal{C}$ an $[n,k]$ RS code over $\mathbb{F}$, then each word $(m_0, \ldots, m_{k-1})$ can be encoded into the codeword $c = (c_1, \ldots, c_n)$ such that $c_i = m(\alpha_i)$ where $m(x) = \sum_{j=0}^{k-1} m_j x^j$ (Shamir's scheme corresponds to the case where $n+1$ is prime and $\alpha_i = i$). The secret to be shared is the information symbol m_0. Table 11 describes the protocol. When r users meet together, they

Table 11. A code based secret sharing scheme

Secret : $m_0 \in \mathbb{F}_q$

Secret sharing :

. Compute the codeword $c = (c_1, \ldots, c_n)$ from the information symbols $(m_0, \ldots, m_{k-1})$, $(m_1, \ldots, m_{k-1})$ being randomly generated.
. Each user receives a pair (i, c_i).

Secret recovering :

. From $r(\geqslant k)$ pairs $(i_1, c_{i_1}), \ldots, (i_r, c_{i_r})$, build an n bits word c' such that $c'_i = c_i$ if $i \in \{i_1, \ldots, i_r\}$, $c'_i = 0$ otherwise.
. Use the erasure decoding algorithm to compute c and then m_0.

know r symbols (and their positions) of the whole codeword c. The remaining $n-r$ symbols are called *erasures*: simply replace them with 0 and they become special errors whose positions are known.

Remark 8. Notice that since the protocol is used over $\mathbb{F}_q$ we have $n = q-1$.

Proposition 3. *Reed-Solomon codes can polynomially decode n_e errors and n_ε erasures provided that $2n_e + n_\varepsilon < n - k + 1$.*

In our case, we have $n_e = 0$ and $n_\varepsilon = n - r$, thus if $r \geqslant k$, every assembly of r users can compute the whole codeword c using the decoding algorithm of RS codes and deduce m_0.

Remark 9. Notice that $m_0 = -\sum_{i=1}^{n} c_i$. Moreover the encoding of RS code can be done in an efficient way without the generator matrix of the code. Hence in this protocol, there is no need to store this matrix.

This protocol has a non-negligible advantage as compared to Shamir's scheme. Suppose that a dishonest party want to denied access to the secret to legitimate users by tampering some of the pieces c_i (or being less paranoiac, just envisage that some c_i's have been tampered with some "natural" phenomena). Let t be the number of invalid c_i. Suppose r users meet together and t of them have corrupted pieces, the whole codeword c can be computed if $2t + n - r < n - k + 1$, i.e. $r \geqslant k + 2t$. Hence, if some pieces are damaged, it is still possible to retrieve the secret. On the other hand, since there are n users, the opponent has to alter more than $\lfloor (n-k)/2 \rfloor$ pieces to ensure that the secret be inaccessible.

A more general situation is to specify some users who have greater privileges of access to the secret than to others. An access structure consists of all subsets of participants that should be able to compute the secret but that contains no proper subset that also could determine it. J.L. Massey proposed to treat this problem using linear codes and the notion of "minimal" codewords [68,3].

10 Conclusion

While code based cryptosystems use only elementary operations over the two elements field, they were not really considered by cryptographic community because of the size of the public data. Since these last years, numerous works have been developed in order to enhance the performance of code based cryptography leading to realistic alternatives to number based theory schemes even in constrained environments such as smart cards or RFID tags. Nowadays code based cryptography has to be considered as a real alternative to number theory based cryptography especially since :

. despite several speedups and improvements, best cryptanalysis against the Syndrome Decoding problem is still exponential whereas it is subexponential for factoring,
. there does not exist a quantum algorithm which can polynomially solve the SD problem while Shor's algorithm can factor an integer N in $\mathcal{O}((\log N)^3)$ operations on a quantum computer.

Steganography

1 Introduction

Steganography (from greek *steganos*, or "covered", and *graphie*, or "writing") is the art and science of hidding a secret message within an ordinary message (the *cover-medium*) in such a way that no one, apart from the sender and intended recipient, even realizes there is a hidden message. While cryptography intends to make a message unreadable from a third party without hiding the secret communication, the aim of steganography is covert communication to hide the message from a third party. As an increasing amount of data is stored on computers and transmitted over networks, multimedia objects like image, audio and video files are today's most common cover-media.

Usually, the sender extracts from the cover-medium some of its components to construct a *cover-data* vector (for example the least significant bit of each byte of the cover medium). Then, the message is embedded into the cover-data to produce the *stego-data*. Finally, the cover-data is

replaced by the stego-data in the cover-medium, which gives the *stego-medium* communicated to the recipient. From the *stego-medium*, the recipient uses a recovering algorithm in order to extract the embedded message. The embedding and recovering algorithms form the *steganographic scheme* (or *stegoscheme*).

Only the sender and the receiver should be able to tell if the stego-medium carries an hidden message or not. This means that the stego-medium should be statistically indistinguishable from the cover-medium. Especially, it is of importance to embed the message while modifying as less components of the cover-data as possible.

2 Definitions, Properties

Definition 9 (Stegoscheme). *Let $\mathcal{A}$ a finite alphabet, $r, n \in \mathbb{N}$ such that $r < n$, $\mathbf{x} \in \mathcal{A}^n$ denote the* cover-data*, $\mathbf{m} \in \mathcal{A}^r$ denote the message to embed, and T be a strictly positive integer. A* stegoscheme *is defined by a pair of functions Ext and Emb such that:*

$$\begin{array}{ll} Emb : \mathbb{F}_2^n \times \mathbb{F}_2^r \longrightarrow \mathbb{F}_2^n & Ext(Emb(\mathbf{x}, \mathbf{m})) = \mathbf{m} \\ Ext \ : \mathbb{F}_2^n \longrightarrow \mathbb{F}_2^r & d(\mathbf{x}, Emb(\mathbf{x}, \mathbf{m})) \leq T \end{array}$$

where $d(.,.)$ denotes the Hamming distance over $\mathcal{A}^n$.

We focus in this paper on binary stegoscheme, i.e. $\mathcal{A} = \mathbb{F}_2$. The efficiency of a stegoscheme is usually evaluated through two quantities : the embedding efficiency and the relative payload.

Definition 10 (Embedding efficiency). *The* average embedding efficiency *of a stegoscheme, is usually defined by the ratio of the number of message symbols we can embed by the average number of symbols changed. We denote it by e.*

Definition 11 (Relative payload). *The* relative payload *of a stegoscheme, denoted by α, is the ratio of the number of message symbols we can embed by the number of (modifiable) symbols of covered data.*

3 LSB embedding

The simplest and most common steganographic algorithm uses LSB (Least Significant Bit) embedding. Let us assume that the cover-medium is an image composed of n pixels. The cover-data is the sequence $x_1, \ldots, x_n$ where x_i is the LSB of the ith pixel of the image. The message to embed is composed of n bits $m_1, \ldots, m_n$. The functions Ext and Emb are defined as :

$$\begin{array}{lll} Emb : \mathbb{F}_2^n \times \mathbb{F}_2^n & \longrightarrow & \mathbb{F}_2^n \\ \quad ((x_1, \ldots, x_n), (m_1, \ldots, m_n)) & \longmapsto & (m_1, \ldots, m_n) \\ Ext \ : \mathbb{F}_2^n & \longrightarrow & \mathbb{F}_2^n \\ \quad (y_1, \ldots, y_n) & \longmapsto & (y_1, \ldots, y_n) \end{array}$$

Hence, each bit of the cover-data conveys one bit of the message, and if the bits of the message are uniformally distributed (which should be the case if it has been encrypted before) then on average one bit over 2 is not modified in the cover-data, i.e. on average we modify only one bit to insert two bits of the message. Hence for this system : $\alpha = 1$ and $e = 2$.

Unfortunately, one can easily detect the presence of a secret message by looking at the image histogram. Let us consider each pixel as an integer, and denote by :

- $h[j]$ the number of pixels whose value is j in the cover-medium,
- $h_s[j]$ the number of pixels whose value is j in the stego-medium.

Notice that if a pixel is equal to $2i$ in the cover-medium and if the bits of the message to hide are uniformally distributed, then in the stego-medium, this same pixel is equal to $2i$ (with probability $1/2$) or $2i+1$ (with probability $1/2$), hence :

$$E(h_s[2i]) = \frac{h[2i] + h[2i+1]}{2}\,.$$

Similarly, a pixel whose value is $2i+1$ gives rise to a pixel equal to $2i+1$ (with probability $1/2$) or $2i$ (with probability $1/2$), hence :

$$E(h_s[2i+1]) = \frac{h[2i] + h[2i+1]}{2} = E(h_s[2i])\,.$$

Such a result shows that LSB embedding has a tendency to even out the histogram within each pair of bin representing a pair $(2i, 2i+1)$. This is the starting point of several powerful attacks against this scheme.

Another drawback of this scheme comes from the embedding efficiency. Let us suppose that the message we aim to hide contains $2n/3$ bits. It is obvious that the size of the message has no impact on the embedding efficiency for the LSB scheme, we will always (on average) modify 1 bit of the cover-data to insert two bits of the message.
Let us now consider the cover-data as a vector composed of $n/3$ blocks (x_0, x_1, x_2) and the message as vector of $n/2$ blocks (m_0, m_1). For each block, apply the following algorithm to compute the stego-data :

If $x_0 \oplus x_2 \neq m_0$ **and** $x_1 \oplus x_2 \neq m_1$ **then** flip x_2
elsif $x_0 \oplus x_2 \neq m_0$ **then** flip x_0
elsif $x_1 \oplus x_2 \neq m_1$ **then** flip x_1

In each block of the cover-data, the probability that one bit is changed is :

$$1 - \Pr(x_0 \oplus x_2 = m_0 \text{ and } x_1 \oplus x_2 = m_1) = \frac{3}{4}\,.$$

Thus the embedding efficiency of this scheme is :

$$e = \frac{2n/3}{(3/4)(n/3)} = \frac{8}{3} > 2\,.$$

Hence, less bits are modified in this scheme to insert the message as compared to the LSB scheme.

4 From LSB embedding to matrix embedding and coding theory

In the preceeding scheme, in order to extract the message m, consider the stego-data as $n/3$ blocks of three bits (y_0, y_1, y_2) and compute for each block :

$$m_0 = y_0 \oplus y_2,\ m_1 = y_1 \oplus y_2\,.$$

Let $y = \begin{pmatrix} y_0 \\ y_1 \\ y_2 \end{pmatrix}$ and $m = \begin{pmatrix} m_0 \\ m_1 \end{pmatrix}$, then for each block (y_0, y_1, y_2), the extraction algorithm computes $m = Hy$ where

$$H = \begin{pmatrix} 1\,0\,1 \\ 0\,1\,1 \end{pmatrix}\,,$$

is a matrix over $\mathbb{F}_2$. This method named matrix embedding has been proposed in 1998 by Crandall [33]. Notice that H is the parity check matrix of the $[3, 1]$ binary Hamming code.

Definition 12. *The binary Hamming code is a linear code whose columns of the parity check matrix are all the non zero vectors of* $\mathbb{F}_2^n$*. Code length is* $2^n - 1$*, dimension is* $2^n - 1 - n$ *and minimal distance is 3.*

To embed any message m in a fixed cover-data x we have to solve the equation $Hy = m$ and we substitute x by y. Since there exists $e \in \mathbb{F}_2^3$ such that $y = x + e$, the embedding process is equivalent to find e such that $He = m - Hx$.
With this method, we can embed 2 bits in 3 pixels and at most one bit is modified. Remember that a good steganographic scheme has to embed as much information as possible in the cover with as few changes as possible. Suppose now that we want to embed any sequence of p bits into a set of s fixed pixels allowing one change at most. What is the minimum value of s ? Since there are 2^p sequences of p bits and since changing at most 1 bit in s gives $s + 1$ new pixels, then we must have $s + 1 \geqslant 2^p$.

Theorem 1. *Let* H *be the parity check matrix of the* $[2^p - 1, 2^p - 1 - p]$ *Hamming code, let* $x \in \mathbb{F}_2^{2^p-1}$*, the system*

$$\begin{aligned} He &= m - Hx\,, \\ \omega(e) &\leqslant 1\,. \end{aligned}$$

where $\omega(e)$ *denotes the Hamming weight of* e*, always admits, for any* $m \in \mathbb{F}_2^p$*, a solution* $e \in \mathbb{F}_2^{2^p-1}$*.*

Proof. Since H contains all the non zero vectors of $\mathbb{F}_2^p$, if $m - Hx \neq 0$, then $m - Hx$ is one of the column of H.

From this, we can deduce the following theorem :

Theorem 2. *Let* H *be the parity check matrix of the* $[2^p - 1, 2^p - 1 - p]$ *Hamming code, the corresponding stegoscheme verifies :*

$$\alpha = \frac{p}{2^p - 1},\ e = \frac{p}{1 - 2^{-p}}\,.$$

Proof. H can be used to embed p bits in $2^p - 1$ pixels, hence the relative payload is $p/(2^p - 1)$. During the embedding process, the $2^p - 1$ bits are not modified with probability $1/2^p$, and exactly one bit is modified with probability $1 - 1/2^p$. The average number of bits modified is thus $0 \times 1/2^p + 1 \times (1 - 1/2^p)$.

From this theorem, we can see that embedding efficiency increases with p while relative payload decreases (see tab. 12). Hamming codes are well suited when the size of the message to embed

Table 12. Relative payload α_p and embedding efficiency e_p for stegoscheme defined from the $[2^p - 1, 2^p - 1 - p]$ Hamming code.

p	α_p	e_p
1	1	2
2	0.667	2.667
3	0.429	3.429
4	0.267	4.267
5	0.161	5.161
6	0.093	6.093
7	0.055	7.055
8	0.031	8.031
9	0.018	9.018

is a small fraction of the cover-data since many bits can be embedded with a single change. For

example, when the size of the message is 18% of the size of the cover-data, 9 bits of information are embedded with a single bit modification. Notice that when $p = 1$, matrix embedding leads to classical LSB embedding. Moreover, for any relative payload α, since one has to choose the largest α_p such that $\alpha_p \geqslant \alpha$ to embed a message using Hamming codes, this method boils down to LSB embedding when $\alpha > 2/3$.

Let us now consider a random binary linear $[n, k]$ code $\mathcal{C}$ and let $x \in \mathbb{F}_2^n$ be a stego-data. To build a stegoscheme from $\mathcal{C}$, we have to solve the following system :

$\forall m \in \mathbb{F}_2^{n-k}$, find $e \in \mathbb{F}_2^n$, such that :

$$\begin{array}{l} He \ = m - Hx\,, \\ \omega(e) \leqslant T\,. \end{array} \tag{9}$$

where H is a $(n-k, n)$ parity check matrix of $\mathcal{C}$, and T must be as "small" as possible in order to minimize the number of changes in x. Hence, for any message m, we have to solve an instance of the well-known SD problem which is NP-complete. In other words, for general linear codes, computing the vector e is a problem whose complexity will exponentially increase with n.

Now for any code $\mathcal{C}$, we have to answer to the following questions :

1. What is the maximum number of changes needed to embed a message m ?
2. What is the relative payload ?
3. What is the embedding efficiency ?

As we are going to show, all these values are well determined by the parameters of the code. The first problem is to determine for a given code $\mathcal{C}$, what is the maximal number of changes needed to embed any message m. In other words, we need an upperbound on T. Let us denote by R the covering radius of $\mathcal{C}$ which is determined by the most distant point y from the code, i.e. :

$$R = \max_{y \in \mathbb{F}_2^n} d(y, \mathcal{C})\,.$$

For any $s \in \mathbb{F}_2^{n-k}$, let $C(s) = \{e \in \mathbb{F}_2^n,\ He = s\}$. This set has 2^{n-k} members.

Definition 13. *A coset leader e_s for s is a member of $C(s)$ with the smallest Hamming weight.*

Proposition 4. *The Hamming weight of any coset leader is at most R.*

Proof. Let $z \in \mathbb{F}_2^n$, and $s = Hz$. From elementary linear algebra, $C(s) = \{x \in \mathbb{F}_2^n \mid x = z - c, c \in \mathcal{C}\}$. Let e_s be a coset leader,

$$R = \max_{y \in \mathbb{F}_2^n} d(y, \mathcal{C}) \geqslant d(z, \mathcal{C}) = \min_{c \in \mathcal{C}} \omega(z - c) = \omega(e_s)\,.$$

The minimum number of changes in the stego-data is obtained when the solution e of the problem (9) is a coset leader of $C(m - Hx)$. Hence, this problem always admits a solution $e \in \mathbb{F}_2^n$ such that $\omega(e) \leqslant R$.

Theorem 3 (Matrix embedding theorem). *A stegoscheme defined from an $[n, k]$ binary code $\mathcal{C}$ whose covering radius is R can embed $n - k$ bits in n pixels by making at most R changes. The relative payload is $(n-k)/n$ and the embedding efficiency is $(n-k)/R_{\mathcal{C}}$ where :*

$$R_{\mathcal{C}} = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} d(x, \mathcal{C})\,,$$

is the average distance to the code.

Proof. As already mentionned, the bound on the number of changes comes from the property that the Hamming weight of any coset leader is bounded by the covering radius R of the code. Next, by definition, we have $\alpha = (n-k)/n$. Now, let us suppose that the messages to embed are uniformally distributed, so that $m - Hx$ is uniformally ditributed in $\mathbb{F}_2^{n-k}$, to find the average number of changes, we thus have to compute the expected weight of a coset leader :

$$\frac{1}{2^{n-k}} \sum_{s \in \mathbb{F}_2^{n-k}} \omega(e_s) = \frac{1}{2^n} \sum_{s \in \mathbb{F}_2^{n-k}} 2^k \omega(e_s)\,.$$

Let $s \in \mathbb{F}_2^{n-k}$, from proposition 4, for any $x \in C(s)$, $d(x, \mathcal{C}) = \omega(e_s)$, hence :

$$\frac{1}{2^n} \sum_{s \in \mathbb{F}_2^{n-k}} 2^k \omega(e_s) = \frac{1}{2^n} \sum_{s \in \mathbb{F}_2^{n-k}} \sum_{x \in C(s)} d(x, \mathcal{C}) = \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} d(x, \mathcal{C})\,.$$

since $\cup_{s \in \mathbb{F}_2^{n-k}} C(s) = \mathbb{F}_2^n$.

To end this section we will give (without proofs, see [51]) asymptotic bounds on optimal matrix embedding schemes when embedding into cover-medium containing n pixels:

Proposition 5. *Let $\mathcal{H}_2(x)$ be the binary entropy function defined by :*

$$\mathcal{H}_2(x) = -x \log_2(x) - (1-x) \log_2(1-x)\,,$$

and $\mathcal{H}_2^{-1}()$ be its inverse function, then :

1. *The maximal number of bits which can be embedded making at most R changes is $n\mathcal{H}_2(R/n)$.*
2. *The average number of embedding changes to embed m bits is $n\mathcal{H}_2^{-1}(m/n)$.*
3. *The maximal embedding efficiency to embed m bits is $\frac{m/n}{\mathcal{H}_2^{-1}(m/n)}$.*

Last property can be generalized to obtain :

Proposition 6 (Sphere-covering bound [49]). *For any binary stegoscheme,*

$$e \leq \frac{\alpha}{\mathcal{H}_2^{-1}(\alpha)},$$

where α is the relative payload associated to the stegoscheme.

Remark 10. These bounds are still valid for q-ary codes using the q-entropy function :

$$H_q(x) = x \log_2(q-1) - x \log_2(x) - (1-x) \log_2(1-x)\,.$$

5 Wet paper codes

Usually, the sender does not use all pixels of the image to embed a message m. He may select part of the image where embedding changes will be more difficult to detect. The set of pixels which can be modified is called the *selection channel*. Most of the time, the selection channel is unknown to the receiver, he may even not know the selection rules used by the sender, we then call it a *non-shared selection channel*.
Wet paper codes, introduced in [50], have been designed to tackle the non-shared selection channel context. The idea is to consider that the cover-medium has been altered (like a sheet of paper) by rain. Hence a subset $\mathcal{W}$ of the components are "wet" and cannot be changed. Only a subset $\mathcal{D}$ of components (the "dry" components) can be modified to embed the message. During the transmission, the cover medium dries out and the receiver cannot determine $\mathcal{D}$ and $\mathcal{W}$.

Let $\mathcal{C}$ be an $[n, k]$ linear binary code, $\mathcal{D} \subset \{1, \ldots, n\}$, $\mathcal{W} = \{1, \ldots, n\} \setminus \mathcal{D}$, to build a stegoscheme

for the non-shared selection channel defined by $\mathcal{D}$, we have to solve the following problem :

Let $x \in \mathbb{F}_2^n$, $\forall m \in \mathbb{F}_2^{n-k}$, find $e \in \mathbb{F}_2^n$, such that :

$$\begin{aligned} He &= m - Hx\,, \\ e_i &= 0\,,\ \forall i \in \mathcal{W}\,. \end{aligned} \tag{10}$$

Notice that in this context, we do not seek for a word of minimum weight, but for a word e whose support is contained in $\mathcal{D}$. Let $H^{\mathcal{D}}$ denote the matrix composed of the columns of H whose index is in $\mathcal{D}$, then (10) is equivalent to :

Let $x \in \mathbb{F}_2^n$, $\forall m \in \mathbb{F}_2^{n-k}$, find $\tilde{e} \in \mathbb{F}_2^{\#\mathcal{D}}$, such that :

$$H^{\mathcal{D}}\tilde{e} = m - Hx\,. \tag{11}$$

The problem is that $H^{\mathcal{D}}$ depends on $\mathcal{D}$, that in turn depends on the cover object, hence even if H comes from some structure code for which the computation of a coset leader is easy, the sender cannot always deduce nice properties on $H^{\mathcal{D}}$. In particular, this means that trying to choose $\tilde{e}$ as a coset leader will constitute a much harder task than computing an arbitrary coset member.

Proposition 7 ([81]). *Problem (10) has a solution if and only the matrix $G_{\mathcal{W}}$ is of full rank, where $G_{\mathcal{W}}$ is the projection over $\mathcal{W}$ of the columns of a generator matrix G of the code $\mathcal{C}$.*

Proof. Let us denote by $\pi_{\mathcal{W}}$ the projection over the set $\mathcal{W}$. Let $x \in \mathbb{F}_2^n$, notice that (10) has a solution, if and only if for any $m \in \mathbb{F}_2^{n-k}$, $\pi_{\mathcal{W}}(x) \in \pi_{\mathcal{W}}(C(m))$, where $C(m) = \{z \in \mathbb{F}_2^n \mid Hz = m\}$. Now, for any m,

$$\#\pi_{\mathcal{W}}(C(m)) = \#\pi_{\mathcal{W}}(\mathcal{C}) = 2^{\mathrm{rank}(G_{\mathcal{W}})}\,,$$

since $C(m) = z + \mathcal{C}$, where z satisfies $Hz = m$. For any x, we must have $\pi_{\mathcal{W}}(x) \in \pi_{\mathcal{W}}(C(m))$, it means that $\pi_{\mathcal{W}}(\mathbb{F}_2^n) \subset \pi_{\mathcal{W}}(C(m))$, hence $\mathrm{rank}(G_{\mathcal{W}}) = \#\mathcal{W}$ (notice that $\#\mathcal{W} \leqslant k$ since we need to embed $n-k$ symbols in $\#\mathcal{D}$ dry symbols).

Proposition 8 ([81]). *$G_{\mathcal{W}}$ is full rank iff there is no word in $\mathcal{C}^\perp$ with support contained in $\mathcal{W}$.*

Proof. Can be easily deduced from the fact that there exists a word of weight δ in $\mathcal{C}^\perp$ iff there are δ linear dependent columns in G.

Proposition 9 ([81]). *Problem (10) has a solution for any $\mathcal{W}$ iff $\#\mathcal{W} < d_{\min}(\mathcal{C}^\perp)$ and in this case the number of solutions is exactly $q^{k-\#\mathcal{W}}$.*

Proof. If $\#\mathcal{W} < d_{\min}(\mathcal{C}^\perp)$ then no codeword of $\mathcal{C}^\perp$ has its support contained in $\mathcal{W}$ hence, from proposition 7 and 8, problem (10) has a solution. Conversely, suppose that problem (10) has a solution for any $\mathcal{W}$ and that $\#\mathcal{W} \geqslant d_{\min}(\mathcal{C}^\perp)$. Choose a set $\mathcal{W}$ and a word c of $C^\perp$ such that its support be contained in $\mathcal{W}$ then, from propostion 8, $\mathrm{rank}(G_{\mathcal{W}}) < \#\mathcal{W}$ which is a contradiction with proposition 7. Last, when $\mathrm{rank}(G_{\mathcal{W}}) = \#\mathcal{W}$, the number of solutions is $\#\mathcal{C}/\#\pi_{\mathcal{W}}(\mathcal{C}) = q^{k-\#\mathcal{W}}$.

From these propositions, we deduce that for a general $[n,k]$ code $\mathcal{C}$, $n-k$ symbols can be embed in a cover medium if there are strictly less than $d^\perp$ wet positions. As an example, using the binary Hamming code, p bits can be embed in $2^p - 1$ bits, if there are at most $2^{p-1} - 1$ wet positions.

Remark 11. A more general result states that, for n large enough, the number of dry symbols needed on average to transmit k informations symbols is roughly equal to k [81].

6 The $\varepsilon + 1$ matrix embedding scheme

In this section we describe how to use wet paper codes to transform an optimal binary matrix embedding scheme into an optimal ternary matrix embedding scheme. Let us suppose that we have a binary code $\mathcal{C}$ with embedding efficiency equal to ε, i.e. k bits can be embed in n bits by making on average k/ε changes. Let $(x_1, \ldots, x_n)$ be the cover-data obtained by taking the LSB of the n pixels of the image. Let us denote by $\mathcal{D} \subset \{1, \ldots, n\}$ the indices of the modified pixels during the embedding process, and let $\mathcal{W} = \{1, \ldots, n\} \setminus \mathcal{D}$. When the sender flips the last bit of the pixel p_i, $i \in \mathcal{D}$, he also adjusts the second LSB of p_i to insert one more bit of information. Here is the description of the embedding process :

1. Let m a message of length k, x the cover-data, find e such that $H(x + e) = m$. Let t be the Hamming weight of e (on average $t \simeq k/\varepsilon$).
2. Let $\tilde{m}$ a message of lenght t and $\tilde{x}$ the cover-data computed from the second LSB of the cover-medium. Find $\tilde{e}$ such that $\tilde{H}(\tilde{x} + \tilde{e}) = \tilde{m}$ and $\tilde{e}_i = 0$ for $i \in \mathcal{W}$ (where $\tilde{H}$ is obtained from the t first rows of H).

The value t must be communicated to the receiver, a small portion of the cover image can be used to embed this value.

Notice that instead of flipping a bit (or adding 1 if the bit is even and -1 is the bit is odd) , we now modify a pixel by adding +1 or -1 regardless its parity. On average, $k + k/\epsilon$ bits are embedded making k/ε modifications, the embedding efficiency is then :

$$\frac{k + k/\varepsilon}{k/\varepsilon} = \varepsilon + 1 \,.$$

From proposition 5, if the binary stegoscheme is optimal than the maximal number of bits which can be embed making at most R changes is $n\mathcal{H}_2(R/n)$. Using this scheme with wet paper trick, we can embed at most $n\mathcal{H}_2(R/n) + R$ bits. Now,

$$n\mathcal{H}_2(R/n) + R = n(\mathcal{H}_2(R/n) + R/n) = n(\mathcal{H}_2(R/n) + R/n \log_2(3-1)) = nH_3(R/n)\,,$$

which is the maximal number of bits that can be embed using an optimal ternary stegoscheme (see remark 10).

The first practical steganographic scheme which incorporates the matrix embedding mechanism is the F5 algorithm [110]. A good starting point on Steganography and matrix embedding is [51]. In [82], steganography is described from a coding theory point of view and numerous bibliographical notes are given about the study of some well known codes in this context (Hamming, Golay, BCH, Reed-Solomon, $\mathbb{Z}_4$ linear codes).

References

1. C.M. Adams and H. Meijer. Security-related comments regarding mceliece's public-key cryptosystem. *IEEE Trans. Inform. Theory.*, 35:454–455, 1989.
2. C. Aguilar Melchor, P.-L. Cayrel, and P. Gaborit. A new efficient threshold ring signature scheme based on coding theory. In *Proceedings of the second international workshop on Post-quantum cryptography - PQCrypto'2008*, volume 5299 of *LNCS*, pages 1–16, 2008.
3. Alexei E. Ashikhmin and Alexander Barg. Minimal vectors in linear codes. *IEEE Transactions on Information Theory*, 44(5):2010–2017, 1998.
4. Daniel Augot, Morgan Barbier, and Alain Couvreur. List-decoding of binary goppa codes up to the binary johnson bound. In *IEEE, ITW'11*, pages 229 – 233, October 2011.
5. Daniel Augot, Matthieu Finiasz, Philippe Gaborit, Stphane Manuel, and Nicolas Sendrier. Sha-3 proposal: Fsb. Submission to NIST, 2008.

6. Daniel Augot, Matthieu Finiasz, and Nicolas Sendrier. A family of fast syndrome based cryptographic hash functions. In Ed Dawson and Serge Vaudenay, editors, *Mycrypt*, volume 3715 of *Lecture Notes in Computer Science*, pages 64–83. Springer, 2005.
7. Marco Baldi and Franco Chiaraluce. Cryptanalysis of a new instance of mceliece cryptosystem based on qc-ldpc codes. In *Information Theory, 2007. ISIT 2007. IEEE International Symposium on*, pages 2591 –2595, june 2007.
8. S. Barg. Some new NP-complete coding problems. *Probl. Peredachi Inf.*, 30:23–28, 1994.
9. P. Barreto, P.-L. Cayrel, R. Misoczki, and R. Niebuhr. Quasi dyadic cfs signature scheme. In *Inscrypt*, Lecture Notes in Computer Science. Springer, 2010.
10. Anja Becker, Antoine Joux, Alexander May, and Alexander Meurer. Decoding random binary linear codes in 2 n/20: How 1 + 1 = 0 improves information set decoding. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology - EUROCRYPT 2012 - 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15-19, 2012. Proceedings*, volume 7237 of *Lecture Notes in Computer Science*, pages 520–536. Springer, 2012.
11. Mihir Bellare, Chanathip Namprempre, and Gregory Neven. Security proofs for identity-based identification and signature schemes. In *EUROCRYPT*, volume 3027 of *Lecture Notes in Computer Science*, pages 268–286. Springer, 2004.
12. Thierry Berger, Pierre-Louis Cayrel, Philippe Gaborit, and Ayoub Otmani. Reducing key length of the mceliece cryptosystem. In Bart Preneel, editor, *Progress in Cryptology, AFRICACRYPT 09*, volume 5580 of *Lecture Notes in Computer Science*, pages 77–97. Springer Berlin / Heidelberg, 2009.
13. E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.
14. Daniel Bernstein, Tanja Lange, Christiane Peters, and Peter Schwabe. Really fast syndrome-based hashing. In Abderrahmane Nitaj and David Pointcheval, editors, *Progress in Cryptology, AFRICACRYPT 2011*, volume 6737 of *Lecture Notes in Computer Science*, pages 134–152. Springer Berlin / Heidelberg, 2011.
15. Daniel J. Bernstein. Grover vs. mceliece. http://cr.yp.to/papers.html, 2008.
16. Daniel J. Bernstein. List decoding for binary goppa codes. http://cr.yp.to/codes/goppalist-20081107.pdf, 2008.
17. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Attacking and defending the mceliece cryptosystem. In *Proceedings of the 2nd International Workshop on Post-Quantum Cryptography*, PQCrypto '08, pages 31–46, Berlin, Heidelberg, 2008. Springer-Verlag.
18. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 743–760. Springer, 2011.
19. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Smaller decoding exponents: Ball-collision decoding. In Phillip Rogaway, editor, *Advances in Cryptology - CRYPTO 2011 - 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011. Proceedings*, volume 6841 of *Lecture Notes in Computer Science*, pages 743–760. Springer, 2011.
20. Daniel J. Bernstein, Tanja Lange, and Christiane Peters. Wild mceliece. In *Selected Areas in Cryptography*, volume 6544 of *Lecture Notes in Computer Science*, pages 143–158. Springer, 2011.
21. J.-L. Beuchat, N. Sendrier, A. Tisserand, and G. Villard. Fpga implementation of a recently published signature scheme. Tech. Rep. 5158, Inria, March 2004.
22. Bhaskar Biswas and Nicolas Sendrier. Mceliece cryptosystem implementation: Theory and practice. In Johannes Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography*, volume 5299 of *Lecture Notes in Computer Science*, pages 47–62. Springer Berlin / Heidelberg, 2008.
23. E.F. Brickell and A.M. Odlyzko. Cryptanalysis: A survey of recent results. *Comtemporary Cryptology - the Science of Information Integrity*, pages 501–540, 1992.
24. Anne Canteaut. *Attaques de cryptosystèmes à mots de poids faible et construction de fonctions t-résilientes*. PhD thesis, Université Paris VI, 1996.
25. Anne Canteaut and Florent Chabaud. A new algorithm for finding minimum-weight words in a linear code: Application to mceliece's cryptosystem and to narrow-sense bch codes of length 511. *IEEE Transactions on Information Theory*, 44(1):367–378, 1998.
26. Anne Canteaut and Nicolas Sendrier. Cryptanalysis of the original mceliece cryptosystem. In *Advances in Cryptology - ASIACRYPT '98*, volume 1514 of *Lecture Notes in Computer Science*, pages 187–199. Springer-Verlag, 1998.
27. P.-L. Cayrel and P. Dusart. Mceliece/niederreiter pkc: sensitivity to fault injection. In *International Workshop on Future Engineering, Applications and Services FEAS*, 2010.

28. Pierre-Louis Cayrel, Sidi Mohamed El Yousfi Alaoui, Gerhard Hoffmann, and Pascal Véron. An improved threshold ring signature scheme based on error correcting codes. In *International Workshop on the Arithmetic of Finite Fields*, volume 7369 of *LNCS*, pages 45–63, Bochum, Germany, July 2012. Springer Verlag.
29. Pierre-Louis Cayrel, Philippe Gaborit, and Marc Girault. Identity-based identification and signature schemes using correcting codes. In D. Augot, N. Sendrier, and J.-P. Tillich, editors, *WCC 2007*. INRIA, 2007.
30. Pierre-Louis Cayrel, Pascal Véron, and Sidi Mohamed El Yousfi Alaoui. A zero knowledge identification scheme based on the q-ary SD problem. In *Selected Areas in Cryptography*, volume 6544 of *LNCS*, pages 171–186, Waterloo, Canada, 2011. Springer.
31. H. Chabanne and B. Courteau. Application de la méthode de décodage itérative d'omura à la cryptanalyse du système de mc eliece. Rapport de Recherche 122, Université de Sherbrooke, October 1993.
32. Nicolas Courtois, Matthieu Finiasz, and Nicolas Sendrier. How to achieve a mceliece-based digital signature scheme. In *Advances in Cryptology - ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 157–174. Springer-Verlag, 2001.
33. R. Crandall. Some notes on steganography, 1998. Posted on the steganography mailing list.
34. Léonard Dallot. Towards a concrete security proof of courtois, finiasz and sendrier signature scheme. In Stefan Lucks, Ahmad-Reza Sadeghi, and Christopher Wolf, editors, *WEWoRC*, volume 4945 of *Lecture Notes in Computer Science*, pages 65–77. Springer, 2008.
35. Léonard Dallot and Damien Vergnaud. Provably secure code-based threshold ring signatures. In Matthew Parker, editor, *Cryptography and Coding*, volume 5921 of *Lecture Notes in Computer Science*, pages 222–235. Springer Berlin / Heidelberg, 2009.
36. Ivan Damgård. A design principle for hash functions. In *Advances in Cryptology - CRYPTO '89*, volume 435 of *Lecture Notes in Computer Science*, pages 416–427. Springer-Verlag, 1990.
37. Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.
38. Rafael Dowsley, Jrn Mller-Quade, and Anderson Nascimento. A cca2 secure public key encryption scheme based on the mceliece assumptions in the standard model. In Marc Fischlin, editor, *Topics in Cryptology CT-RSA 2009*, volume 5473 of *Lecture Notes in Computer Science*, pages 240–251. Springer Berlin / Heidelberg, 2009.
39. Thomas Eisenbarth, Tim Gneysu, Stefan Heyse, and Christof Paar. Microeliece: Mceliece for embedded devices. In Christophe Clavier and Kris Gaj, editors, *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 49–64. Springer Berlin / Heidelberg, 2009.
40. Sidi Mohamed El Yousfi Alaoui, Özgür Dagdelen, Pascal Véron, David Galindo, and Pierre-Louis Cayrel. Extended Security Arguments for Signature Schemes. In *Africacrypt 2012*, volume 7374 of *LNCS*, pages 19–34, Ifrane, Morocco, July 2012. Springer Verlag.
41. Jean-Charles Faugère, Ayoub Otmani, Ludovic Perret, and Jean-Pierre Tillich. A distinguisher for high rate mceliece cryptosystems. IACR Eprint archive, 2010/331, 2010.
42. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In *Advances in Cryptology - CRYPTO '86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer-Verlag, 1987.
43. Matthieu Finiasz. *Nouvelles constructions utilisant des codes correcteurs d'erreurs en cryptographie à clé publique*. PhD thesis, Ecole Polytechnique, 2004.
44. Matthieu Finiasz. Parallel-cfs: strengthening the cfs mceliece-based signature scheme. In *Proceedings of the 17th international conference on Selected areas in cryptography*, SAC'10, pages 159–170, Berlin, Heidelberg, 2011. Springer-Verlag.
45. Matthieu Finiasz, Philippe Gaborit, and Nicolas Sendrier. Improved fast syndrome based cryptographic hash function. In *ECRYPT Hash Workshop 2007*, 2007.
46. Matthieu Finiasz and Nicolas Sendrier. Security bounds for the design of code-based cryptosystems. In Mitsuru Matsui, editor, *Advances in Cryptology ASIACRYPT 2009*, volume 5912 of *Lecture Notes in Computer Science*, pages 88–105. Springer Berlin / Heidelberg, 2009.
47. Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Advances in Cryptology - EUROCRYPT '96*, volume 1070 of *Lecture Notes in Computer Science*, pages 245–255. Springer-Verlag, 1996.
48. Pierre-Alain Fouque and Gaëtan Leurent. Cryptanalysis of a hash function based on quasi-cyclic codes. In Tal Malkin, editor, *Topics in Cryptology - CT RSA 2008*, volume 4964 of *Lecture Notes in Computer Science*, pages 19–35. Springer Berlin / Heidelberg, 2008.

49. J. Fridrich. Asymptotic behavior of the ZZW embedding construction. *IEEE Transactions on Information Forensics and Security*, 4(1):151–153, 2009.
50. J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal. Writing on wet paper. *IEEE Trans. on Signal Processing*, 53(10):3923 – 3935, October 2005.
51. Jessica Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, New York, NY, USA, 1st edition, 2009.
52. Philippe Gaborit. Shorter keys for code based cryptography. In *Proceeedings of WCC'05*, pages 81–90, 2005.
53. Philippe Gaborit and Marc Girault. Lightweight code-based identification and signature. In *Proceeedings of ISIT'07*, 2007.
54. Philippe Gaborit, Cedric Laudauroux, and Nicolas Sendrier. Synd : a fast code-based stream cipher with a security reduction. In *Proceeedings of ISIT'07*, 2007.
55. Philippe Gaborit and G. Zémor. Asymptotic improvement of the gilbert-varshamov bound for linear codes. In *Proceeedings of ISIT'06*, pages 287–291, 2006.
56. Michael R. Garey and David S. Johnson. *Computers and Intractability, A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, New York, 1979.
57. Valerie Gauthier Umana and Gregor Leander. Practical key recovery attacks on two mceliece variants. IACR Eprint archive, 2009/509, 2009.
58. J. K. Gibson. Equivalent goppa codes and trapdoors to mceliece's public key cryptosystem. In *Advances in Cryptology - EUROCRYPT '91*, volume 547 of *Lecture Notes in Computer Science*, pages 517–521. Springer-Verlag, 1991.
59. Marc Girault. A (non-practical) three-pass identification protocol using coding theory. In *Advances in Cryptology, Auscrypt'90*, volume 453 of *Lecture Notes in Computer Science*, pages 265–272. Springer-Verlag, 1990.
60. Marc Girault and Jacques Stern. On the length of cryptographic hash-values used in identification schemes. In *Advances in Cryptology - CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 202–215. Springer-Verlag, 1994.
61. S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof systems. *SIAM, Journal of Computing*, 18:186–208, 1989.
62. V. D. Goppa. A new class of linear error correcting codes. *Probl. Pered. Inform.*, pages 24–30, 1970.
63. Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. ACM.
64. Lov K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Phys. Rev. Lett.*, 79(2):325–328, Jul 1997.
65. Sami Harari. A new authentication algorithm. In *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 91–105. Springer-Verlag, 1988.
66. Stefan Heyse, Amir Moradi, and Christof Paar. Practical power analysis attacks on software implementations of mceliece. In Nicolas Sendrier, editor, *Post-Quantum Cryptography*, volume 6061 of *Lecture Notes in Computer Science*, pages 108–125. Springer Berlin / Heidelberg, 2010.
67. R. Housley. Using advanced encryption standard (aes) counter mode with ipsec encapsulating security payload (esp). RFC 3686, Network Working Group, January 2004.
68. Massey J.-L. Minimal codewords and secret sharing. In *6th Joint Swedish-Russian Workshop on Information Theory*, pages 276–279, 1993.
69. Thomas Johansson and Fredrik Jönsson. On the complexity of some cryptographic problems based on the general decoding problem. *IEEE Transactions on Information Theory*, 48(10):2669–2678, 2002.
70. Pil Joong Lee and Ernest F. Brickell. An observation on the security of mceliece's public-key cryptosystem. In *Advances in Cryptology - EUROCRYPT '88*, volume 330 of *Lecture Notes in Computer Science*, pages 275–280. Springer-Verlag, 1988.
71. Jeffrey S. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Transactions on Information Theory*, 34(5):1354–1359, 1988.
72. Yuan Xing Li, Robert H. Deng, and Xin mei Wang. On the equivalence of mceliece's and niederreiter's public-key cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–, 1994.
73. Pierre Loidreau and Nicolas Sendrier. Weak keys in the mceliece public-key cryptosystem. *IEEE Transactions on Information Theory*, 47(3):1207–1211, 2001.
74. F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error-Correcting Code*. North-Holland, 1977.
75. Alexander May, Alexander Meurer, and Enrico Thomae. Decoding random linear codes in $o(2^{0.054n})$. In Dong Hoon Lee and Xiaoyun Wang, editors, *Advances in Cryptology - ASIACRYPT 2011 - 17th*

International Conference on the Theory and Application of Cryptology and Information Security, Seoul, South Korea, December 4-8, 2011. Proceedings, volume 7073 of *Lecture Notes in Computer Science*, pages 107–124. Springer, 2011.
76. R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *JPL DSN Progress Report*, pages 114–116, 1978.
77. R. J. McEliece and D. V. Sarwate. On sharing secrets and Reed-Solomon codes. *Communications of the ACM*, 24(9):583–584, 1981.
78. R.C. Merkle and M.E. Hellman. Hiding information and signatures in trapdoor knapsacks. *IEEE Trans. Inform. Theory*, 24:525–530, 1978.
79. Rafael Misoczki and Paulo Barreto. Compact mceliece keys from goppa codes. In Michael Jacobson, Vincent Rijmen, and Reihaneh Safavi-Naini, editors, *Selected Areas in Cryptography*, volume 5867 of *Lecture Notes in Computer Science*, pages 376–392. Springer Berlin / Heidelberg, 2009.
80. H. Molter, Marc Stttinger, Abdulhadi Shoufan, and Falko Strenzke. A simple power analysis attack on a mceliece cryptoprocessor. *Journal of Cryptographic Engineering*, 1:29–36, 2011.
81. C. Munuera and M. Barbier. Wet paper codes and the dual distance in steganography. *Advances in Mathematics of Communications. Vol. 6, number 3*, pages 237 – 285, August 2012.
82. Carlos Munuera. *Steganography from a coding theory point of view*, volume 8 of *Series on Coding Theory and Cryptology*. World Scientific Publishing Co Pte Ltd, 2013. 2013; 1.
83. H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems Control Inform. Theory*, 15(2):159–166, 1986.
84. Ryo Nojima, Hideki Imai, Kazukuni Kobara, and Kirill Morozov. Semantic security for the mceliece cryptosystem without random oracles. *Designs, Codes and Cryptography*, 49:289–305, 2008. 10.1007/s10623-008-9175-9.
85. Ayoub Otmani, Jean-Pierre Tillich, and Lonard Dallot. Cryptanalysis of two mceliece cryptosystems based on quasi-cyclic codes. *Mathematics in Computer Science*, 3:129–140, 2010.
86. Christiane Peters. Information-set decoding for linear codes over f_q. In Nicolas Sendrier, editor, *Post-Quantum Cryptography, Third International Workshop, PQCrypto 2010, Darmstadt, Germany, May 25-28, 2010. Proceedings*, volume 6061 of *Lecture Notes in Computer Science*, pages 81–94. Springer, 2010.
87. E. Prange. The use of information sets in decoding cyclic codes. *IRE Trans.*, IT-8:85–89, 1962.
88. Jean-Jacques Quisquater and Louis Guillou. How to explain zero-knowledge protocols to your children. In *Advances in Cryptology-Crypto '89*, volume 435 of *Lecture Notes in Computer Science*, pages 628–631. Springer-Verlag, 1990.
89. R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 26(1):96–99, 1983.
90. Markku-Juhani O. Saarinen. Linearization attacks against syndrome based hashes. In *Proceedings of the cryptology 8th international conference on Progress in cryptology*, INDOCRYPT'07, pages 1–9, Berlin, Heidelberg, 2007. Springer-Verlag.
91. N. Sendrier. Efficient generation of binary words of given weight. In *Cryptography and Coding - 5th IMA Conference*, volume 1025 of *Lecture Notes in Computer Science*, pages 184–187. Springer Verlag, 1995.
92. Nicolas Sendrier. On the structure of a randomly permuted concateneted code. In *EUROCODE '94*, pages 169–173. Inria, 1994.
93. Nicolas Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.
94. Adi Shamir. How to Share a Secret. *Communications of the ACM*, 22(11):612–613, November 1979.
95. Adi Shamir. Identity-based cryptosystems and signature schemes. In *CRYPTO'84*, pages 47–53, 1984.
96. Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 20–22, 1994.
97. Abdulhadi Shoufan, Falko Strenzke, H. Molter, and Marc Stttinger. A timing attack against patterson algorithm in the mceliece pkc. In Donghoon Lee and Seokhie Hong, editors, *Information, Security and Cryptology ICISC 2009*, volume 5984 of *Lecture Notes in Computer Science*, pages 161–175. Springer Berlin / Heidelberg, 2010.
98. V.M. Sidelnikov and S.O. Shestakov. On cryptosystems based on generalized reed-solomon codes. *Diskretnaya Math*, 4:57–63, 1992.
99. Jacques Stern. A method for finding codewords of small weight. In *Coding Theory and Applications*, volume 388 of *Lecture Notes in Computer Science*, pages 106–113. Springer-Verlag, 1988.

100. Jacques Stern. A new identification scheme based on syndrome decoding. In *Advances in Cryptology - CRYPTO '93*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer-Verlag, 1993.
101. Falko Strenzke. A smart card implementation of the mceliece pkc. In Pierangela Samarati, Michael Tunstall, Joachim Posegga, Konstantinos Markantonakis, and Damien Sauveron, editors, *Information Security Theory and Practices. Security and Privacy of Pervasive Systems and Smart Devices*, volume 6033 of *Lecture Notes in Computer Science*, pages 47–59. Springer Berlin / Heidelberg, 2010.
102. Falko Strenzke, Erik Tews, H. Molter, Raphael Overbeck, and Abdulhadi Shoufan. Side channels in the mceliece pkc. In Johannes Buchmann and Jintai Ding, editors, *Post-Quantum Cryptography*, volume 5299 of *Lecture Notes in Computer Science*, pages 216–229. Springer Berlin / Heidelberg, 2008.
103. Y. Sugiyama, M. Kasahara, S. Hirasawa, and T. Namekawa. Further results on goppa codes and their applications to constructing efficient binary codes. *IEEE Transactions on Information Theory*, 22:518–526, 1976.
104. Johan van Tilburg. On the mceliece public-key cryptosystem. In *Advances in Cryptology - CRYPTO '88*, volume 403 of *Lecture Notes in Computer Science*, pages 119–131. Springer-Verlag, 1988.
105. Pascal Véron. Cryptanalysis of harari's identification scheme. In *Cryptography and Coding, 5th IMA Conference.*, volume 1025 of *Lecture Notes in Computer Science*, pages 264–269. Springer-Verlag, 1995.
106. Pascal Véron. Improved identification schemes based on error-correcting codes. *Appl. Algebra Eng. Commun. Comput.*, 8(1):57–69, 1996.
107. Pascal Véron. Public key cryptography and coding theory. In I. Woungang, S. Misra, and S.C. Misra, editors, *Selected Topics in Information and Coding Theory*, volume 7. World Scientific Publications, March 2010.
108. D. Wagner. A generalized birthday problem. In *Advances in Cryptology - CRYPTO '02*, volume 2442 of *Lecture Notes in Computer Science*, pages 288–304. Springer-Verlag, 2002.
109. Andreas Westfeld. F5 - A steganographic algorithm. In Ira Moskowitz, editor, *Information Hiding*, volume 2137 of *LNCS*, pages 289–302. Springer, 2001.