SYND: a Fast Code-Based Stream Cipher with a Security Reduction

Philippe Gaborit
XLIM-DMI, Université de Limoges
123 av. Albert Thomas
87000, Limoges, France
gaborit@unilim.fr

Cedric Lauradoux
INRIA Rocquencourt, projet CODES
Domaine de Voluceau, B.P. 105
78153, Le Chesnay Cedex, France
Cedric.Lauradoux@inria.fr

Nicolas Sendrier
INRIA Rocquencourt, projet CODES
Domaine de Voluceau, B.P. 105
78153, Le Chesnay Cedex, France
Nicolas.Sendrier@inria.fr

***Abstract*— In this note we reconsider the code-based pseudo-random generator proposed by Fischer and Stern. This generator is proven as secure as the syndrome decoding problem but has two main drawbacks: it is slow (3000 bits/s) and a large size of memory is needed (88 kiloBytes). We propose a variation on the scheme which avoid them: the use of regular words speeds the system up and the use of quasi-cyclic codes allows a decrease of the memory requirements. We eventually obtain a generator as fast as AES in counter mode using only about 8000 bits of memory. We also give a more precise security reduction.**

I. Introduction

Pseudo-random generator are very important in cryptography and can be used for one-time-pad cryptosystems. One of the main desired figure of such pseudo-random generators (PRNG) is to be fast, at least as fast as the best block cipher scheme since it is possible by using the OFB mode to turn any block into a stream cipher.

Meanwhile another kind of property is also interesting for stream ciphers: proven stream ciphers, i.e. PRNG proven as secure as particular difficult problems. This problem has received much attention beginning in the 80's and a series of paper concluded that a necessary and sufficient condition for the existence of a PRNG is the existence of one way function [15]. One way functions are functions which are easy to compute but hard to invert.

The first construction of provably secure stream cipher is due to Blum and Micali [6] which relates the existence of a PRNG to modular exponentiation, later Blum, Blum and Shoub [5] proposed a system based on quadratic residuosity and Alexi,Chor, Goldreich and Schnorr proposed a system based on the RSA assumption. All the previous PRNG are related to number theory problems, the first PRNG related to a non number theoretic problem (the subset problem) was proposed by Impagliazzo and Naor [16], this construction was followed by a construction by Fischer-Stern based on syndrome decoding problem [16]. At last very recently Berbain, Gilbert and Patarin [3] proposed the system QUAD based on multivariate equations (MQ problem).

In practice all PRNG based on number theory are rather slow with a speed not exceeding a few thousand bits per second (a hundred time slower than AES). The fastest system with a security reduction is the recent QUAD system which has a speed of a few Megabits per second, but has a memory usage of 4 Mb. Even if this size is not an issue for today computers it may become one for low cost devices like smartcards or RFID.

The PRNG proposed by Fischer and Stern based on syndrome decoding seems more interesting at first sight since it uses only linear equations but with a weight constraint. Unfortunately this system has two main drawbacks: first, dealing with the set of words of given weight necessitates the use of a quadratic algorithm which slows the whole process considerably, second the matrix needed to evaluate the syndrome is very large. Eventually these drawbacks make it as slow as number theoretic based PRNG.

A solution to the first drawback is the use of regular words. Regular words were introduced by Augot, Finiasz and Sendrier in a hash function context [1], they are words of given weight w which have a fixed number of 1's in each sequenced subset of columns of fixed size. This set of word does not describe the whole set of words of given weight but is very easy to generate and permits to avoid the use of a quadratic algorithm to generate words of given weight. The second drawback is avoided by the use quasi-cyclic codes, it was recently showed by Gaborit and Zemor [12] that in terms of syndrome these codes behaved like random codes with a small constraint on the length.

Hence in this note we reconsider the Fischer-Stern PRNG by modifying several points: first we use regular words for words of higher weights that what consider Fischer and Stern, it permits both to generate them easily and to obtain a better rate, second we use quasi-cyclic codes. Eventually we obtain a PRNG as fast as AES (ie: hundred times faster than QUAD) with a memory requirement of only 8000 bits.

In term of security, following the proof of the QUAD and the Fischer-Stern system we give a reduction proof for our system and associated parameters.

The paper is organized as follows: Section 2 recalls basic facts and definition of code-based cryptography, Section 3 gives a description of our system, Section 4 deals with the accurate security proof and at last Section 5 proposes parameters, peformances and considers practical security of our scheme.

II. Code-based cryptography

In this section we recall basic facts about code-based cryptography. We refer to the work of Nicolas Sendrier in [21] for a more general context on these problems and to [19] for a general context on coding theory.

Though most of the statements below hold for larger alphabets, we will consider binary linear codes of length n and dimension k, we will denote $r = n - k$ the codimension.

A. Syndrome decoding

The fundamental primitive we will consider is the syndrome mapping

$$\begin{array}{rccc} f: & \mathbf{F}_2^n & \rightarrow & \mathbf{F}_2^r \\ & x & \mapsto & xH^T \end{array}$$

defined for any $r \times n$ matrix ($r < n$). When the defining set is restricted to the words of Hamming weight less than some $w > 0$ this linear mapping becomes one-way (in the cryptographic sense). Inverting f for an upper bounded weight is strongly related to decoding in the binary linear code of parity check matrix H. In fact decoding in a linear code is at least as hard as Syndrome Decoding:

Problem 1 (Syndrome Decoding - SD):

Instance: A binary $r \times n$ matrix H, a word s of $\mathbf{F}_2^r$ and an integer $w > 0$.

Question: Is there a word x in $\mathbf{F}_2^n$ of weight $\leq w$ such that $xH^T = s$?

The problem SD is NP-complete [4]. Indeed, this is only a worst-case complexity result. But, though no formal reduction is known [17], [14], decoding in a random linear code is believed to be hard in the average case.

In practice, decoding a general linear code has recieved a lot attention, both in cryptology and in coding theory (see [2] for a state of the art). All known algorithms have an exponential complexity. Several cryptosystems based on coding theory, including the famous McEliece encryption scheme [18], have been proposed ([21] for references).

B. Best known decoding attack

As far as cryptography is concerned, the best known decoding technique is "information set decoding". Many papers have been published on this topic, leading to Canteaut-Chabaud algorithm [7] which is the best known cryptanalysis so far.

Computing the work factor of the Canteaut-Chabaud algorithm requires the computation of the stationnary distribution of (several) Markov processes. Thus it is not given by a closed formula, however, for reasonnable parameters, decoding t errors in a binary linear code of length n and codimension r, requires about $(n/r)^t = 2^{t \log_2(n/r)}$ binary operations (up to a polynomial factor). Table I gives the binary work factor (WF) of Canteaut-Chabaud algorithm for various parameters.

n	$n-k$	t	$\log_2 WF$
1024	500	50	62
2048	330	30	87
8192	256	32	152
8192	512	32	131

TABLE I

Cost for finding a word of weight t in a binary (n, k) code

C. Regular words

Given a binary $r \times n$ matrix H, computing eH^T for a word e of length n and weight t is extremely fast, as it consists in xoring (adding $\mod 2$) t columns of H. One flaw of this mapping lies in the fact that most, if not all, information systems deal with binary data. Somehow we need to transform binary strings into words of constant weight and length. Solutions to that problem exist [10], [22], but are significantly more expensive than the syndrome mapping alone.

Regular words are much more convenient in an algorithmic point of view. The drawback is in the security reduction, which relies on a much less studied problem.

1) Definition: We consider binary words of length n and we divide the coordinates in t blocs of n/t positions. A binary *regular word* of length n and weigth t ((n, t)-regular word) has exactly one non zero coordinate in each of those blocs.

There are exactly $(n/t)^t$ such words, which is less than $\binom{n}{t}$ the number of words of length n and weight t. If n and t are chosen such that $n/t = 2^\ell$, then there is an easy mapping

$$\theta_{n,t} : \mathbf{F}_2^{\ell t} \rightarrow \mathbf{F}_2^n \tag{1}$$

such that the image of $\theta_{n,t}$ is exactly the set of (n, t)-regular words.

2) Security: The problem corresponding to SD restricted to regular words is still NP-complete [1]:

Problem 2 (Regular Syndrome Decoding - RSD):

Instance: An integer $w > 0$, a sequence of w binary $r \times n$ matrices H_i, $1 \leq i \leq w$, and a word s of $\mathbf{F}_2^r$.

Question: Is there a set of w columns, one in each H_i, adding to s?

Regular Syndrome Decoding is probably also difficult in the average case, though we do not have decades of research to assess that claim. The best known attack is the generalized birthday attack [23] as demonstrated in [9] for the cryptanalysis of the syndrome-based hash function. However, in the context of this paper, the instances we consider have only one solution and this attack becomes a "simple" birthday attack of complexity $n2^{(n-k)/2}$ for decoding t errors in a binary (n, k) code (note that this cost is independant of t).

Notice moreover that there is a reduction from RSD to SD. A black box which decodes regular errors of weight t can be used to correct random errors of weight t in the same code. The coordinates are permuted until the error pattern becomes regular. The number of call to the black box is the ratio between the number of words of weight t and the number

of regular words, that is $\binom{n}{t}(t/n)^t \approx t^t/t! \approx \sqrt{t/2\pi}\exp(t)$. So, decoding a regular error cannot be more than about $\exp(t)$ easier than decoding a random error of same weight. This reduction is probably not tight.

Now, in practice, besides this theoretical reduction, the best known attack remains the usual attack for searching for a word of a given weight. Notice that because of the special form of the regular word and since for our case on the average there exists only one such codeword, the work factor still remains approximatively $(n/r)^t$ even though t lies beyond the Gilbert-Varshamov bound.

D. Quasi-Cyclic random codes

In this part we recall basic facts about quasi-cyclic (QC) codes.

Definition 1: A code of length n is called *quasi-cyclic* of order s, for n a multiple of s if every cyclic shift of a codeword by s coordinates is again a codeword.
Remark 1: A cyclic code is a quasi-cyclic code with $s = 1$.
Remark 2: Equivalently QC codes of order s can be seen as codes invariant under the concatenation of s cyclic shifts of length $\frac{n}{s}$.

A particular class of QC codes is the class of QC codes obtained by concatenation of cyclic (circulant) codes. Circulant matrices of size $r \times r$ are given by a random first row and the $r-1$ next rows are obtained by cyclic shifts of the first row.

It is well known that in term of miminum distance, random codes are good (this is the Gilbert-Varshamov bound). Starting from a $[n, k, d]$ the result comes from the fact that since the code is random its dual code with a generator matrix of size $(n-k) \times n$ is also random and that the probability that a random element of F_2^n belongs to the code is $2^{-(n-k)}$. The Gilbert-Varshamov implies that there exists a code with minimum distance d such that:

$$\sum_{i=0}^{d-1} \binom{n}{i} < 2^{n-k}$$

For random QC codes such a result does not hold directly but if we accept a small constraint on the length, it is possible to get a similar result:

Consider a length r such that r is prime and such that 2 is primitive root of Z/rZ, then $x^r - 1 = (x+1)(1 + x + x^2 + \cdots + x^{r-1}$ and in that case the circulant matrix generated by any random word of odd weight is invertible. This ensures that basically one gets the same type of random properties in term of minimum weights than for linear codes. This result is showed by Chen, Peterson and Weldon in [8] (see also [19],p.507 and [12]). Notice that it is no known whether there exist an infinite number of such r (this is the Dirichlet Theorem), but such r are known to exist at least up to 10^{50} and for the cases we are interested in (small r under 1024) many are known.

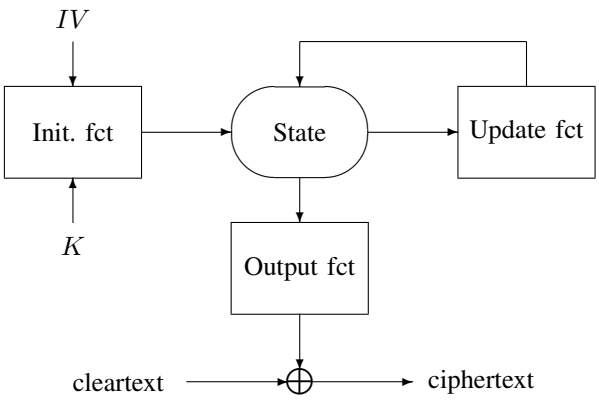
Hence if we start from a random QC code C $[n, n-r]$ given by its dual code of length $n = sr$ constructed as a concatenation of s random circulant matrices of length r such that 2 is a primitive root of Z/rZ we obtain that the probability that a random codeword belongs to C is close to 2^{-r} and hence such codes behave like purely random codes with the same lengths and dimension (in fact it is even a little better, see [12] for more details). In particular it means that such codes satisfy the Gilbert-Varshamov bound.

Notice that at the difference of purely random codes it is not known whether decoding a QC codes is NP-complete, meanwhile, as far as we know, all classical algorithms used to decode random codes are not more efficient for decoding random QC codes. The best algorithms to decode them are also algorithms based on information set decoding which more efficient is [7]. Moreover if there existed an algorithm which was able to decode up to the GV bound in polynomial or sub-exponential time it would certainly be a major breakthrough in coding theory. Hence we can be fairly confident that decoding QC codes up to the GV bound and higher has the same complexity than for random codes.

III. Description of the stream cipher

We consider a model of stream cipher functionning as a finite state machine with three procedures:

1) Key and initial value (IV) injection
2) State uptate
3) Output extraction



Let $|K|$ and $|IV|$ denote the key size and initial value size, let r_1 denote de state size and r_2 the output size. In order to fully describe the stream cipher we have to provide:

Initialization function	$g:$	$\mathbf{F}_2^{\|K\|} \times \mathbf{F}_2^{\|IV\|}$	$\rightarrow$	$\mathbf{F}_2^{r_1}$
Update function	$f_1:$	$\mathbf{F}_2^{r_1}$	$\rightarrow$	$\mathbf{F}_2^{r_1}$
Output function	$f_2:$	$\mathbf{F}_2^{r_1}$	$\rightarrow$	$\mathbf{F}_2^{r_2}$

The initialization function computes an initial state e_1 from the key K and the initial value IV. Starting from e_1 and at each time unit $i \geq 1$, the machine

- computes the new state $e_{i+1} = f_1(e_i)$
- extracts some bits from the current state $s_i = f_2(e_i)$

Let n and t be two integers chosen such that $n/t = 2^\ell$. The functions f_1 and f_2 will be syndrome mappings acting on (n,t)-regular words. From the definition of regular words, we have $f_1 : \mathbf{F}_2^{\ell t} \rightarrow \mathbf{F}_2^{r_1}$ and $f_2 : \mathbf{F}_2^{\ell t} \rightarrow \mathbf{F}_2^{r_2}$. This implies $r_1 = \ell t$. In practice we will also choose $r_1 = r_2$.

Let H_1 and H_2 denote two binary matrices of size respectively $r_1 \times n$ and $r_2 \times n$. Let $\theta_{n,t}$ be the mapping defined in (1) which transform a binary string of length ℓt into a (n,t)-regular word. We define for $i = 1, 2$

$$\begin{array}{rccc} f_i: & \mathbf{F}_2^{\ell t} & \rightarrow & \mathbf{F}_2^{r_i} \\ & x & \mapsto & \theta_{n,t}(x) H_i^T \end{array}$$

A. Update and output

The state will have a size of $\ell t = r_1$ bits. The update and output functions are respectively f_1 and f_2. We also define the mapping f defined for all $x \in \mathbf{F}_2^{\ell t}$ by $f(x) = f_1(x) \parallel f_2(x)$ ($\parallel$ denotes the concatenation). Note that f is also a syndrome mapping depending on a matrix H obtained by stacking H_1 and H_2. Though inverting this function f do not appear to be be meaningful in practical attacks, it is meaningful in the formal security reduction.

Note that matrix H (and thus matrices H_1 and H_2) can be obtained by concatenating circulant matrices. In that case, only the first row of H is needed to describe both f_1 and f_2.

B. Initialisation

For that step we will assume $r = r_1 = r_2$ the initialization phase will take as arguments a key of length $r/2$ and an initial value of length $r/2$. It consists of a three round Feistel scheme using successively the functions f_1, f_1 and f_2:

$$g(K, IV) = z \oplus f_2(y \oplus f_1(z)),$$

where $y = K \parallel IV$ and $z = f_1(y) \oplus y$. This initialization requires three function evaluation.

C. Security

A formal security reduction is discussed in the next section. In practice, the parameters we propose in the last section will be such that state recovery or key recovery by invertion of either f_1 or f_2 is difficult given the present state of art.

We assume $r_1 = r_2 = r$. As we mentionned, the best technique known for inverting the syndrome mapping for regular word is a birthday attack whose cost is about $r2^{r/2}$, decoding attack will have a higher cost. For instance, for $t = 32$, $n = 8192$ and $r = 256$ finding a given regular word costs about 2^{152} binary operations by decoding (see table I) and about 2^{136} by a birthday attack. So, for that particular set of parameters, no attack for recovering the state or the key is known that would be significantly better than exhaustive search.

IV. Security of the scheme

In this part we consider the security of our scheme. Fischer and Stern have showed in [11], how the security of their scheme could be related to the security of the syndrome decoding problem, but they did not give a precise reduction.

In this section, based on their proof and the proof of QUAD [3], we give a reduction for our construction and a more precise reduction.

The idea of the proof is to start from a difficult problem (syndrome decoding here) and to relate the existence of a distinguisher between the keystream produced and a random sequence, to the possibility of inverting a difficult problem.

All these proofs are related to the Goldreich-Levin [13] result linking the existence of pseudo-random generators to the existence of one-way function.

The reduction of [11] can be directly adapted to our case. In our case the security of the generator is reduced to the capacity of finding a regular word x of given weight w which syndrome xH^T is known where H can be either a random matrix or a quasi-cyclic matrix as described in Section 2.

The proof has three main steps which are linked together to obtain the main result.

In the following we outline the idea of the proof and only give the final reduction result. The details of the proofs are straightforward generalization of [11] and [3].

The first step relates the existence of distinguisher between an iterated sequence produced and a random string, to a distinguisher for only an basic iteration. Then the second step shows that if one has a distinguisher for a unique iteration then it is possible to convert it into an algorithm which predicts the result of the product by any vector. At least the final step relates the existence of such a predictor to the possibility of inverting the hard to invert basic function (here the syndrome decoding for regular word and random or quasi-cyclic random matrix).

We denote by T the maximum computation capacity allowed for the distinguisher (typically 2^{80}) to obtain an advantage ϵ.

The final reduction theorem is as follows:

Theorem 2: Let $L = \lambda n$ be the number of keystream bits produced by the scheme in time λT_S using λ iterations. Suppose there exists an algorithm A that distinguishes the L-bit keystraem sequence associated with a known randomly (or quasi-cyclic randomly) chosen $r \times n$ matrix H and an unknown randomly chosen initial internal state x (corresponding to a regular word) of weight w of $\{0,1\}^n$ from a random L-bits sequence in time T with advantage ϵ. Then there exists an algorithm C, which given the image xH^T of a randomly chosen regular word (unknown) x of weight w of $\{0,1\}^n$ by a randomly (or QC randomly) chosen $r \times n$ matrix H permits to recover x with a probability at least $\frac{\epsilon}{2^{3\lambda}}$ over all possible values of x and H in time upper bounded by T', such that

$$T' \approx \frac{2^7 n^2 \lambda^2}{\epsilon^2} T \tag{2}$$

V. Parameters and performance

A. Practical parameters

We suggest the parameters $n = 8192$, $t = 32$, $\ell = 8$, $n/t = 2^\ell = 256$ and $r = r_1 = r_2 = \ell t = 256$. The size of the state is 256 bits. Key and initial value both have 128 bits.
Remark: this case corresponds to the case when the matrix H is randomly chosen. It also possible to choose parameters (for a similar computational cost) such that r is a prime and 2 is primitive root of Z/rZ (like $r = 509$ or $r = 523$) in order to

get the random-like properties of QC matrices, but we do not develop it, in this short version.

We give in Table II different type of parameters besides the set of parameters suggested (in bold). As we can see, we obtain a speed and a security comparable to the AES. The program computing our stream cipher is extremely small but requires a lot of data (matrix H has a size of 512 kilobytes), much larger than the table lookup of AES.

A specific version with circulant matrices has not been implemented yet, however, it would only require one line of the matrix (1 kilobyte) from which all the columns of the matrix could be deduced by circular shift. Hopefully, we could have an efficient implementation, comparable in speed with the AES, but using less ROM and RAM than AES, which could be an advantage when implemented on constrained environement.

	security ($\log_2$)				speed
t	f_1 (or f_2)	f	r	key size	cycle/byte
16	81	71	128	64	22
24	119	102	192	96	46
32	**152**	**131**	**256**	**128**	**27**
48	223	191	384	192	47
64	294	252	512	256	53
128	575	493	1024	512	83
AES-CTR	-	-	-	128	26

TABLE II

PERFORMANCE (PENTIUM IV) AND SECURITY FOR $\ell = 8$

Comments on table II :

- The speed we give is for the implementation of the whole stream cipher. It compares well with the AES in counter mode with a similar software implementation. On a 3.4 Ghz PC, the encryption/decryption rate is about 1 Gbit/s.
- Concerning the security of f_1 (or f_2), there always exists an attack that costs about $2^{|K|}$ computations of f_1 or f_2 (where $|K|$ is the binary size of the key) which enables to recover the key, or, almost as good, the initial state.
- Though practical attacks cannot be built from an inversion attack on f, we give its difficulty: the difficulty of the decoding attack.
- If we apply the security reduction result (equation (2)) for $L = 2^{40}, T = 2^{80}$ and $\epsilon = 1/100$ (like in [3]), we compare the value obtained by the reduction of (2) and search for a contradiction when this complexity is less than the best known attack on f. Taking $t = 32$ is clearly not sufficient. Higher values $t = 64$ or $t = 128$ are required in order to obtain a contradiction. Meanwhile, if we consider the best existing attack, we admit that $t = 32$ is sufficient to assure a good heuristic security for the system.

VI. CONCLUSION

We have presented a new stream cipher with a security reduction. Compared with the previous proposals based on codes [11] or quadratic equations [3], it possesses some very interesting features

- it is as fast as AES in counter mode
- it has a security reduction
- the program as well as the memory requirement are very small.

Admittedly, the security analysis has to be pursued, in particular when regular words and quasi-cyclic codes are concerned. Also, the state of the art for decoding random linear codes might not be sufficient to consider all the security aspects of a stream cipher. However this system, whose design is derived from existing and unbroken systems, combines many interesting properties.

REFERENCES

[1] D. Augot, M. Finiasz and N. Sendrier, A family of fast syndrome based cryptographic hash functions, In MyCrypt 2005, LNCS 3715, (2005), p. 64-83.

[2] A. Barg. Complexity issues in coding theory. In V. S. Pless and W. C. Huffman, editors, *Handbook of Coding theory*, volume I, chapter 7, pages 649–754. North-Holland, 1998.

[3] C. Berbain, H. Gilbert and J. Patarin, QUAD: A practical stream cipher with provable security. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, number 4004 in LNCS, pages 109–128. Springer-Verlag, 2006.

[4] E. Berlekamp, R. McEliece and H. van Tilborg, On the inherent intractability of certain coding problems, IEEE Transactions on Information Theory, IT-24(3) (1978).

[5] L. Blum, M. Blum and M. Shub, A simple unpredictable pseudo-random number generator, *SIAM J. comput.* 15(2), (1986), p. 164-383.

[6] M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudorandom bits, *SIAM J. Comput.*, 15(2),(1984), p. 850-864.

[7] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to primitive narrow-sense BCH codes of length— 511. IEEE Transactions on Information Theory, IT-44(1998), 367-378.

[8] C. Chen, W. Peterson and E. Weldon, *Some results on quasi-cyclic codes*, Information and Control 15 (1969) pp. 407–423.

[9] J.-S. Coron and A. Joux. Cryptanalysis of a provably secure cryptographic hash function. Cryptology ePrint Archive, 2004. http://eprint.iacr.org/2004/013/.

[10] T. Cover. Enumerative source encoding. *IEEE Transactions on Information Theory*, 19(1):73–77, January 1973.

[11] J.-B. Fischer and J. Stern, An efficient pseudo-ra,dom generator provably as secure as syndrome decoding, EuroCrypt 1996, (1996) p.245-255.

[12] P. Gaborit and G. Zémor, Asymptotic improvement of the Gilbert-Varshamov bound for linear codes, Proceedings of ISIT 2006, Seattle, USA (2006), 287-291.

[13] O. Goldreich and L. Levin, Hard core predicate for any one-way function. In Proc of STOC'89, ACM press p. 25-32.

[14] Y. Gurevich. Average case completeness. *Journal of Computer and System Sciences*, 42(3):346–398, 1991.

[15] J. Hastad, R. Impagliazzo,L. Levin and M. Luby, Pseudo-random generator from one-way functions, *SIAM J. Comput.*, 28(4), (1999), p. 1364-1396.

[16] R. Impagliazzo and M. Naor, Efficient cryptographic schemes provably as secure as subset sum, *J. Cryptology*, 9(4), 199-216 (1999).

[17] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[18] R.J. McEliece, "A public-key cryptosystem based on algebraic coding theory," JPL DSN Progress Report 42-44, 114-116 (1978).

[19] F.J. MacWilliams, N.J.A. Sloane, *The Theory of Error Correcting Codes*, North-Holland (1977).

[20] J. N. Pierce, *Limit distributions of the minimum distance of random linear codes*, IEEE Trans. Inf. theory, Vol IT-13 (1967), pp. 595-599.

[21] N. Sendrier, *Cryptosystèmes à clé publique basés sur les codes correcteurs d'erreurs*, Mémoire d'habilitation, Inria 2002, available at: http://www-rocq.inria.fr/codes/Nicolas.Sendrier/pub.html

[22] N. Sendrier. Encoding information into constant weight words. In *IEEE Conference, ISIT'2005*, Adelaide, Australia, September 2005.

[23] D. Wagner. A generalized birthday problem. In M. Yung, editor, *CRYPTO'02*, number 2442 in LNCS, pages 288–303. Springer-Verlag, 2002.