The cryptographic security of the syndrome decoding problem for rank distance codes

F. Chabaud[1]   J. Stern

Florent.Chabaud@ens.fr   Jacques.Stern@ens.fr

Laboratoire d'informatique
de l'École Normale Supérieure[2]
45, rue d'Ulm
F-75230 Paris cedex 05

phone: (33-1) 44 32 20 47
fax: (33-1) 44 32 20 80

May 25, 1996

Abstract. We present an algorithm that achieves general syndrome decoding of a (n, k, r) linear rank distance code over $GF(q^m)$ in $O((nr + m)^3 q^{(m-r)(r-1)})$ elementary operations. As a consequence, the cryptographic AI schemes [Che94, Che96] which rely on this problem are not secure with the proposed parameters. We also derive from our algorithm a bound on the minimal rank distance of a linear code which shows that the parameters from [Che94] are inconsistent.

---

[1] On leave from *Délégation Générale de l'Armement*

[2] *Unité de Recherche Associée n° 1327 du Centre National de la Recherche Scientifique*

The cryptographic security of the syndrome decoding problem for rank distance codes

May 25, 1996

—

Abstract. We present an algorithm that achieves general syndrome decoding of a (n, k, r) linear rank distance code over $GF(q^m)$ in $O((nr + m)^3 q^{(m-r)(r-1)})$ elementary operations. As a consequence, the cryptographic AI schemes [Che94, Che96] which rely on this problem are not secure with the proposed parameters. We also derive from our algorithm a bound on the minimal rank distance of a linear code which shows that the parameters from [Che94] are inconsistent.

1 Introduction

It is known that the problem of finding a codeword of given weight in a linear binary code is $\mathcal{NP}$-complete [BMT78]. Furthermore, the problem remains difficult when the code is chosen at random and the weight is close to the Gilbert-Varshamov bound (see the discussion in [Ste, FS96]). Recently, several cryptographic schemes aimed at entity identification and based on this property [Gir90, Har89, Ste90, Ste94, Vér95b] have been proposed[3]. They have low computational requirements and high speed. The counterpart is that the communication complexity is significant.

In an attempt to improve the performances of the above systems, Kefei Chen has suggested the idea of using rank metric codes [Gab85] instead of Hamming metric codes in cryptographic schemes. He has designed two authentication schemes [Che94, Che96] with claimed better performances than the above systems. The security of these protocols relies on the following informal assumption:

The syndrome decoding problem for rank distance codes appears even more difficult than for Hamming distance codes.

In this paper, we first recall the definition of rank distance codes and how they are used in K. Chen's protocols. Then we present our attack on these protocols. Accordingly, we modify their parameters to achieve security. It is debatable whether or not the original schemes proposed by K. Chen achieve better performances than their analogues based on standard error-correcting codes. But taking into account the loss in the efficiency of the protocols resulting from underestimating the necessary sizes, it appears that rank distance codes are not better than usual codes.

[3] The authentication scheme [Har89] was broken by P. Véron [Vér95a].

2 Background

2.1 Rank distance codes

The rank distance codes were introduced by E.M. Gabidulin [Gab85] and rely on the following observation.

Let $\bar{x} = (x_1, \dots, x_n)$ be a n-dimensional vector over $GF(q^m)$, where q is the power of a prime. Let $b_1, \dots, b_m$ be a basis of $GF(q^m)$. Write each element $x_j \in GF(q^m)$ as $x_j = \beta_{1,j} b_1 + \dots + \beta_{m,j} b_m$, where $\beta_{i,j} \in GF(q)$ for all i. Then the rank of

$$A(\bar{x}) = \begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,n} \\ \vdots & & \vdots \\ \beta_{m,1} & \cdots & \beta_{m,n} \end{pmatrix}$$

is uniquely determined by $\bar{x}$, and defines a metric on the n-dimensional vector space V over $GF(q^m)$. Following [Gab85], we will denote this metric by $r(\bar{x}, q)$. Generally speaking, given a linear code over $GF(q^m)$, that is to say a k-dimensional subspace of V, the rank distance decoding problem can be stated as follows:

Rank distance decoding problem Let H be a parity check matrix over $GF(q^m)$ of the code C (*i.e.* $\bar{x} \in C \Leftrightarrow H\bar{x}^t = 0$), given a $(n-k)$-vector $\bar{\sigma}$ over $GF(q^m)$, find a n-vector $\bar{s} \in V$ of smallest rank $r(\bar{s}, q)$ such that

$$H\bar{s}^t = \bar{\sigma}^t. \tag{1}$$

In coding theory, vector $\bar{\sigma}$ is called the *syndrome* of the *error vector* $\bar{s}$.

Mutatis mutandis, as for Hamming distance codes, if the error vector has rank r smaller than half the minimum rank distance d of the code, then equation (1) has a unique solution of rank less than $\frac{d}{2}$.

2.2 Minimum Rank Distance codes

The above metric can be used to formulate a theory analogous to the theory of Minimum Distance Separable codes [MS83]. In particular, if H is a $(n-k) \times n$ parity check matrix of a linear code over a finite field, the minimum rank distance d of the code, *i.e.* the minimum of the non-zero ranks of the codewords, verifies

$$d \leq n - k + 1. \tag{2}$$

This bound is called the Singleton bound [MS83] in the theory of linear Hamming distance codes and a code that achieves equality is called *Minimum Distance Separable (MDS)*. Following [Gab85], we similarly call *Minimum Rank Distance (MRD)*-code a linear rank distance code that achieves equality

$$d = n - k + 1.$$

Such codes exist. Some of them are constructed in [Gab85] together with coding and decoding algorithms.

Prover		Verifier
Chooses a random n-vector $\bar{x}$ over $GF(q^m)$, a random $n \times n$ permutation matrix P, and computes the products $\bar{c} = HP\bar{x}$ and $\bar{c}' = H\bar{x}$.	$\xrightarrow{\bar{c},\ \bar{c}'}$	
Computes the $(n-k)$-vector $\bar{w} = \bar{x} + \lambda P^{-1}\bar{s}$	$\xleftarrow{\lambda}$	Chooses randomly $\lambda \in GF(q^m)$
	$\xrightarrow{\bar{w}}$	
	$\xleftarrow{b}$	Chooses $b \in \{0,1\}$
If $b = 0$ sends P, otherwise sends $\bar{x}$.	$\xrightarrow{P \text{ or } \bar{x}}$	If $b = 0$, checks if $HP\bar{w} = \bar{c} + \lambda\bar{\sigma}$ and $H\bar{x} = \bar{c}'$.

Otherwise checks the rank: $r(\bar{w} - \bar{x}, q) = \begin{cases} r, & \text{if } \lambda \neq 0; \\ 0, & \text{if } \lambda = 0. \end{cases}$

Fig. 1. Kefei Chen's authentication protocol [Che96].

2.3 K. Chen's authentication schemes

We now briefly describe the identification scheme [Che96] which is an improvement of [Che94]. Both schemes use as public data a $(n-k) \times n$ parity check matrix H of a rank-distance code over $GF(q^m)$ with error-capacity t[4] and an integer $r \leq t$.

Each user chooses a random n-vector $\bar{s}$ over $GF(q^m)$ of rank r and computes the syndrome $\bar{\sigma} = H\bar{s}^t$. The $(n-k)$-vector $\bar{\sigma}$ over $GF(q^m)$ is the public key for authentication. The interactive protocol of figure 1 can now be repeated a certain number of times to achieve "security". This protocol is zero-knowledge. The proposed parameters are

$$q = 2, n = 32, k = m = 16 \text{ and } r = 4.$$

We now show that the underlying problem is too weak for this set of parameters.

3 Algorithm A

3.1 Principle

We now present an algorithm which solves the following problem:

[4] Every error vector of rank less than t can be successfully corrected.

Fixed rank codeword search problem Given an integer r and a parity check matrix H of a linear rank-distance code over $GF(q^m)$, find a n-vector $\bar{s}$ over $GF(q^m)$ of rank less than r such that

$$H\bar{s}^t = 0. \tag{3}$$

First we see how such an algorithm can solve the *rank distance decoding problem* described by equation (1). Given H and $\bar{\sigma}$, we can add a column to matrix H and form the matrix

$$H' = (\ H \quad | \quad \bar{\sigma}^t\ ).$$

Every solution of the equation

$$H'\bar{s}'^t = 0 \tag{4}$$

can be split into two parts $\bar{s}'^t = (\ \bar{s_0} \quad | \quad s_1\ )$, with $\bar{s_0}$ a n-vector over $GF(q^m)$ and s_1 an element of $GF(q^m)$. For every solution $\bar{s}'$ of equation (4), either $H\bar{s_0}^t = s_1\bar{\sigma}^t$ or $\bar{s_0}$ is a solution of equation (3) and $s_1 = 0$.

But, if we know *a priori* that the error vector $\bar{s}$, such that $H\bar{s}^t = \bar{\sigma}^t$, has rank r smaller than half the minimum rank distance d of the code H, which is the case for the rank distance decoding problem, then we know that a solution of equation (4) of rank less than $r+1$ cannot be a codeword of matrix H. Hence, we can obtain a solution for equation (1) which is $s_1^{-1} \times \bar{\sigma}$. Therefore, if we have an algorithm that solves problem described by equation (4), it will solve the rank distance decoding problem of equation (1).

We note that this adaptation is made at the cost of increasing the rank parameter of the problem by one. If we want to solve an instance of the problem (1) with a searched rank r, we will have to solve a derived instance of problem (4) with a searched rank $r+1$[5].

3.2 Brute force algorithm

Assume there exists a solution $\bar{s} = (s_1, \ldots, s_n)$ of equation (3) of rank r. Then, there exists r elements $\theta_0, \ldots, \theta_{r-1}$ of $GF(q^m)$, linearly independent over $GF(q)$, and nr coefficients $\alpha_{j,k} \in GF(q)$, such that for all j, $1 \le j \le n$,

$$s_j = \sum_{k=0}^{r-1} \alpha_{j,k}\theta_k.$$

We denote by $(h_{i,j})_{\substack{1 \le i \le n-k \\ 1 \le j \le n}}$ the coefficients of matrix H. We can then rewrite equation (3), and obtain a system of $(n-k)$ relations over $GF(q^m)$.

$$\forall i,\ 1 \le i \le n-k,\ \ \sum_{j=1}^{n}\sum_{k=0}^{r-1} h_{i,j}\alpha_{j,k}\theta_k = 0. \tag{5}$$

[5] In the very special case of $s_1 = 1$, the rank remains unchanged, but this is of no importance as we will see later.

As soon as $(\theta_0, \dots, \theta_{r-1})$ are known, the above system gives a redundant linear system over $GF(q)$, with nr unknowns $(\alpha_{j,k})_{\substack{1 \le j \le n \\ 0 \le k \le r-1}}$ and at most $(n-k)m$ independent equations.

Therefore, our brute force algorithm enumerates all bases $(\theta_0, \dots, \theta_{r-1})$ over $GF(q^m)$ and tries to solve the linear system over $GF(q)$ resulting from system (5).

There are at most q^{mr} different bases $(\theta_0, \dots, \theta_{r-1})$ over $GF(q^m)$. The resulting complexity is too high for the proposed parameters of K. Chen's authentication scheme, but note that this exhaustive search is not even mentioned in [Che96]. The rest of the paper is devoted to the study of a better search algorithm.

3.3 Bound for minimal rank-distance codes

We now fix for this subsection the r elements $(\theta_0, \theta_1, \dots, \theta_{r-1})$ of $GF(q^m)$. System (5) can give us solutions to equation (3) as soon as it has more unknowns than equations, that is to say if $nr > (n-k)m$. In all (n,k) linear code, this inequality implies that we can find a codeword of rank r if $r \ge \frac{(n-k)m+1}{n}$. Hence, we obtain a bound on the minimal rank-distance of a (n,k,d) linear rank distance code

$$d \le \left\lceil \frac{(n-k)m+1}{n} \right\rceil \tag{6}$$

Theorem 1. *No MRD code can exist for $m < n$.*

Proof. An MRD code achieves $d = n - k + 1$. But we can obtain a codeword for $r = n - k$ as soon as $n(n-k) > (n-k)m$.

Note 2. Speaking in terms of coding theory, that means that our bound (6) is better than the Singleton bound for the case $m < n$. One should note that all the MRD codes of Gabidulin's paper are given for $n \ge m$. Therefore, there is no contradiction between the above result and [Gab85].

3.4 Selective enumeration

We now show how to decrease the cost of our enumerative search. The principle of the following algorithm remain the same. We just want to reduce the number of bases $(\theta_0, \theta_1, \dots, \theta_{r-1})$ over $GF(q^m)$ for which we have to solve a linear system over $GF(q)$.

First, we can notice that for all $\theta \in GF(q^m)$, if $(s_1, \dots, s_n)$ is a solution of equation (3), then $(\theta s_1, \dots, \theta s_n)$ is also a solution. We can therefore only enumerate the bases of the form $(1, \theta_1, \dots, \theta_{r-1})$.

Let $(b_1, \dots, b_m)$ be a basis of $GF(q^m)$ over $GF(q)$. For every element B in $GF(q^m)$, there exists m elements $\beta_1, \dots, \beta_m$ of $GF(q)$ such that

$$B = \sum_{\ell=1}^{m} \beta_\ell b_\ell.$$

Such a basis can for instance be the canonical polynomial representation of $GF(q^m)$, in which case we have $b_\ell = X^{\ell-1}$. For simplicity we symbolize this particular representation by the notation

$$B = [\beta_m \cdots \beta_1] = \beta_m X^{m-1} + \cdots + \beta_2 X + \beta_1,$$

and we call *digits* the particular coefficients of this representation.

As $\theta_0 = 1$, we have $\theta_0 = [0 \cdots 01]$. Therefore, the last digit of every θ_i can be arbitrarily set to zero without loss of generality as

$$[\theta_{i,1} \cdots \theta_{i,m-1}\theta_{i,m}] = [\theta_{i,1} \cdots \theta_{i,m-1}0] + \theta_{i,m}[0 \cdots 01].$$

We now formalize these ideas and estimate the number of bases to enumerate.

Lemma 3 [LN83, page 455]. *The number of $m \times r$ matrices of rank r over $GF(q)$ is*

$$N_q(m,r) = q^{\frac{r(r-1)}{2}} \prod_{i=0}^{r-1} (q^{m-i} - 1).$$

Corollary 4. *The number $C_q(r)$ of invertible matrices or size r over $GF(q)$ is*

$$C_q(r) = q^{\frac{r(r-1)}{2}} \prod_{i=1}^{r} (q^i - 1).$$

Definition 5. A *strict basis* of rank r, is a basis $\Theta = (1, \theta_1, \ldots, \theta_{r-1})$ for which the last digits of the θ_i are all zeros.

Definition 6. Two strict bases Θ and Θ' are *equivalent* if there exists an invertible matrix T over $GF(q)$ of size r such that

$$\Theta' = \begin{pmatrix} 1 \\ \theta'_1 \\ \vdots \\ \theta'_{r-1} \end{pmatrix} = T \begin{pmatrix} 1 \\ \theta_1 \\ \vdots \\ \theta_{r-1} \end{pmatrix} = T\Theta.$$

It is clear that we only have to enumerate one element in each equivalence class. We now count the number of elements to enumerate.

Lemma 7. *The number of bases in a class is*

$$C_q(r-1) = q^{\frac{(r-1)(r-2)}{2}} \prod_{i=1}^{r-1} (q^i - 1).$$

The proof of this lemma uses a block-wise representation of transition matrix T and is given in A. We set

$$D_q(m,r) = \frac{N_q(m,r)}{C_q(r)} = \frac{\prod_{i=m-(r-1)}^{m} (q^i - 1)}{\prod_{i=1}^{r} (q^i - 1)}. \tag{7}$$

Lemma 7 means that using for instance lexicographic order, we only need to enumerate $D_q(m-1, r-1)$ strict bases in order to find a solution of equation (5). Appendix B gives a way to enumerate such bases.

The following theorem means that this solution is unique. Therefore, we have essentially no better strategy than enumerating one basis in every class and check if the corresponding linear system over $GF(q)$ resulting from system (5) has a solution.

Theorem 8. *Let n be an integer greater than r. Let Θ and Θ' be two bases such that there exists two $n \times r$ matrices over $GF(q)$ A and A' of maximal rank r for which $A\Theta = A'\Theta'$. Then Θ and Θ' are equivalent.*

Proof. As A and A' are of maximal rank, by Gaussian elimination there exists two invertible matrices S and S' of size n over $GF(q)$ and two permutation matrices P and P' of size r such that, using a block-wise representation for S and S' we have

$$A = \begin{pmatrix} S_1 & S_2 \\ S_3 & S_4 \end{pmatrix} \begin{pmatrix} Id_r \\ 0 \end{pmatrix} P = \begin{pmatrix} S_1 \\ S_3 \end{pmatrix} P \text{ and } A' = \begin{pmatrix} S'_1 & S'_2 \\ S'_3 & S'_4 \end{pmatrix} \begin{pmatrix} Id_r \\ 0 \end{pmatrix} P' = \begin{pmatrix} S'_1 \\ S'_3 \end{pmatrix} P'.$$

As A and A' are of rank r, these relations imply in particular that the two $r \times r$ matrices over $GF(q)$ S_1 and S'_1 are invertible.

These relations are also true over $GF(q^m)$. Hence, we have

$$S \begin{pmatrix} Id_r \\ 0 \end{pmatrix} P\Theta = S' \begin{pmatrix} Id_r \\ 0 \end{pmatrix} P'\Theta',$$

from which we can extract $S_1 P\Theta = S'_1 P'\Theta'$. This completes the proof.

3.5 Implementation

We have implemented our algorithm using the ZEN C-library [CL96]. We enumerate the strict bases as described in appendix B and for each of these bases, we solve the linear system over $GF(q)$ of $(n-k)m$ equations and $n(r+1)$ unknowns resulting from system (5). This second step is in fact a little optimized using a trick described in appendix C. This trick performs a kind of parallelization of successive Gaussian eliminations resulting in an improvement by a factor 2.

The above algorithm was successfully tested and the results are summarized in figure 2. The parameters are chosen to match with those proposed in Kefei Chen's schemes. We now discuss the security of these schemes.

4 Application to K. Chen's schemes for authentication

At this point, let's recall the parameters of K. Chen's protocols. The first one [Che94] used the parameters

$$q = 2, n = 32, k = 16, m = 8 \text{ and } r \geq 4.$$

r	Number of basis	$(nr)^3$	CPU time by iteration (ms)	Estimated CPU max.
2	$2^{15} = 32767$	2^{18}	6.5	200 s
3	$2^{27.4} = 178940587$	$2^{19.8}$	8.5	18 days
4	$2^{37.6} = 209386049731$	$2^{21.0}$	10.5	70 years
5	$2^{45.7} = 57162391576563$	$2^{22.0}$	12.5	22,400 years

Fig. 2. Finding a codeword of given rank for $q = 2$, $n = 32$, $k = 16$, $m = 16$ on PC-486 100MHz

These parameters are inconsistent. System (5) is not redundant with these parameters. We obtain $(n-k)m = 128$ equations with $n(r+1) \geq 160$ unknowns over $GF(q)$. Therefore, given a public key syndrome, one can easily find a secret key for these parameters. This means that the minimal rank distance of this code is smaller than r. Indeed, bound (6) gives in this case $d \leq 5$. Hence, as we should have $r < \frac{d}{2}$, possible parameters are

$$q = 2, n = 32, k = 16, m = 8 \text{ and } r = 2.$$

With these parameters, there will be no two secret keys with the same public key. But, given a public key, our algorithm need at most $D_q(m-1,r) = 2^{11.4}$ Gaussian eliminations each with $(n(r+1))^3 = 2^{20}$ further elementary operations. This clearly defeats the scheme.

The second scheme [Che96] uses the following parameters:

$$q = 2, n = 32, k = m = 16 \text{ and } r = 4.$$

In this case, our bound (6) gives $d \leq 9$. Codes necessary for this scheme can therefore exist.

Our algorithm leads to at most $2^{45.7}$ Gaussian eliminations each with about 2^{15} operations using the trick described in appendix C.

We can estimate the overall complexity of our algorithm A for solving fixed rank codeword search problem (3). On one hand, the number of strict bases grows asymptotically as $O(q^{(m-r)(r-1)})$. On the other hand, it is well known that the number of elementary operations in a Gaussian elimination over $GF(q)$ is $O((nr)^3)$. We therefore obtain for the complexity of our algorithm

$$O\left((nr)^3 q^{(m-r)(r-1)}\right).$$

This first approach has the disadvantage to increase by one the rank of the word to find for solving problem (1). This is necessary to convert the problem in the form of equation (3). Hence, the overall complexity of our algorithm A for solving problem 1 is

$$O\left((n(r+1))^3 q^{(m-r-1)r}\right).$$

In table 2, for instance, one can see that the Kefei Chen's authentication scheme with proposed parameters [Che96] is solved by algorithm A in about 22,000 years.

It would be better to avoid this increase in the rank, because we would then obtain a more realistic search in about 70 years.

We now present a modification of the above algorithm that solves directly the initial problem (1) without increasing the search ranked. This results in a better algorithm for solving the initial rank distance decoding problem.

5 General syndrome decoding problem for rank-distance linear codes

5.1 Algorithm B

We now suppose given a non null syndrome $\bar{\sigma} = (\sigma_1, \ldots, \sigma_{n-k})$ over $GF(q^m)$. In this case, system (5) is replaced by:

$$\begin{aligned} \forall i, 1 \leq i \leq n-k, \quad & \sum_{j=1}^{n} \sum_{k=0}^{r-1} h_{i,j} \alpha_{j,k} \theta_k = \sigma_i, \\ h_{i,j}\alpha_{j,0} + & \sum_{j=1}^{n} \sum_{k=1}^{r-1} h_{i,j} \alpha_{j,k} \theta'_k = \sigma_i \theta_0^{-1}, \end{aligned} \tag{8}$$

with $\theta'_k = \theta_k / \theta_0$. Let $(b_1, \ldots, b_m)$ be a basis of $GF(q^m)$ over $GF(q)$. There exists m elements $\beta_1, \ldots, \beta_m$ of $GF(q)$ such that

$$\frac{-1}{\theta_0} = \sum_{\ell=1}^{m} \beta_\ell b_\ell.$$

Hence, we obtain

$$h_{i,j}\alpha_{j,0} + \sum_{j=1}^{n} \sum_{k=1}^{r-1} h_{i,j} \alpha_{j,k} \theta'_k + \sum_{\ell=1}^{m} \beta_\ell b_\ell \sigma_i = 0. \tag{9}$$

This system of $n-k$ relations over $GF(q^m)$ gives over $GF(q)$ a system of at most $(n-k)m$ independent equations with $nr+m$ unknowns. The remaining of our discussion is the same, and our algorithm B will therefore consist in enumerating the same strict bases and solving for each one a linear system over $GF(q)$. The resulting system has only a little more unknowns than before, but the important point is that there is no more need to increase by one the searched rank, that is to say the number of elements in each basis.

Hence the number of bases to enumerate remains the same. In particular, asymptotically algorithm B performs $O(q^{(m-r)(r-1)})$ Gaussian eliminations. As a Gaussian elimination takes $O((nr+m)^3)$ operations, we eventually obtained the claimed complexity

$$O\left((nr+m)^3 q^{(m-r)(r-1)}\right).$$

Using this algorithm, we obtain for the second scheme proposed by K.Chen an exhaustive search of $D_q(m-1, r-1) = 2^{37.6}$ Gaussian eliminations. With the same trick as before (see Appendix C) that parallelizes Gaussian eliminations, we need on average 2^15 operations to perform a Gaussian elimination. Thus, we can obtain an exhaustive attack of K. Chen's protocol that discloses a secret key from a public one, in less than 2^{53} elementary operations.

This is less than the time needed for an exhaustive search of a DES-key, and the scheme should therefore be considered unsecure according to current standards.

5.2 Implementation

Algorithm B was also implemented using the ZEN C-library [CL96] and successfully tested. We present in figure 3 our experimentations on same dimensions as in figure 2. One should note that the estimated maximal CPU times are increased by relatively small values which confirms our estimation.

The estimated time of computation to break an instance of Kefei Chen's authentication scheme appears to be 78 years. This figure may still seem high but we note the following:

1. Using a faster machine like a sparc 20, the estimated time falls to 20 years.
2. Our estimation is made in the worst case. On average, we only need half this time to solve a random instance of the problem.
3. The algorithm can be easily distributed on a network. Suppose we have about a thousand machines (like the RSA-130 breaking project), then a secret key would be found in less than a week.

r	Number of basis	$(nr)^3$	CPU time by iteration (ms)	Estimated CPU max.
2	$2^{15} = 32767$	2^{18}	7.5	250 s.
3	$2^{27.4} = 178940587$	$2^{19.8}$	10	20 days
4	$2^{37.6} = 209386049731$	$2^{21.0}$	12	78 years
5	$2^{45.7} = 57162391576563$	$2^{22.0}$	13	24,000 years

Fig. 3. Solving syndrome decoding problem for $q = 2$, $n = 32$, $k = 16$, $m = 16$ on PC-486 100MHz

6 Conclusion

We have presented an attack against the general syndrome decoding of linear rank distance codes problem, and shown that the authentication schemes described in [Che94, Che96] are unsecure with the proposed parameters. Besides, the attack can be easily distributed on a network of stations. Thus, one should be very careful in choices for K. Chen's protocols parameters.

References

[BMT78] E.R. Berlekamp, R.J. McEliece, and H.C.A. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inform. Theory*, IT-24(3):384–386, May 1978.

[Che94] K. Chen. Improved Girault identification scheme. *IEE Electronic Letters*, 30(19):1590–1591, sep 1994.

[Che96] K. Chen. A new identification algorithm. In *Cryptography Policy and Algorithms conference*, volume 1029. LNCS, 1996.

[CL96] F. Chabaud and R. Lercier. Zen: A new toolbox for finite extensions in finite fields. Rapport de recherche, Laboratoire d'Informatique de l'Ecole Polytechnique, 91128 Palaiseau Cedex, France, 1996. in preparation.

[FS96] J.-B. Fischer and J. Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In *Advances in Cryptology – EUROCRYPT '96*, volume to appear. LNCS, 1996.

[Gab85] E.M. Gabidulin. Theory of codes with maximum rank distance. *Problems of Information Transmission*, 21:1–12, 1985.

[Gir90] M. Girault. A (non practical) three-pass identification protocol using coding theory. In *Proc. Auscrypt'90*, volume 453, pages 265–272. LNCS, 1990.

[Har89] S. Harari. A new authentication algorithm. In *Coding Theory and Applications*, volume 388, pages 204–211. LNCS, 1989.

[LN83] R. Lidl and H. Niederreiter. Finite fields. In Gian-Carlo Rota, editor, *Encyclopedia of Mathematics and its applications*. Addison-Wesley Publishing Company, 1983.

[MS83] F.J. MacWilliams and N.J.A. Sloane. *The Theory of Error-correcting Codes*. North-Holland, 1983.

[Ste] J. Stern. A new paradigm for public key identification. *IEEE Trans. Inform. Theory*. to be published.

[Ste90] J. Stern. An alternative to the Fiat-Shamir protocol. In *Advances in Cryptology – EUROCRYPT '89*, pages 173–180. LNCS, 1990.

[Ste94] J. Stern. A new identification scheme based on syndrome decoding. In *Advances in Cryptology – CRYPTO '93*, volume 773. LNCS, 1994.

[Vér95a] P. Véron. Cryptanalysis of Harari's identification scheme. In *Cryptography and Coding*, volume 1025, pages 264–269. LNCS, 1995.

[Vér95b] P. Véron. *Problème SD, Opérateur Trace, Schémas d'identification et Codes de Goppa*. PhD thesis, Université de Toulon et du Var, juillet 1995.

A Proof of lemma 7

We use a block-wise representation of the transition matrix T between two strict bases Θ and Θ'.

$$T = \begin{pmatrix} \Delta & D \\ C & B \end{pmatrix}$$

with

$$B = \begin{pmatrix} \beta_{1,1} & \cdots & \beta_{1,r-1} \\ \vdots & & \vdots \\ \beta_{r-1,1} & \cdots & \beta_{r-1,r-1} \end{pmatrix}, C = \begin{pmatrix} \gamma_1 \\ \vdots \\ \gamma_{r-1} \end{pmatrix}, D = \begin{pmatrix} \delta_1 & \cdots & \delta_{r-1} \end{pmatrix},$$

and $\Delta \in GF(q)$. Matrices B, C and D have coefficients over $GF(q)$.

As Θ' is a strict basis, we must have $\theta'_0 = 1$. Therefore

$$\theta'_0 = [0 \cdots 01] = [0 \cdots 0\Delta] + \sum_{i=1}^{r-1} \delta_i [\theta_{i,m} \cdots \theta_{i,2} 0].$$

That gives a redundant linear system of m equations over $GF(q)$ with $r \leq m$ unknowns $\Delta, \delta_1, \ldots, \delta_{r-1}$. This system implies $\Delta = 1$ and $D = 0$.

We also have $\theta'_j = \gamma_j + \sum_{k=1}^{r-1} \beta_{j,k}\theta_k$. As the last digit of every θ'_i is zero, we have

$$[\theta'_{j,m} \cdots \theta'_{j,2} 0] = [0 \cdots 0\gamma_j] + \sum_{k=1}^{r-1} \beta_{j,k} [\theta_{k,m} \cdots \theta_{k,2} 0],$$

from which we deduce that vector C is all-zeros. Hence, $\theta'_j = \sum_{k=1}^{r-1} \beta_{j,k}\theta_k$ and B must be an invertible matrix over $GF(q)$.

Clearly, matrix T is invertible and we have

$$T = \begin{pmatrix} 1 & 0 \\ 0 & B \end{pmatrix} \text{ and } T^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & B^{-1} \end{pmatrix}.$$

We can deduce from corollary 4 the number of invertible matrices B which is the number of transition matrices T between strict bases:

$$C_q(r-1) = q^{\frac{(r-1)(r-2)}{2}} \prod_{i=1}^{r-1} (q^i - 1).$$

This complete the proof.

B Selective bases enumeration

B.1 Principle

We can uniquely represent a strict basis $\Theta = (1, \theta_1, \ldots, \theta_{r-1})$ by a $(r-1) \times (m-1)$ matrix over $GF(q)$ using the digits of the θ_i

$$\Theta = \begin{pmatrix} \theta_{1,m} & \ldots & \theta_{1,2} \\ \vdots & & \vdots \\ \theta_{r-1,m} & \ldots & \theta_{r-1,2} \end{pmatrix}.$$

In the following, we denote $\succ$ the lexicographic order that one can define on $GF(q^m)$ using a polynomial representation of this set over $GF(q)$.

Our problem is to enumerate all represent-ants $(\theta_1, \ldots, \theta_{r-1})$ of classes of section 3.4. Without loss of generality we can impose

$$\theta_1 \succ \ldots \succ \theta_{r-1}, \tag{10}$$

and every most significant digit of the θ_i can be set to one.

First, let's consider the matrix

$$\Theta_{0,0} = \begin{pmatrix} Id_{r-1} & 0 \end{pmatrix}.$$

This matrix respects condition 10. Besides, for all $(m-r) \times (r-1)$ matrix A_0 over $GF(q)$, the bases

$$\Theta_{0,A} = \begin{pmatrix} Id_{r-1} & A_0 \end{pmatrix}$$

are all of distinct classes.

Our enumeration will therefore take as radix R all the matrices that are permutations of $\Theta_{0,0}$ with respect to condition 10, and enumerate for every radix the possible completion matrices A_R. It is simpler to understand this enumeration with an example.

B.2 Example

We take as a small example the parameters

$$q = 3, m-1 = 3 \text{ and } r-1 = 2.$$

We first have

$$R_0 = \begin{pmatrix} 1 & 0 & x \\ 0 & 1 & y \end{pmatrix}.$$

For this radix, we enumerate the completion matrices $\begin{pmatrix} x \\ y \end{pmatrix}$. This gives $3^2 = 9$ matrices:

$$\Theta_{0,0} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}.$$

$$\Theta_{0,1} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \Theta_{0,2} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 0 \end{pmatrix}, \Theta_{0,3} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}, \Theta_{0,4} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix},$$

$$\Theta_{0,5} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 1 \end{pmatrix}, \Theta_{0,6} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 2 \end{pmatrix}, \Theta_{0,7} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{pmatrix}, \Theta_{0,8} = \begin{pmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \end{pmatrix}.$$

The second radix obtained according to the lexicographic order is

$$R_1 = \begin{pmatrix} 1 & x & 0 \\ 0 & y & 1 \end{pmatrix}.$$

This second radix only gives 3 more matrices, because lexicographic order implies $y = 0$:

$$\Theta_{1,0} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}. \Theta_{1,1} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \Theta_{1,2} = \begin{pmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

We then obtain a last radix

$$\Theta_{2,0} = \begin{pmatrix} x & 1 & 0 \\ y & 0 & 1 \end{pmatrix},$$

that gives no more matrix because lexicographic order implies $x = y = 0$.

One can check that the number of classes in this case is indeed

$$\frac{N_3(3,2)}{C_3(2)} = 13.$$

C The Gaussian elimination trick

During the enumeration of the bases, the last element θ_{r-1} is the only one that changes at each step. It is therefore natural to perform the beginning of the Gaussian elimination (corresponding to the first $n(r-1)$ columns) one for all. However, memory limitations only allow to manage a certain number N of different possible θ_{r-1}.

In binary case, the beginning of the Gaussian elimination takes about $n(r-1)\frac{1}{2}(n-k)m(N+r-1)n$ operations. Then, N completing eliminations take each about $n\frac{1}{2}(n-k)mrn$ more operations. We finally obtain

$$\mathcal{N}_1(N) = n(r-1)\frac{1}{2}(n-k)m(N+r-1)n + Nn\frac{1}{2}(n-k)mrn.$$

Without the trick, the number of operations for N iterations is about

$$\mathcal{N}_0(N) = N\mathcal{N}_1(1) = Nnr\frac{1}{2}(n-k)mrn.$$

Hence, the resulting gain is

$$\frac{\mathcal{N}_0(N)}{N} - \frac{\mathcal{N}_1(N)}{N} = K(1 - \frac{1}{N}),$$

with K a constant on the other parameters:

$$K = \frac{nr}{2}(n-k)mrn(1-\frac{1}{r})^2.$$

The asymptotic behavior is reached quite rapidly so that the value $N = 128$ seems reasonable to avoid loss of performance due to too large memory requirements.

As an example, with the parameters

$$N = 128, q = 2, n = 32, k = m = 16 \text{ and } r = 5,$$

one has

$$\mathcal{N}_0(N)/N = 2^{21.6} \text{ and } \mathcal{N}_1(N)/N = 2^{20.2}.$$

This is an improvement by a factor $2^{1.4} \simeq 2$. Besides, in binary case, we can take advantage of the binary representation of integers on 32 or 64 bits to divide the overall complexity of the Gaussian elimination by 2^5 or 2^6. That gives us an average complexity of about 2^{15} operations for one Gaussian elimination.

This figure remains unchanged for algorithm B because the m supplementary unknowns can be included in the parallelization trick.