MinRank problem and Zero-knowledge authentication

Nicolas Courtois
Systèmes Information Signal (SIS), Université de Toulon et du Var
BP 132, F-83957 La Garde Cedex, France
e-mail : courtois@minrank.org
http://www.minrank.org

Abstract

A zero-knowledge protocol provides provably secure authentication based on a computational problem. Several such schemes have been proposed since 1984, and the most practical ones rely on problems such as factoring that are unfortunately subexponential. Still there are several other (practical) schemes based on NP-complete problems. Among them, the problems of coding theory are in spite of some 20 years of significant research effort, still exponential to solve.

The problem MinRank recently arouse in cryptanalysis of HFE (Crypto'99) and TTM (Asiacrypt'2000) public key cryptosystems. It happens to ba a strict generalization of those hard problems of (decoding) error correcting codes.

We propose a new Zero-knowledge scheme based on MinRank. We prove it to be Zero-knowledge by black-box simulation. We compare it to other practical schemes based on NP-complete problems.

The MinRank scheme allows also an easy setup for a public key shared by a few users, and thus allows anonymous group signatures.

Key words:

Zero-knowledge protocol, Identification, MinRank problem, NP-complete problems, Multivariate cryptography, Rank-distance codes, Error correcting codes, Syndrome decoding, Anonymous group signatures.

1 Introduction

The general problem we address is the classical problem of interactive authentication. It is well known that solving this problem also allows in prac-

tice non-interactive authentication, for example digital signatures, if only combined with a hash function (supposed to behave as a random oracle).

The notion of Zero-knowledge identification has been formalized by Goldwasser, Micali and Rackoff in [11]. In such a scheme a prover proves his identity to a verifier. Provided the underlying problem is difficult, we prove that there is no interactive strategy for the verifier communicating with the Prover, to extract any information whatsoever on the prover's secret.

Several such schemes have been proposed since the original Fisher-Micali-Rackoff scheme (1984), and the most practical ones are Fiat-Shamir and Guillou-Quisquater schemes. Unfortunately they rely on factoring which is subexponential.

In contrast, several schemes based on NP-complete problems have been proposed, and they are still practical. First such a scheme was the scheme PKP by Shamir [19]. Then we have several schemes related to coding, the best being those of Stern [22, 28]. Another scheme proposed by Chen [5] has been shown less secure then expected in [25]. There are also schemes CLE by Stern [23] and PPP by Pointcheval [17].

In the present paper we present a new Zero-knowledge protocol, for another NP-complete problem, along with good reasons to believe it is very hard, maybe even harder that the previously mentioned.

We argue that the schemes related to coding theory seem the most secure. Indeed, the underlying decoding problem(s) have been believed intractable since 1978 [1]. They have withstood since a significant research effort and are still exponential, see [22, 28, 3, 2, 24]. The basic problem called Syndrome Decoding, is equivalent to another problem called the Minimal Weight problem.

The Minimal Weight problem consists of finding a linear combination of given vectors over a finite filed that has a small Hamming weight. All known attacks for this problem are exponential. The MinRank problem is a problem of finding a linear combination of given matrices that has a small rank. It is at least as difficult as the previous problem, thus probably also exponential.

The problem of Minimal Weight is NP-complete. It is very important in cryptography as it gives the complexity of best attacks known for the McEliece and Niederreiter trapdoor functions [28].

The problem of MinRank is also NP-complete and recently proved relevant in cryptography. In [20] Shamir and Kipnis reduced the cryptanalysis of Hidden Fields Equations (HFE) scheme [15] to MinRank. In [12] Goubin and Courtois showed that the TTM cryptosystem is not secure, and gives solvable MinRank instances. In both these attacks the actual rank was small (respectively logarithmic and constant, unlike the general problem MinRank with $r \approx n$.

In the present paper we show have to prove in zero-knowledge an ability to compute (or have) MinRank solutions for **any** instance of MinRank. It

can be used not only to prove in Zero-knowledge a solution to MinRank itself, but also for many other problems: the aforementioned Minimal Weight problem (corresponding to a diagonal MinRank), and in fact any other problem (e.g factoring) as there is an effective way to write any system of multivariate equations (over a field) as a MinRank.

The paper is organized as follows: The next chapter recalls the basic requirements of a Zero-knowledge protocol. The section 3 defines MinRank, explains why it is hard and studies several related problems.

The section 4 proposes instances of MinRank for practical use, evaluated with 4 attacks currently known for MinRank.

In the section 5 we describe key generation and setup of the MinRank authentication which is described in section 6.

The following section 7 gives proofs of completeness, soundness and Zero-knowledge.

Finally in 8 we analyse the performance of the scheme. In 9 we compare MinRank to other known schemes that are based on NP-complete problems. We conclude that they are all roughly similar in terms of required resources.

In Appendix 1 we show how to realize anonymous group signatures with MinRank.

In Appendix 2 we show how to transform the scheme to achieve the cheating probability of 1/2 in 3 moves instead of 2/3.

2 Zero-knowledge protocols

An interactive protocol involves two entities/strategies, the Prover (P) and the Verifier (V). Formally we say that the Prover (P) and Verifier (V) are two interactive Turing machines that work in probabilistic polynomial time. They have access to some common data that includes x of length n (x is the public key of the Prover). At the end of the protocol the Verifier gives one of the two answers: Accept or Refuse.

We recall the three basic properties that are required:
the (efficient) **completeness**, the (computational) **soundness** and (black-box) **Zero-knowledge**.

(Efficient) Completeness. The Verifier that interacts with a legitimate Prover, will accept after a bounded number of questions.

The security is based on the impossibility by the prover to predict Verifier's questions. Thus he must be prepared to answer all of them. However one question only is asked at each time, and the protocols are made in such a way that an answer to one question gives no information (Zero-knowledge), while answering all the questions is proved to reveal Prover's secret (Soundness property).

(Computational) Soundness. If the Prover is illegitimate, he will be rejected with an overwhelming probability. We usually prove that if the Prover can always fool the Verifier, it can be used to extract the Prover's secret that allows further successful executions of the algorithm (a knowledge extractor).

Zero-knowledge. It is in fact more that the fact that the Verifier learns nothing about the secret. A much stronger notion demands that no Verifier strategy, no matter how smart, can extract any information from the Prover, even in several interactions. It gives provable security against active attacks.

Proofs are made by simulation using the Verifier as an oracle, or black-box, and therefore this definition has been called black box (computational) Zero-knowledge, as formalized by Goldreich, and Oren [10].

Definition 2.0.1 (Black box Zero-knowledge (BSZK), [10]) *A strategy P is told to be black box Zero-knowledge on inputs from S (common input) if there exists an efficient simulating algorithm U so that for every feasible Verifier strategy V, the two following probability ensembles are computationally indistinguishable:*

- $\{(P,V)(x)\}_{x\in S} \stackrel{def}{=}$ *all the outputs of V when interacting with P on a common input $x \in S$.*
- $\{U(V)(x)\}_{x\in S} \stackrel{def}{=}$ *the output of U using V as a black box, on input $x \in S$.*

The definition above is strong and still realistic: all well-known Zero-knowledge protocols are proven in this model.

3 The MinRank problem

Let $M_i, i = 1..m$ be a set of $n \times n$ matrices over a ring R. The problem MinRank(r) [14] is to find (at least one) solution $\alpha \in R^m$ such that:

$$Rank(\sum_i \alpha_i M_i) \leq r.$$

3.1 Related problems

The problem defined here is a sightly modified version of the MinRank problem defined in [14], still the complexity results of [14] hold. MinRank as defined here, can also be seen as a (strict) generalization of the Rank distance coding problem introduced by Gabidulin, see [9, 5, 25] which in

turn generalizes the well known syndrome decoding and minimal weight problems of error correcting codes [1, 22, 28, 13, 3, 2, 24].

The problem MinRank also appears (though it is not named) in the Shamir-Kipnis attack of the HFE cryptosystem presented at Crypto'99 [20].

3.2 Encoding NP problems as MinRank

It is possible to **effectively** encode any set of multivariate polynomial modular equations as a MinRank. It works for any ring [14]. The encoding is based on the universality of the determinant:

Theorem 3.2.1 (Determinant Universality, Valiant 79) *Any set of multivariate equations can be encoded as a determinant of a matrix.*

The result was first shown by Valiant [26]. For a simpler, and still effective proof see [14]. The sizes of matrices obtained in those constructions [26, 14] seem hard to improve. For instance in order to encode m equations of degree d with n variables we need matrices of width about mn^d.

Since solving multivariate quadratic equations over a field is an NP-complete problem, as shown in [16] and partially in [**?**], it follows that:

3.3 MinRank is NP-nard

The proof of [14], however, was made for a different vision of MinRank. It gives instances of MinRank in which the size of the matrices will be polynomial in the number of matrices, and therefore it might seem that MinRank is less secure with m matrices $n \times n$ and m and n being of the same order of magnitude.

We give a direct proof that MinRank is NP-complete that gives instances with m and n being of the same order of magnitude and moreover the rank will be smaller too.

We reduce from the Minimal Weight problem of a linear error correcting code. Let (n, k, d) be an error correcting code. The encoding is trivial: each of the lines of the generating matrix will be put on the diagonal of a $n \times n$ matrix M_i that will have all 0's elsewhere. Solving MinRank with rank $r = d$ is equivalent to finding a word with the Hamming weight d.

The minimal weight problem of the error correcting codes, is not only NP-complete [1], but it is so hard that hardly any code has its minimal weight known and the best known algorithms are exponential, both in theory and in practice. For comprehensive reference see [28], and also [1, 2, 3, 24].

3.4 MinRank over $\mathbb{Z}$

MinRank over $\mathbb{Z}$ might be broken by the widely-used LLL algorithm, and it is clearly, when the M_i are all diagonal of size up to 300×300. Still, it

is undecidable in general [14] because it can encode diophantine equations (Tenth Hilbert's problem).

Let us show that MinRank over $\mathbb{Z}$ is as hard as factoring. We write the equation $N = xy$ and in order to assure $x > 1$ and $y > 1$ we write a non-negative integer as a sum of four squares. A well known theorem of Lagrange states that it is always possible. Then we encode the resulting equation as $N = (1 + x_1^2 + x_2^2 + x_3^2 + x_4^2)(1 + y_1^2 + y_2^2 + y_3^2 + y_4^2)$ and as explained in 3.2, we may encode it as a determinant of a matrix. If the resulting MinRank is broken, it gives factors of N.

4 MinRank instances and attacks

4.1 Proposed hard instances

There are vague reasons to advocate $r \leq n - 2$, as it has been shown that some very special subcases of MinRank$(n - 1)$ are not NP-hard [14].

We propose here three sets of parameters that have been chosen to respectively withstand attacks (below) in respectively 2^{64}, 2^{80} and 2^{128}.

By inspection we have found that MinRank schemes with small n and a big q are at the constant security more practical that schemes with with small q and a big n. Thus we use $GF(65521)$, 65521 is the greatest prime that fits in 16 bits.

Parameter set A: $m = 6, n = 5, r = 3$, $R = GF(65521)$.

Parameter set B: $m = 7, n = 7, r = 5$, $R = GF(65521)$.

Parameter set C: $m = 10, n = 11, r = 9$, $R = GF(65521)$.

4.2 Unique solution assumption

We conjecture that the parameters in MinRank should be chosen in such a way that the probability it has a solution (apart form those we might put by construction) should be close to 0.

4.3 Known Attacks

There are four attacks known for the problem MinRank. We suppose that ω is the exponent of the Gaussian reduction $2 \leq \omega < 3$.

4.3.1 Attack using sub-matrices

The basic idea is simple and has already been used by Coppersmith, Stern and Vaudenay in [6]. We express the MinRank in terms of determinants of all submatrices $(r + 1) \times (r + 1)$. The attack works only when roughly

$$\sum_{i=0}^{r+1} \binom{m+i-1}{i} < \binom{n}{r+1}^2$$

and it is not the case at all for [A],[B], [C].

4.3.2 The Kernel attack

This attack has been proposed by Louis Goubin in [12]. The complexity of the attack is basically

$$q^{\lceil \frac{m}{n} \rceil r} \cdot m^{\omega}$$

However several variants and improvements exist in [8]. For our settings it gives about respectively [A]:2^{71}, [B]:2^{88}, [C]:2^{138}.

4.3.3 MQ-solving attacks

MinRank can be written as an overdefined system of multivariate degree $r+1$ equations, and unless n is also very small, there will be enough equations to solve them by Gröbner bases or XL algorithm [21]. This possibility of solving MinRank in polynomial time with a small r was first discovered by Shamir and Kipnis [20]. The complexity of this attack has been evaluated in [8]:

$$(n')^{\frac{\omega}{\sqrt{e}}} \approx (rn)^{\omega r^2} \approx n^{\mathcal{O}(r^2)}$$

For our settings it gives about respectively [A]:2^{106}, [B]:2^{385}, [C]:2^{1612}.

4.3.4 Attacks when m is big

This attack is described in [8]. It an birthday-paradox version of a trivial attack and gives the complexity of about

$$q^{\frac{1}{2}Max(0,(n-r)*n-m)} \left((n-r)*n\right)^{\omega}.$$

For our settings it gives about respectively [A]:2^{66}, [B]:2^{91}, [C]:2^{133}.

5 Setup of MinRank authentication

5.1 Key setup

5.1.1 The public key

are m non-singular matrices $n \times n$ over a finite field $GF(q)$, $M_1, \ldots, M_m$. All of them but one will not be transmitted but (re)generated out of a pseudo-random generator.

Let $r < n$. To generate a random hard instance we choose a random M of rank r, then random non-singular matrices α and $M_1, \ldots, M_{m-1}$. Then we "adapt" M_m: we pick a random $\alpha \in R^m$ such that $\alpha_m \neq 0$ and M_m is computed as:

$$M_m = (M - \sum \alpha_i \cdot M_i)/\alpha_m$$

We know that (if it behaves as a random matrix), with a probability close to 1 it is non-singular, otherwise we try again with a different α.

In practice, we generate M and $M_1, \ldots, M_{m-1}$ out of a pseudo-random generator with a seed of 80 bits. For this we use the well-known LU method to generate a deterministic pseudo-random invertible matrices $M_1, \ldots, M_{m-1}$. It is slightly more complex for M. First we generate first a random invertible matrix M' which is a random invertible matrix $r \times r$ completed with 0's. Then we apply a random couple of invertible matrices S and T: $M = SM'T$.

5.1.2 The secret key

It is the chosen (above) $\alpha \in GF(q)^n$, and it has the desired property $M = \sum \alpha_i \cdot M_i$ has the rank r.

5.1.3 Key sizes

The size of the public key is thus reduced to $80 + n^2 \log_2 |R|$ bits.

The secret key requires only additional $m \log_2 |R|$ bits to store α.

5.2 Preliminaries

We need a hash function H, used only for commitments. It may be implemented by a collision-intractable cryptographic hash function such as MD5.

The Prover is going to convince the Verifier of his knowledge of α (and M). The Verifier will fail with the probability 1/3 unless he has the solution α (or an equivalent one if exists, proof of ability).

We need the following lemma:

Lemme 5.2.1 *Let M be a $n \times n$ matrix of rank r. Let S and T be two uniformly distributed random invertible matrices $n \times n$. Then TMS is uniformly distributed among all matrices of rank r.*

Proof sketch: It comes from the fact that every M of rank r can be written as:

$$M = S' \cdot \begin{pmatrix} Id_{r\times r} & 0_{r\times n-r} \\ 0_{n-r\times r} & 0_{n-r\times n-r} \end{pmatrix} \cdot T'$$

with S' and T' invertible. All the $n \times n$ matrices of rank r are equivalent modulo invertible variable changes.

6 MinRank authentication scheme

The Prover chooses two random non-singular $n \times n$ matrices S and T.

He also chooses a random $n \times n$ matrix X.

Then, he chooses a random combination β_1 of M_i:

$$N_1 = \sum \beta_{1i} \cdot M_i$$

And he also uses the expression of M to get an expression of $N_2 = M + N_1$ as:

$$N_2 = \sum \beta_{2i} \cdot M_i$$

The matrices has been done in such a way that $N_2 - N_1 = M$ and each of them (taken separately) is a random combination of the M_i.

β_1 is random and uniformly distributed, it is the same for the translated distribution of β_2.

One round of MinRank authentication:

1. The Prover sends to the Verifier:

$$\longrightarrow$$
$$H(X),\ H(TN_1S + X),\ H(TN_2S + X),\ H(S, T)$$

2. The Verifier chooses a query $q \in \{0, 1, 2\}$ and sends q to P.

$$\longleftarrow$$
$$q \in \{0, 1, 2\}$$

3. If $q = 0$ the Prover gives the following values:

$$\longrightarrow$$
$$(TN_1S + X),\ (TN_2S + X)$$

   Verification $q = 0$**:** The Verifier accepts if $H(TN_1S+X)$, $H(TN_2S+X)$ are correct and

$$(TN_2S + X) - (TN_1S + X) = TMS$$

   is indeed a matrix of rank r.

3' If $q = 1, 2$ the Prover reveals:

$$\longrightarrow$$
$$S,\ T,\ \beta_q, X$$

   Verification $q = 1, 2$**:** The Verifier checks if S and T are invertible and $H(S, T)$ is correct. Then he computes

$$TN_qS = \sum \beta_{qi}\ \ TM_iS$$

   and verifies $H(TN_qS + X)$ along with $H(X)$.

6.1 Completeness

It is obvious that if the Prover really knows α, he will always be able to follow the protocol.

6.2 Soundness

We show that a false Prover is rejected with probability $(\frac{1}{3})$.

Let C (Charlie or the cheater), be an expected polynomial time Turing machine. We prove that there is no such a false Prover C that can answer all the questions q unless C knows α (or an equivalent).

Since C commits (with H) the values of X, $X + TN_1S$ and $X + TN_2S$, he must choose them before he receives q.

The only way the Prover can answer the questions $q = 1$ and $q = 2$ is to have them indeed in the form $X + N_1'$ and $X + N_2'$ with the same X. Then each of N_1', N_2' must be indeed in the form TN_1S and TN_1S with the same S and T, as $H(S,T)$ is verified between $q = 1$ and $q = 2$.

Moreover to answer $q = 1..2$ he has to choose each of N_1, N_2 in the form $N_b = \sum \beta_{bi} \cdot M_i$ for $b = 0, 1$.

We check for $q = 0$ that the rank of the following matrix is r:

$$(TN_2S + X) - (TN_1S + X) = TN_2S - TN_1S = \sum_{i=1}^{m} (\beta_{2i} - \beta_{1i}) \cdot TM_iS$$

Since we check for $q = 1$ or 2 that S and T are invertible, $N_2 - N_1$ is also of rank r.

Thus the Prover knows the following matrix of rank r that is a linear combination of the M_i:

$$N_2 - N_1 = \sum_{i=1}^{m} (\beta_{2i} - \beta_{1i}) \cdot M_i.$$

The Prover knows $(\beta_1 + \beta_2)$, and it is indeed a secret key (or an equivalent one) that allows to impersonate the Verifier.

It is easy to see that in several rounds of MinRank, the cheating probability is in general:

$$\mathbb{P}_{cheat} = \left(\frac{2}{3}\right)^{\#rounds}.$$

7 Security of MinRank scheme

7.1 Black-box Zero-knowledge of MinRank

The simplicity of MinRank makes very easy to show it is Zero-knowledge.

Let the Prover strategy P be a probabilistic average polynomial time Turing machine. We suppose that H is a random function and therefore the committed H are considered random.

- In cases $q = 1, 2$ we only disclose random unrelated variables S, T, β_q, X.
- The case $q = 0$: disclosing $(TN_1S + X)$ and $(TN_2S + X)$ is equivalent to disclosing $(TN_1S + X)$ and $TMS = TN_2S - TN_1S$.

  Since X is completely random, $(TN_1S + X)$ is a random matrix independent from TMS.

  TMS, and as we have shown before, is for random invertible S and T an uniformly distributed matrix of rank r.

The exact construction of the simulator $U(V)$ is the following:

1. $U(V)$ chooses a random query $q = 1, 2$. He will prepare to answer to questions 0 **and** q.
2. He chooses an M' according to the probability distribution $M' = \sum \delta_i M_i$ with a random δ.
3. He chooses a random matrix X.
4. He chooses two random invertible matrices S and T and a random matrix M" of rank r.
5. He puts $N_q = M'$ and $N_{3-q} = (-1)^{q+1}M" + N_q$.
6. He asks for Verifier's query on his commitment:

   $$q' = V(H(X),\ H(TN_1S + X),\ H(TN_2S + X),\ H(S, T)) \in \{0, 1, 2\}.$$

7. He repeats steps 1-6 about 2 times (**rewinding**), until he does get one of the two queries he has prepared to:

   $$q' \in \{0, q\}$$

8. If $q' = 0$ the simulator $U(V)$ reveals $(TN_1S + X)$ and $(TN_2S + X)$.

8' If $q' \neq 0$, then $q' = q$ and the simulator $U(V)$ reveals S, T, δ, X.

8 Performance of the scheme

The computations are carried respectively for the parameter sets specified earlier as A/B/C.

Communications We assume that hash values are on 128 bits computed with MD5. Then the average communication complexity is of respectively [A]:1645, [B]:2680 and [C]:5784 bits per round.

For example for the set [A] we have $m = 6, n = 5, r = 3, R = GF(65521)$. We compute as follows: $4 * 128 + 2$ bits for the first two passes. The last pass requires $2 * 25 * 16$ bits in the case $q = 0$. In the two other cases it requires $(3 * 25 + 6) * 16$ bits. The weighted average for the last round will be of 1131 bits. In all $1131 + 512 + 2$ gives 1645 bits.

Key sizes Following 5.1.3 the size of the secret key is only respectively [A]:480, [B]:864 and [C]:2016 bits.

The size secret key consists of only m additional elements of R, thus respectively [A]:96, [B]:112 and [C]:160 bits, plus the size of the public key.

9 Comparison with other schemes

The following table compares different Zero-knowledge protocols based on NP-complete problems and is based on previous work by Pointcheval [18].

	PKP Shamir	SD Stern	Chen.2 Chen	CLE Stern	PPP Pointcheval	MinRank Courtois
matrix	16 x 32	256 x 512	32 x 16	24 x 48	101 x 117	**5 x 5**
field	$\mathbb{F}_{251}$	$\mathbb{F}_2$	$\mathbb{F}_{65535}$	$\mathbb{F}_{257}$	$\mathbb{F}_2$	$\mathbb{F}_{65521}$
passes	5	3	5	3/5	3/5	3
impersonation probability	$\frac{1}{2}$	$\frac{2}{3}$	$\frac{1}{2}$	$\frac{2}{3}/\frac{1}{2}$	$\frac{3}{4}/\frac{2}{3}$	$\frac{2}{3}/\frac{1}{2}$
rounds	20	35	20	35/20	48/35	35/20
impersonation global	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}	10^{-6}
public key [bits]	256	256	256	80	149	864
secret key [bits]	98	512	512	80	117	112+864
best attack	2^{52}	2^{70}	2^{53}	2^{73}	2^{61}	2^{66}
bits send/round	523	954	1553	940/824	896/1040	1645/1823
global [Kbytes]	1.28	4.08	3.79	4.01/2.01	5.25/4.44	7.02/4.44

Remark: In Appendix 2 we explain how the cheating probability in MinRank could become 1/2 instead of 2/3.

10 Conclusion and Perspectives

MinRank is proven Zero-knowledge and relies on a linear algebra problem MinRank. This NP-hard problem contains in a very natural way many famous problems such as syndrome decoding. There are good reasons to believe that MinRank is exponential.

MinRank is a very powerful scheme that can prove in Zero-knowledge a knowledge of a MinRank solution over a finite field.

The performances of the scheme are similar to the known schemes such as SD, PKP or PPP.

MinRank can be used to prove in Zero-knowledge a knowledge of a solution for any set of multivariate equations over a finite field. However, the encoding for an arbitrary problem will not be practical anymore.

References

[1] E.R. Berlekamp, R.J. McEliece, H.C.A. Van Tilborg: *On the inherent intractability of certain coding problems*; IEE Trans. Inf. Th., IT-24(3), pp. 384-386, May 1978.

[2] Anne Canteaut, Florent Chabaud: *A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to BCH Codes of length 511;*

[3] Florent Chabaud: *Asymptotic analysis of probabilistic algorithms for finding short codewords* in Proceedings of Eurocode'92, LNCS, Springer-Verlag.

[4] Kefei Chen: *Improved Girault Identification scheme*; Electronic Letters, 15 September 1994, Vol. 30, NO... 19, pp. 1590-1591.

[5] Kefei Chen: *A new identification algorithm*. Cryptography Policy and algorithms conference, vol. 1029, LNCS, Springer-Verlag, 1996.

[6] Don Coppersmith , Jacques Stern , Serge Vaudenay : *Attacks on the birational permutation signature schemes*; CRYPTO 93, Springer-Verlag, pp. 435-443.

[7] Nicolas Courtois: *The security of Hidden Field Equations (HFE)*; Cryptographers' Track Rsa Conference 2001, San Francisco 8-12 April 2001, to appear in Springer-Verlag.

[8] Nicolas Courtois: *The security of cryptographic primitives based on multivariate algebraic problems: MQ, MinRank, IP, HFE*; PhD thesis, to appear in 2001, Paris 6 University, France.

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

[10] Oded Goldreich, Y. Oren. Definitions and properties of Zero-knowledge proof systems. Journal of Cryptology 1994, vol.7, no.1, pp.1-32.

[11] S. Goldwasser, S. Micali and C. Rackoff, *The knowledge Complexity of interactive proof systems*; SIAM Journal of computing, 1997, Vol. 6, No.1, pp.84.

[12] Louis Goubin, Nicolas Courtois *Cryptanalysis of the TTM Cryptosystem*; Advances of Cryptology, Asiacrypt'2000, 3-9 December 2000, Kyoto, Japan, Springer-Verlag.

[13] Sami Harari. *A new authentication algorithm*. In Coding Theory and Applications, volume 388, pp.204-211, LNCS, 1989.

[14] Jeffrey O. Shallit, Gudmund S. Frandsen, Jonathan F. Buss: *The Computational Complexity of Some Problems of Linear Algebra problems*, BRICS series report, Aaarhus, Denmark, RS-96-33, available on the net http://www.brics.dk/RS/96/33/.

[15] Jacques Patarin: *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): two new families of Asymmetric Algorithms*; Eurocrypt'96, Springer Verlag, pp. 33-48.

[16] Jacques Patarin, Louis Goubin, Nicolas Courtois, + papers of Eli Biham, Aviad Kipnis, T. T. Moh, et al.: *Asymmetric Cryptography with Multivariate Polynomials over a Small Finite Field*; known az 'orange script', compilation of different papers with added materials. Available from J.Patarin@frlv.bull.fr.

[17] David Pointcheval: *A new Identification Scheme Based on the Perceptrons Problem*; In Advances in Cryptology, Proceedings of Eurocrypt'95, LNCS 921, pp.319-328, Springer-Verlag.

[18] David Pointcheval: *Les preuves de connaissance et leurs preuves de sécurité*, PhD thesis, December 1996, Caen University, France.

[19] Adi Shamir: *An efficient Identification Scheme Based on Permuted Kernels*, In Advances in Cryptology, Proceedings of Crypto'89, LNCS 435, pp.606-609, Springer-Verlag.

[20] Adi Shamir, Aviad Kipnis: *Cryptanalysis of the HFE Public Key Cryptosystem*; In Advances in Cryptology, Proceedings of Crypto'99, Springer-Verlag, LNCS.

[21] Adi Shamir, Nicolas Courtois, Alexander Klimov , Jacques Patarin:*Efficient Algorithms for Solving Overdefined Systems of Multivariate Polynomial Equations*, accepted at Eurocrypt 2000.

[22] Jacques Stern: *A new identification scheme based on syndrome decoding*; In Advances in Cryptology, Proceedings of Crypto'93, LNCS 773, pp.13-21, Springer-Verlag.

[23] Jacques Stern: *Designing identification schemes with keys of short size*; In Advances in Cryptology, Proceedings of Crypto'94, LNCS 839, pp.164-73, Springer-Verlag.

[24] Jacques Stern: *A method for finding codewords of small weight*; Coding Theory and Applications, LNCS 434, pp.173-180, Springer-Verlag.

[25] Jacques Stern, Florent Chabaud: *The cryptographic security of the syndrome decoding problem for rank distance codes*. In Advances in Cryptology, Asiacrypt'96, LNCS 1163, pp. 368-381, Springer-Verlag.

[26] L.G. Valiant: *Completeness classes in algebra*. In Proc. Eleventh Ann. ACM Symp. Theor. Comp., pp. 249-261, 1979.

[27] Pascal Véron, *Cryptanalysis of Harari's identification scheme*; Cryptography and Coding, vol. 1025, pp.264-269, LNCS 1025, Springer Verlag, 1995.

[28] Pascal Véron, *Problème SD, Opérateur Trace, Schémas d'Identification et Codes de Goppa*; PhD thesis in french, l'Université de Toulon et du Var, France, july 1995.

Appendix 1 - Multi-user setting

It is quite easy to produce almost totally random instances of MinRank with several users, each of which has one solution to MinRank and no information about other people's solutions. We pick m random matrices, and then each user has the right to add a matrix that is of small rank modulo linear combination of the others. This setting is secure as long as the number of users is smaller than about $m/2$.

Anonymous Group signatures with MinRank

A classical way of transforming any Zero-knowledge protocol into a signature scheme works with MinRank. If combined with the above multi-user setting, it gives group signatures with the following properties:

- Each user signs with his own secret key on behalf of the group.
- The verification is the same for each user.
- **Anonymity:** The verifier ignores which user signed a document.
- Security will be guaranteed by the NP-complete problem MinRank under the random oracle assumption.

Appendix 2 - achieving cheating probability 1/2

We found a new non-trivial trick to achieve the cheating probability 1/2 instead of 2/3 with exactly the same protocol. It has the following interesting features:

- It can be adapted for other Zero-knowledge schemes.
- It requires additional assumption of type one-way function.
- We base it on an NP-compete problem but we also present a version based on factoring.
- Should this assumption fail, the scheme is still at least as secure as before, only with the old (worse) probability.

The principle of the "trick" is to add an additional "verifiable" requirement on generation of (random) values involved in the protocol, in order to eliminate some fraud scenarios (but not others). Thus we can modify the probabilities of different questions in order to balance the probabilities (only) for the remaining fraud scenarios.

Given a Zero-knowledge protocol in which a prover picks up 2 values β_1 and β_2 such that $\beta_2 - \beta_1 = \alpha$ is a given (usually secret) value. Usually we will generate β_1 at random and compute β_2, which enables fraud scenarios in which the adversary may chose β_1 or β_2. We avoid this.

Let F be a function with a following properties:

1. It is very hard to compute $F^{-1}(y)$.
2. It is very easy to compute two solutions x and x' such that $F(x')-F(x)$ is a given value Δy and $x' = x + \Delta x$ with a given constant Δx.

Example 1: $F : x \mapsto x^2 mod N$, N being an RSA modulus. The inversion problem 1. is provably based on intractability of integer factorisation.

Example 2: F is a set of random quadratic equations over a finite field, the inversion problem 1. is NP-complete and hard in practice, see [21].

In both examples, 2. is a linear problem easily solved.

We note that each of the above examples is applied with an operation '+' that belong to a different group. Only the first example can be used for MinRank, as our '+' will be the component-by-component addition in the finite field.

10.1 Application to MinRank scheme

In the modified protocol, the Prover computes the output difference for F, Δy as a pseudo-random generator value $\Delta y = Expand(z)$ for a random 80-bit seed z he picks.

Then he computes β_i by solving

$$F(\beta_2) - F(\beta_1) = \Delta y.$$

Thus he can't chose any of β_i.

The latter equation is linear in β_1 after substitution of $\beta_2 = \beta_1 + \alpha$. In practice we need to try several times if there is no such solutions.

In the previous version of MinRank scheme possible fraud scenarios were:

1. Try to be able to answer $q = 0$ and 1.
2. Try to be able to answer $q = 0$ and 2.
3. Try to be able to answer $q = 1$ and 2.

It is easy to see that the new version excludes the scenarios (1) and (2). Thus we may modify the probabilities. The question $q = 0$ is asked with probability $1/2$ and questions $q = 1, 2$ with probabilities of $1/4$. In such case, a false Prover is detected with probability $1/2$.

How do we verify that P follows the described scenario ? In the first step of the Minrank scheme P will send z and $F(\beta_1)$ along with the hash values already specified. It gives additional $80 + m \log_2 |R|$ bits. For parameters [A] the transmission increases therefore from 1645 to 1823 bits, however we need only 20 rounds to achieve the security of 10^{-6}.

Appendix 3 - the affine MinRank

All the results of the present paper can be exteded to the affine case of MinRank defined as the following: The problem is, as before to find (at least one) solution $\alpha \in R^m$ such that:

$$Rank(\sum_i \alpha_i M_i + M_0) \leq r.$$

To prove in Zero-knowledge this (slightly) more general problem we modify (slightly) our scheme. The only difference it the term $+\mathbf{TM_0S}$ printed in **bold characters** which is added in some places.

One round of Affine MinRank authentication:

1. The Prover sends to the Verifier:

$$\longrightarrow$$

$$H(X),\ H(TN_1S + X),\ H(TN_2S + X + \mathbf{TM_0S}),\ H(S, T)$$

2. The Verifier chooses a query $q \in \{0, 1, 2\}$ and sends q to P.

$$\longleftarrow$$

$$q \in \{0, 1, 2\}$$

3. If $q = 0$ the Prover gives the following values:

$$\longrightarrow$$
$$(TN_1S + X),\ (TN_2S + X + \mathbf{TM_0S})$$

Verification $q = 0$**:** The Verifier accepts if $H(TN_1S+X)$, $H(TN_2S+X+\mathbf{TM_0S})$ are correct and

$$(TN_2S + X + \mathbf{TM_0S}) - (TN_1S + X) = TMS$$

is indeed a matrix of rank r.

3' If $q = 1, 2$ the Prover reveals:

$$\longrightarrow$$
$$S,\ T,\ \beta_q, X$$

Verification $q = 1, 2$**:** The Verifier checks if S and T are invertible and $H(S, T)$ is correct. Then he computes

$$TN_qS = \sum \beta_{qi}\ \ TM_iS$$

and verifies $H(TN_1S+X)$ or $H(TN_2S+X+\mathbf{TM_0S})$ as well as $H(X)$.

By inspection we verify that all results in the present paper hold for this modified scheme, and that it does not increase the communication complexity of the MinRank authentication scheme.