23rd International Symposium on Mathematical Theory of Networks and Systems
Hong Kong University of Science and Technology, Hong Kong, July 16-20, 2018

Weight Two Masking of the Reed-Solomon Structure in Conjunction with List Decoding

Karan Khathuria Joachim Rosenthal Violetta Weger

Abstract*— We present a code-based cryptosystem, in which we use Reed-Solomon codes as secret codes and a weight two matrix for masking, to make the system secure against attacks based on the Schur product. We combine this with the Guruswami-Sudan list decoding for decryption to get lower key sizes. As a consequence, we obtain a key size reduction of 21.8% compared to the standard McEliece cryptosystem proposed by Bernstein *et al.

***Index Terms*— Code-based Cryptography, McEliece Cryptosystem, List Decoding, Reed-Solomon codes**

2010 Mathematics Subject Classification: 94B05, 94A60

I. Introduction

Code-based cryptography first came up with the McEliece system [19], which uses a binary irreducible Goppa code as secret code. To hide this code a permutation matrix and an invertible matrix are used for scrambling. Eventhough there is no structural attack on this original proposal, the large key size (see for example [5]) makes it desirable to search for systems with smaller key sizes. The Reed-Solomon (RS) code is the most prefered alternative to reduce the key size. The proposal to use RS codes directly in the McEliece system is broken by the attack of Sidelnikov and Shestakov [26]. To thwart the attack of Sidelnikov and Shestakov the BBCRS scheme ([1], [2]) is using the sum $T+R$ for masking, where T is a matrix of weight m and R is a matrix of rank z. For key size and complexity reasons they focus on small m and z. This proposal was attacked for some parameters by Couvreur *et al.* ([10], [11]), where they use the Schur product of the public code to find a permutation equivalent code to the secret RS code. Since the attack works for $m < 2$, in the weight two masking proposal by Bolkema *et al.* in [6], it is claimed that a matrix of constant row weight two will hide the structure of the secret RS code, even under the Schur product.

Another idea to reduce the key size came from Barbier and Barreto in [3] by using list decoding in the decryption step of the original McEliece system using Goppa codes. They use the Guruswami-Sudan list decoding algorithm [16], which assymptotically corrects up to $n - \sqrt{kn}$ errors, where n is the length of the code and k is the dimension. In our cryptosystem we use the Guruswami-Sudan list decoding for RS codes along with improvements by Kötter [20] and Roth-Ruckenstein [24].

In this article we propose a cryptosystem, where we combine the idea of the weight two masking on the Reed-Solomon structure and using list decoding in the decryption.

K. Khathuria, J. Rosenthal, and V. Weger are with the Institute of Mathematics, University of Zurich, Switzerland.

Corresponding author: Joachim Rosenthal, Email: rosenthal@math.uzh.ch

In Section II we explain the Guruswami-Sudan list decoding algorithm with the number of errors it can correct, the list size and the complexity of the algorithm. In Section III we present the cryptosystem in detail with the hash function used in the encryption and the list decoding used in the decryption. In the Section IV we make a cryptoanalysis of the proposed system, where we give experimental results providing strong evidence for security against the Schur product attack. In the Section V we compare the key size of the proposed cryptosystem against the key size using unique decoding and against the standard McEliece system in [5].

II. List decoding of GRS codes

We first recall the definition of a generalized Reed-Solomon (GRS) code. Let $\mathbb{F}_q$ be a finite field and let $1 \leq k < n \leq q$ be integers. Let $\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ be an n-tuple of distinct elements and $\beta = (\beta_1, \ldots, \beta_n) \in \mathbb{F}_q^n$ be an n-tuple of nonzero elements, then the generalized Reed-Solomon code $\mathrm{GRS}_{n,k}(\alpha, \beta)$ of dimension k is the set of vectors $(\beta_1 p(\alpha_1), \ldots, \beta_n p(\alpha_n))$, where p ranges over all polynomials of degree less than k having coefficients in $\mathbb{F}_q$. Thus

$$\mathrm{GRS}_{n,k}(\alpha, \beta) = \left\{ (\beta_1 p(\alpha_1), \ldots, \beta_n p(\alpha_n)) \;\middle|\; p \in \mathbb{F}_q[x],\ \deg(p) < k \right\}.$$

The GRS code has the minimum Hamming distance $d = n - k + 1$, and hence can uniquely correct upto $d/2$ errors using efficient algorithms such as the Berlekamp-Massey algorithm ([4], [18]). In 1999, Guruswami and Sudan [16] published a polynomial time list decoding algorithm for Reed-Solomon codes that can correct errors beyond the $d/2$ error-correcting bound.

The Guruswami-Sudan (GS) decoding algorithm has an internal parameter m, called the interpolation multiplicity. The bound on the number of errors the GS algorithm can correct is associated to m and is given by

$$t_m = n\left(1 - \sqrt{R\left(\frac{m+1}{m}\right)}\right),$$

where n is the length of the Reed-Solomon code and R is its rate.

Let $\mathcal{C}$ be a Reed-Solomon code $RS_{n,k}(\alpha)$ of length n and dimension k over a finite field $\mathbb{F}_q$. Given $z = (z_1, \ldots, z_n)$ in $\mathbb{F}_q^n$, the GS algorithm finds all the polynomials $p(x)$ of degree less than k, such that the codeword $(p(\alpha_1), \ldots, p(\alpha_n))$ has Hamming distance $\leq t_m$ from z. The GS algorithm involves two major steps:

1) (Interpolation step) Construct a bivariate polynomial $Q(x,y) = \sum_{i,j} a_{i,j}x^i y^j$ such that Q has a zero of multiplicity m at each of the points (α_i, z_i) and the $(1, k-1)$-weighted degree of $Q(x,y)$ is minimal.
2) (Factorization step) Compute the factors of $Q(x,y)$ of the form $y - p(x)$ with degree of $p(x)$ less than k.

The output of the algorithm is a list $\mathcal{L}_m$ of codewords of $\mathcal{C}$, which includes all the codewords with Hamming distance $\leq t_m$ from z. The size of the list is bounded above by

$$\ell_m = \left(m + \frac{1}{2}\right)\sqrt{\frac{n}{k-1}}.$$

In [16] the main aim of Guruswami and Sudan was to show the existence of a polynomial-time list decoding algorithm and not the efficiency of the algorithm. However, several authors have contributed to improve the efficiency of the key steps in the GS algorithm. Some noteworthy contributions are by Kötter, described by McEliece in [20], for the interpolation step and by Roth-Ruckenstein [24] for the factorization step. Using Kötter's improvement, inspired by the Feng-Tzeng algorithm [13], the interpolation algorithm takes $\mathcal{O}(n^2m^4)$ field operations. Whereas the factorization algorithm, using the Roth-Ruckenstein improvement, takes $\mathcal{O}(n^2m^2)$ field operations. Hence the overall complexity of the GS algorithm is $\mathcal{O}(n^2m^4)$ field operations. For more details on the GS algorithm and improvements by Kötter and Roth-Ruckenstein we refer the reader to [20].

In the following cryptosystem we use the GS algorithm for decryption with an aim to reduce the size of the keys. Although the running time of the GS list decoding algorithm is high, the trade-off between the running time and the key size can easily be achieved. In the following proposed cryptosystem we used the interpolation multiplicity $m = \lfloor n^{1/2} \rfloor$ and achieved nearly 7 % reduction in the key sizes compared to unique decoding, see Section V.

III. The cryptosystem

In this section we will present the proposed cryptosystem in the McEliece version. The Niederreiter version of the proposed cryptosystem is similar to the McEliece version, using the $(n-k)\times n$ parity check matrix H of $\mathcal{C}$ and in the encryption one computes the syndrome of the message.

For the key generation take a GRS code $\mathcal{C} = \mathrm{GRS}_{n,k}(\alpha, \beta)$ of dimension k and length n over the finite field $\mathbb{F}_q$ and choose a generator matrix G of $\mathcal{C}$, given in the canonical form

$$G = \begin{pmatrix} \beta_1 & \cdots & \beta_n \\ \beta_1\alpha_1 & \cdots & \beta_n\alpha_n \\ \vdots & & \vdots \\ \beta_1\alpha_1^{k-1} & \cdots & \beta_n\alpha_n^{k-1} \end{pmatrix}.$$

Choose a random $k \times k$ invertible matrix S and an invertible $n \times n$ matrix Q of constant row weight two, both over $\mathbb{F}_q$. Then we compute $G' = SGQ^{-1}$. Let $R = \frac{k}{n}$ be the rate of the code $\mathcal{C}$. Let m be the interpolation multiplicity of the GS algorithm, which we use during decryption. Since Q is of constant weight two, the amount of the errors we add in the encryption step is then given by

$$t = \left\lfloor \frac{t_m}{2} \right\rfloor = \left\lfloor \frac{n}{2}\left(1 - \sqrt{R\left(\frac{m+1}{m}\right)}\right)\right\rfloor.$$

Since the GS algorithm gives us a list of possible messages, we also send hash of the message in the cipher in order to recover the sent message. Let $\mathcal{H}$ be a fixed hash function, globally known, with output size of h bits. The value of h depends on the list size ℓ_m in such a way that we do not encounter second pre-images in the list of hash values of possible messages.

The public key is given by (G', t, h).

The encryption step works as follows. Let $x \in \mathbb{F}_q^k$ be the message. Then compute

$$y = xG' + e,$$

where $e \in \mathbb{F}_q^n$ is an error vector of weight less than or equal to t. The cipher is then given by $(y, \mathcal{H}(x))$. For the decryption one computes

$$y' = yQ = xSG + eQ.$$

Since $\mathrm{wt}(eQ) \leq \lfloor t_m \rfloor$, we list decode y' to get a list $\mathcal{L}_m$ of possible messages, say

$$\mathcal{L}_m = \{z_1, \ldots, z_{\ell_m}\}.$$

In order to recover the original message x from the list, we compute $\mathcal{H}(z_i)$ for all $i \in \{1, \ldots, \ell_m\}$ and compare it with $\mathcal{H}(x)$. The sent message x is given by the z_j for which $\mathcal{H}(z_j) = \mathcal{H}(x)$.

The output size h of the hash function should be chosen in such a way that the probability of finding a second pre-image of $\mathcal{H}(x)$ in the list $\{\mathcal{H}(z) | z \in \mathcal{L}_m\}$ is negligible. In order to achieve that, h should be chosen sufficiently larger than $\log_2(\ell_m)$.

Let $z \in \mathcal{L}_m$ with $z \neq x$. With an ideal hash function, the probability that $\mathcal{H}(z) = \mathcal{H}(x)$ is 2^{-h}. Hence the probability of finding a second pre-image of $\mathcal{H}(x)$ is $1 - \left(1 - 2^{-h}\right)^{(\ell_m - 1)}$. From Section II we know that $\ell_m = \mathcal{O}(n^{1/2})$. Assuming that the rate $R > 1/4$ we get $\ell_m \leq 2\lfloor n^{1/2} \rfloor + 1$. Hence h should be chosen sufficiently larger than $\log_2(2n^{1/2} + 1)$. In practice, if $n = 2^{10}$, then $h = 14$ should be sufficient. In this case, the probability of finding a second pre-image is less than 0.0039, hence 0.4 %. Note that the probability of decryption failure can further be reduced by simply increasing the value of h.

Since we are taking $m = \lfloor n^{1/2} \rfloor$ by Section II, we get the decryption complexity to be $\mathcal{O}(n^4)$ field operations.

IV. Security

In this section we will discuss the security of the proposed cryptosystem in Section III. In theory the Niederreiter version has equivalent security to the McEliece version by [17]. However, the McEliece cryptosystem has a disadvantage when the same message is encrypted multiple times (see [7], [8]). In the following we consider the Niederreiter version

of the proposed cryptosystem. Note that in the Niederreiter system we take a $(n-k)\times n$ parity check matrix H of a GRS code and Q an invertible $n\times n$ matrix of constant row weight two. We do not consider the influence of the invertible matrix S, since this gives the same code. Hence the public matrix is given by HQ. Clearly the attack of Sidelnikov and Shestakov [26] can not be applied, since the public code is not permutation equivalent to the secret GRS code.

The security of the weight two masking is already discussed in [1], [2], a scheme of which the weight two masking is a special case of. The only structural attack to the scheme of ([1], [2]) are the attacks based on the Schur product ([14], [11]).

A. Attack based on Schur product of public matrix

For the attack based on the Schur product we need to introduce some definitions and notations.

Definition 1 (Schur product)**.** Let $x, y \in \mathbb{F}_q^n$. We denote by the Schur product of x and y their componentwise product

$$x \star y = (x_1y_1, \ldots, x_ny_n).$$

Remark 2. The Schur product is symmetric and bilinear.

Definition 3 (Schur product of codes and square code)**.** Let $\mathcal{A}, \mathcal{B}$ be two codes of length n. The Schur product of two codes is the vector space spanned by all $a \star b$ with $a \in \mathcal{A}$ and $b \in \mathcal{B}$:

$$\langle \mathcal{A} \star \mathcal{B} \rangle = \langle \{ a \star b \mid a \in \mathcal{A}, b \in \mathcal{B} \} \rangle.$$

If $\mathcal{A} = \mathcal{B}$, then we call $\langle \mathcal{A} \star \mathcal{A} \rangle$ the square code of $\mathcal{A}$ and denote it by $\langle \mathcal{A}^2 \rangle$.

Definition 4 (Schur matrix)**.** Let G be a $k \times n$ matrix, with rows $(g_i)_{1\le i\le k}$. The Schur matrix of G, denoted by $S(G)$ consists of the rows $g_i \star g_j$ for $1 \le i \le j \le k$.

Using Remark 2, we observe that if G is a generator matrix of a code $\mathcal{C}$ then its Schur matrix $S(G)$ is a generator matrix of the square code of $\mathcal{C}$. Let s be the following map

$$\begin{array}{rcl} s : \mathbb{N} & \to & \mathbb{N} \\ k & \mapsto & \frac{1}{2}\left(k^2+k\right). \end{array}$$

For a $k \times n$ matrix A, we observe that $S(A)$ has the size $s(k) \times n$.

In [1], [2], Baldi *et al.* proposed the BBCRS scheme which uses GRS codes as secret codes and as scrambling matrix the sum $T + R$, where T is of row weight m and R is of rank z. In [14], Gauthier-Umaña *et al.* were able to attack this proposal for $m = 1$, $z = 1$ and $k < \frac{n-2}{2}$ or $k > \frac{n+2}{2}$. This attack is based on the fact that the square code of a GRS code has small dimension. Even after the scrambling with $T + R$, the square code dimension is still low whereas for a random secret code the dimension is with high probability maximal (see [9], [12], [22]). With this they can construct a subcode of the public code, which is also a subcode of a permutation equivalent GRS code to the secret code. In [11] Couvreur *et al.* were able to extend this attack for $m \le 1 + k/n < 2$. In this extended attack it is observed that in the Niederreiter version of the BBCRS scheme puncturing the public code gives a small square code dimension. This helps to detect the weights of the rows of T and reduce to the case $z = 1$ and $m = 1$.

Altough these attacks are only for certain values of the scheme, it is not excluded that the whole scheme is vulnerable to the Schur product attacks. The purpose of the weight two masking is to be a countermeasure to these attacks. More precisely we claim that raising m to 2 is enough for the Schur product attacks to fail, which aims in proving that under the weight two masking the square code of the public code has maximal dimension and thus it behaves like a random code. We provide experimental results, which give evidence that this is indeed the case with high probability.

The security of the proposed cryptosystem against the attack based on the Schur product relies on the following; for a parity check matrix H of any GRS code and a random matrix Q of constant row weight two, the Schur matrix of HQ has with high probability maximal rank.

In the experiments, for sufficiently large n the Schur matrix of HQ always had maximal rank. As a consequence, we conjecture the following statement.

Conjecture 5. *Let H be a parity check matrix of a random GRS code of length n and dimension k over a finite field $\mathbb{F}_q$. Let Q represent a weight two matrix having variables $x_1, \ldots, x_n, y_1, \ldots, y_n$ as the nonzero entries. Then the Schur matrix $S(HQ) \in \mathbb{F}_q[x_1, \ldots, x_n, y_1, \ldots, y_n]^{s(n-k)\times n}$ has maximal rank, i.e. there exist a nontrivial $u \times u$ minor of $S(HQ)$, where $u = \min\{s(n-k), n\}$.*

Note that each entry in the i^{th} column of $S(HQ)$ is a homogeneous polynomial of degree 2 in the variables x_i and y_i. Since the variables $y_1, \ldots, y_n$ are representing nonzero elements of $\mathbb{F}_q$, we can normalize y_i in each column. Hence we can assume that $S(HQ) \in \mathbb{F}_q[x_1, \ldots, x_n]^{s(n-k)\times n}$.

If Conjecture 5 holds, we can assume that the nontrivial $u \times u$ minor is the leading $u \times u$ minor. Let $p(x_1, \ldots, x_u)$ be this nontrivial $u \times u$ minor. The total degree of p is at most $2u$ and each individual degree $\deg_{x_i}(p)$ is at most 2. We use the Schwartz-Zippel lemma to get a bound on the number of points in $\left(\mathbb{F}_q^\times\right)^n$ where p is non-zero.

Theorem 6 (Schwartz-Zippel lemma [25], [29])**.** *Let $f \in \mathbb{F}_q[x_1, x_2, \ldots, x_n]$ be a nontrivial polynomial of total degree d over a finite field $\mathbb{F}_q$. Let S be a subset of $\mathbb{F}_q$. Then f is nonzero on at least a fraction $\left(1 - \frac{d}{|S|}\right)$ of points in S.*

We apply the Schwartz-Zippel lemma iteratively on each variable of p with $S = \mathbb{F}_q^\times$, to get the following corollary.

Corollary 7. *Let $p(x_1, \ldots, x_u)$ be the nontrivial $u \times u$ minor of $S(HQ)$. Then p is nonzero on at least a fraction $\left(1 - \frac{2}{q-1}\right)^u$ of points in $\left(\mathbb{F}_q^\times\right)^u$.*

Let $P_H(q, n)$ denote the fraction of weight two matrices giving maximal $S(HQ)$ rank for a given parity check matrix

H.

For example, let H be a parity check matrix of a random GRS code of length $n = 8$ and dimension 4 over the field $\mathbb{F}_9$. Since all the Reed-Solomon codes of length 8 over $\mathbb{F}_9$ are permutation equivalent to each other, $P_H(9,8)$ is invariant of H. Then the Corollary 7 says that $P(9,8) \geq (1-2/8)^8 = 0.1001$. However we computed the exact value of $P(9,8) \approx 0.988$, which is much higher than the bound given by Corollary 7.

In Section IV-B, we see that for fixed field size q and length of the code n, the smallest key size is achieved at the rate $1/2$. Thus for any $n \geq 8$, we have $u = \min\{s(n-k), n\} = n$. The lower bound on $P_H(q,n)$ is then $(1-2/(q-1))^n$. This implies that for a fixed n, the lower bound tends to 1 as q increases.

For fixed $n = 8$ and $n = 9$ respectively, we performed Monte-Carlo experiments to get an estimate on the fraction $P_H(q,n)$ for increasing q. Let $\widetilde{P}_H(q,n)$ denote an estimate of $P_H(q,n)$ computed on Sage by taking 10^7 random constant row weight two matrices Q. By randomly varying the parity check matrix H, we compute the average of $\widetilde{P}_H(q,n)$, denoted by $\mu(\widetilde{P}_H(q,n))$. In Figure 1 and Figure 2, corresponding to $n = 8$ and $n = 9$ respectively, we observe that $\mu(\widetilde{P}_H(q,n))$ tends to 1 much faster than the Schwartz-Zippel lower bound.

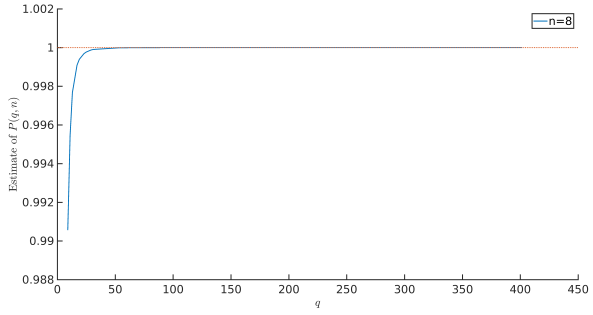

Fig. 1: Estimate of $\mu(\widetilde{P}_H(q,8))$ obtained from Monte-Carlo tests on 10^7 weight two matrices

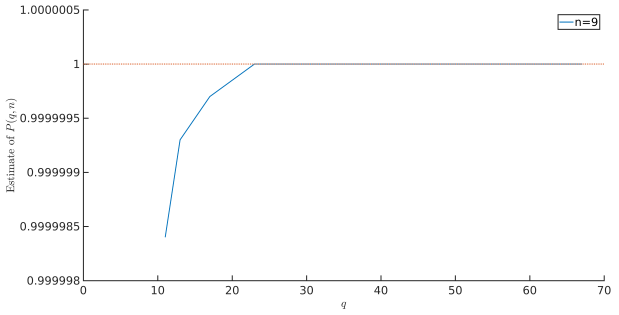

Fig. 2: Estimate of $\mu(\widetilde{P}_H(q,9))$ obtained from Monte-Carlo tests on 10^7 weight two matrices

In further experiments for $n \geq 12$, we noticed the rank of Schur matrix $S(HQ)$ for a randomly chosen weight two matrix Q is always maximal. Experimental and therotical ananlysis on the rank of Schur matrix is also presented in detail by Weger in [28]. These computations infer that, like random linear codes, the rank of Schur matrix $S(HQ)$ is maximal with high probability. As a conclusion, these experiments tend to imply that the proposed cryptosystem is not vulnerable to the attacks based on Schur product of the public matrix.

B. ISD

The information set decoding (ISD) attack (see for example [23], which is a generalization of Stern's algorithm [27]) is a non-structural attack, it decodes a random code without exploiting any structural property of the code, hence it is non-polynomial in the dimension of the code.

The ISD attack takes as input q, n, k, t, where t is the weight of the error vector. In the case of the weight two masking without list decoding we have

$$t = \left\lfloor \frac{n-k}{4} \right\rfloor.$$

In the case of the weight two masking with list decoding we introduce the rate $R = \frac{k}{n}$ and the interpolation multiplicity $m = \lfloor n^{1/2} \rfloor$, and we have

$$t = \left\lfloor \frac{n}{2}\left(1 - \sqrt{R\left(\frac{m+1}{m}\right)}\right)\right\rfloor.$$

We propose to use the following parameters in the proposed cryptosystem to achieve a 80 bit security against the ISD attack; $q = 401, n = 400$ and $k = 200$, hence the rate is $R = \frac{1}{2}$, the interpolation multiplicity is $m = 20$ and $t = 55$. This gives a key of size 360000 bits. Observe that we choose rate $1/2$ as this gives the smallest key size:

Rate	q	n	k	Key Size (bits)
0.3	457	456	136	391680
0.35	431	430	150	378000
0.4	419	418	167	377253
0.45	409	408	183	370575
0.5	401	400	200	360000
0.55	409	408	224	370944
0.6	421	420	252	381024
0.65	439	438	284	393624
0.7	479	478	334	432864

TABLE I: Comparing key sizes for different rates having 80 bit security against ISD attack

For 128 bit security against the ISD attack, we propose to use the parameters $q = 701, n = 700$ and $k = 350$, hence $R = \frac{1}{2}, m = 26$ and $t = 97$. This gives a key of size 1225000 bits. Again rate $1/2$ was the rate achieving smallest key size, see Table II.

The tables are for fixed 2^{80} resp. 2^{128} binary operations costs of the ISD attack, this was computed by a PARI/GP script provided by Peters in [23].

Since we are proposing to use rate $1/2$ observe that there is no key size advantage in using the Niederreiter version instead of the McEliece version.

Rate	q	n	k	Key Size (bits)
0.3	811	810	243	1377810
0.35	761	760	266	1314040
0.4	729	728	291	1271670
0.45	709	708	318	1240200
0.5	701	700	350	1225000
0.55	709	708	389	1240910
0.6	727	726	435	1265850
0.65	751	750	487	1280810
0.7	797	796	557	1331230

TABLE II: Comparing key sizes for different rates having 128 bit security against ISD attack

V. Key size

In this section we compare the key sizes of the different proposals. Table III compares key sizes for fixed 2^{80} binary operations costs of the ISD attack, this was computed by a PARI/GP script provided by Peters in [23]. For 80 bit security we propose to use the weight two masking with list decoding with the parameters $q = 401, n = 400, k = 200$, which gives a key of size 360000 bits. Whereas the weight two masking with unique decoding is proposed for the parameters $q = 479, n = 478, k = 358$, which gives a key size of 386640 bits. The proposed parameters for the McEliece system using Goppa codes by Bernstein *et al.* in [5] are $q = 2^{11}, n = 1632, k = 1269$, which gives a key size of 460647 bits.

Cryptosystem	q	n	k	Key Size
Weight Two, List Decoding	401	400	200	360000
Weight Two, Unique Decoding	479	478	358	386640
McEliece with Goppa Codes	2048	1632	1269	460647

TABLE III: Comparing key sizes (in bits) for different cryptosystems having 80 bit security against ISD attack

We observe that the weight two masking with list decoding reduces the key size of the weight two masking with unique decoding by 6.9% and it reduces the key size of the proposed McEliece system with Goppa codes by 21.8%.

Table IV compares key sizes for fixed 2^{128} binary operations costs of the ISD attack. For 128 bit security we propose to use the weight two masking with list decoding with the parameters $q = 701, n = 700, k = 350$, which gives a key of size 1225000 bits. Whereas the weight two masking with unique decoding is proposed for the parameters $q = 907, n = 906, k = 724$, which gives a key size of 1317680 bits. The proposed parameters for the McEliece system using Goppa codes by Bernstein *et al.* in [5] are $q = 2^{12}, n = 2960, k = 2288$, which gives a key size of 1537536 bits.

Cryptosystem	q	n	k	Key Size
Weight Two, List Decoding	701	700	350	1225000
Weight Two, Unique Decoding	907	906	724	1317680
McEliece with Goppa Codes	4096	2960	2288	1537536

TABLE IV: Comparing key sizes (in bits) for different cryptosystems having 128 bit security against ISD attack

We observe that the weight two masking with list decoding reduces the key size of the weight two masking with unique decoding by 7.0% and it reduces the key size of the proposed McEliece system with Goppa codes by 20.3%.

It is also noteworthy to mention that the weight two masking even with unique decoding obtains nearly 16% and 14% smaller key size compared to the proposed McEliece cryptosystem for 80 bit and 128 bit security, respectively.

VI. Conclusion

In this work we presented a code-based cryptosystem, which uses a Reed-Solomon code as the secret code and an invertible matrix of constant row weight two for masking. This masking appears to be hiding the algebraic structure of the private Reed-Solomon code against all known attacks. In particular, we analysed the effect of the weight two masking on the security against the attack based on the Schur product, which has become an enormous threat to code-based cryptosystems. Furthermore, with a view to reduce the key size, we used the Guruswami-Sudan list decoding algorithm in the decryption step. We recovered the original message from the list by including hash of the message in the cipher. The list decoding allowed us to correct more errors compared to unique decoding and hence results in smaller key size. For example, for 80-bit security level against ISD attack, the key size of the proposed cryptosystem is 360000 bits, which is 21.8% less than the key size of the standard McEliece cryptosystem proposed by Bernstein et al. in [5].

Acknowledgement

This work has been supported by the Swiss National Science Foundation under grant no. 169510. We would like to thank Davide Schipani and Heide Gluesing-Luerssen for useful discussions during the preparation of this paper. We would also like to thank Alessandro Neri for helpful suggestions on the security analysis of the cryptosystem against the Schur product attack.

Notes

The Guruswami-Sudan error correction capacity bound can be improved, for example with the Parvaresh-Vardy [21] or Guruswami-Rudra algorithm [15], when using folded Reed-Solomon codes. A folded Reed-Solomon code is a Reed-Solomon code viewed over an extension field. We observed that the folded Reed-Solomon code cannot be used directly in the key generation, since it is a non-linear code. Nevertheless, one can use a Reed-Solomon code during encryption and fold the received cipher with a folding parameter m. To get a better error correction bound, one needs to bundle the error positions in the encryption step, and in order not to destroy this bundling, one should also use a weight two matrix of block diagonal form. We noted, that the public key is then vulnerable to ISD attack on the smaller subcodes.

REFERENCES

[1] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani. A Variant of the McEliece Cryptosystem with Increased Public Key Security. In *Proceedings of the Seventh International Workshop on Coding and Cryptography (WCC) 2011*, pages 173 – 182, 2011.

[2] M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, and D. Schipani. Method and Apparatus for Public-Key Cryptography Based on Error Correcting Codes, November 17 2015. US Patent 9,191,199.

[3] M. Barbier and P. SLM Barreto. Key Reduction of McEliece's Cryptosystem using List Decoding. In *Information Theory Proceedings (ISIT), 2011 IEEE International Symposium on*, pages 2681–2685. IEEE, 2011.

[4] E. R. Berlekamp. *Algebraic Coding Theory*. McGraw-Hill, New York, 1968.

[5] D. Bernstein, T. Lange, and C. Peters. Attacking and defending the McEliece cryptosystem. *Post-Quantum Cryptography*, pages 31–46, 2008.

[6] J. Bolkema, H. Gluesing-Luerssen, C.A. Kelley, K.E. Lauter, B. Malmskog, and J. Rosenthal. Variations of the McEliece Cryptosystem. In *Algebraic Geometry for Coding Theory and Cryptography*, pages 129–150. Springer, 2017.

[7] A. Canteaut. *Attaques de cryptosystèmes à mots de poids faible et construction de fonctions t-résilientes*. PhD thesis, Univ. Paris 6, 1996.

[8] A. Canteaut and F. Chabaud. A new algorithm for finding minimum-weight words in a linear code: application to McEliece's cryptosystem and to narrow-sense BCH codes of length 511. 44(1):367–378, 1998.

[9] I. Cascudo, R. Cramer, D. Mirandola, and G. Zémor. Squares of random linear codes. *IEEE Transactions on Information Theory*, 61(3):1159–1173, 2015.

[10] A. Couvreur, P. Gaborit, V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich. Distinguisher-Based Attacks on Public-Key Cryptosystems using Reed-Solomon Codes. *Designs, Codes and Cryptography*, 73(2):641–666, 2014.

[11] A. Couvreur, A. Otmani, J.-P. Tillich, and V. Gauthier-Umaña. A Polynomial-Time Attack on the BBCRS Scheme. *Public-key cryptography—PKC 2015*, 9020:175–193, 2015.

[12] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A Distinguisher for High-Rate McEliece Cryptosystems. *IEEE Transactions on Information Theory*, 59(10):6830–6844, 2013.

[13] G. L. Feng and K. K. Tzeng. A Generalized Euclidean Algorithm for Multisequence Shift-Register Synthesis with Appl. to Decoding Cyclic Codes. *IEEE Trans. Inform. Theory*, IT-37(5):1274–1287, 1991.

[14] V. Gauthier-Umaña, A. Otmani, and J.-P. Tillich. A Distinguisher-Based Attack on a Variant of McEliece's Cryptosystem Based on Reed-Solomon Codes. *arXiv preprint arXiv:1204.6459*, 2012.

[15] V. Guruswami and A. Rudra. Explicit Codes achieving List Decoding Capacity: Error-Correction with Optimal Redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, Jan 2008.

[16] V. Guruswami and M. Sudan. Improved Decoding of Reed-Solomon and Algebraic-Geometry Codes. 45(6):1757–1767, 1999.

[17] Y. X. Li, R. H. Deng, and X. M. Wang. On the Equivalence of McEliece's and Niederreiter's Public-Key Cryptosystems. *IEEE Transactions on Information Theory*, 40(1):271–273, 1994.

[18] J. L. Massey. Shift-Register Synthesis and BCH Decoding. *IEEE Trans. Inform. Theory*, IT-15:122–127, 1969.

[19] R. J. McEliece. A Public-Key Cryptosystem Based on Algebraic Coding Theory. Technical report, DSN Progress report, Jet Propulsion Laboratory, Pasadena, 1978.

[20] R. J. McEliece. The Guruswami–Sudan Decoding Algorithm for Reed–Solomon Codes. *Interplanetary Network Progress Report*, 153:1–60, 2003.

[21] F. Parvaresh and A. Vardy. Correcting Errors Beyond the Guruswami-Sudan Radius in Polynomial Time. *Foundations of Computer Science, 2005. FOCS 2005. 46th Annual IEEE Symposium on*, pages 285–294, Oct. 2005.

[22] R. Pellikaan and I. Márquez-Corbella. Error-Correcting Pairs for a Public-Key Cryptosystem. In *Journal of Physics: Conference Series*, volume 855, page 012032. IOP Publishing, 2017.

[23] C. Peters. Information-Set Decoding for Linear Codes over $\mathbb{F}_q$. *PQCrypto*, 2010:81–94, 2010. `http://christianepeters.wordpress.com/publications/tools/`.

[24] R. M. Roth and G. Ruckenstein. Efficient Decoding of Reed-Solomon Codes Beyond Half the Minimum Distance. *IEEE Transactions on Information Theory*, 46(1):246–257, Jan 2000.

[25] J. T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980.

[26] V. M. Sidelnikov and S. O. Shestakov. On Insecurity of Cryptosystems Based on Generalized Reed-Solomon Codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992.

[27] J. Stern. A Method for Finding Codewords of Small Weight. *Coding Theory and Applications*, pages 106–113, 1989.

[28] V. Weger. A Code-Based Cryptosystem using GRS Codes. Master Thesis at the University of Zürich (Switzerland), 2016.

[29] R. Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, London, UK, UK, 1979. Springer-Verlag.