Random Linear Code Based Public Key Encryption Scheme RLCE

Yongge Wang

Abstract

Lattice based encryption schemes and linear code based encryption schemes have received extensive attention in recent years since they have been considered as post-quantum candidate encryption schemes. Though LLL reduction algorithm has been one of the major cryptanalysis techniques for lattice based cryptographic systems, cryptanalysis techniques for linear code based cryptographic systems are generally scheme specific. In recent years, several important techniques such as Sidelnikov-Shestakov attack, filtration attacks, and algebraic attacks have been developed to crypt-analyze linear code based encryption schemes. Though most of these cryptanalysis techniques are relatively new, they prove to be very powerful and many systems have been broken using these techniques. Thus it is important to design linear code based cryptographic systems that are immune against these attacks. This paper proposes linear code based encryption scheme RLCE which shares many characteristics with random linear codes. Our analysis shows that the scheme RLCE is secure against existing attacks and we expect that the security of the RLCE scheme is equivalent to the hardness of decoding random linear codes. Example RLCE scheme parameters for different security levels are included in the paper also.

Key words: Random linear codes; McEliece Encryption scheme; secure public key encryption scheme; linear code based encryption scheme

MSC 2010 Codes: 94B05; 94A60; 11T71; 68P25

I. Introduction

With rapid development for quantum computing techniques, our society is concerned with the security of current Public Key Infrastructures (PKI) which are fundamental for Internet services. The core components for current PKI infrastructures are based on public cryptographic techniques such as RSA and DSA. However, it has been shown that these public key cryptographic techniques could be broken by quantum computers. Thus it is urgent to develop public key cryptographic systems that are secure against quantum computing.

Since McEliece encryption scheme [18] was introduced more than thirty years ago, it has withstood many attacks and still remains unbroken for general cases. It has been considered as one of the candidates for post-quantum cryptography since it is immune to existing quantum computer algorithm attacks. The original McEliece cryptographic system is based on binary Goppa codes. Several variants have been introduced to replace Goppa codes in the McEliece encryption scheme. For instance, Niederreiter [20] proposed the use of generalized Reed-Solomon codes and later, Berger and Loidreau [3] proposed the use of sub-codes of generalized Reed-Solomon codes. Sidelnikov [23] proposed the use of Reed-Muller codes, Janwa and Moreno [11] proposed the use of algebraic geometry codes, Baldi et al [1] proposed the use of LDPC codes, Misoczki et al [19] proposed the use of MDPC codes, and Löndahl and Johansson [14] proposed the use of convolutional codes. Most of them have been broken though MDPC/LDPC code based McEliece encryption scheme [1], [19] and the original binary Goppa code based McEliece encryption scheme are still considered secure.

Goppa code based McEliece encryption scheme is hard to attack since Goppa codes share many characteristics with random codes. Motivated by Faugere et al's [10] algebraic attacks against quasi-cyclic and dyadic structure based compact variants of McEliece encryption scheme, Faugere et al [9] designed an efficient algorithm to distinguish a random code from a high rate Goppa code. Márquez-Corbella and Pellikaan [15] simplified the distinguisher in [9] using Schur component-wise product of codes. The Schur product codes are used to define the square of a code which can be used to distinguish a high rate Goppa code from a random code.

Sidelnikov and Shestakov [24] show that for the generalized Reed-Solomon code based McEliece encryption scheme, one can efficiently recover the private parameters for the generalized Reed-Solomon code from the public

Part of this work was done when the author was visiting Qatar University.

key. Using component-wise product of codes and techniques from [24], Wieschebrink [29] shows that Berger and Loidreau's sub codes [3] of Niederreiter's scheme could be broken efficiently also. Couvreur et al [6] proposed a general distinguisher based filtration technique to recover keys for generalized Reed-Solomon code based McEliece scheme and Couvreur, Márquez-Corbella, and Pellikaan [7] used filtration attacks to break Janwa and Moreno's [11] algebraic geometry code based McEliece encryption scheme. The filtration attack was recently used by Couvreur et al [8] to attack Bernstein et al's [4] wild Goppa code based McEliece scheme.

General Goppa code based McEliece schemes are still immune from these attacks. However, based on the new development of cryptanalysis techniques against linear code based cryptographic systems in the recent years, it is important to systematically design random linear code based cryptographic systems defeating these attacks. Motivated by this observation, this paper presents a systematic approach of designing public key encryption schemes using any linear codes. For example, we can even use Reed-Solomon codes to design McEliece encryption scheme while it is insecure to use Reed-Solomon codes in the original McEliece scheme. Since our design of linear code based encryption scheme embeds randomness in each column of the generator matrix, it is expected that, without the corresponding private key, these codes are as hard as random linear codes for decoding.

The most powerful attacks on McEliece cryptosystem is the information-set decoding attack which was introduced by Prange [22]. In an information-set decoding approach, one finds a set of coordinates of a received ciphertext which are error-free and that the restriction of the code's generator matrix to these positions is invertible. The original message can then be computed by multiplying the ciphertext with the inverse of the sub-matrix. Improvements of the information-set decoding attack have been proposed by Lee-Brickell [12], Leon [13], Stern [25], May-Meurer-Thomae [16], Becker-Joux-May-Meurer [2], and May-Ozerov [17]. Bernstein, Lange, and Peters [5] presented an exact complexity analysis on information-set decoding attack against McEliece cryptosystem. The attacks in [2], [5], [12], [13], [16], [17], [25] are against binary linear codes and are not applicable when the underlying field is $GF(p^m)$ for a prime p. Peters [21] presented an exact complexity analysis on information-set decoding attack against McEliece cryptosystem over $GF(p^m)$. These information-set decoding techniques (in particular, the exact complexity analysis in [5], [21]) are used to select example parameters for RLCE scheme in Section V.

Unless specified otherwise, we will use $q = 2^m$ or $q = p^m$ for a prime p and our discussion are based on the field $GF(q)$ through out this paper. Bold face letters such as $\mathbf{a}, \mathbf{b}, \mathbf{e}, \mathbf{f}, \mathbf{g}$ are used to denote row or column vectors over $GF(q)$. It should be clear from the context whether a specific bold face letter represents a row vector or a column vector.

II. Goppa codes and McEliece Public Key Encryption scheme

In this section, we briefly review Goppa codes and McEliece scheme. For given parameters $q, n \leq q$, and t, let $g(x)$ be a polynomial of degree t over $GF(q)$. Assume that $g(x)$ has no multiple zero roots and $\alpha_0, \cdots, \alpha_{n-1} \in GF(q)$ be different non root elements for $g(x)$. The following subspace $\mathcal{C}_{Goppa}(g)$ defines the code words of an $[n, k, d]$ binary Goppa code where $d \geq 2t + 1$. This binary Goppa code $\mathcal{C}_{Goppa}(g)$ has dimension $k \geq n - tm$ and correct t errors.

$$\mathcal{C}_{Goppa}(g) = \left\{ c \in \{0,1\}^n : \sum_{i=0}^{n-1} \frac{c_i}{x - \alpha_i} \equiv 0 \bmod g(x) \right\}.$$

Furthermore, if $g(x)$ is irreducible, then $\mathcal{C}_{Goppa}(g)$ is called an irreducible Goppa code. The parity check matrix H for the Goppa codes looks as follows:

$$V_t(\mathbf{x}, \mathbf{y}) = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_0^1 & \alpha_1^1 & \cdots & \alpha_{n-1}^1 \\ & & \ddots & \\ \cdots & \cdots & & \cdots \\ \alpha_0^t & \alpha_1^t & \cdots & \alpha_{n-1}^t \end{pmatrix} \begin{pmatrix} \frac{1}{g(\alpha_0)} & & \\ & \ddots & \\ & & \frac{1}{g(\alpha_{n-1})} \end{pmatrix} \tag{1}$$

where $\mathbf{x} = [\alpha_0, \ldots, \alpha_{n-1}]$ and $\mathbf{y} = \left[\frac{1}{g(\alpha_0)}, \ldots, \frac{1}{g(\alpha_{n-1})}\right]$.

The McEliece scheme [18] is described as follows. For the given parameters n and t, choose a binary Goppa code based on an irreducible polynomial $g(x)$ of degree t. Let G be the $k \times n$ generator matrix for the Goppa code. Select a random dense $k \times k$ nonsingular matrix S and a random $n \times n$ permutation matrix P. Then the public

key is $G' = SGP$ which generates a linear code with the same rate and minimum distance as the code generated by G. The private key is G.

Encryption. For a k-bit message block $\mathbf{m}$, choose a random row vector $\mathbf{z}$ of length n and weight t. Compute the cipher text $\mathbf{x} = \mathbf{m}G' + \mathbf{z}$

Decryption. For a received ciphertext $\mathbf{x}$, first compute $\mathbf{x}' = \mathbf{x}P^{-1}$. Next use an error-correction algorithm to recover $\mathbf{m}' = \mathbf{m}S$ and compute the message $\mathbf{m}$ as $\mathbf{m} = \mathbf{m}'S^{-1}$.

III. Random linear code based secure encryption scheme RLCE

In this section, we present the design of Random Linear Code based Encryption scheme RLCE. Let n, $k, d, t > 0$, and $1 \leq r < \sqrt{k}$ be given parameters such that $n - k + 1 \geq d \geq 2t + 1$. Let $G = [\mathbf{g}_0, \cdots, \mathbf{g}_{n-1}]$ be a $k \times n$ generator matrix for an $[n, k, d]$ linear code. Assume that there is an efficient decoding algorithm to correct at least t errors for this linear code given by G. The private and public keys are generated using the following steps.

1) Let $C_0, C_1, \cdots, C_{n-1} \in GF(q)^{k\times r}$ be $k \times r$ matrices drawn uniformly at random and let
$$G_1 = [\mathbf{g}_0, C_0, \mathbf{g}_1, C_1 \cdots, \mathbf{g}_{n-1}, C_{n-1}] \tag{2}$$
be the $k \times n(r + 1)$ matrix obtained by inserting the random matrices C_i into G.
2) Let $A_0, \cdots, A_{n-1} \in GF(q)^{(r+1)\times(r+1)}$ be dense nonsingular $(r + 1) \times (r + 1)$ matrices chosen uniformly at random and let
$$A = \begin{pmatrix} A_0 & & & \\ & A_1 & & \\ & & \ddots & \\ & & & A_{n-1} \end{pmatrix} \tag{3}$$
be an $n(r + 1) \times n(r + 1)$ nonsingular matrix.
3) Let S be a random dense $k \times k$ nonsingular matrix and P be an $n(r + 1) \times n(r + 1)$ permutation matrix.
4) The public key is the $k \times n(r + 1)$ matrix $G' = SG_1AP$ and the private key is S, G_1, A, P.

Encryption. For a row vector message block $\mathbf{m} \in GF(q)^k$, choose a random row vector $\mathbf{e} = [e_0, \ldots, e_{(n(r+1)-1}] \in GF(q)^{n(r+1)}$ such that the Hamming weight of $\mathbf{e}$ is at most t. The cipher text is $\mathbf{x} = \mathbf{m}G' + \mathbf{e}$.

Decryption. For a received cipher text $\mathbf{x} = [x_0, \ldots, x_{n(r+1)-1}]$, compute
$$\mathbf{x}P^{-1}A^{-1} = [y_0, \ldots, y_{n(r+1)-1}] = \mathbf{m}SG_1 + \mathbf{e}P^{-1}A^{-1}$$
where
$$A^{-1} = \begin{pmatrix} A_0^{-1} & & & \\ & A_1^{-1} & & \\ & & \ddots & \\ & & & A_{n-1}^{-1} \end{pmatrix} \tag{4}$$
Let $\mathbf{y} = [y_0, y_{r+1}, \cdots, y_{(n-1)(r+1)}]$ be the row vector of length n selected from the length $n(r + 1)$ row vector $\mathbf{x}P^{-1}A^{-1}$. Then it is straightforward to check that $\mathbf{y} = \mathbf{m}SG + \mathbf{e}'$. For $i \leq n - 1$, the element $y_{i(r+1)}$ of $\mathbf{y}$ contains error if and only if the sub-vector $[e_{i(r+1)}, \ldots, e_{i(r+1)+r}]$ of $\mathbf{e}$ is a non-zero vector. Since $\mathbf{e}$ contains at most t errors, it follows that there are at most t non-zero sub-vector $[e_{i(r+1)}, \ldots, e_{i(r+1)+r}]$. Thus $\mathbf{y}$ contains at most t errors and the Hamming weight of $\mathbf{e}' \in GF(q)^n$ is at most t. Using the efficient decoding algorithm, one can compute $\mathbf{m}' = \mathbf{m}S$. Finally, $\mathbf{m}$ is computed by $\mathbf{m} = \mathbf{m}'S^{-1}$.

IV. Robustness of RLCE codes against existing attacks

A. Randomness of generator matrix columns

We first use the following theorem to show that any single column of the underlying generator matrix G could be completely randomized in a RLCE public key G'.

Theorem 4.1: Let $\mathbf{g}_0 \in GF(q)^k$ be a non-zero column vector of the generator matrix G. For any randomly chosen $k \times (r + 1)$ matrix $R_0 \in GF(q)^{k\times(r+1)}$, there exists a $k \times k$ nonsingular matrix S, a $(r + 1) \times (r + 1)$ matrix A_0, and a $k \times r$ matrix $C_0 \in GF(q)^{k\times r}$ such that
$$R_0 = S[\mathbf{g}_0, C_0]A_0 \tag{5}$$

Proof. The theorem could be proved using the fundamental properties of matrix equivalence and the details are omitted here. □

Let $R = [R_0, \ldots, R_{n-1}] \in GF(q)^{k \times n(r+1)}$ be a fixed random linear code generator matrix. Theorem 4.1 shows that for any generator matrix G (e.g., a Reed-Solomon code generator matrix), we can choose matrices S and A_0 so that the first $r+1$ columns of the RLCE scheme public key G' (constructed from G) are identical to R_0. However, we cannot use Theorem 4.1 to continue the process of choosing $A_1, \ldots, A_{n-1}$ to obtain $G' = R$ since S is fixed after A_0 is chosen. Indeed, this observed property is essential for the security of RLCE scheme. If one could continue the process of Theorem 4.1 and obtain matrices $S, A_0, A_1, \ldots, A_{n-1}$ such that $G' = R$, then the adversary may use the same process to recover an equivalent private key. In the following, we present a general theorem regarding this observation.

Theorem 4.2: Let $R = [R_0, \ldots, R_{n-1}] \in GF(q)^{k \times n(r+1)}$ and $G = [\mathbf{g}_0, \ldots, \mathbf{g}_{n-1}] \in GF(q)^{k \times n}$ be two fixed linear code generator matrices. If $(r+1)^2 > k$, then one can find $A_0, \ldots, A_{n-1} \in GF(q)^{r \times r}$ and $C_0, \ldots, C_{n-1} \in GF(q)^{k \times r}$ such that $R = [\mathbf{g}_0, C_0, \ldots, \mathbf{g}_{n-1}, C_{n-1}]A$ where A is in the format of (3).

Proof. For each $0 \leq i \leq n-1$, one can construct $k(r+1)$ linear equations in $(r+1)^2 + kr$ unknown variables using the identity

$$R_i A_i^{-1} = [\mathbf{g}_i, C_i].$$

Since $(r+1)^2 > k$, we have $kr + (r+1)^2 > k(r+1)$. This implies that there are more unknown variables than equations. Thus the theorem is proved. □

Theorem 4.2 shows that in the RLCE scheme, we must have $(r+1)^2 \leq k$. Otherwise, for a given public key $G' \in GF(q)^{k \times n(r+1)}$, the adversary can choose a Reed-Solomon code generator matrix $G \in GF(q)^{k \times n}$ and compute $A_0, \ldots, A_{n-1} \in GF(q)^{r \times r}$ and $C_0, \ldots, C_{n-1} \in GF(q)^{k \times r}$ such that $G' = [\mathbf{g}_0, C_0, \ldots, \mathbf{g}_{n-1}, C_{n-1}]A$. In other words, the adversary can use the decryption algorithm corresponding to the generator matrix G to break the RLCE scheme.

Theorem 4.2 gives an efficient algorithm for decrypting random $[n, k]$ linear codes with sufficiently small number t of errors. Specifically, for an $[n, k]$ linear code with generator matrix $R \in GF(q)^{k \times n}$, if $t \leq \frac{n-k(\sqrt{k}+1)}{2(\sqrt{k}+1)}$, then one can divide R into $m = 2t + k$ blocks $R = [R_0, \ldots, R_{m-1}]$. Theorem 4.2 can then be applied to construct an equivalent $[m, k]$ Reed-Solomon code with generator matrix $G \in^{k \times m}$. Thus it is sufficient to decrypt the equivalent Reed-Solomon code instead of the original random linear code. For McEliece based encryption scheme, Bernstein, Lange, and Peters [5] recommend the use of 0.75 ($= k/n$) as the code rate. Thus Theorem 4.2 has no threat on these schemes.

For $t \leq \frac{n-k(\sqrt{k}+1)}{2(\sqrt{k}+1)}$, the adversary is guaranteed to succeed in breaking the system. Since multiple errors might be located within the same block R_i with certain probability, for a given t that is slightly larger than $\frac{n-k(\sqrt{k}+1)}{2(\sqrt{k}+1)}$, the adversary still has a good chance to break the system using the above approach. It is recommended that t is significantly larger than $\frac{n-k(\sqrt{k}+1)}{2(\sqrt{k}+1)}$. For the RLCE scheme, this means that t should be significantly smaller than $\sqrt{k}$. We recommend the use of $r \leq 5$ in RLCE encryption scheme. Generally it is suffiicent to use $r = 1$.

In following sections, we list heuristic and experimental evidences that G' shares the properties of random linear codes. Thus the security of the RLCE scheme is believed to be equivalent to decoding a random linear code which is **NP**-hard.

B. Niederreiter's scheme and Sidelnikov-Shestakov's attack

Sidelnikov and Shestakov's cryptanalysis technique [24] was used to analyze Niederreiter's scheme which is based on generalized Reed-Solomon codes. Let $\alpha = (\alpha_0, \ldots, \alpha_{n-1})$ be n distinct elements of $GF(q)$ and let $v = (v_0, \ldots, v_{n-1})$ be nonzero (not necessarily distinct) elements of $GF(q)$. The generalized Reed-Solomon (GRS) code of dimension k, denoted by $GRS_k(\alpha, v)$, is defined by the following subspace.

$$GRS_k(\alpha, v) = \{(v_0 f(\alpha_0), \ldots, v_{n-1} f(\alpha_{n-1})) : f(x) \in GF(q)[x]_k)\}$$

where $GF(q)[x]_k$ is the set of polynomials in $GF(q)[x]$ of degree less than k. Alternatively, we can interpret $GF(q)[x]_k$ as a vector space of dimension k over $GF(q)$. For each code word $c = (v_0 f(\alpha_0), \ldots, v_{n-1} f(\alpha_{n-1}))$, $f(x) = f_0 + f_1 x + \ldots + f_{k-1} x^{k-1}$ is called the associate polynomial of the code word c that encodes the message $(f_0, \ldots, f_{k-1})$. $GRS_k(\alpha, v)$ is an $[n, k, d]$ MDS code where $d = n - k + 1$.

Niederreiter's scheme [20] replaces the binary Goppa codes in McEliece scheme using GRS codes as follows. For given security parameters n and k, one first chooses GRS code parameters α and v. Let G be the $k \times n$ generator matrix for this GRS code. Choose a random $k \times k$ nonsingular matrix S over $GF(q)$ and the public key is $G' = SG$ and $t = \lfloor \frac{n-k}{2} \rfloor$. G' generates a linear code with the same rate and minimum distance as the code generated by G. The encryption and decryption process are the same as in the original McEliece scheme.

The best attack on Niederreiter scheme is presented by Sidelnikov and Shestakov [24]. In Sidelnikov-Shestakov attack, one recovers an equivalent private key (α, v) from a public key G' for the code $GRS_k(\alpha, v)$ as follows. For the given public key G', one first computes the systematic form $E(G') = [I|G'']$ (also called echelon form) using Gaussian elimination. An equation system is then constructed from $E(G')$ to recover a decryption key. Wieschebrink [28] revised Niederreiter's scheme by inserting random column vectors into random positions of G before obtaining the public key G'. Couvreur et al [6] showed that Wieschebrink's revised scheme is insecure under the product code attacks.

Berger and Loidreau [3] recommend the use of sub codes of Niederreiter's scheme to avoid Sidelnikov and Shestakov's attack. Specifically, in Berger and Loidreau's scheme, one uses a random $(k-l) \times k$ matrix S' of rank $k-l$ instead of the $k \times k$ matrix S to compute the public key $G' = S'G$.

For smaller values of l, Wieschebrink [29] shows that a private key (α, v) for Berger and Loidreau scheme [3] could be recovered using Sidelnikov-Shestakov algorithm. For larger values of l, Wieschebrink used product code to recover the secret values for Berger-Loidreau scheme. Let $G' = SG$ be the $(k-l) \times n$ public generator matrix for Berger-Loidreau scheme, $\mathbf{r}_0, \cdots, \mathbf{r}_{k-l-1}$ be the rows of G', and $f_0, \cdots, f_{k-l-1}$ be the associated polynomials to those rows. For two row vector $\mathbf{a}, \mathbf{b} \in GF(q)^n$, the component wise product $\mathbf{a} * \mathbf{b} \in GF(q)^n$ is defined as

$$\mathbf{a} * \mathbf{b} = (a_0 b_0, \cdots, a_{n-1} b_{n-1}) \tag{6}$$

By the definition in (6), it is straightforward to observe that

$$\mathbf{r}_i * \mathbf{r}_j = (v_0^2 f_i(\alpha_0) f_j(\alpha_0), \cdots, v_{n-1}^2 f_i(\alpha_{n-1}) f_j(\alpha_{n-1})). \tag{7}$$

For $2k - 1 \leq n - 2$, if the code generated by $\mathbf{r}_i * \mathbf{r}_j$ equals to $GRS_{2k-1}(\alpha, \mathbf{v}')$ for $\mathbf{v}' = (v_0^2, \cdots, v_{n-1}^2)$, then the Sidelnikov-Shestakov algorithm could be used to recover the values α and v. For $2k - 1 \leq n - 2$, if the code generated by $\mathbf{r}_i * \mathbf{r}_j$ does not equal to $GRS_{2k-1}(n, \mathbf{v}')$, then the attack fails. Wieschebrink shows that the probability that the attack fails is very small. For the case of $2k - 1 > n - 2$, Wieschebrink applied Sidelnikov-Shestakov algorithm on the component wise product code of a shortened code of the original $GRS_k(\alpha, \mathbf{v})$.

The crucial step in Sidelnikov and Shestakov attack is to build an equation system from the echelon form $E(G') = [I|G'']$ obtained from the public key. In the encryption scheme RLCE, each column of the public key G' contains mixed randomness. Thus the echelon form $E(G') = [I|G'']$ obtained from the public key G' could not be used to build any useful equation systems. In other words, it is expected that Sidelnikov and Shestakov attack does not work against the RLCE scheme.

C. Filtration attacks

Using distinguisher techniques [9], Couvreur et al. [6] designed a filtration technique to attack GRS code based McEliece scheme. The filtration technique was further developed by Couvreur et al [8] to attack wild Goppa code based McEliece scheme. In the following, we briefly review the filtration attack in [8]. For two codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of length n, the star product code $\mathcal{C}_1 * \mathcal{C}_2$ is the vector space spanned by $\mathbf{a} * \mathbf{b}$ for all pairs $(\mathbf{a}, \mathbf{b}) \in \mathcal{C}_1 \times \mathcal{C}_2$ where $\mathbf{a} * \mathbf{b}$ is defined in (6). For $\mathcal{C}_1 = \mathcal{C}_2$, $\mathcal{C}_1 * \mathcal{C}_1$ is called the square code of $\mathcal{C}_1$. It is showed in [8] that

$$\dim \mathcal{C}_1 \times \mathcal{C}_2 \leq \left\{ n, \dim \mathcal{C}_1 \dim \mathcal{C}_2 - \binom{\dim(\mathcal{C}_1 \cap \mathcal{C}_1)}{2} \right\}. \tag{8}$$

Furthermore, the equality in (8) is attained for most randomly selected codes $\mathcal{C}_1$ and $\mathcal{C}_2$ of a given length and dimension. Note that for $\mathcal{C} = \mathcal{C}_1 = \mathcal{C}_2$ and $\dim \mathcal{C} = k$, the equation (8) becomes $\dim \mathcal{C}^{*2} \leq \min \left\{ n, \binom{k+1}{2} \right\}$.

Couvreur et al [8] showed that the square code of an alternant code of extension degree 2 may have an unusually low dimension when its dimension is larger than its designed rate. Specifically, this happens for wild Goppa codes over quadratic extensions. Using a shortening trick, Couvreur et al showed that the square of a shortened wild Goppa code of extension degree 2 is contained in an alternant code of non trivial dimension. Based on those

observations, Couvreur et al designed the code filtration techniques. Specifically, create a family of nested codes defined for any $a \in \{0, \cdots, n-1\}$ as follows.

$$\mathcal{C}^a(0) \supseteq \mathcal{C}^a(1) \supseteq \cdots \supseteq \mathcal{C}^a(q+1) \tag{9}$$

This family of nested codes is called a filtration. Roughly speaking, $\mathcal{C}^a(j)$ consists in the codewords of $\mathcal{C}$ which correspond to polynomials which have a zero of order j at position a. It is shown that the first two elements of this filtration are just punctured and shortened versions of $\mathcal{C}$ and the rest of them can be computed from $\mathcal{C}$ by computing star products and solving linear systems. Furthermore, the support values $\alpha_0, \cdots, \alpha_{n-1}$ for the Goppa code could be recovered by this nested family of codes efficiently. Thus the private keys for wild Goppa code based McEliece scheme could be recovered from the public keys.

The crucial part of the filtration techniques is the efficient algorithm to compute the nested family of codes in (9). For our RLCE scheme, the public generator matrix G' contains random columns. Thus linear equations constructed in Couvreur et al [8] could not be solved and the nested family (9) could not be computed correctly. Furthermore, the important characteristics for a code $\mathcal{C}$ to be vulnerable is that one can find a related code $\mathcal{C}_1$ of dimension k such that the dimension of the square code of $\mathcal{C}_1$ has dimension significantly less than $\min\left\{n, \binom{k+1}{2}\right\}$.

To get experimental evidence that RLCE codes share similarity with random linear codes with respect to the above mentioned filtration attacks, we carried out several experiments. In particular, we used Reed-Solomon codes over $GF(257)$ as the underlying code and we used the value $r = 1$. That is, we inserted one random column after each generator matrix column. For each given 10×30 generator matrix G of Reed-Solomon code, we selected another random 10×30 matrix $C \in GF(257)^{10\times 30}$ and selected 2×2 matrices $A_0, \ldots, A_{29}$. Each column $\mathbf{c}_i$ in C is inserted in G after the column $\mathbf{g}_i$. The extended generator matrix is multiplied by $A = \text{diag}[A_0, \ldots, A_{29}]$ from the right hand side to obtain the public key matrix $G' \in GF(257)^{10\times 60}$. Then we delete columns $0, \ldots, 59$ from G' one at a time. The resulting matrix is used to compute the product code. For 1000 experiments that we have done, the resulting product codes have dimension from 54 to 55. Since $\min\{60, \binom{11}{2}\} = 55$, the experimental results meet our expectation that RLCE behaves like a random linear code. We did the same experiments for the dual code of the above code. That is, for a 20×30 generator matrix G of the dual code, the same procedure has been taken. In this time, after deleting one column from the resulting public key matrix, the product code always has dimension from 58 to 60 which meets our expectation also. The experiments have been carried out using Maple 2015 with its built in Mersenne Twister pseudorandom generator. We are aware of the fact that the parameters used in the experiments are significantly smaller than the recommended parameters for RLCE scheme in Section V. The goal of the experiments is to get some heuristic (experimental) evidence that RLCE scheme behaves like a random linear code and the experiments did confirm our expectation.

D. Algebraic attacks

Faugere, Otmani, Perret, and Tillich [10] developed an algebraic attack against quasi-cyclic and dyadic structure based compact variants of McEliece encryption scheme. In a high level, the algebraic attack from [10] tries to find $\mathbf{x}^*, \mathbf{y}^* \in GF(q)^n$ such that $V_t(\mathbf{x}^*, \mathbf{y}^*)$ is the parity check matrix for the underlying alternant codes of the compact variants of McEliece encryption scheme. $V_t(\mathbf{x}^*, \mathbf{y}^*)$ can then be used to break the McEliece scheme. Note that this $V_t(\mathbf{x}^*, \mathbf{y}^*)$ is generally different from the original parity check matrix $V_t(\mathbf{x}, \mathbf{y})$ in (1).

The parity check matrix $V_t(\mathbf{x}^*, \mathbf{y}^*)$ could be found by solving an equation system constructed from

$$V_t(\mathbf{x}^*, \mathbf{y}^*){G'}^T = 0. \tag{10}$$

where G' is the public key. The authors of [10] employed the special properties of quasi-cyclic and dyadic structures (which provide additional linear equations) to rewrite the equation system obtained from (10) and then calculate $V_t(\mathbf{x}^*, \mathbf{y}^*)$ efficiently.

Faugere, Gauthier-Umana, Otmani, Perret, and Tillich [9] used the algebraic attack in [10] to design an efficient Goppa code distinguisher to distinguish a random matrix from the matrix of a Goppa code whose rate is close to 1. For instance, [9] showed that the binary Goppa code obtained with $m = 13$ and $r = 19$ corresponding to a 90-bit security key is distinguishable.

It is challenging to mount the above mentioned algebraic attacks on our RLCE encryption scheme even if the underlying code is a Reed-Solomon code. The parity check matrix for a Reed-Solomon code is in the format of

$$V_t(\alpha) = \begin{pmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^2 & \alpha^4 & \cdots & \alpha^{2(n-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha^{t+1} & \alpha^{2(t+1)} & \cdots & \alpha^{(t+1)(n-1)} \end{pmatrix} \tag{11}$$

In order to mount the algebraic attacks in [10], [9], one first constructs an equation system using the identity

$$V_t(\alpha)(G'P^{-1}A^{-1})^T = V_t(\alpha)[\mathbf{g}_0, C_0, \cdots, \mathbf{g}_{n-1}, C_{n-1}]^T S^T = [\mathbf{0}, C_0, \mathbf{0}, \cdots, \mathbf{0}, C_{n-1}]^T S^T \tag{12}$$

with an unknown α and an unknown $n(r+1)\times n(r+1)$ nonsingular matrix A which consists of n dense nonsingular $(r+1)\times(r+1)$ matrices $A_0, \cdots, A_{n-1} \in GF(q)^{(r+1)\times(r+1)}$ as defined in (3). In the above discussion, we do not use equations corresponding to the matrices $C_0, \cdots, C_{n-1}$. Since A contains $4(r+1)^2$ unknown variables, an equation system constructed from the identity (12) does not have enough equations to enable the attacks developed in [9], [10]. In other words, it is expected that the algebraic attack does not work against our encryption scheme RLCE.

V. Practical considerations

Following discussions in Section IV-A, it is preferred to use smaller r for the RLCE encryption scheme. We recommend the use of $r \leq 5$. Furthermore, we assume that Reed-Solomon code over $GF(q)$ with $q = p^m \leq 2^{12}$ or $q = 2^m \leq 2^{12}$ is used as the underlying linear code. As we have mentioned in the introduction section, the most powerful attack on McEliece encryption schemes is the information-set decoding attack. For RLCE encryption scheme, the information-set decoding attack is based on the number of columns in the public key G' instead of the number of columns in the private key G. For the same error weight t, the probability to find error-free coordinates in $(r+1)n$ coordinates are higher than the probability to find error-free coordinates in n coordinates. Thus for the same security level, larger error weight t is needed. Generally, this is not an issue since any efficient MDS error correction code such as Reed-Solomon code could be used to construct the RLCE scheme.

The recommended parameters for RLCE in Table I are generated using the PARI/GP script `isdfq.gp`[1] that is based on the exact information-set decoding complexity analysis by Peters [21]. For the purpose of comparison, we also list the recommended parameters from [5] for binary Goppa code based McEliece encryption scheme. The authors in [5], [21] proposed the use of semantic secure message coding approach so that one can store the public key as a systematic generator matrix. For binary Goppa code based McEliece encryption scheme, the systematic generator matrix public key is $k(n-k)$ bits. For RLCE encryption scheme over $GF(q)$, the systematic generator matrix public key is $k(n(r+1)-k)\log q$ bits. It is observed that RLCE scheme generally has large public key size. When the original binary Goppa code based McEliece scheme is used to construct the RLCE scheme, one needs to double or triple the degree t in the parameter under the "binary Goppa code [5]" column in Table I. The RLCE public key size is generally the triple of the corresponding McEliece scheme public key size. For example, for the security level 60, binary Goppa code based RLCE scheme has a public key of $3 \times 19.8 = 59.4$KB.

TABLE I
Parameters for RLCE: n, k, t, q, key size ($q = 2$ for Binary Goppa code omitted)

Security	RLCE-MDS code	binary Goppa code [5]
60	360,240, 60, 2^9, 126KB	1024, 524, 50, 19.8KB
80	560, 380, 90, 2^{10}, 343KB	1632, 1269, 34, 56.2KB
128	1020, 680, 170, 2^{10}, 1.1MB	2960, 2288, 57, 187.7KB
191	1560, 1000, 280, 2^{11}, 2.8MB	4624, 3468, 97, 489.4KB
256	2184, 1400, 392, 2^{12}, 5.9MB	6624, 5129, 117, 0.9MB

[1]available from https://christianepeters.wordpress.com/publications/tools/.

VI. Conclusions

In this paper, we presented techniques for designing general random linear code based public encryption schemes using any linear codes. Heuristics and experiments are used to show that the proposed schemes are immune against existing attacks on linear code based encryption schemes such as Sidelnikov-Shestakov attack, filtration attacks, and algebraic attacks.

References

[1] M. Baldi, M. Bodrato, and F. Chiaraluce. A new analysis of the McEliece cryptosystem based on QC-LDPC codes. In *Security and Cryptography for Networks*, pages 246–262. Springer, 2008.

[2] A. Becker, A. Joux, A. May, and A. Meurer. Decoding random binary linear codes in $2^{n/20}$: How $1 + 1 = 0$ improves information set decoding. In *EUROCRYPT 2012*, pages 520–536. Springer, 2012.

[3] T Berger and P Loidreau. How to mask the structure of codes for a cryptographic use. *Designs, Codes and Cryptography*, 35(1):63–79, 2005.

[4] D. Bernstein, T. Lange, and C. Peters. Wild McEeliece. In *Selected Areas in Cryptography*, pages 143–158. Springer, 2011.

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

[6] 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*, pages 1–26, 2013.

[7] A. Couvreur, I. Márquez-Corbella, and R. Pellikaan. A polynomial time attack against algebraic geometry code based public key cryptosystems. In *Proc. ISIT*, pages 1446–1450. IEEE, 2014.

[8] A. Couvreur, A. Otmani, and J.-P. Tillich. Polynomial time attack on wild McEliece over quadratic extensions. In *EUROCRYPT 2014*, pages 17–39. Springer, 2014.

[9] J.-C. Faugere, V. Gauthier-Umana, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high-rate McEliece cryptosystems. *IEEE Trans. Information Theory*, 59(10):6830–6844, 2013.

[10] J.-C. Faugere, A. Otmani, L. Perret, and J.-P. Tillich. Algebraic cryptanalysis of McEliece variants with compact keys. In *Eurocrypt 2010*, pages 279–298. Springer, 2010.

[11] H. Janwa and O. Moreno. McEliece public key cryptosystems using algebraic-geometric codes. *Designs, Codes and Cryptography*, 8(3):293–307, 1996.

[12] P. J. Lee and E. F. Brickell. An observation on the security of McEliece's public-key cryptosystem. In *EUROCRYPT'88*, pages 275–280. Springer, 1988.

[13] J. Leon. A probabilistic algorithm for computing minimum weights of large error-correcting codes. *IEEE Trans. Information Theory*, 34(5):1354–1359, 1988.

[14] C. Löndahl and T. Johansson. A new version of McEliece PKC based on convolutional codes. In *Information and Communications Security*, pages 461–470. Springer, 2012.

[15] I. Márquez-Corbella and R. Pellikaan. Error-correcting pairs for a public-key cryptosystem. *arXiv preprint arXiv:1205.3647*, 2012.

[16] A. May, A. Meurer, and E. Thomae. Decoding random linear codes in $\tilde{O}(2^{0.054n})$. In *ASIACRYPT 2011*, pages 107–124. Springer, 2011.

[17] A. May and I. Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes. In *EUROCRYPT 2015*, pages 203–228. Springer, 2015.

[18] R.J. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN progress report*, 42(44):114–116, 1978.

[19] R. Misoczki, J.-P. Tillich, N. Sendrier, and P. Barreto. MDPC-McEliece: New McEliece variants from moderate density parity-check codes. In *Proc. ISIT*, pages 2069–2073. IEEE, 2013.

[20] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Informaiton Theory*, 15(2):159–166, 1986.

[21] C. Peters. Information-set decoding for linear codes over F_q. In *Post-Quantum Cryptography*, pages 81–94. Springer, 2010.

[22] Eugene Prange. The use of information sets in decoding cyclic codes. *IRE Trans. Information Theory*, 8(5):5–9, 1962.

[23] V Sidelnikov. A public-key cryptosystem based on binary Reed-Muller codes. *Discrete Mathematics and Applications*, 4(3):191–208, 1994.

[24] V. M Sidelnikov and S. Shestakov. On insecurity of cryptosystems based on generalized Reed-Solomon codes. *Discrete Mathematics and Applications*, 2(4):439–444, 1992.

[25] J. Stern. A method for finding codewords of small weight. In *Coding theory and applications*, pages 106–113. Springer, 1989.

[26] Y. Wang. Resource bounded randomness and computational complexity. *Theoret. Comput. Sci.*, 237:33–55, 2000.

[27] Y. Wang. Insecure "provably" secure network coding" and homomorphic authentication schemes for network coding. *IACR Cryptology ePrint Archive*, 2010:60, 2010.

[28] C. Wieschebrink. Two NP-complete problems in coding theory with an application in code based cryptography. In *Proc. IEEE ISIT*, pages 1733–1737. IEEE Press, 2006.

[29] C. Wieschebrink. Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes. In *Post-Quantum Cryptography*, pages 61–72. Springer, 2010.