IET Information Security

Research Article

Security of generalised Reed–Solomon code-based cryptosystems

ISSN 1751-8709
Received on 13th September 2017
Revised 9th May 2018
Accepted on 2nd November 2018
E-First on 1st April 2019
doi: 10.1049/iet-ifs.2018.5207
www.ietdl.org

Marco Baldi[1] ✉, *Franco Chiaraluce*[1], *Joachim Rosenthal*[2], *Paolo Santini*[1], *Davide Schipani*[2]

[1] *DII, Università Politecnica delle Marche, Ancona, Italy*

[2] *Department of Mathematics, University of Zurich, Zurich, Switzerland*

✉ *E-mail: m.baldi@univpm.it*

Abstract: In this study, the authors elaborate on a recently proposed variant of the public-key McEliece and Niederreiter cryptosystems using generalised Reed–Solomon (GRS) codes as private codes. The use of these codes brings known advantages in terms of public key size, but particular care is needed in the choice of parameters not to endanger the system security. In fact, the considered system exploits a strong disguising technique of the private code within the public code. However, it has recently been pointed out that some new attacks exist which may threaten some instances of such a system, therefore the choice of parameters needs to consider some further constraints compared to the original version. After outlining these constraints, the authors propose a new modification of the system achieving greater flexibility in the parameter choice. Moreover, the new system exhibits a lower complexity than the original GRS code-based system. Its very competitive features such as key size and encryption rate are highlighted with respect to classic systems.

1 Introduction

In recent years, research on code-based public-key cryptography has been boosted by the fact that code-based cryptosystems may offer shelter to attacks exploiting quantum computers. In fact, it is known that quantum algorithms such as Shor's algorithm [1] are able to turn into polynomial-time solvable some hard mathematical problems that are at the basis of many widespread public-key cryposystems, such as Rivest-Shamir-Adleman (RSA), ElGamal, digital signature algorithm (DSA), elliptic curve digital signature algorithm (ECDSA), Diffie–Hellman, and others. While quantum computers represented a mostly theoretical threat until a few years ago, they are rapidly becoming real nowadays [2, 3]. Therefore, research efforts devoted to the definition of efficient post-quantum replacements of quantum vulnerable systems have dramatically increased in the last few years. Also, National Institute of Standards and Technology (NIST)'s post-quantum crypto project [4] goes in this direction, by encouraging the community to select secure and efficient post-quantum cryptosystems to be used in commercial applications.

According to NIST [5], code-based cryptosystems are among the most promising solutions to replace quantum vulnerable public-key cryptosystems. Code-based public key cryptography has been initiated by Robert McEliece, who proposed the first public-key cryptosystem based on codes in 1978 [6]. The hard problem exploited by the McEliece cryptosystem is that of decoding a random-like linear block code with no visible structure, which is known to be a non-polynomial-time solvable problem [7]. Instead, many families of non-random or quasi-random codes exist which are provided with efficient and fast decoding algorithms working on suitable representations of the codes. McEliece chose to use Goppa codes of length n able to correct t errors to form the secret key. By knowing the secret representation of these codes, they can be decoded through Patterson's algorithm running in time $O(n \cdot t)$. Then, McEliece proposed to use a linear transformation to disguise the secret code having a random-like representation, which prevents the use of efficient decoders and creates the trapdoor. While McEliece used generator matrices to represent private and public codes, Niederreiter later introduced an equivalent cryptosystem exploiting parity-check matrices [8], which is still secure when Goppa codes are used as secret codes.

However, when Goppa codes are used, these systems have the drawback of requiring rather large public keys. This is mainly due to the random-like nature of Goppa codes characteristic matrices and of their disguised versions, which does not facilitate their storage. A common approach to face this problem is to replace Goppa codes with other families of codes characterised by more structured representations allowing more compact storage of the public keys. In fact, several families of codes have been considered for this purpose, but often with security drawbacks [9–11]. Also, low-density parity-check (LDPC) codes have a long history in the framework of these cryptosystems [12–15], and some secure LDPC code-based instances have been found [16, 17].

The use of generalised Reed–Solomon (GRS) codes as replacements of Goppa codes in the McEliece and Niederreiter cryptosystems is a challenging matter as well. On the one hand, GRS codes would allow us to reduce the public key size (KS) thanks to their optimum error correction capability since they are maximum distance separable codes. In addition, GRS codes are widespread in many applications, and much research has been done on their encoding and decoding, with the aim to reduce complexity and facilitate implementation. On the other hand, GRS codes have more structure than Goppa codes (and wild Goppa codes), therefore it is harder to avoid attacks aimed at recovering the secret code from the public one. The first successful attack against GRS code-based cryptosystems was the Sidelnikov–Shestakov attack [18], which was addressed to the Niederreiter system. This highlighted the fact that, differently from Goppa codes, exposing a public code which is permutation-equivalent to the secret GRS code does not provide security.

A GRS code-based variant of the McEliece and Niederreiter cryptosystems has been proposed in [19], with the purpose of providing strong protection of the secret GRS code. The rationale of the proposal is to replace, in the transformation from the private to the public code, the classic permutation matrix with a more general matrix obtained as the sum of a low rank matrix ($\boldsymbol{R}$) and a sparse matrix ($\boldsymbol{T}$), having average row and column weight m. Recently, an attack procedure already devised in [20] has been improved in [21], resulting in a polynomial-time attack against the system parameters proposed in [19]. This induces to increase the rank of $\boldsymbol{R}$, which entails a higher decryption complexity, and/or the density of $\boldsymbol{T}$, which can weaken the advantage brought by the high correction capability of GRS codes. Nevertheless, some choices of the system parameters still avoid the mentioned attack (e.g. using codes with rate $R \simeq 1/2$, fixing the rank of $\boldsymbol{R}$ to 1 and the density of $\boldsymbol{T}$ larger than or equal to $1 + R$), so they can be proposed and combine security with an acceptable decryption complexity. A few instances will be given in Section 4, where it is shown that the KS for this system is a fraction (0.67 and 0.61, respectively) of that required by the Goppa code-based solution to achieve the same security level (SL).

In this study, we introduce a modification of the scheme proposed in [19], which allows increasing the rank of $\boldsymbol{R}$ without any effect on the decryption complexity, which is even considerably reduced. This is paid in terms of the ciphertext and public key lengths, which may be somewhat increased with respect to [19]. However, in several applications of public-key cryptosystems, having long ciphertexts is less detrimental than having large public keys, and this solution can still allow achieving considerably shorter keys than the classic binary Goppa code-based counterparts. Moreover, the new system incurs a very small penalty (or even no penalty in some cases) in terms of encryption rate with respect to the binary Goppa code-based counterparts. On the other hand, the secret GRS code disguising technique is weakened by the new modification, in that a subcode of the public code is now exposed. In some instances of the system (i.e. when $m = 1$) this subcode is permutation-equivalent to a subcode of the secret code, which introduces some new constraints and guides the choice of parameters that avoid attacks on the subcode. In fact, in this case, the security of the system can be reduced to the security of the system in [22]. Choosing $m > 1$ instead provides a higher level of protection and at the same time allows achieving more competitive parameters in terms of KS.

The paper is organised as follows. In Section 2, we describe the original GRS code-based cryptosystem that we consider as a starting point. In Section 3, we describe the proposed new variant. In Section 4, we compare the original system and the new variant with the classic binary Goppa code-based systems in terms of KS and complexity. In Section 5, we provide some final remarks and hints for future works.

2 Original system

In the following, we focus mainly on the Niederreiter version of the system in [19], which is reminded next, but the same arguments can be applied to the McEliece version, with minor modifications. A few differences, e.g. in encryption rate, between the two versions are anyway pointed out later in Section 4.

2.1 Key generation

Bob's secret key is formed by three matrices, $\{\boldsymbol{H}, \boldsymbol{S}, \boldsymbol{Q}\}$. The first one is the $r \times n$ parity-check matrix of a GRS code with length n and dimension $k = n - r$, defined over $\mathbb{F}_q$, able to correct $t = \frac{r}{2}$ errors. The matrix $\boldsymbol{S}$ is a non-singular $r \times r$ scrambling matrix and $\boldsymbol{Q}$ is a non-singular $n \times n$ transformation matrix. The latter is obtained as $\boldsymbol{Q} = \boldsymbol{R} + \boldsymbol{T}$, where $\boldsymbol{R}$ is a dense $n \times n$ matrix with rank $z \ll n$ and $\boldsymbol{T}$ is a sparse $n \times n$ matrix with average row and column weight $m \ll n$. Both $\boldsymbol{R}$ and $\boldsymbol{T}$ take values in $\mathbb{F}_q$. The matrix $\boldsymbol{R}$ is obtained as $\boldsymbol{R} = \boldsymbol{a}^{\mathrm{T}} \cdot \boldsymbol{b}$, where $\boldsymbol{a}$ and $\boldsymbol{b}$ are two matrices defined over $\mathbb{F}_q$, having size $z \times n$ and rank z. Therefore, $\boldsymbol{R}$ has rank z as well. The matrix $\boldsymbol{T}$ instead is an $n \times n$ non-singular sparse matrix having average row and column weight equal to m, where m is not necessarily an integer value. More details on its construction are provided in [19, Section 2.4]. Bob's public key $\boldsymbol{H}'$ is then obtained as

$$\boldsymbol{H}' = \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{Q}^{\mathrm{T}}. \tag{1}$$

2.2 Encryption

In order to perform encryption, Alice maps the cleartext vector into an error vector $\boldsymbol{e} = [e_1, e_2, \ldots, e_n]$, having weight $t_{\mathrm{p}} \le t$. Different from the original McEliece and Niederreiter systems, where $t_{\mathrm{p}} = t$, this system uses $t_{\mathrm{p}} = \lfloor t/m \rfloor$. Alice then computes the ciphertext as the syndrome

$$\boldsymbol{x} = \boldsymbol{H}' \cdot \boldsymbol{e}^{\mathrm{T}}. \tag{2}$$

2.3 Decryption

In order to decrypt $\boldsymbol{x}$, Bob first computes

$$\begin{aligned} \boldsymbol{x}' &= \boldsymbol{S} \cdot \boldsymbol{x} \\ &= \boldsymbol{H} \cdot \boldsymbol{Q}^{\mathrm{T}} \cdot \boldsymbol{e}^{\mathrm{T}} \\ &= \boldsymbol{H} \cdot (\boldsymbol{e} \cdot \boldsymbol{Q})^{\mathrm{T}} \\ &= \boldsymbol{H} \cdot [\boldsymbol{e} \cdot (\boldsymbol{R} + \boldsymbol{T})]^{\mathrm{T}} \\ &= \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \gamma + \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{e}^{\mathrm{T}}, \end{aligned} \tag{3}$$

where $\gamma = \boldsymbol{a} \cdot \boldsymbol{e}^{\mathrm{T}}$. Bob then has to guess the entries of γ, which are unknown to him. This means discovering the values of z elements of $\mathbb{F}_q$, which can be done through a sequential search over the whole set of q^z possible solutions, requiring $q^z/2$ attempts on average. As shown in [19, Sect. 3.2], Bob is able to tell if each attempt is successful or not. After having correctly guessed the entries of γ, Bob computes

$$\begin{aligned} \boldsymbol{x}'' &= \boldsymbol{x}' - \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \gamma \\ &= \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{e}^{\mathrm{T}} \\ &= \boldsymbol{H} \cdot \boldsymbol{e}_{\mathrm{T}}^{\mathrm{T}}. \end{aligned}$$

Since $\boldsymbol{e}_{\mathrm{T}} = \boldsymbol{e} \cdot \boldsymbol{T}$ has weight $\le m \cdot t_{\mathrm{p}} \le t$, $\boldsymbol{x}''$ is a correctable syndrome through the secret GRS code, and Bob can recover $\boldsymbol{e}_{\mathrm{T}}$, having weight $\le t$, by performing syndrome decoding. Then, he computes

$$\boldsymbol{e} = \boldsymbol{e}_{\mathrm{T}} \cdot \boldsymbol{T}^{-1} \tag{4}$$

and finally, demaps $\boldsymbol{e}$ into its associated cleartext vector to get the secret message.

2.4 Parameters design

In this system, the two values m and z must be small, since

- for a given t_{p}, determined by the target SL against decoding attacks, increasing m requires to increase t and, hence, the code size and the public KS;
- in order to guess the value of γ, Bob needs to perform $q^z/2$ attempts on average, therefore increasing z increases the decryption complexity.

Keeping both z and m too small exposes the system to polynomial-time attacks [21], therefore a security/performance trade-off arises. In fact, the attack in [21] can be applied only if m and z are chosen in such a way as to verify both the following conditions

$$\begin{cases} 1 \le m \le 1 + R - \dfrac{1}{n} - \sqrt{\dfrac{8}{n}R + \dfrac{1}{n^2}} < 2, \\ z = 1. \end{cases} \tag{5}$$

The special choice $m = z = 1$ exposes the system to the original version of the attack, introduced in [20]. The improved version of the attack proposed in [21] is built upon a distinguisher of GRS codes based on computing the dimension of the square of shortenings of the public code. The squares of some of these shortenings have a smaller dimension than the squares of shortened random codes of the same size, due to the structure of the hidden private code. The core of the attack in [21] is an algorithm to distinguish between rows of $\boldsymbol{T}$ with Hamming weight 1 and rows of $\boldsymbol{T}$ with Hamming weight 2. In fact, for $1 < m < 2$, the rows of $\boldsymbol{T}$

have Hamming weight 1 or 2. Then, the effect of weight-2 columns of $\boldsymbol{T}$ is reverted to that of weight-1 columns through linear combinations of columns of the public parity-check matrix. Through these steps, the public key of an alternative system with the same private code but with $m = 1$ is recovered by the opponent, who can then mount the attack in [20] against such an alternative system to recover the private key.

In order to counter these attacks, the system parameters must be chosen such that conditions (5) on m and z are not verified. This, however, has a detrimental effect on the public KS and complexity when the original system is used. In the following, a new variant of this system is presented that aims at preventing known vulnerabilities by increasing z while, at the same time, avoiding increasing the decryption complexity by publishing $\boldsymbol{a}$. This leaks some information on $\boldsymbol{R}$, which affects the work factor (WF) of decoding attacks and in some cases leads us to choose a set of parameters that thwart some known algebraic attacks. However, as stated above, choosing high values of z allows avoiding the attack presented in [21], which can be applied successfully only if conditions (5) on m and z are verified, thus resulting in some instances of the new system that bring significant advantages over Goppa code-based systems and other GRS code-based systems.

3 New system

In the variant we propose, $\boldsymbol{a}$ is made public together with $\boldsymbol{H}'$, so Bob's public key is now the pair $(\boldsymbol{H}', \boldsymbol{a})$. Moreover, during encryption, Alice computes $\gamma = \boldsymbol{a} \cdot \boldsymbol{e}^{\mathrm{T}}$ and sends it along with the ciphertext. In this case, Bob no longer needs to guess the value of γ, which significantly reduces the decryption complexity. Basically, the new system works as follows.

3.1 Encryption

Alice maps the cleartext vector into an error vector $\boldsymbol{e} = [e_1, e_2, \ldots, e_n]$, having weight $t_{\mathrm{p}} = \lfloor t/m \rfloor$. Alice then sends the pair $(\boldsymbol{x}, \gamma)$, where $\boldsymbol{x}$ is given by (2) and

$$\gamma = \boldsymbol{a} \cdot \boldsymbol{e}^{\mathrm{T}}. \tag{6}$$

3.2 Decryption

As in the original system (see Section 2.3), in order to decrypt $\boldsymbol{x}$, Bob first computes

$$\boldsymbol{x}' = \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \gamma + \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{e}^{\mathrm{T}}.$$

Now, knowing γ, he can directly compute

$$\begin{aligned}\boldsymbol{x}'' &= \boldsymbol{x}' - \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \gamma \\ &= \boldsymbol{H} \cdot \boldsymbol{e}_{\mathrm{T}}^{\mathrm{T}},\end{aligned}$$

and then recover $\boldsymbol{e}_{\mathrm{T}}$ by performing syndrome decoding in order to get the secret message.

3.3 Parameters design

The parameters of the new system variant are chosen as follows to avoid potential vulnerabilities. In fact, when $\boldsymbol{a}$ is public, the system is exposed to the subcode vulnerability described in [19, Sect. 3.1]. More precisely, from (1) it follows that

$$\begin{aligned}\boldsymbol{H}' &= \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{R}^{\mathrm{T}} + \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \\ &= \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \boldsymbol{a} + \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}}.\end{aligned} \tag{7}$$

An attacker could consider the following alternative parity-check matrix

$$\boldsymbol{H}_{\mathrm{S}} = \begin{bmatrix} \boldsymbol{H}' \\ \boldsymbol{a} \end{bmatrix} = \begin{bmatrix} \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \boldsymbol{a} + \boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \\ \boldsymbol{a} \end{bmatrix}. \tag{8}$$

Compared to $\boldsymbol{H}'$, $\boldsymbol{H}_{\mathrm{S}}$ includes an additional set of parity-check equations, defined by $\boldsymbol{a}$, which imply that any codeword $\boldsymbol{c}$ belonging to the code defined by $\boldsymbol{H}_{\mathrm{S}}$ satisfies $\boldsymbol{a} \cdot \boldsymbol{c}^{\mathrm{T}} = \boldsymbol{0}$. This, in turn, implies that $\boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{b}^{\mathrm{T}} \cdot \boldsymbol{a} \cdot \boldsymbol{c}^{\mathrm{T}} = \boldsymbol{0}$. Therefore, the constraint imposed by the set of parity-check equations appearing in (8) due to $\boldsymbol{H}'$ becomes $\boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{c}^{\mathrm{T}} = \boldsymbol{0}$, for any codeword $\boldsymbol{c}$ belonging to the code defined by $\boldsymbol{H}_{\mathrm{S}}$. In summary, $\boldsymbol{H}_{\mathrm{S}}$ as in (8) defines a subcode of the public code where any codeword $\boldsymbol{c}$ satisfies $\boldsymbol{S}^{-1} \cdot \boldsymbol{H} \cdot \boldsymbol{T}^{\mathrm{T}} \cdot \boldsymbol{c}^{\mathrm{T}} = \boldsymbol{0}$. Hence, the effect of the dense matrix $\boldsymbol{R}$ is removed and, when $\boldsymbol{T}$ is a permutation matrix (i.e. when $m = 1$), the subcode defined by $\boldsymbol{H}_{\mathrm{S}}$ is permutation-equivalent to a subcode of the secret code. A cryptosystem exposing a subcode of a private GRS code is the Berger–Loidreau (BL) scheme [22]. Based on the above considerations, for the case $m = 1$, the security of the new system is equivalent to that of a BL scheme with the same subcode parameters. This also makes the attack presented in [10] applicable, and designing parameters for the BL scheme that are secure against known attacks is related to designing secure parameters for the new variant we propose in the case $m = 1$.

However, the dimension of such a subcode is equal to $n - \mathrm{rank}\{\boldsymbol{H}_{\mathrm{S}}\}$ and we can choose parameters which avoid known attacks on the subcode (which apply in case $m = 1$): if we look at [10, Eq. 8], for dimension $k \geq n/2$, we can avoid the attack by requiring $k - z - 1 < 2k - n + 1$. In particular, we will show examples for $m = 1$ with high rate R and $z \geq n - k$, while for $m > 1$, we can consider lower values of z. For this case, we will choose a few values of $m > 1 + ((r-3)/n)$ as in [19] to avoid distinguishers, and see how the KS changes for a few values of z in the range of interest.

Another point to take into account is that, although this type of subcode attack may be avoided, knowing $\boldsymbol{a}$ and γ facilitates decoding attacks. In fact, $\begin{bmatrix} \boldsymbol{x} \\ \gamma \end{bmatrix} = \boldsymbol{H}_{\mathrm{S}} \cdot \boldsymbol{e}^{\mathrm{T}}$ and an attacker could perform syndrome decoding on the code defined by $\boldsymbol{H}_{\mathrm{S}}$, rather than the public code. Such a code has rate $(k - z)/n < k/n$, and this facilitates attacks exploiting general decoding algorithms such as information set decoding (ISD). The attack complexity decreases as long as z increases (when $z \geq k$ the attack becomes very simple since Eve could find a full rank $\boldsymbol{H}_{\mathrm{S}}$ with size $n \times n$ and just invert it). Indeed, we will choose parameters with large dimension k and relatively small z.

It is interesting to remark that a first attempt in using a disguising matrix such as $\boldsymbol{R}$ together with publishing $\boldsymbol{a}$ was reported in [23], but this involved choosing $\boldsymbol{e}$ such that $\boldsymbol{a} \cdot \boldsymbol{e}^{\mathrm{T}} = 0$. This constraint on the choice of $\boldsymbol{e}$ forced to choose very small values of z, otherwise the WF of decoding attacks would be significantly reduced. In the case of the present proposal, there is no constraint on $\boldsymbol{e}$, which allows choosing higher values of z. The price to pay is some loss in encryption rate due to the transmission of γ along with the ciphertext.

4 Comparison with classic systems

Let us compare the new system with related previous systems. As shown in Section 3.3, designing secure parameters for the BL scheme is related to designing secure parameters for the new variant we propose in the case $m = 1$. However, differently from our scheme, the BL system uses the subcode to encrypt messages. As we will see next, this implies some difference in encryption rate when compared to our system instances with $m = 1$ and the same SL.

In the following, we first compare the new variant with $m = 1$ (and $z = n - k$) with the BL system based on GRS subcodes and the classic cryptosystem based on binary Goppa codes, for some fixed SLs. We then take m slightly larger than 1 as already suggested in [19]. In this case, the exposed subcode is no longer permutation-equivalent to a subcode of the secret code, therefore security is higher than the security of BL. The reduction in error correction capability is counterbalanced by the possibility of decreasing z, so that overall the effect on KS is favourable. We, therefore, add new instances (choosing a few different values for m and z) in the comparison.

4.1 Public KS

In the Niederreiter cryptosystem, we can use a systematic version of the public key, i.e.

$$\boldsymbol{H}'' = (\boldsymbol{H}'_{\rm l})^{-1} \cdot \boldsymbol{H}' = \left[\boldsymbol{I}\middle|(\boldsymbol{H}'_{\rm l})^{-1} \cdot \boldsymbol{H}'_{\rm r}\right] = [\boldsymbol{I}|\boldsymbol{H}''_{\rm r}],$$

where $\boldsymbol{H}'_{\rm l}$ and $\boldsymbol{H}'_{\rm r}$ are the left $r \times r$ and right $r \times k$ submatrices of $\boldsymbol{H}'$, i.e. $\boldsymbol{H}' = [\boldsymbol{H}'_{\rm l}|\boldsymbol{H}'_{\rm r}]$. By considering public parity-check matrices in such a form, the public KS is kr bits for the binary Goppa code-based system. In the new system, $kr + z(n - z)$ values over $\mathbb{F}_q$, i.e. $(kr + zn - z^2)\log_2 q$ bits are required to store the public key $\{\boldsymbol{H}', \boldsymbol{a}\}$, with $\boldsymbol{a}$ in systematic form, while in the BL scheme using a subcode of dimension $k - z$, the public KS is $(k - z)(r + z)\log_2 q$ bits. It follows that, for a given z, when k increases the difference in KS between this new system and BL decreases, although BL always has a smaller KS. When $m = 1$, the new system requires the same value of z used in BL to achieve the same SL. Therefore, in such a case the BL system exhibits some advantage over the new system in terms of public KS. However, choosing $m > 1$ allows reducing the value of z in the new system with respect to BL for the same SL, and this yields significant reductions in the public KS of the new system with respect to BL.

The values of the KS are the same for the McEliece versions of the above systems using matrices in systematic form.

4.2 Encryption rate

Due to the increased storage capabilities in recent years, the encryption rate $R_{\rm e}$ is currently often considered more critical than the public KS. In the Niederreiter versions, the information is encrypted through the intentional error vector, so the encryption rates are

$$R_{\rm e} = \frac{\log_2\binom{n}{t}}{(n - k)} \tag{9}$$

for the Goppa code-based system,

$$R_{\rm e} = \frac{\log_2\binom{n}{t_{\rm p}} + t_{\rm p}\log_2(q - 1)}{(n - k + z)\log_2 q} \tag{10}$$

for the new system (considering γ of length z as part of the ciphertext), and

$$R_{\rm e} = \frac{\log_2\binom{n}{t} + t\log_2(q - 1)}{(n - k + z)\log_2 q} \tag{11}$$

for the BL system. In the McEliece versions, the encryption rates are k/n for the Goppa code-based system, $(k - z)/n$ for BL, and $k/(n + z)$ for the new system. From (10) and (11), we observe that, with the Niederreiter version, when the same set of parameters is chosen and $m = 1$ (hence, $t_{\rm p} = t$), the encryption rate of the new system equals that of BL with the same SL, while with the McEliece version it is higher.

4.3 Security analysis

Cryptanalysis of the new system has to study two possible scenarios. First, the system should be secure against the best decoding algorithm capable of decoding a general linear code. As we will explain next, ISD is the best known algorithm in the literature for decoding a general linear code over a finite field $\mathbb{F}_q$. Second, we should consider possible structural attacks involving some distinguishers.

ISD was first introduced by Prange [24], then improved by Lee and Brickell [25], Leon [26] and Stern [27], and in recent years has benefited from a number of further improvements [28–30]. In the general area of code-based cryptography, ISD represents the most dangerous non-polynomial-time attack against general code-based systems. In this study, we estimate the cost of the algorithm as it can be found in [28], where the extension to non-binary fields (that are of interest here) is also provided, and we use it for both the binary and the non-binary cases in order to perform a fair comparison. In fact, there are no simple generalisations of the most recent algorithms [30] to work over non-binary fields. However, as shown in [19], we can suppose that their possible application could result in a WF reduction in the order of 2^9 or less with respect to the algorithm in [28]. Hence, such an advance would result in a general, slight reduction of the SL of the systems we consider and, more importantly, would not significantly affect the results of our comparative assessment.

Let us now discuss the possibility of a structural attack. Starting with the work of Sidelnikov and Shestakov [18], it has been recognised that McEliece-type systems publishing a disguised version of a private GRS code are insecure. The main reason for this is that the Schur square code of a GRS code has, in general, a very small dimension compared to a random code of the same dimension and the dimension of the Schur square remains invariant under monomial transformations. In other words, the Schur square can serve as a distinguisher. Couvreur *et al.* [21] used the Schur square to attack the system in [19], and Marquez-Corbella *et al.* explained in [31] how the BL system [22] can be attacked using the Schur square. More recently Couvreur *et al.* [32] came up with general polynomial-time attacks against a large class of algebraic geometric codes and their subcodes (and GRS codes are algebraic geometric).

The question, therefore, is if the Schur square of the proposed code could be dimension deficient. For this, let us consider once more the relevant equations

$$\boldsymbol{x} = \boldsymbol{H}' \cdot \boldsymbol{e}^{\rm T} \text{ and } \gamma = \boldsymbol{a} \cdot \boldsymbol{e}^{\rm T}.$$

Combining these equations one can study a related parity-check matrix, i.e.

$$\tilde{\boldsymbol{H}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{H}' \\ -\gamma & \boldsymbol{a} \end{bmatrix}, \tag{12}$$

where $\boldsymbol{0}$ is an all-zero matrix with size $r \times z$. Clearly, if one has efficient decoding for this parity-check matrix the system becomes insecure. For this, we wish to comment on two extreme cases.

i. Assume that $m = 1$, i.e. $\boldsymbol{H}$ simply represents a disguised GRS code. Then, it is fairly clear that the Schur square of the defined code is dimension deficient unless the size of z is chosen sufficiently large. So, it is important that $m > 1$, what we will do in all our examples.
ii. Consider the other extreme case where $z = 0$. Here, Couvreur *et al.* [21] derived a polynomial-time attack in a case that $1 < m < 2$. Bolkema *et al.* [33] studied the case of $m = 2$ and they called this situation 'the weight two masking of the GRS system'. Using extensive simulations, they conjectured that over large fields the Schur square of the public code has the expected dimension of a random code. More recently, Weger [34] has shown that the probability that the Schur square has maximal dimension approaches 1 as the field size q goes to infinity.

These remarks should make it clear that a distinguisher attack using the Schur square computation becomes out of reach when m and z are chosen sufficiently large. When $z = 0$ it seems that m should be chosen at least two in order to avoid distinguisher attacks. When $z > 0$ we do not know what the minimal value for m should be to avoid the Schur square distinguisher attack and it will be a question of future research to come up with a precise value for this.

4.4 Examples

On the basis of the above performance and security metrics, let us compare the new system based on GRS codes with the classic binary Goppa code-based system and the BL system based on GRS subcodes. For such a purpose, we consider some instances of these systems approximately achieving the same SL and compare their features. For GRS code-based systems, we consider full length GRS codes defined over $\mathbb{F}_q$, with $q = n + 1$ being a prime. Goppa code-based systems instead exploit irreducible binary Goppa codes with length equal to a power of two.

In Tables 1 and 2, we, respectively, consider codes with a SL of at least 2^{180} and 2^{260}, estimated as the WF of attacks based on ISD, computed according to [28]. On the basis of an exhaustive search performed over the range of parameters of interest, we report the solutions achieving the smallest public KS expressed in kibibytes (KiB) for the classic binary Goppa code-based cryptosystem, for the new variant of GRS code-based cryptosystem with $m = 1$, and for the BL cryptosystem, using $z = n - k$. We can notice that the new system with $m = 1$ is able to achieve smaller key sizes than the Goppa code-based solution, and it may have a higher encryption rate in the McEliece version. The new system has always higher encryption rate than BL, particularly for the McEliece version, but larger KS.

As mentioned, $m = 1$ was considered as in this case the new system is comparable to BL, but an m slightly larger than 1 is certainly preferable, to better protect the secret code. In Tables 3–5, a few values of m and z are tested and the instances with the smallest key sizes are presented. It is evident how it is possible to achieve very interesting parameters, namely high encryption rates and compact public keys. Considering the instances in the tables, the reduction in public KS with respect to the Goppa code-based solution with the same SL can reach 73%.

To conclude the comparison, in Table 6 we report two instances of the original GRS code-based cryptosystem in [19], achieving smallest key sizes for a SL of 2^{180} and 2^{260}, respectively, and satisfying $m = 1.1 \cdot (1 + R) > 1 + R$ to avoid the attack in [21]. Through the comparison with the previous tables, we observe that, for the same SLs, the new system is able to achieve a reduction in public KS by more than 50%.

Table 1 System performance comparison for $\mathrm{SL} = 2^{180}$: (a) Goppa code-based system, (b) new system with $m = 1$ and $z = n - k$, (c) BL

Variant	n	k	$t = t_\mathrm{p}$	WF	KS, KiB	R_e (Niederreiter)	R_e (McEliece)
(a)	4096	3004	91	180.06	400.44	0.5724	0.7334
(b)	1282	1146	68	180.01	392.89	0.3850	0.8082
(c)	1212	1062	75	180.05	342.15	0.3806	0.7525

Table 2 System performance comparison for $\mathrm{SL} = 2^{260}$: (a) Goppa code-based system, (b) new system with $m = 1$ and $z = n - k$, (c) BL

Variant	n	k	$t = t_\mathrm{p}$	WF	KS, KiB	R_e (Niederreiter)	R_e (McEliece)
(a)	8192	6957	95	260.57	1048.82	0.6012	0.8492
(b)	1950	1754	98	260.10	917.37	0.3798	0.8173
(c)	1788	1560	114	260.11	801.13	0.3732	0.7450

Table 3 New system performance for $m = 1.2$, $\mathrm{SL} = 2^{180}$ and $\mathrm{SL} = 2^{260}$

SL	z	n	k	t	t_p	WF	KS, KiB	R_e (Niederreiter)	R_e (McEliece)
180	10	796	634	81	67	181.03	130.09	0.5870	0.7866
180	30	886	722	82	68	180.50	172.24	0.5304	0.7882
180	50	918	740	89	74	180.19	210.43	0.4879	0.7644
260	10	1162	928	117	97	260.64	284.27	0.5894	0.7918
260	30	1222	976	123	102	260.21	435.37	0.5467	0.7796
260	50	1276	1018	129	107	260.03	408.04	0.5128	0.7677

Table 4 New system performance for $m = 1.3$, $\mathrm{SL} = 2^{180}$ and $\mathrm{SL} = 2^{260}$

SL	z	n	k	t	t_p	WF	KS (KiB)	R_e (Niederreiter)	R_e (McEliece)
180	10	760	544	108	83	180.04	146.06	0.5399	0.7065
180	30	810	576	117	90	180.16	186.60	0.4989	0.6857
180	50	852	594	129	99	180.11	229.80	0.4671	0.6585
260	10	1122	810	156	120	260.56	326.36	0.5399	0.7155
260	30	1200	888	156	120	260.14	389.81	0.5104	0.7220
260	50	1236	898	169	130	260.30	454.98	0.4843	0.6983

Table 5 New system performance for $m = 1.8$, $\mathrm{SL} = 2^{180}$ and $\mathrm{SL} = 2^{260}$

SL	z	n	k	t	t_p	WF	KS (KiB)	R_e (Niederreiter)	R_e (McEliece)
180	10	1092	804	144	80	180.19	298.65	0.4042	0.7296
180	30	1116	792	162	90	180.05	357.44	0.3789	0.6911
180	50	1180	852	164	91	180.00	418.54	0.3594	0.6927
260	10	1600	1168	216	120	260.12	676.31	0.4012	0.7255
260	30	1626	1154	236	131	260.03	771.67	0.3828	0.7054
260	50	1722	1268	227	126	260.04	865.19	0.3691	0.7156

Table 6 Original GRS code-based system performance for $\mathrm{SL} = 2^{180}$ and $\mathrm{SL} = 2^{260}$

SL	n	k	m	t	t_p	WF	KS, KiB	R_e (Niederreiter)	R_e (McEliece)
180	946	504	1.686	221	131	180.24	268.87	0.4209	0.5328
260	1422	786	1.708	318	186	260.26	639.19	0.4111	0.5527

Table 7 Complexity comparison between the binary Goppa code-based Niederreiter cryptosystem, the original GRS code-based cryptosystem and the new variant

	Binary Goppa code-based	GRS code-based original	GRS code-based new variant 2
n	4096	946	796
k	3004	504	634
q	2	947	797
t	91	221	81
t_p	91	131	67
E	115	3520	2009
D	23,024	20,818,690	47,710

4.5 Complexity

In this section, we assess the encryption and decryption complexity of the new system and compare it with that of the original version of the GRS code-based system and the classic binary Goppa code-based system. We restrict our comparison to the Niederreiter versions, as looking at other instances will not change significantly the sense of this comparison.

As done in [19, 35], let us assume that one addition between elements of $\mathbb{F}_q$ costs $\mathscr{S} = \lceil \log_2(q) \rceil$ binary operations, while one multiplication costs $\mathscr{M} = 2\mathscr{S}^2$ binary operations. We also consider that an inversion over $\mathbb{F}_q$ has the same cost as a multiplication. The right (or left, respectively) multiplication of an $x \times y$ matrix by a vector having w non-null elements requires to sum w columns (or rows, respectively) of the matrix, i.e. $(w-1)x\mathscr{S}$ (or $(w-1)y\mathscr{S}$, respectively) binary operations, plus $wx\mathscr{M}$ (or $wy\mathscr{M}$, respectively) binary operations to multiply each element of the vector by the corresponding matrix column (or row, respectively). The latter term must be considered only when working over $\mathbb{F}_q$ with $q > 2$. Computing the element-wise sum or product of two vectors with length x requires, on average, $x((q-1)/q)$ sums or multiplications. Since in GRS code-based systems q is in the order of some hundreds or more, the term $(q-1)/q$ can be set equal to 1 with a negligible impact on the numerical results.

Concerning encryption, if we consider the systematic version of the public parity-check matrix and split the vector $\boldsymbol{e}$ into its left and right parts, $\boldsymbol{e} = [\boldsymbol{e}_l | \boldsymbol{e}_r]$, the encryption function becomes $\boldsymbol{x} = \boldsymbol{e}_\mathrm{l}^\mathrm{T} + \boldsymbol{H}''_\mathrm{r} \cdot \boldsymbol{e}_\mathrm{r}^\mathrm{T}$, where $\boldsymbol{e}_\mathrm{r}$ has a weight equal to $w = (k/n)t_\mathrm{p}$ on average. Hence, if we neglect the transformation of the information vector into a constant weight vector and back, we obtain an encryption cost equal to

- $[(w-1)r + t_\mathrm{p} - w]$ binary operations for the classic binary code-based system (with codes over $\mathbb{F}_2$),
- $[(w-1)r + t_\mathrm{p} - w]\mathscr{S} + wr\mathscr{M}$ binary operations for the original GRS code-based system (with codes over $\mathbb{F}_q$, $q > 2$).

Concerning the proposed variant, we must consider the number of binary operations needed to compute $\gamma = \boldsymbol{a} \cdot \boldsymbol{e}^\mathrm{T}$, i.e. $(t_\mathrm{p} - 1)z\mathscr{S} + t_\mathrm{p}z\mathscr{M}$ binary operations.

Concerning decryption, as done in [19], we consider the classic GRS syndrome decoding algorithm [35], requiring

i. $4t(2t+2)\mathscr{M} + 2t(2t+1)\mathscr{S}$ binary operations for the key equation solver,
ii. $n(t-1)\mathscr{M} + nt\mathscr{S}$ binary operations for the Chien search, and
iii. $(2t^2 + t)\mathscr{M} + t(2t-1)\mathscr{S}$ binary operations for Forney's formula.

After decoding, Bob needs to compute $\boldsymbol{x}' = \boldsymbol{S} \cdot \boldsymbol{x}$, requiring further $(r-1)r\mathscr{S} + r^2\mathscr{M}$ binary operations, and $\boldsymbol{x}'' = \boldsymbol{x}' - \boldsymbol{H} \cdot \boldsymbol{b}^\mathrm{T} \cdot \gamma$, requiring further $r\mathscr{S} + (z-1)r\mathscr{S} + zr\mathscr{M}$ binary operations (taking into account that $\boldsymbol{H} \cdot \boldsymbol{b}^\mathrm{T}$ can be pre-computed only once). The original GRS code-based system also requires Bob to guess the value of γ. Each guessing attempt requires repeating the key equation solver and the computation of $\boldsymbol{x}''$ [19]. Therefore, for the GRS code-based systems, we can estimate the overall decryption complexity as

$$D_{\mathrm{GRS}} = \left\{[4t(2t+2) + zr]g + 2t^2 + (2n+1)t + r^2 - n\right\}\mathscr{M} + \left\{[2t(2t+1) + zr]g + 2t^2 + (2n-1)t + (r-1)r - n\right\}\mathscr{S}, \quad (13)$$

where g is the number of times the key equation solver needs to be executed.

For the original system, g equals the average number of attempts needed for correct guessing by Bob, i.e. $g = q^z/2$. For the proposed variants, instead, we have $g = 1$, since no repeated attempts are needed to obtain γ. Finally, for the classic binary Goppa code-based Niederreiter cryptosystem, the decryption complexity is [36]

$$D_{\mathrm{Goppa}} = n + 2[\log_2(n)]^2 t(2t+1) + \log_2(n)n(2t+1) + \frac{r^2}{2}. \quad (14)$$

In Table 7, we provide the values of the encryption (E) and decryption (D) complexity per information bit for the classic binary Goppa code-based system and instances of the old (Table 6) and new variant (Table 3, $z = 10$) of the GRS code-based system considered in this study for achieving 180-bit security.

By looking at the table, we observe that the gain in public KS with respect to the classic binary Goppa code-based system is paid in terms of complexity. This already occurred for the original GRS code-based system, and still holds for the new variant. However, it has to be noted that the new system achieves a considerable reduction in the decryption complexity with respect to the original GRS code-based system. In fact, the decryption complexity is reduced by more than two orders of magnitude. This makes the new system more balanced from the complexity standpoint since the encryption and decryption complexities are now more similar.

5 Conclusion

A new type of GRS code-based cryptosystem has been presented, which improves over previous solutions by allowing a larger set of instances featuring competitive parameters, such as KS and encryption rate, as well as resilience with respect to any known attack. This is also achieved without affecting complexity, which is even reduced with respect to the original GRS code-based scheme.

6 Acknowledgments

We would like to thank Alain Couvreur and Ludovic Perret for fruitful discussion on subcode attacks and the relations with the BL scheme. The Swiss National Science Foundation partially supported the work of Joachim Rosenthal under the grant no. 169510 and the work of Davide Schipani under the project IZK0Z2_173546.

7 References

[1] Shor, P.W.: 'Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer', *SIAM J. Comput.*, 1997, **26**, (5), pp. 1484–1509

[2] Mitchell, R.: 'IBM to sell use of its new 17-qubit quantum computer over the cloud', May 2017. Available at https://www.allaboutcircuits.com/news/ibm-to-sell-use-of-its-new-17-qubit-quantum-computer-over-the-cloud/

[3] Hardesty, L.: 'Toward mass-producible quantum computers', May 2017. Available at http://news.mit.edu/2017/toward-mass-producible-quantum-computers-0526

[4] National Institute of Standards and Technology: 'Post-quantum crypto project', December 2016. Available at https://csrc.nist.gov/projects/post-quantum-cryptography

[5] Chen, L., Liu, Y.-K., Jordan, S., *et al.*: 'Report on post-quantum cryptography'. Tech. Rep. NISTIR 8105, National Institute of Standards and Technology, 2016

[6] McEliece, R.J.: 'A public-key cryptosystem based on algebraic coding theory'. DSN Progress Report, 1978, pp. 114–116

[7] Berlekamp, E., McEliece, R., van Tilborg, H.: 'On the inherent intractability of certain coding problems', *IEEE Trans. Inf. Theory*, 1978, **24**, (3), pp. 384–386

[8] Niederreiter, H.: 'Knapsack-type cryptosystems and algebraic coding theory', *Probl. Control Inf. Theory*, 1986, **15**, (2), pp. 159–166

[9] Overbeck, R.: 'Structural attacks for public key cryptosystems based on Gabidulin codes', *J. Cryptol.*, 2008, **21**, (2), pp. 280–301

[10] Wieschebrink, C.: 'Cryptanalysis of the Niederreiter public key scheme based on GRS subcodes'. Post-Quantum Cryptography (PQCrypto 2010), Darmstadt, Germany, May 2010 (LNCS, **6061**), pp. 61–72

[11] Umana, V.G., Leander, G.: 'Practical key recovery attacks on two McEliece variants'. Proc. 2nd Int. Conf. on Symbolic Computation and Cryptography, Egham, UK, June 2010, pp. 27–44

[12] Monico, C., Rosenthal, J., Shokrollahi, A.: 'Using low density parity check codes in the McEliece cryptosystem'. Proc. IEEE Int. Symp. on Information Theory (ISIT 2000), Sorrento, Italy, June 2000, p. 215

[13] Baldi, M., Chiaraluce, F.: 'Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes'. Proc. IEEE Int. Symp. on Information Theory (ISIT 2007), Nice, France, March 2007, pp. 2591–2595

[14] Baldi, M., Chiaraluce, F., Garello, R., *et al.*: 'Quasi-cyclic low-density parity-check codes in the McEliece cryptosystem'. Proc. IEEE Int. Conf. on Communications (ICC 2007), Glasgow, Scotland, June 2007, pp. 951–956

[15] Otmani, A., Tillich, J.P., Dallot, L.: 'Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes'. Proc. First Int. Conf. on Symbolic Computation and Cryptography (SCC 2008), Beijing, China, April 2008, pp. 129–140

[16] Baldi, M., Bianchi, M., Chiaraluce, F.: 'Security and complexity of the McEliece cryptosystem based on QC-LDPC codes', *IET Inf. Sec.*, 2013, **7**, (3), pp. 212–220

[17] Misoczki, R., Tillich, J.-P., Sendrier, N., *et al.*: 'MDPC-McEliece: new McEliece variants from moderate density parity-check codes'. Proc. IEEE Int. Symp. on Information Theory (ISIT 2013), Istanbul, Turkey, July 2013, pp. 2069–2073

[18] Sidelnikov, V.M., Shestakov, S.O.: 'On insecurity of cryptosystems based on generalized Reed–Solomon codes', *Discrete Math. Appl.*, 1992, **2**, (4), pp. 439–444

[19] Baldi, M., Bianchi, M., Chiaraluce, F., *et al.*: 'Enhanced public key security for the McEliece cryptosystem', *J. Cryptol.*, 2016, **29**, (1), pp. 1–27

[20] Couvreur, A., Gaborit, P., Gauthier-Umaña, V., *et al.*: 'Distinguisher-based attacks on public-key cryptosystems using Reed-Solomon codes', *Des. Codes Cryptogr.*, 2014, **73**, (2), pp. 641–666

[21] Couvreur, A., Otmani, A., Tillich, J.-P., *et al.*: 'A polynomial-time attack on the BBCRS scheme'. Public-Key Cryptography (PKC 2015), Gaithersburg, MD, USA, March 30–April 1 2015 (LNCS, **9020**), pp. 175–193

[22] Berger, T., Loidreau, P.: 'How to mask the structure of codes for a cryptographic use', *Des. Codes Cryptogr.*, 2005, **35**, (1), pp. 63–79

[23] Baldi, M., Bianchi, M., Chiaraluce, F., *et al.*: 'A variant of the McEliece cryptosystem with increased public key security'. Proc. 19th Int. Workshop on Coding and Cryptography (WCC), Paris, France, April 2011, pp. 173–182

[24] Prange, E.: 'The use of information sets in decoding cyclic codes', *IRE Trans. Inf. Theory*, 1962, **8**, (5), pp. 5–9

[25] Lee, P., Brickell, E.: 'An observation on the security of McEliece's public-key cryptosystem'. Advances in Cryptology (EUROCRYPT 88), Davos, Switzerland, May 1988 (LNCS, **330**), pp. 275–280

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

[27] Stern, J.: 'A method for finding codewords of small weight'. Coding Theory and Applications, Toulon, France, November 1988 (LNCS, **388**), pp. 106–113

[28] Peters, C.: 'Information-set decoding for linear codes over $\mathbb{F}_q$'. Post-Quantum Cryptography (PQCrypto 2010), Darmstadt, Germany, May 2010 (LNCS, **6061**), pp. 81–94

[29] Bernstein, D.J., Lange, T., Peters, C.: 'Smaller decoding exponents: ball-collision decoding'. Advances in Cryptology (CRYPTO 2011), Santa Barbara, CA, USA, August 2011 (LNCS, **6841**), pp. 743–760

[30] Becker, A., Joux, A., May, A., *et al.*: 'Decoding random binary linear codes in $2^{n/20}$: how $1+1=0$ improves information set decoding'. Advances in Cryptology (EUROCRYPT 2012), Cambridge, UK, April 2012 (LNCS, **7237**), pp. 520–536

[31] Márquez-Corbella, I., Martinez-Moro, E., Pellikaan, R.: 'The non-gap sequence of a subcode of a generalized Reed-Solomon code', *Des. Codes Cryptogr.*, 2013, **66**, (1–3), pp. 317–333

[32] Couvreur, A., Márquez-Corbella, I., Pellikaan, R.: 'Cryptanalysis of McEliece cryptosystem based on algebraic geometry codes and their subcodes', *IEEE Trans. Inf. Theory*, 2017, **63**, (8), pp. 5404–5418

[33] Bolkema, J., Gluesing-Luerssen, H., Kelley, C.A., *et al.*: 'Variations of the McEliece cryptosystem', in '*Algebraic geometry for coding theory and cryptography*' (Springer, 2017), pp. 129–150

[34] Weger, V.: 'A code-based cryptosystem using GRS codes'. Master thesis, University of Zürich, Switzerland, 2016

[35] Chen, N., Yan, Z.: 'Complexity analysis of Reed-Solomon decoding over GF(2^m) without using syndromes', *EURASIP J. Wirel. Commun. Netw.*, 2008, Article ID 843634

[36] Canteaut, A.: 'Attaques de cryptosystemes à mots de poids faible et construction de fonction t-résilentes'. PhD thesis, Université Paris, October 1996