DE GRUYTER
OPEN

TATRA
MOUNTAINS
Mathematical Publications

DOI: 10.1515/tmmp-2017-0025
Tatra Mt. Math. Publ. **70** (2017), 163–177

CONNECTING THE COMPLEXITY OF MQ- AND CODE-BASED CRYPTOSYSTEMS

Pavol Zajac

ABSTRACT. We study the connection between the MQ problem and the decoding problem, through the intermediate MRHS representation. The main goal of this study is to explicitly bound the complexity of solving MQ systems with decoding tools. The main observation is that although the MQ problem over $GF(2)$ can be efficiently transformed to syndrome decoding, the existing general decoding methods are not suitable to solve the system as efficiently as expected from the MQ representation.

1. Introduction

Both multivariate and code based post-quantum systems (over $\mathbb{Z}_2$) rely on the difficulty of particular NP-hard problems. It is known theoretically that there are polynomial-time reductions between NP-hard problems. However, it is not clear how the instances of the problem used for different types of cryptosystems are related in practice, and what effect this has on security/complexity trade-offs of various types of systems.

The main goal of this article is to stimulate a research that will provide explicit reductions and direct comparison of different types of post-quantum systems in terms of the estimated security. We are able to provide an explicit polynomial transformation between the MQ problem and the decoding problem. We can do this over $\mathbb{Z}_2$ using a process involving MRHS representation of a system of polynomial equations, and a transformation of the MRHS system to a decoding problem from [14, 16].

Unfortunately, the basic upper bound of the estimated decoding complexity is $O(2^{2cn^2})$, with $c \doteq 1/20$, which is not very tight. However, it is comparative with some proposed parameters of code-based schemes (see section 2.2 for discussion),

2010 Mathematics Subject Classification: 94A60,68P25,06E30.
Keywords: decoding problem, MQ problem, MRHS.
This work was supported by project VEGA 1/0159/17.

albeit with special code parameters not typically used in code based systems. A tighter complexity bound is given by $O(2^{c\mu})$, where μ is a multiplicative complexity of the transformation represented by the MQ system. This result is theoretical, as it requires the attacker to actually find a representation with the minimal number of products (leading to a MinRank problem).

It remains an open question, whether we can provide a similar transformation over general fields. Furthermore, it can also be interesting to find an opposite polynomial transformation from the decoding problem back to a compact MQ instance.

2. Notation and basic definitions

Symbol $\mathbb{F}$ denotes a finite field, and $\mathbb{Z}_2$ denotes finite field $GF(2)$. All vectors in this article are row vectors. Hamming weight of vector $v \in \mathbb{F}^n$, denoted by $w_H(v)$ is the number of non-zero coordinates of v. By $d_H(x, y)$ we understand Hamming distance of vectors x, y (Hamming weight of their difference).

Matrices are typed in boldface. Matrix $\mathbf{I}_n$ is the $n \times n$ identity matrix (if n is not specified, it should be clear from the context). Let $\mathbf{A}$ be a $n \times m$ matrix over some field $\mathbb{F}$, and let S be a set of vectors from $\mathbb{F}^n$. Then $S\mathbf{A} = \{x\mathbf{A}; x \in S\}$ will denote a set of vectors from $\mathbb{F}^m$. When S is a set of m vectors from $\mathbb{F}^n$, then $\mathbf{S}$ will denote an $(m \times n)$ matrix with rows from S (in some specified order).

2.1. MRHS equation systems

Definition 1. [11] Let $\mathbb{F}$ be a finite field. A Multiple-Right-Hand-Sides (MRHS) equation is an expression of the form

$$x\mathbf{M} \in S, \tag{1}$$

where $\mathbf{M} \in \mathbb{F}^{(n \times l)}$ is an $(n \times l)$ matrix, and $S \subset \mathbb{F}^l$ is a set of l-bit vectors. We say that $x \in \mathbb{F}^n$ is a solution of MRHS equation (1), if and only if $x\mathbf{M} \in S$.

A system of MRHS equations $\mathbb{M}$ is a set of m MRHS equations with the same dimension n, i.e.,

$$\mathbb{M} = \{x\mathbf{M}_i \in S_i; i = 1, 2, \ldots, m\}, \tag{2}$$

with $\mathbf{M}_i \in \mathbb{F}^{(n \times l_i)}$, and $S_i \subset \mathbb{F}^{l_i}$, respectively. A MRHS system can be written as one MRHS equation with the right-hand side given as the Cartesian product of the right-hand sides of the individual equations in the MRHS system, i.e.:

$$x\,(\mathbf{M}_1|\mathbf{M}_2|\cdots|\mathbf{M}_m) \in S_1 \times S_2 \times \cdots \times S_m, \tag{3}$$

where $\mathbf{M} = (\mathbf{M}_1|\mathbf{M}_2|\cdots|\mathbf{M}_m)$ is the matrix obtained by concatenating the individual left-hand side matrices.

A MRHS system is considered to be polynomially sized, if the total number of right hand sides in its representation is bounded by a polynomial function of n (this usually means that both m, and each $|S_i|$ in the Cartesian product representation are polynomially bounded in n). In such a system, the verification that x is a solution of the system is also polynomial in n. We will typically work with MRHS systems with fixed sizes and dimensions of the sets S_i, which we will denote $l = dim(v_{i,j}), v_{i,j} \in S_i$, and $k = |S_i|$, respectively.

Definition 2. [15] We define MRHS solution decision problem as: Given a MRHS system $x\mathbf{M} \in S$, decide whether there exists any solution $x \in \mathbb{F}^n$ of this system.

As was shown in [15], this problem is NP-hard (NP-complete for a family of polynomially sized MRHS systems). In practical applications, we are interested in finding some concrete solution of the system, not just that it exists. MRHS oracle O_{MRHS} is an (oracular) algorithm, that given MRHS system $x\mathbf{M} \in S$ produces any solution $x \in \mathbb{F}^n$ of this system, or $\perp$, if none exists.

Suppose there exists oracle O for the MRHS solution decision problem. Oracle O_{MRHS} can be constructed as follows: If O returns “NO”, return $\perp$. Else fix each value $x_1, x_2, \dots, x_n$ in sequence: Compute $r = (x_1, x_2, \dots, \hat{x}_i, 0, \dots, 0)\mathbf{M}$, where $x_1, \dots, x_{i-1}$ is already fixed. Verify with O for which $\hat{x}_i$ as a potential value for x_i, system $x'\mathbf{M}' \in S + r$, where $\mathbf{M}'$ contains last $n - i$ rows of $\mathbf{M}$, has a solution. This requires at most $n \cdot |\mathbb{F}|$ calls to O. Alternatively, if $|\mathbb{F}|$ is large, we can test with O a sequence of systems that are obtained by removing all but one vector from each S_i in the sequence. The final solution x is then computed by linear algebra. This requires at most $\sum |S_i|$ calls to O.

2.2. Decoding problem

Let $\mathbf{G}$ be an $k \times n$ matrix over a finite field $\mathbb{F}$. A linear (n, k, d)-code generated by $\mathbf{G}$ is the set

$$\mathcal{C}_{\mathbf{G}} = \left\{ u\mathbf{G}; u \in \mathbf{F}^k \right\},$$

where $d = \min \{w_H(v); v \in \mathcal{C} \setminus \{0\}\}$ is a code distance. We omit $\mathbf{G}$ in the subscript, if the generator matrix is obvious from the context.

Given $\mathbf{G}$, there exists an $(n-k) \times n$ parity check matrix $\mathbf{H}$ such that $\mathbf{G}\mathbf{H}^T = 0$. An arbitrary vector $c \in \mathbf{F}^n$ is a code word of $\mathcal{C}$ if and only if $c\mathbf{H}^T = 0$. Thus, a parity check matrix also uniquely defines a linear code.

In general, we call $s = w\mathbf{H}^T$ a syndrome. The vector space $\mathbb{F}^n$ can be factored into cosets $\mathcal{C} + w$, where every vector $c + w$ gives the same syndrome

$$(c + w)\mathbf{H}^T = w\mathbf{H}^T = s.$$

Let $t = \lfloor \frac{d-1}{2} \rfloor$. Let $e \in \mathbb{F}^n$, with $w_H(e) \leq t$, and let $s = e\mathbf{H}^T$. Then for every vector $x \in \mathbb{F}^T$, with $x\mathbf{H}^T = s$, we can find a unique codeword $c \in \mathcal{C}$ such that $d_H(x, c) = w_H(e)$. For two vectors $e_1 \neq e_2$, with corresponding weights at most t, the cosets $\mathcal{C} + e_1$ and $\mathcal{C} + e_2$ must be distinct, and the corresponding syndromes as well. In general, not all cosets contain vectors of weight at most t.

The syndrome decoding decision problem is defined as follows: Given linear (n, k)-code $\mathcal{C}$, $t \in \mathbb{Z}$, and syndrome s, decide whether there exists

$$e \in \mathbb{F}^n \quad \text{with} \quad w_H(e) \leq t \quad \text{such that} \quad e\mathbf{H}^T = s.$$

The syndrome decoding decision problem was proven to be NP-complete [4]. In real-world applications we are mostly interested in the non-decision version of the problem, that asks for a vector e (if it exists). If $t \leq \lfloor \frac{d-1}{2} \rfloor$, this question can also be formulated as a question of finding the shortest vector in the code generated by $\mathbf{G} \cup \{w\}$, where w is an arbitrary solution of $w\mathbf{H}^T = s$.

We define a syndrome decoding oracle O_{syn} as an oracular algorithm, that given inputs $s, \mathbf{H}, t$ produces any vector $e \in \mathbb{F}^n$, such that $e\mathbf{H}^T = s$, with $w_H(e) \leq t$, or returns $\perp$, if no such vector exists.

A syndrome decoding oracle can also be created from a decision version of the oracle O. If the decision oracle O returns “no”, O_{syn} returns $\perp$. Otherwise, we start by adding vectors from H to s till O returns “no”. We now know that final addition crossed the threshold t on Hamming distance of the (hidden) solution e. By reverting the last change and testing each position individually, we can reconstruct the entire vector e in $cn|\mathbb{F}|$ steps (for some small constant c).

Families of codes, for which the syndrome decoding problem is easy to solve, are used for the construction of error correcting codes. A code word is transmitted through the channel. A syndrome is computed from the received word, and vector e identifies the errors added during the transmission. Low weight criterion corresponds to a model of the transmission channel with random errors and a low error rate. We can however imagine many different models for error distribution, and the corresponding decoding problems.

The regular decoding decision problem[1] is defined as follows: Given matrix $\mathbf{H}$, $t \in \mathbb{Z}$, $n = mt$, and syndrome s, decide whether there exists

$$\in \mathbb{F}^n \;\; \text{with} \;\; e\mathbf{H}^T = s, \;\; w_H(e) = t,$$

and

$$w_H(e_{mi+1}, e_{mi+2}, \ldots, e_{m(i+1)}) = 1 \quad \text{for each} \;\; i = 0, 1, \ldots, t-1.$$

Similarly to the syndrome decoding oracle, we can define a regular decoding oracle O_{reg}: given inputs $s, \mathbf{H}, t$, return any vector $e \in \mathbb{F}^n$, such that $e\mathbf{H}^T = s$, with $w_H(e) = t$, and $w_H(e_{mi+1}, e_{mi+2}, \ldots, e_{m(i+1)}) = 1$ for each $i = 0, 1, \ldots, t-1$, or return $\perp$, if no such vector exists. A regular decoding oracle can also be constructed from the decision version, similarly to the syndrome decoding version.

[1] We use the word regular in the sense defined in [1], and later used, e.g., in [5].

The regular decoding problem models a situation where the codeword contains t blocks of size m, and to each block we add a single error (to an unknown position). This problem arises in connection with solving MRHS equations, as shown in [14, 16]. In dedicated subsections of section 3 we revisit this connection, and show also a correspondence between the classical syndrome decoding and the regular decoding.

2.3. MQ problem

Let $\mathcal{F} = \{f_1, f_2, \ldots, f_m\}$, $f_i \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, be a set of multivariate polynomials of degree at most 2. The multivariate quadratic (MQ) decision problem is: given $y = (y_1, y_2, \ldots, y_m) \in \mathbb{F}^m$, decide whether there exists $x \in \mathbb{F}^n$, such that $y_i = f_i(x)$, for every $i = 1, 2, \ldots, m$.

We can again define an MQ oracle O_{MQ} : Given y, $\mathcal{F}$, return x such that $y_i = f_i(x)$, for every $i = 1, 2, \ldots, m$, or $\perp$ if no such solution exists. Similarly to MRHS case, we can build oracle O_{MQ} from decision version, by testing values x_i in sequence.

It seems obvious that there is a strong connection between MRHS decision problem and MQ decision problem, as both ask whether some algebraic variety is non-empty. On the other hand, the variety is defined in a different way:

- MQ problem requires a sparse polynomial representation (limit on the degree),
- MRHS problem requires a sparse representation as an intersection of unions of affine spaces.

In Section 3, we show the correspondence between the MQ problem and the MRHS problem, and through the MRHS representation, with the decoding problem.

3. Connecting MQ problem and decoding problem

In the following section we show a polynomial transformation that can be used to get from the MQ problem to the MRHS problem, then to the regular decoding problem and finally to the syndrome decoding problem. We can only construct the whole transformation for systems over $\mathbb{F} = \mathbb{Z}_2$, a general algorithm is still an open question. We separate all these transformations to the corresponding subsections, and then connect the results in the final section.

3.1. From MQ problem to MRHS problem

Let x be a solution of MQ system $y_i = f_i(x)$. Let us substitute each term $x_i x_j$ (if $i = j$, we get x_i^2) which has a non-zero coefficient anywhere in the system by $z_{i,j}$. We will denote the number of such terms by N. Then x can be computed as a solution of the combination of linear system $y_i = (x|z)\mathbf{M}$ and non-linear equations $z_{i,j} - x_i x_j = 0$. Using $y_i = (x|z)\mathbf{M}$, we can express $rank\,(\mathbf{M})$ of the variables as linear combinations of the remaining $n + N - \mathrm{rank}\,(\mathbf{M})$ variables u and constants from y.

We can suppose that $\mathrm{rank}\,(\mathbf{M}) = m < n + N$. If not, we can remove some linearly dependent equations from the system (this cannot decrease N, because the equation that contains a singular term is not linearly dependent on others). This means that we can find matrix $\mathbf{M}_{i,j}$ and constants $c = (c_0, c_1, c_2)$, such that $(x_i, x_j, z_{i,j})$ can be written as $u\mathbf{M}_{i,j} + c$. Thus, each non-linear equation $z_{i,j} - x_i x_j = 0$ corresponds to a MRHS equation

$$u\mathbf{M}_{i,j} \in S_{i,j} = \{(a, b, ab) + c; a, b \in \mathbb{F}\}.$$

After collecting all such non-linear equations, we can write the final MRHS system as

$$u\,(\cdots \mathbf{M}_{i,j} \cdots) \in S_{1,1} \times S_{1,2} \times \cdots$$

To simplify the notation, we will simply write the system as

$$u\mathbf{M}' \in S,$$

where $\mathbf{M}'$ denotes the concatenation of matrices $\mathbf{M}_{i,j}$, and S denotes the Cartesian product of the corresponding sets $S_{i,j}$.

The dimensions of the final MRHS system are $n' = N + (n - m)$ (more exactly: $n' = n + N - \mathrm{rank}\,(\mathbf{M})$), $m' = N$ (the number of non-linear equations), $l' = 3$ (each non-linear equation relates three linear combinations of unknowns), and $k' = |\mathbb{F}|^2$ (each choice of $a, b \in \mathbb{F}$ provides a distinct right-hand side vector). We have $N \leq \frac{n(n-1)}{2} + n$ (note that for $\mathbb{F} = \mathbb{Z}_2$, we can also remove n terms x_i^2), thus (for small fields $\mathbb{F}$) the MRHS system is polynomially sized.

Up till now, we were only doing linear algebra transformations on the original MQ system. Thus, given oracle O_{MRHS}, we can construct O_{MQ} with a single call to O_{MRHS} as follows:

(1) If O_{MRHS} returns $\perp$, return $\perp$;
(2) If O_{MRHS} returns u, compute x from u, return x.

The overhead cost consists only of linear algebra operations with the matrix of size $(n + N) \times m$ over $\mathbb{F}$.

3.2. From MRHS problem to regular decoding (and back)

Given a (polynomially sized) MRHS system $x\mathbf{M} \in S_1 \times \cdots S_m$, with parameters n, m, l, k, we can use regular decoding oracle O_{reg} to find solutions of the MRHS problem.

Let

$$\mathbf{S} = \begin{pmatrix} \mathbf{S}_1 & 0 & \cdots & 0 \\ 0 & \mathbf{S}_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{S}_m \end{pmatrix},$$

be a block diagonal matrix, with each $\mathbf{S}_i$ composed of vectors in sets S_i. Dimensions of $\mathbf{S}$ are $(mk \times ml)$. Let $\mathbf{H}$ be a $(ml - n) \times ml$ matrix such that $\mathbf{M}\mathbf{H}^T = \mathbf{0}$. It is thus a parity check matrix for the linear code $\mathcal{C}$ generated by $\mathbf{M}$. Now x is a solution of the MRHS system if and only if there exists some $c \in \mathbb{F}^{ml}$, such that $x\mathbf{M} = c$, and $c \in S_1 \times \dots S_m$.

We can define a one to one mapping between vectors

$$c \in S_1 \times \dots S_m,$$

and regular codewords

$$r = (e^{r_1},\ \ e^{r_2}, \dots, e^{r_m}) \in \mathbb{F}_k^m,$$
$$w_H(e^i) = 1,\ \ e^i \in \mathbb{F}_k,\ \ k = |S_i|.$$

Each r_i has value from $\{1, 2, \dots, k\}$, corresponding to an index of the projection of c onto S_i in some specified order. We can use the same order which is used to convert S_i to $\mathbf{S}_i$. Thus, $r \cdot \mathbf{S} = c$.

Let $\mathbf{V} = \mathbf{S}\mathbf{H}^T$. Any regular solution of $\hat{r}\mathbf{V} = 0$ will provide vector $\hat{c} = \hat{r}\mathbf{S}$, that is a codeword of $\mathcal{C}$. If $\mathbb{F} = GF(2)$, we have also found a solution of MRHS system, as

$$\hat{r}\mathbf{S} \in S_1 \times \dots \times S_m.$$

Thus for $GF(2)$, we can use O_{reg} to solve the MRHS problem. Block matrix $\mathbf{V}$, with m blocks and total dimensions $(mk) \times (ml - n)$, is the input of the regular decoding oracle. If the O_{reg} answers $\perp$, the MRHS system does not have a solution. Otherwise, for output r of O_{reg}, compute $c = r\mathbf{S}$, and find $x\mathbf{M} = c$ by linear algebra (in polynomial time).

Suppose that we have started with an MQ problem over $GF(2)$ with parameters (n, m, N). We have transformed this problem to MRHS problem with parameters

$$n' = N + (n - m), \quad m' = N, \quad l' = 3, \quad \text{and} \quad k' = |\mathbb{F}|^2 = 4.$$

The parameters of the regular decoding problem would be

$$n'' = m'k' = 4N,$$
$$r'' = n'' - k'' = m'l' - n' = 3N - (N + n - m) = 2N - n + m,$$

or

$$k'' = 2N + n - m, \quad \text{and} \quad t = m' = N.$$

3.3. Connection between regular decoding and syndrome decoding

Let us now explore the exact nature of the connection between the syndrome decoding and the regular decoding oracles. It can be trivially seen that if O_{reg} returns some vector e of weight t, then e is also a solution of the corresponding syndrome decoding problem with bound t. Conversely, if there is no e of weight at most t with $e\mathbf{H} = s$, then there cannot be any regular decoding with the same parameter t.

On the other hand, the number of errors t for regular decoding problem can be much higher than corresponds to a code distance. As such, a regular solution to the decoding problem can be only one of many solutions for the syndrome decoding problem. Moreover, if there is a unique word with the given distance t to a codeword that solves the syndrome decoding problem, there is only a very low chance that it is also regular (each of t parts has weight 1). However, for each e with $w_H(e) \leq m$, there is a set of permutation matrices $\mathcal{P}$, such that $e\mathbf{P}$ is regular for $\mathbf{P} \in \mathcal{P}$. Thus, if there is a solution e to a syndrome decoding problem $(\mathbf{H}, s, t)$, there exists some matrix $\mathbf{P}$, such that $e\mathbf{P}$ is a solution to a regular decoding problem $(\mathbf{P}^{-1}\mathbf{H}, s, t)$. Unfortunately, we do not know the set $\mathcal{P}$, until we find e.

What we want is a polynomial transformation between a syndrome decoding and a regular decoding that is one-to-one. We propose one such transformation from a regular to a syndrome decoding over $\mathbb{F} = \mathbb{Z}_2$. We are given input $\mathbf{H}, s, t$ to a regular decoding problem:

(1) From $n \times r$ matrix $\mathbf{H}$, with $n = mt$, construct a new $n \times (r + t)$ matrix $\mathbf{H}'$, such that

$$\mathbf{H}' = \left(\begin{array}{c|c} & \mathbf{J}_1 \\ & \mathbf{J}_2 \\ \mathbf{H} & \vdots \\ & \mathbf{J}_t \end{array} \right),$$

where each $\mathbf{J}_i$ is an $m \times t$ matrix, that contains all ones in the ith column, and zeroes in other columns.

(2) The new syndrome will be $s' = (s|11\cdots 1)$.

If $O_{reg}(\mathbf{H}, s, t)$ returns some e, the same e is a solution of $O_{reg}(\mathbf{H}', s', t)$. This is because additional parity checks in each block sum to one (the syndrome value) due to the regularity of the solution (this is also why this construction does not work for general $\mathbb{F}$). As e is a solution of $O_{reg}(\mathbf{H}', s', t)$, it is also one of solutions of $O_{syn}(\mathbf{H}', s', t)$. We would like to show that there is no solution of $O_{syn}(\mathbf{H}', s', t)$, that is not regular.

Suppose that some e' is a solution of $e'\mathbf{H}' = s'$ of weight at most t, which is not regular. If e' has weight less than t, there is at least one block that has weight zero. If e' has weight t, but is not regular, it must contain more than one non-zero coordinate in some block. Because $n = mt$ there exists some i, such that $w_H(e'_{mi+1}, e'_{mi+2}, \dots, e'_{m(i+1)}) = 0$. But this means that we cannot get s' by multiplying $e'\mathbf{H}'$, as the product of e' and ith added column would be zero.

Thus we have proven the following lemma:

Lemma 1. *Over $\mathbb{F} = \mathbb{Z}_2$, there exists a polynomial equivalent transformation from O_{reg} with parameters t, $n = mt$, r, to O_{syn} with parameters t, $n = mt$, $r + t$.*

It remains an open question, whether Lemma 1 can be generalized to all finite fields.

The corollary of Lemma 1 is that, for binary fields, we can use the existing syndrome decoding algorithms to solve regular decoding problems. We slightly modify the underlying code. While the code length n remains the same, code rate is decreased from $\frac{n-r}{n}$ to $\frac{n-r-t}{n}$. The construction of $\mathbf{H}'$ guarantees that if there is a unique solution of the regular decoding problem it would also be a unique solution of the syndrome decoding problem, and if there are more solutions of the syndrome decoding, they always have weight t, and are regular.

The opposite transformation does not seem to be so well defined. Given a regular decoding oracle, we would like to obtain an arbitrary error vector of weight at most t for some instance of the syndrome decoding problem $(\mathbf{H}, s, t)$. The first problem is that the regular decoding is only well defined for $n = mt$, where t is fixed, and m is some integer. If we know the exact number of errors to be t, we can compute $m = \lceil \frac{n}{t} \rceil$, and extend the code to the size $n' = mt$ by adding arbitrary extra rows to $\mathbf{H}$. If we do not know the exact t, we must try all potential values separately.

If a regular decoding oracle produces a negative answer, it does not mean that there is no solution for the syndrome decoding problem. Given some solution e of the syndrome decoding problem, we know that some $e\mathbf{P}$ is a solution of the regular decoding problem $(\mathbf{P}^{-1}\mathbf{H}, s, w_H(e))$. The problem is to find $\mathbf{P}$, or to show that no such $\mathbf{P}$ exists (meaning that there is no solution to the syndrome decoding problem).

If $t = 1$, the transformation is trivial, as there is only one block and the regular decoding oracle behaves in the same way as the syndrome decoding oracle. With $t = 2$ we have two blocks. Either we get a regular solution (in which case O_{reg} returns the correct answer), or both errors are located in one of the blocks (in which case we get answer $\perp$). In the second case, we can try to obtain the solution by cutting each block in half and exchange one of the halves. We can continue this process recursively. In $\log_2 n$ steps we either obtain a solution or get a proof that no such solution exists. It is not clear to us, whether this process can be extended to an arbitrary t (but this issue is out of scope of the present paper).

4. Complexity of solving MQ systems with syndrome decoding algorithms

Let $\mathbb{F} = GF(2)$. We are given MQ system with m linearly independent MQ equations in n variables and N quadratic terms with non-zero coefficients. Value N can be bounded by $\binom{n}{2}$.

We can solve a MQ problem with parameters n, m, N via the MRHS problem with parameters $n' = N + (n - m)$, $m' = N$, $l' = 3$, $k' = 4$. Obviously, if $N \leq m - n$, the system can be solved by linearisation directly in the MQ form, so we can consider that $n' > 0$. If $l'm' \leq n'$, or $2N \leq n - m$, the MRHS system can be solved trivially: for every potential right-hand side there exists a solution (space) obtained by a linear algebra [15]. Thus, we are only interested in non-trivial systems with $N > \frac{n-m}{2}$, and $N > m - n$.

We can solve a MRHS problem with parameters n', m', k', l' via the regular decoding problem with parameters

$$\hat{t} = m', \quad \hat{n} = k'm', \quad \hat{r} = l'm' - n'.$$

After substituting the values from the original MQ problem we get

$$\hat{n} = 4N, \quad \hat{r} = 2N - n + m, \quad \hat{t} = N.$$

We can solve a regular decoding problem with parameters (t, n, r) via the syndrome decoding problem with parameters $(t, n, r + t)$. After substituting the values from the original MQ problem we get a decoding problem with parameters $(N, 4N, 3N - n + m)$. This means, that to solve a MQ problem in $\mathbb{F} = \mathbb{Z}_2$, we can use syndrome decoding algorithms for $(4N, N + n - m)$-linear code, looking for error vector of weight N. If the MQ problem is dense, with every term present, we have $N = n(n - 1)/2$. This gives an asymptotic complexity bound $O(2^{2cn(n-1)})$ for the decoding problem, where c is some constant. According to [3], we can take $c = 1/20$ as an upper bound. This result is clearly unsatisfactory, as MQ problem can be solved by faster methods with $O(2^{0.792n})$ asymptotic complexity [2].

On the other hand, we can try to compare cryptosystem parameters across various types of post-quantum systems. A public key of the signature scheme QUARTZ [8] is a system of $m = 100$ equations in $n = 107$ $GF(2)$ variables for security level 80. The basic decoding attack would have to work with code of size 22684 and dimension 5678, trying to find a codeword of weight 5671. The same security level was expected from the code-based signature scheme CFS , with proposed code length 65536, dimension 65392, correcting 9 errors [7]. Security level 80 is also expected of the McEliece cryptosystem with code length 1702, dimension 1219, correcting 45 errors [10]. Similar in dimensions is the code for QC-MDPC system [9] with code length 27212 and dimension 6803, but it is designed for 128 bit security, and should correct 68 errors. It would be nice to have a simple closed formula, even if not exactly precise, that we can use to compare these various cases.

4.1. Reducing the size of the decoding problem

An alternative approach can be used to (slightly) reduce the decoding complexity. Note that MRHS systems can be used to efficiently represent generic Boolean functions $F : \mathbb{Z}_2^{l_1} \to \mathbb{Z}_2^{l_2}$ that depend only on a small number of variables (such as S-boxes). The left-hand side contains $(l_1 + l_2)$ input and output variables (or any linear combinations of variables), the right-hand side contains 2^{l_1} pairs of values $(x, F(x))$.

When we return to the transformation from a MRHS system to a decoding problem, we can see that the codeword size is actually the total number of right-hand sides (RHS) in the MRHS system. For a general MQ system we expect $N = n(n-1)/2$ product equations, with 4 RHS each. Without the loss of generality, let the system contain the following 3 products: x_1x_2, x_1x_3, x_2x_3. These terms can be substituted by 3 product equations with 12 RHSs (altogether). We can also construct a single MRHS equation that contains all three substitutions:

$$\begin{array}{ccc|ccc}
x_1 & x_2 & x_3 & z_{1,2} & z_{1,3} & z_{2,3} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
 & & & \vdots & & \\
0 & 1 & 1 & 0 & 0 & 1 \\
1 & 1 & 1 & 1 & 1 & 1
\end{array}$$

Instead of 12 RHSs we get only 8. In general, we can group s variables to produce $\binom{s}{2}$ substitutions with a single MRHS equation with 2^s RHSs . This is only useful for $s < 6$, because of the exponential growth of the number of RHSs with s.

There is another problem with grouping variables in this way. If we express all terms x_1, x_2, x_3 in one MRHS equation with 8 RHSs, we save 4 RHSs. However, we must also express combinations of x_1, x_2, x_4. If we grouped these variables again, we would be repeating the product x_1x_2. Thus, we would cover only two new products x_1x_4, and x_2x_4, with the total number of RHSs staying at 8. This gets worse with larger s, because of the larger number of combinations of variables we must consider.

We can instead group variables x_1, x_2, x_3, then separately group variables x_3, x_4, x_5, variables x_5, x_6, x_7, etc., and leave the cross-products intact. Unfortunately, this saves only 4 RHSs per 2 variables in the system, leading to a decoding complexity of $O(2^{2c(n-1)^2})$. It is an open question, how can we express the attacked MQ system in the most efficient way in terms of the total number of RHSs of the MRHS system.

4.2. Tighter complexity bounds based on product equations

Let us consider a system of m MQ equations that can be written in the form $l_{i,1} \cdot l_{i,2} = l_{i,3} + c_i$, where $l_{i,j}(x)$ are linear functions and $c_i \in \mathbb{Z}_2$. We can substitute each product using $y_i = l_{i,1}(x) \cdot l_{i,2}(x)$ (a product equation). This gives us a MRHS system, and further a decoding instance with $N = m$. The asymptotic complexity of solving a system of this type can be bounded by $O(2^{4cm})$. Depending on the ratio of n/m, the decoding algorithm can be faster than the brute-force approach in this case.

The complexity of the transformed problem depends on the number of product equations, but does not depend on the affine part of the system. Thus we can ask: For a given MQ problem, what is the smallest number of product equations N_{min} that can be used to represent the problem? Clearly, $N_{min} \leq n(n-1)/2$, as we can simply take all different terms of degree 2 as product equations. Let us consider a different approach: In each of m equations, separate terms with x_1 to produce the following system:

$$x_1\mathbf{M}_1 x^T = F_1(x_2, x_3, \ldots, x_n) + \mathbf{A}_1 x^T + c_1.$$

Function F_1 is a quadratic function in the remaining variables. Matrix $\mathbf{M}_1$ represents (in rows) individual linear functions that are multiplied by x_1 in each of m equations. We can use equivalent row operations on the system to produce $\mathrm{rank}\,(\mathbf{M}_1)$ product equations on the left hand side (replace each non-zero row of reduced $\mathbf{M}_1$ by y_i, and add $y_i = x_1 m_i(x)$ to a set of product equations). We are working in $\mathbb{Z}_2$, so $x_1^2 = x_1$, and the first column of $\mathbf{M}_1$ can contain only zeroes. This means that $\mathrm{rank}\,(\mathbf{M}_1) \leq \min(m, n-1)$. After we have separated all quadratic terms with x_1, we can proceed similarly with $x_2, \ldots, x_{n-1}$. When processing x_i, all quadratic terms with x_j, $j < i$ are already substituted, so we get $\mathrm{rank}\,(\mathbf{M}_i) \leq \min(m, n-i)$. If $m \geq n$, the total number of equations can be again bounded by $N_{min} \leq n(n-1)/2$. On the other hand, if $m = n - d$ for some $0 < d < n$, we get a smaller bound $N_{min} \leq (n-d)(\frac{n-1}{2} + \frac{d}{2})$. It is however still quadratic in n.

Another approach is to write the system in the symbolic form

$$\mathfrak{M}x^T = \mathbf{M}x^T + c,$$

where $\mathfrak{M}$ is a $m \times n$ matrix of linear forms in

$$x_1, x_2, \ldots, x_n.$$

This corresponds to: moving the affine part to the right, and separating variables x_i from the remaining products in each of the MQ equations in some well defined way (e.g., x_1 first, etc.). When linearising the system, each non-zero linear form $l_{i,j}$ in $\mathfrak{M}$ produces one product equation $z_{i,j} = l_{i,j}x_j$. Thus, the total number of product equations depends on the number of non-zero linear forms in $\mathfrak{M}$, further reduced by the number of repeated linear forms in each column of $\mathfrak{M}$. We can do equivalent row and column operations on $\mathfrak{M}$, to reduce this number, as follows:

$$(\mathbf{R} \cdot \mathfrak{M} \cdot \mathbf{C})(\mathbf{C}^{-1}x^T) = (\mathbf{RM})x^T + (\mathbf{R}c).$$

The question becomes: find a non-singular $m \times m$ matrix $\mathbf{R}$ and a non-singular $n \times n$ matrix C, such that $(\mathbf{R} \cdot \mathfrak{M} \cdot \mathbf{C})(\mathbf{C}^{-1}x^T)$ produces the minimal number of product equations.

The theoretical minimal bound can be obtained by rewriting the system in any way with the minimal number of products and unlimited number of sums. This minimal number is the multiplicative complexity of the associated quadratic boolean function. This problem was already studied by Boyar et.al. in [6]. The multiplicative complexity $\mu(f)$ of a single Boolean function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ is given by $\mu(f) = 1/2\text{rank}\,(\mathbf{A} \oplus \mathbf{A}^T)$, where $\mathbf{A} \oplus \mathbf{A}^T$ characterizes the quadratic form corresponding to f. This means that for a single Boolean function $\mu(f) \leq \lfloor n/2 \rfloor$. Their hypothesis for a set of Boolean forms is similar, and is related to a rank of a set of matrices corresponding to all component functions. For random MQ systems, this would lead to similar estimates as shown in the previous paragraph. Note that results based on the multiplicative complexity are theoretical, and essentially violate the main assumption of MQ systems: we know that the examined MQ system has a low complexity representation (trapdoor/private key system), but we do not know (or should not be able to know) how to reconstruct it.

4.3. Supporting experiments

We illustrate the theoretical results with experiments on 20-variable toy examples from Fukuoka MQ Challenge [13].[2] We use four ToyExample-type1-n20 instances, with 20 unknowns and 40 equations. It is possible to find the solution by trying at most 2^{20} variable assignments, for each of them computing 180 ANDs, and 8000 XORs (schoolbook version, around 2^{33} basic operations).

After transforming one of these systems to MRHS form, we obtain a system with $n = 170$ variables, and $m = 190$ MRHS (product) equations (each term, no reductions in size attempted). With our yet unpublished MRHS solver [12], we can solve these systems with 7039894-7481030 XORs and 7732366–8390297 table lookups (about 2^{23} basic operations), which takes 0.2 seconds on our experimental PC (single core process on Intel i7–3820 CPU, 3.60GHz).

The code generated by left-hand side of the MRHS system has parity check matrix $\mathbf{H}^T$ with dimensions 400×570. After multiplying the right-hand side block matrix, we get $\mathbf{V} = \mathbf{SH}^T$, a matrix with 190 blocks of four $GF(2)^{400}$ vectors. A regular decoding problem is equivalent to finding one vector in each of these blocks, such that their sum is a 0 vector. We can use the transformation

[2] `https://www.mqchallenge.org/`

from Section 3.3 to get a syndrome decoding problem: find the shortest solution of $c\mathbf{H}' = s$, where $\mathbf{H}'$ is a matrix with dimensions 760×590, and the weight of c is 190. We are not aware of an algorithm that can find such solution (without knowing the original MQ /MRHS problem) within the 2^{20} basic "operations" order of magnitude.

5. Concluding remarks

In this article we have explored the connection between various NP-hard problems related to post-quantum cryptography research. We have provided a way to explicitly transform MQ problem to decoding problem with the help of MRHS equation systems. We have used term-by-term representation of MQ problem, which does not seem optimal. The question of optimal representation of MQ problem as a decoding problem is related to the corresponding multiplicative complexity of the system, which is in itself a difficult open problem.

It would be interesting if we were also able to reverse the process, and find a compressed MRHS or MQ representation of the decoding problem. This would enable us to employ various tools and techniques that were already developed for solving algebraic systems for decoding problems and unify the security levels of these types of post-quantum systems.

REFERENCES

[1] AUGOT, D.—FINIASZ, M.—GABORIT, P.—MANUEL, S.—SENDRIER, N.: *Sha-3 proposal: FSB*, Submission to NIST (2008), 81–85.
`https://www.rocq.inria.fr/secret/CBCrypto/fsbdoc.pdf`

[2] BARDET, M.—FAUGÈRE, J.-C.—SALVY, B.—SPAENLEHAUER, P.-J.: *On the complexity of solving quadratic boolean systems*, J. Complexity **29** (2013), 53–75.

[3] BECKER, A.—JOUX, A.—MAY, A.—MEURER, A.: *Decoding random binary linear codes in $2^{n/20}$: How $1+1 = 0$ improves information set decoding.* In: *Adv. in Cryptology—EUROCRYPT 2012*, Lect. Notes in Comput. Sci. Vol. 7237, Springer-Verlag, 2012, pp. 520–536.

[4] BERLEKAMP, E.—MCELIECE, R.—VAN TILBORG, H.: *On the inherent intractability of certain coding problems (Corresp.)*, IEEE Trans. Inform. Theory **24** (1978), 384–386.

[5] BERNSTEIN, D. J.—LANGE, T.—PETERS, C.—SCHWABE, P.: *Faster 2-Regular Information-Set Decoding.* In: Coding and Cryptology: Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings (Y. M. Chee, Z. Guo, S. Ling, F. Shao, Y. Tang, H. Wang, C. Xing, eds.), Springer-Verlag, Berlin, 2011, pp. 81–98.

[6] BOYAR, J.—PERALTA, R.—POCHUEV, D.: *On the multiplicative complexity of boolean functions over the basis* $(\wedge, \oplus, 1)$, Theoret. Computer Sci. **235** (2000), 43–57.

[7] COURTOIS, N. T.—FINIASZ, M.—SENDRIER, N.: *How to achieve a McEliece-based digital signature scheme.* In: International Conference on the Theory and Application of Cryptology and Information Security, Springer-Verlag, Berlin, 2001, pp. 157–174.

[8] COURTOIS, N. T.—GOUBIN, L.—PATARIN, J.: *Quartz, an asymmetric signature scheme for short signatures on PC*, Primitive specification and supporting documentation (second revised version) (2001).
`http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.10.6603`

[9] MISOCZKI, R.—TILLICH, J.-P.—SENDRIER, N.—BARRETO, P. S.: *MDPC-McEliece: New McEliece variants from moderate density parity-check codes.* In: IEEE International Symposium on Information Theory - ISIT 2013, Istanbul, Turkey, Information Theory Proceedings (ISIT), 2013, pp. 2069–2073.

[10] NIEBUHR, R.—MEZIANI, M.—BULYGIN, S.—BUCHMANN, J.: *Selecting parameters for secure McEliece-based cryptosystems*, Int. J. Inf. Sec. **11** (2012), 137–147.

[11] RADDUM, H.—SEMAEV, I.: *Solving Multiple Right Hand Sides linear equations*, Design, Codes and Cryptography **49** (2008), 147–160.

[12] RADDUM, H.—ZAJAC, P.: *MRHS Solver Based on Linear Algebra and Exhaustive Search.* 2017. `https://eprint.iacr.org/2018/111.pdf`

[13] YASUDA, T.—DAHAN, X.—HUANG, Y.-J.—TAKAGI, T.—SAKURAI, K.: *MQ Challenge: Hardness Evaluation of Solving Multivariate Quadratic Problems.*, IACR Cryptology ePrint Archive **2015** (2015), p. 275.

[14] ZAJAC, P.: *A new method to solve MRHS equation systems and its connection to group factorization*, J. Math. Cryptol. **7** (2013), 367–381.

[15] ______ *MRHS equation systems that can be solved in polynomial time*, Tatra MT. Math. Publ. **67** (2016), 205–219.

[16] ______ *Upper bounds on the complexity of algebraic cryptanalysis of ciphers with a low multiplicative complexity*, Des. Codes and Cryptogr. **82** (2017), 43–56.

Received May 24, 2017

Institute of Comput. Sci. and Math.
Faculty of Electrical Engineering and
Information Technology
Slovak University of Technology
in Bratislava
Ilkovičova 3
SK–812-19 Bratislava
SLOVAKIA
E-mail: pavol.zajac@stuba.sk