

# Public-Key Cryptography from New Multivariate Quadratic Assumptions

Yun-Ju Huang<sup>1,3</sup>, Feng-Hao Liu<sup>2</sup>, Bo-Yin Yang<sup>3</sup>

<sup>1</sup> Faculty of Mathematics, Kyushu University, Japan

<sup>2</sup> Computer Science, Brown University, USA

<sup>3</sup> Institute of Information Science, Academia Sinica, Taiwan

**Abstract.** In this work, we study a new multivariate quadratic (MQ) assumption that can be used to construct public-key encryptions. In particular, we research in the following two directions:

- We establish a precise *asymptotic* formulation of a family of hard MQ problems, and provide empirical evidence to confirm the hardness.
- We construct public-key encryption schemes, and prove their security under the hardness assumption of this family. Also, we provide a new *perspective* to look at MQ systems that plays a key role to our design and proof of security.

As a consequence, we construct the *first* public-key encryption scheme that is *provably secure* under the MQ assumption. Moreover, our public-key encryption scheme is efficient in the sense that it only needs a ciphertext length  $L + \text{poly}(k)$  to encrypt a message  $M \in \{0, 1\}^L$  for any un-prespecified polynomial  $L$ , where  $k$  is the security parameter. This is essentially *optimal* since an additive overhead is the best we can hope for.

## 1 Introduction

Exploring different types of assumptions has been an important direction in the agenda of cryptography research. For robustness, this reduces the risk of a new mathematical/algorithmic/hardware breakthrough that breaks a particular assumption and renders all its following constructions insecure; for versatility, different assumptions usually have advantages for different applications. However, over the past 30 years, only a few candidates of computational problems are built as foundations on which more exciting cryptographic applications can build; for example, some well-structured algebraic, coding, or geometric problems (and their variants): DDH [17], Pairing (some are instantiated by elliptic curves) [10], RSA [46], McEliece [38], LWE [1, 43, 45], and some recent works for combinatorial problems [2].

This work is in a step of this agenda. We study a new type of assumption inspired from the field of solving multivariate quadratic (MQ) equations. In particular, we give the first asymptotic formulation of a family of MQ problems that enjoy some good mathematical structures and hardness. Thus one can use this formulation as a base to construct more interesting crypto primitives, such

as public-key encryption schemes. Our assumption considers a family of problems that can be viewed as solving MQ equations described as the followings (informally) :

**Definition 1 (The Hard Task (Informal)).** *Let  $\mathbb{F}_q$  be a finite field, and  $H$  be some subset of  $\mathbb{F}_q$ . Let  $S$  be a multivariate quadratic system with  $n$  variables and  $m$  polynomials whose coefficients are sampled from some distribution  $\chi$ .*

*Then a solver  $A$ , given  $(S, \mathbf{y} = S(\mathbf{x}))$  where  $\mathbf{x}$  is sampled uniformly from  $H^n$ , is asked to output some  $\mathbf{x}'$  such that  $S(\mathbf{x}') = \mathbf{y}$ .*

Actually, solving systems of non-linear equations is not a new topic, for it has been studied in commutative algebra and algebraic geometry, at least since Francis Sowerby Macaulay [36] (1902). Around the turn of the millennium, these techniques [14] were also found that they can be used as a cryptanalytic step. Claims (e.g. **XSL** [15]) concerning such techniques, today called “algebraic cryptanalysis”, were often over-optimistic, but equation-solvers over different finite fields such as **XL** [14], **F4**, **F5** [23, 24] are now significant topics for crypto.

The fundamental reason that algebraic cryptanalysis is not all-powerful is that solving systems of non-linear equations does not scale well with the parameters even with Moore’s Law. Theoretically, solving multivariate non-linear systems, or even just multivariate quadratic (MQ) equations has been proven to be NP-hard [25, 41] in the worst case, and practically, all the proposed solvers fail to solve the systems efficiently (i.e. in polynomial-time) for *most* non-trivial distributions [4, 35].

The above approach hints at inherent hardness in solving MQ equations, and consequently MQ could be a good choice as a base for designing crypto systems. Although this direction in fact has been considered for the last 20 years, however, it has had a rocky history. Many schemes were proposed, broken, sometimes patched, and sometimes broken again (see [18, 20, 21, 37, 39, 42], and [5, 6, 12, 40]). One objection frequently voiced is that the security of these systems is often ad-hoc, and thus hard to evaluate. Fundamentally, these approaches mostly were designed with a practical goal in mind. As a result, they considered concrete and fixed-parameter constructions, with a design security of, e.g.,  $2^{80}$ , with specialization to signatures with 160-bit hashes and optimizing for speed. Since MQ was examined not as a hardness basis but only as the most obvious attack or even some sanity check, the designers’ mindsets were not focusing on how to construct a reduction for their security proof, nor about extending their schemes in an asymptotic way. Thus, it seems that using the hardness to construct crypto construction remains an interesting open direction.

Berbain, Gilbert, and Patarin [4] explored this and constructed efficient pseudorandom generators (PRGs) based on the hardness of solving MQ equations. Berbain *et al.* considered fixed and concrete-parameter constructions, yet an asymptotic formulation of hard problems is implicit in their work. Consequently, many primitives such as pseudorandom functions (PRFs), symmetric encryptions, etc., in the Minicrypt world (i.e., one way functions exist) [33] can be constructed based on this formulation of hard problems. For the more sophisticated

Cryptomania world (i.e., public-key crypto systems exist) [33], the possibilities have not yet been explored in the MQ literature. This line of research will be our main focus in the rest of this paper.

*Our Main Results.* In this work, we study a new MQ assumption that can be used to construct more sophisticated primitives such as public-key encryptions in the Cryptomania world [33]. In particular, we research in the following two directions:

- On the one hand, we establish a precise *asymptotic* formulation of a family of hard problems, and provide empirical evidence to confirm the hardness. Since there are many practical solvers studied and implemented during the studies of algebraic attacks, we use these to examine the hardness of the problems.
- On the other hand, we construct public-key encryption schemes, and prove their security under the hardness assumption of the said family. Also, we provide a new *perspective* to look at MQ systems that plays a key role to our design and proof of security.

As a consequence, we construct the first public-key encryption scheme that is *provably secure* under the MQ assumption. Moreover, our public-key encryption scheme is efficient in the sense that it only needs a ciphertext length  $L + \text{poly}(k)$  to encrypt a message  $M \in \{0, 1\}^L$  for any un-prespecified polynomial  $L$ .<sup>4</sup> This is essentially *optimal* since an additive overhead is the best we can hope for.

The MQ assumption has some interesting properties for its potential. In the following, we will discuss that the MQ problems share some structures with the learning with error (LWE) problems [26, 44, 45]. Thus the MQ assumption may also enjoys the versatility as LWE. On the other hand, there are many experiences or fast implementations under a variety of hardwares [4, 9, 11] in the MQ literature, and thus this can be a good basis for practical applications.

Note: we are unaware of any reductions between our MQ assumption or indeed any MQ-type assumptions and lattice-related ones such as LWE. Furthermore, lattice problems have been studied for a much shorter period of time than equation-solving, and new methods such as BKZ 2.0 [13] are still proposed. So it is difficult to compare PKC constructions based on lattice-related hard problems and MQ problems. The comparison is a very interesting research direction but outside the scope of this paper. This paper will simply focus on the MQ assumption and its consequent constructions. In section 3.1, we give a brief remark on the difference between MQ and LWE assumptions. More detailed discussions will appear in the full version of this paper.

*A Closer Look at Our Assumption.* In the following, we take a closer look at our assumption and techniques, and still maintain a high-level perspective for intuitions. First, we give some notation for convenience of exposition. Let  $\mathbb{F}_q$  be a field which we use in the following discussion, and let  $S$  describe a multivariate

---

<sup>4</sup>  $k$  is the security parameter.

quadratic system with  $n$  variables and  $m$  polynomials. For example, the following system is one with 3 variables and 2 polynomials, and for a concrete explanation we set  $q = 13$ .

$$S \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \stackrel{\text{def}}{=} \begin{cases} x_1 x_3 + x_2^2 + 3x_1 + 2 \\ x_1 x_2 + 2x_1 + 2x_2 + 7 \end{cases} \quad (1)$$

In addition to viewing  $S$  as a set of polynomials, we can view the above system  $S$  as a function mapping from  $\mathbb{F}_q^3$  to  $\mathbb{F}_q^2$ . For example,  $S([1, 2, 3]^T) = [12, 2]^T$ , where  $T$  denotes transposes of vectors. In the rest of the paper, we use  $S[\cdot]$  to denote a system of polynomials, and  $S(\cdot)$  to denote the corresponding function. Now we are ready to describe the hard problem of our assumption with more details (still informally). Note that here the system  $S$  includes quadratic terms, linear terms and constant terms. Throughout the paper, we will use  $S$  to denote a system with all quadratic, linear and constant terms.

**Definition 2 (The Hard Task (Informal)).** *Let  $q$  be a large enough prime, and  $H$  be some small subset of  $\mathbb{F}_q$ . Let  $S$  be a multivariate quadratic system with  $n$  variables and  $m = \Theta(n)$  polynomials sampled from a distribution where the coefficients of linear and constant terms are uniformly random, and the quadratic terms come from independent Gaussian distributions with means 0 and moderately large standard deviations.*

*Then a solver  $A$ , given  $(S, \mathbf{y} = S(\mathbf{x}))$  where  $\mathbf{x}$  is sampled uniformly from  $H^n$ , is asked to output some  $\mathbf{x}'$  such that  $S(\mathbf{x}') = \mathbf{y}$ .*

To make the seemingly intimidating parameters more reader-friendly, we give an intuitive-level discussion as follows. First, we observe that depending on the parameters, solving MQ equations can be easy or hard. As discussed in [4], when  $m$  is significantly larger or smaller than  $n$ , solving the problem is easy. The interesting hard instances fall on the cases when  $m$  is close to  $n$ , as stated in the above definition that  $m = \Theta(n)$ . Moreover, the problem is believed to be not only hard in the worst case, but hard on average over random instance of  $S$ , and random input  $\mathbf{x}$ . Under a series of empirical studies and theoretical studies [3, 16, 47, 48] for the best known solvers, the best known algorithms still remain exponential-time.

Previously, [35] observed (from experiments) that even if the instance  $S$  is drawn from a biased distribution (whose quadratic coefficients are not uniform but instead sparse), solving the problem is still hard. This result hints at an intuition that MQ problems are hard for most (non-trivial) distributions from which  $S$  is drawn. In this work, we further test this intuition by investigating the case that the instance  $S$  is drawn from a distribution whose quadratic coefficients come from Gaussian distributions with moderately large standard deviation, and the input  $\mathbf{x}$  is drawn from a smaller subset  $H^n$ . Our experiment results (in the full version of this paper) confirm our intuition that the problem does not become significantly easier. In the following paragraphs, we explain how and why this type of assumption and hardness help our design.

We remark that here we only give a structural description of the problem, and leave the precise quantitative statement in Section 3. Before going to the detailed calculation of numbers, we first focus on the structural properties of the hard problem and maintain a high-level perspective.

*Overview of Our Construction.* Inspired by the recent constructions of public-key crypto systems by learning with error (LWE) problems [45], we observe that the problem in Definition 2 also shares the same structure with LWE. We can take advantage of this similarity for our construction of public-key encryption schemes. This is a new perspective of how we can view MQ equations.

First, let us take a look at the LWE problem, which can be stated as the following: let  $A \in \mathbb{F}_q^{m \times n}$  be a matrix, and  $\mathbf{b}$  be a vector  $\mathbf{b} = A \cdot \mathbf{s} + \mathbf{e}$ , where  $\mathbf{s} \in \mathbb{F}_q^n$  is some secret, and  $\mathbf{e}$  comes from some error distribution. The task of the LWE problem is to find out  $\mathbf{s}$  given a random  $A$ , and an induced  $\mathbf{b}$ .

We highlight the similarity by way of the following observation: recall that the task of the problem in Definition 2 is to invert  $\mathbf{y} = S(\mathbf{x})$  given  $S, \mathbf{y}$ . We can rewrite  $\mathbf{y}$  into  $S(\mathbf{x}) = L \cdot \mathbf{x} + \mathbf{d} + R(\mathbf{x})$ , where  $L$  is the matrix of the terms of linear coefficients,  $\mathbf{d}$  is the coefficient vector of constant terms, and  $R(\mathbf{x})$  are the mapping by the quadratic terms. Take Equation 1 for example, we can rewrite the expression of  $S(\mathbf{x})$  as:

$$S(\mathbf{x}) = \begin{cases} x_1x_3 + x_2^2 \\ x_1x_2 \end{cases} + \begin{matrix} 3x_1 \\ 2x_1 + 2x_2 \end{matrix} + \frac{2}{7} = R(\mathbf{x}) + \begin{pmatrix} 3 & 0 & 0 \\ 2 & 2 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} + \begin{pmatrix} 2 \\ 7 \end{pmatrix}$$

In this expression,  $S(\mathbf{x})$  is a combination of an affine transformation ( $L \cdot \mathbf{x} + \mathbf{d}$ ) plus some quadratic mapping  $R(\mathbf{x})$ . We remark that without loss of generality, we can assume  $\mathbf{d} = 0$ , since solving a multivariate system with all 0s for the constant coefficients is equivalent to solving that with random constant coefficients.<sup>5</sup> Then if we view the quadratic terms as *noise* (analogous to the vector  $\mathbf{e}$ ), the shared structure becomes apparent. Thus, the ideas that come from using LWE may be translated into candidates of constructions by MQ problems.

However, to bridge the two problems, we need to deal with some subtleties. In the LWE problems, the noise (error vector  $\mathbf{e}$ ) comes from a Gaussian distribution that has “moderately” large standard deviation. Intuitively, if the standard deviation is too small, then the problems become easier; on the other hand, if it is too large, then the ciphertexts (constructed from LWE) become undecryptable. Thus, in this series of works [26, 44, 45], certain ranges of parameters for stds have been identified such that both the hardness of the problems and the correctness of the decryption hold simultaneously.

When MQ problems are viewed in this way, we also need to argue that the noise  $R(\mathbf{x})$  is also “moderate.” To achieve this, we use the structure of the assumption that the coefficients of each quadratic term come from Gaussian

<sup>5</sup> There is a simple reduction showing that solving  $(S, \mathbf{y} = S(\mathbf{x}))$  for  $S$  contains random constant coefficients is equivalent to solving  $(S', \mathbf{y}' = S'(\mathbf{x}))$ , where  $S'$  has the same distribution as  $S$ , except for the 0 constant coefficients.

distributions with moderately large standard deviations, and the input  $\mathbf{x}$  comes from a small subset  $H^n \subseteq \mathbb{F}_q^n$ . That property allows us to bound the size of the noise  $R(\mathbf{x})$ . On the other hand, we need to examine the hardness of the problem for these parameters. To do so, we conduct experiments under what to our knowledge the best quadratic equation solver. Our experiment results confirm our intuitions that MQ problems do not become significantly easier under any (non-trivial) particular distribution of the inputs  $S$  and  $\mathbf{x}$ . This particularly gives us evidence of the hardness of the problem in Definition 2, which we can use to construct public-key encryptions.

*Our First Construction of Encryption for Bits.* In our first attempt, we construct a public-key encryption scheme for bits. This construction is similar in spirit to those LWE-based constructions [26, 44, 45]. Because of the similarity, here we omit discussions of intuitions and refer the curious readers to [26, 44, 45]. Here we give an informal outline of the construction:

- In key generation, the algorithm samples an MQ system  $S$  with  $n$  variables and  $m = c \cdot n$  polynomials, and  $\mathbf{x} \in H^n$ . Then it sets the public key to be  $(S, \mathbf{y} = S(\mathbf{x}))$ , and the secret key to be  $\mathbf{x}$ .
- To encrypt a bit  $b$ , the encryption algorithm samples  $\mathbf{r} \in H^m$ , and computes  $(c_1, c_2) = (\mathbf{r}^T \cdot L, \mathbf{r}^T \cdot (\mathbf{y} - \mathbf{d}) + b \cdot [q/2])$ . Recall that  $L$  is an  $m \times n$  matrix, and  $m > n$ . Thus, given  $\mathbf{r}^T \cdot L$ ,  $\mathbf{r}$  is still hidden information theoretically.
- To decrypt, the algorithm computes  $t = c_2 - c_1^T \cdot \mathbf{x}$ . It outputs 1 if and only if  $|t - q/2| \leq q/4$ .

*Security Proof.* The key to the security proof of the bit-encryption scheme is based on a proof that relates the hardness of the assumption to some pseudorandom distribution. Namely, suppose the problem in Definition 2 is hard, then  $(S, S(\mathbf{x}))$  is indistinguishable from  $(S, U_m)$  where  $U_m$  is uniform over  $\mathbb{F}_q^m$ . Moreover, we prove a more general theorem that suppose there exist a distribution over the quadratic terms of  $S$ , and a subset  $H \subseteq \mathbb{F}_q$  such that the problem is hard, then  $(S, S(\mathbf{x}))$  is indistinguishable from  $(S, U_m)$ . The crux of our proof is a new application of the new version of Goldreich-Levin Theorem by Dodis *et. al* [19].

We remark that this general theorem also, as a consequence, implies Theorem 2 plus 3 in [4], and Proposition 5 plus 6 in [35] as its special cases.<sup>6</sup>

*Improving Efficiency Using KEM.* Feasibility results for bit-encryptions are nice but not quite satisfactory. One general technique to improve efficiency is to use *Key Encapsulation Mechanism* (KEM). We know that to use KEM, it is sufficient to have an efficient symmetric encryption scheme or a pseudorandom generator (PRG). (Note that a pseudorandom generator implies an efficient symmetric encryption scheme.) Although there are many implementations of PRGs and

---

<sup>6</sup> We present our theorem and assumption in asymptotic forms, and both [4, 35] presented their theorems in concrete parameters.

thus symmetric encryptions as well [7, 8, 22, 30–32, 34], the constructions are either not practically efficient, or require some additional assumption(s).

Here we further observe that the MQ assumption (Definition 2) already gives us an efficient construction of a certain form of PRG<sup>7</sup> that is sufficient to implement the KEM technique. As a consequence, in the resulting scheme, we are able to achieve a public-key encryption scheme that only needs a ciphertext length  $L + \text{poly}(k)$  to encrypt a message  $M \in \{0, 1\}^L$  for any un-prespecified polynomial  $L$ , where  $k$  is the security parameter. This is essentially *asymptotically optimal* since we know the ciphertext length must be at least as large as the message (otherwise there will be decryption errors), and an additive overhead in the security parameter is the the best we can hope for.

## 2 Preliminary

### 2.1 Notation

All vectors are assumed to be column vectors. Unless stated otherwise, all scalar and vector operations are performed modulo  $q$ . We use arrow notation to represent a vector, and subscripts to represent the corresponding element, i.e.  $\mathbf{r} \in \mathbb{F}_q^n$  means  $\mathbf{r}$  is a vector of  $n$  elements in  $\mathbb{F}_q$  and  $r_i$  means the  $i$ -th element of the vector. We denote the transpose of a vector  $\mathbf{r}$  as  $\mathbf{r}^T$ .

For simplicity we will assume that  $q$  is an odd prime. We represent elements in  $\mathbb{F}_q$  by integers within the range  $[-(q-1)/2, (q-1)/2]$ . We denote the inner product of  $\mathbf{a}$  and  $\mathbf{b}$  as  $\langle \mathbf{a}, \mathbf{b} \rangle$ , or  $\mathbf{a}^T \cdot \mathbf{b}$ .

Let  $m, n, q$  be numbers. Though out the paper, we will use  $S = (R, L, \mathbf{d})$  to denote a MQ system with  $n$  variables and  $m$  equations, where  $R \in \mathbb{F}_q^{m \times n \times n}$  denotes the quadratic coefficients, and  $L \in \mathbb{F}_q^{m \times n}$  denotes the linear coefficients and  $\mathbf{d} \in \mathbb{F}_q^m$  denotes the constant coefficients. In particular  $R_{i,j,k}$  denotes the coefficient of  $x_j x_k$  in the  $i$ -th equation, and  $L_{i,j}$  denotes the coefficient of  $x_j$  in the  $i$ -th equation, and  $\mathbf{d}_i$  denotes the constant coefficient in the  $i$ -th equation.

**Definition 3 (Multivariate Quadratic Problems).** *Let  $n, m, q \in \mathbb{N}$  be parameters such that  $q$  is a prime, let  $\chi$  be a distribution between  $\mathbb{F}_q^{m \times n \times n}$ , and let  $H \subseteq \mathbb{F}_q$ . The goal for a solver  $A$  to the (average-case) multivariate quadratic problem  $MQ(n, m, q, \chi, H)$  is that  $A$  on a random instance  $(S, S(\mathbf{x}))$  tries to output some  $\mathbf{x}' \in \mathbb{F}_q^n$  such that  $S(\mathbf{x}') = S(\mathbf{x})$ , where  $S = (R, L, \mathbf{d})$  with  $R \leftarrow \chi$ ,  $L \leftarrow \mathbb{F}_q^{m \times n}$ ,  $\mathbf{d} \leftarrow \mathbb{F}_q^m$ , and  $\mathbf{x} \leftarrow H^n$ . If  $A$  does so, we say it successfully solves the instance.*

**Definition 4 (Hardness of a MQ Family).**<sup>8</sup> *Let  $k$  be the security parameter,  $n, m, q : \mathbb{N} \rightarrow \mathbb{N}$  be efficiently computable and polynomially bounded such that  $q$*

<sup>7</sup> The PRG constructed by the MQ assumption is somewhat non-standard but is sufficient for KEM. See Section 5 for further discussions.

<sup>8</sup> To lend more credence to our contention that our family is hard, we attach logarithmic plots in the appendix in which we compare the behavior under MAGMA-2.17 of systems with  $m/n = 2$  in cases (A) random systems in GF(3) and GF(5); (B)

is an odd prime. Let  $\chi$  be a distribution over  $\mathbb{F}_q^{m \times n \times n}$  and  $H \subseteq \mathbb{F}_q$ . We say that the family  $MQ(n, m, q, \chi, H)$  is hard to solve if for every PPT solver  $A$ , there exists some negligible function  $\text{ngl}(\cdot)$  such that the following holds for all sufficiently large  $k$ :

$$\Pr_{\substack{S \leftarrow MQ(n, m, q, \chi, H) \\ \mathbf{x} \leftarrow H^n}} [\mathbf{x}' \leftarrow A(S, S(\mathbf{x})) : S(\mathbf{x}') = S(\mathbf{x})] < \text{ngl}(k).$$

### 3 Public-Key Encryption Schemes For Bits

In this section, we show a construction of public-key encryption schemes (for bits) under the hardness of some specialized MQ problem. We present our results in the following order: (1) the hardness assumption, (2) the construction of the scheme, and (3) the analysis.

#### 3.1 The Assumption

**Definition 5 (MQ Hardness Assumption).** *Let  $k$  be the security parameter. For every constant  $c > 1 \in \mathbb{N}$ , every efficiently computable and polynomially bounded  $n, m, q : \mathbb{N} \rightarrow \mathbb{N}$ ,  $\alpha : \mathbb{N} \rightarrow [-q/2, q/2]$  and every  $0 < \beta \leq [q/2]$  such that (1)  $m = cn$ , (2)  $q$  is prime, (3)  $\alpha = O(1)$ , let  $\Phi_\alpha$  be the distribution of  $m \times n \times n$  identical independent discrete Gaussian distribution  $D_\alpha$ 's with mean 0, standard deviation  $\alpha$ , namely, each  $D_\alpha$  samples  $z \leftarrow N(0, \alpha^2)$  (normal distribution with mean 0, and standard deviation  $\alpha$ ), and then outputs  $\lfloor z \rfloor \pmod{q}$ , and let  $H_\beta = \{-\beta, -\beta + 1, \dots, \beta - 1, \beta\}$ .*

*Then the problem  $MQ(n, m, q, \Phi_\alpha, H_\beta)$  is hard to solve.*

As discussed in the introduction, we need to choose the parameters  $\alpha$  such that  $|R(\mathbf{x})|$  is “moderate” for two aspects. First,  $\alpha$  cannot be too large, otherwise there will be decryption errors. On the other hand, if  $\alpha$  is too small, then with high probability, most coefficients are 0, so the system becomes sparse. There are known attacks for sparse systems where there are only  $o(1)$  non-zero coefficients, so in our assumption, the  $\alpha$  cannot fall into this region. In our setting,  $\alpha \geq O(1)$  implies that each quadratic terms has at least a constant probability not being zero, and thus there will be  $O(n^2)$  quadratic terms in expectation. In the full version of this paper, we will discuss more details about the parameters and how they influence the hardness of the problem.

*Remark 1.* As we discussed in the introduction, the MQ assumption has a similar structure to the LWE assumption. Here we do a brief comparison of the two assumptions for different range of parameters.

---

systems in larger fields but with variables restricted to  $\{-1, 0, 1\}$  and the equations  $x_i^3 = x_i$  included for every  $i$ ; (C) systems in larger fields but with variables restricted to  $\{-2, -1, 0, 1, 2\}$  and the equations  $x_i(x_i^2 - 1)(x_i^2 - 4) = 0$  included for every  $i$ . The trend looks quite exponential. For more discussion see the full version of this paper.



For  $q$  being superpolynomial, we can show that an MQ instance  $(S, S(\mathbf{x}))$  can be transformed to  $(L, b)$  that is statistically close to an LWE instance. The transformation just sets  $L$  as the linear part of  $S$ , and sets  $b = S(\mathbf{x}) + \mathbf{e}'$ , where each coordinate of  $\mathbf{e}'$  comes from some i.i.d. Gaussian with a small std. For  $q = \text{superpoly}(k)$ , one can show that  $b$  is statistically close to  $L \cdot \mathbf{x} + \mathbf{e}''$  where each coordinate of  $\mathbf{e}''$  comes from i.i.d. Gaussian with a slightly bigger std. Thus,  $(L, b)$  is statistically close to an LWE instance, and consequently, there is a simple reduction from MQ to LWE.

In this paper, we need  $q = \text{poly}(k)$  for our construction. For this range of parameters, the above argument does not work. In fact, an MQ instance and an LWE instance can be statistically far. Thus, a straightforward reduction from MQ to LWE does not work. We are not aware of any other reduction from any one to the other, and leave this issue as an interesting open question.

Under the above assumption, we are able to obtain the following lemma, which is a key to the security proof of our construction of public-key encryption scheme. In the following section, we are going to prove a more general result as Theorem 2, which directly implies this lemma. Thus, we only put the statement of the lemma.

**Lemma 1.** *Let  $k$  be the security parameter, and assuming  $MQ(n, m, q, \Phi_\alpha, H_\beta)$  be the hard problem as stated in Definition 5. Then  $(S, S(\mathbf{x}))$  is computationally indistinguishable from  $(S, U_m)$ , where  $S \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$ ,  $\mathbf{x} \leftarrow H_\beta^n$ ,  $U_m$  is the uniform distribution over  $\mathbb{F}_q^m$ .*

Here we remark that the MQ hardness assumption in Definition 5 can be generalized in the following sense.

*Remark 2.* Actually all we need for our construction is to bound the quantity  $R(\mathbf{x})$ . Thus any distribution of  $S$ , and  $\mathbf{x}$  that has the following properties (1) the problem of equation solving is hard, and (2) we are able to bound  $R(\mathbf{x})$ , are sufficient for us to construct public-key encryptions. Here for concreteness, we present study  $\Phi_\alpha$  and  $H_\beta^n$  as a candidate for the hard problem.

### 3.2 Construction of A Public-Key Encryption Scheme for Bits

In this section we present our construction of a public-key bit-encryption scheme.

*Construction of the Scheme  $\mathcal{E} = (\text{KeyGen}(\cdot), \text{Enc}(\cdot), \text{Dec}(\cdot))$ :*

- $\text{KeyGen}(1^k)$ : choose public parameters  $n, m, q, \alpha, \beta$ , and  $\lambda \in \mathbb{N}$  satisfying the following constraints:
  1.  $k \cdot \alpha \cdot n^{(2+\lambda)} \cdot m \cdot \beta^2 \leq q/4$ .
  2.  $m \cdot \log(2n^\lambda + 1) \geq (n + 1) \cdot \log q + 2k$ .
  3.  $n, m, q, \alpha, \beta$  satisfy the condition in the MQ assumption such that  $MQ(n, m, q, \Phi_\alpha, H_\beta)$  is hard to solve.

Then it samples a random instance  $(S, S(\mathbf{x})) \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$ , and deontes  $\mathbf{y} = S(\mathbf{x})$ . Then it sets  $\text{pk} = (S, \mathbf{y}) = ((R, L, d), \mathbf{y})$ ,  $\text{sk} = \mathbf{x}$ .

- Enc( $b$ ) for  $b \in \{0, 1\}$ : sample  $\mathbf{r} \in H_{n^\lambda}^m$ , and outputs  $(c_1, c_2) = (\mathbf{r}^T \cdot L, \mathbf{r}^T \cdot (\mathbf{y} - \mathbf{d}) + b \cdot [q/2])$ .
- Dec( $c_1, c_2$ ): compute  $t = c_2 - c_1^T \cdot \mathbf{x}$ . If  $|t - q/2| \leq q/4$  then output 1, otherwise 0.

The intuition of the construction and analysis of security are similar to the case of the work [45]. Thus we only state the theorem and leave the discussions in the full version of this paper.

**Theorem 1.** *Assume the MQ assumption holds for the above parameters. Then the scheme  $\mathcal{E}$  is a semantically secure encryption scheme.*

## 4 Hardness of MQ Problems Implies Pseudorandom Distributions

Recall that in the previous section, we claimed that the hardness of some family of MQ problems implies a pseudorandom distribution (Lemma 1). In this section, we are going to show that the hardness of more general families of MQ problems also implies a pseudorandom distribution. In particular, we obtain the following theorem.

**Theorem 2.** *Let  $k$  be the security parameter,  $n, m, q$  be efficiently computable and polynomially bounded such that  $q$  is an odd prime,  $\chi$  is a distribution over  $\mathbb{F}_q^{m \times n \times n}$ , and  $H \subseteq \mathbb{F}_q$ .*

*Suppose for these parameters the problem  $MQ(n, m, q, \chi, H)$  is hard to solve, then the following two distributions are computationally indistinguishable.  $D_1 = (S, S(\mathbf{x}))$ ,  $D_2 = (S, U_m)$ , where  $S \leftarrow MQ(n, m, q, \chi, H)$ ,  $\mathbf{x} \leftarrow H^n$ , and  $U_m$  is a uniform distribution over  $\mathbb{F}_q^m$ .*

If we set  $H$  to be  $H_\beta$ , and  $\chi$  to be  $\Phi_\alpha$  as the setting in Definition 5, then this version of the theorem directly becomes Lemma 1.

We prove the theorem by contradiction. For intuition, first we state our high level ideas and then delve into details. Suppose there exists a distinguisher  $A$  that distinguishes  $D_1$  and  $D_2$ , from here we want to construct an inverter  $B$  that solves the MQ problem  $(S, S(\mathbf{x}))$ , which leads to a contradiction. We achieve this goal using the following strategy:

- First we show that from  $A$ , we can construct another algorithm  $A'$  that distinguishes  $D'_1 = (S, S(\mathbf{x}), \mathbf{r}, \langle \mathbf{r}, \mathbf{x} \rangle)$  and  $D'_2 = (S, S(\mathbf{x}), \mathbf{r}, U)$  where  $\mathbf{r} \in \mathbb{F}_q^n$  is a random vector, and  $U$  is uniform over  $\mathbb{F}_q$ . For any  $\mathbf{r} \in \mathbb{F}_q^n$ , we can view  $\langle \mathbf{r}, \mathbf{x} \rangle$  as the  $\mathbf{r}$ 's location of the (Hadamard) encoding of  $\mathbf{x}$ . The ability to distinguish  $D'_1$  and  $D'_2$  gives us a somewhat corrupted codeword of  $\mathbf{x}$ , i.e., the codeword is correct in at least a noticeable fraction of places over all  $\mathbf{r}$ 's.
- Then from  $A'$ , we construct an inverter  $B$  that applies the list-decoding algorithm by the Goldreich-Levin Theorem to recover  $\mathbf{x}$ . We remind the reader that the Goldreich-Levin Theorem is essentially a decoding algorithm for the Hadamard code, which says (informally) that if given  $f(\mathbf{x})$ , for random

$\mathbf{r}$ 's one can distinguish  $\langle \mathbf{r}, \mathbf{x} \rangle$  from a uniform element with noticeable probability, then one can invert  $f$  with noticeable probability (for any function  $f$ ).

However, when applying the Goldreich-Levin Theorem here, we encountered some subtleties. First the classical theorem [28] deals with the boolean field only (i.e.  $q = 2$ ); thus it is not applicable in general cases. A generalized version of [29] handles the case for large  $q$ 's, but it works only for the case where the input  $\mathbf{x} \in \mathbb{F}_q^n$ . It remains unclear for the case where  $\mathbf{x}$  comes from a subset  $H_\beta^n \subseteq \mathbb{F}_q^n$ . Recently, Dodis *et al.* [19] proved a new version of the theorem that is essentially what we need in our setting. With it, we are able to implement the list-decoding algorithm in the second bullet above, and this completes the proof. The formal proof will appear in the full version of this paper.

## 5 Key Encapsulation Mechanism

In the previous section, we constructed a public-key encryption for bits. However, this approach is not satisfactory when we want to encrypt a long message  $M \in \{0, 1\}^L$  for some large  $L$ . As discussed in the introduction, we can use a key encapsulation mechanism (KEM) to achieve better efficiency.

First, we recall how we can achieve this by the KEM technique: let  $\text{Enc}$  be any public-key encryption scheme for bits, and let  $G : \{0, 1\}^k \rightarrow \{0, 1\}^{k+t}$  be a pseudorandom generator. To encrypt a long message  $M \in \{0, 1\}^L$ , we first sample a seed  $s \in \{0, 1\}^k$  for the PRG, and then stretch the generator  $G^g$  to get a pseudorandom string  $G'(s) \in \{0, 1\}^L$ . Then we encrypt the seed by the public-key scheme and use the pseudorandom string as a one-time pad to XOR  $M$ . The resulting ciphertext becomes  $(\text{Enc}_{\text{pk}}(s), G'(s) \oplus M)$ .

In this paper, we observe that the MQ assumption implies a certain form of PRG. Thus, we can implement KEM under the same assumption as the one from which we construct the public-key encryption scheme. However, this type of PRG is somewhat non-standard, so we avoid using this term formally. We will discuss this issue in the full version of this paper.

In the next section, we are going to show how we can obtain the desired long pseudorandom string  $G'(s)$ , and then present the entire scheme in Section 5.2. Finally we sketch the proof of security, which follows from the folklore.

*Remark 3.* We remark that KEM is a generic way to construct efficient public-key encryption schemes. As discussed in the introduction and the above, we know that a PRG plus any bit-encryption public encryption scheme is sufficient to achieve the task. In this paper, we observe that the MQ assumption implies an efficient constructions of PRGs and a public-key bit-encryption scheme, so we can obtain an efficient public-key encryption under one single assumption.

---

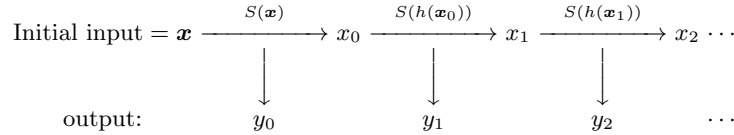
<sup>9</sup> We refer the readers to [27] for the details of how to stretch a PRG.

## 5.1 Longer Pseudorandom Strings

Recall that Lemma 1 states that  $(S, S(\mathbf{x})) \approx_c (S, U_m)$ . This means we can get a pseudorandom string  $S(\mathbf{x}) \in \mathbb{F}_q^m$  by only sampling a shorter seed  $\mathbf{x} \in H_\beta^n$ . Note:  $m > n$ , and  $H \subseteq \mathbb{F}_q$ . To get a longer pseudorandom string, we can use the following iterative method (analogous to how we can stretch a PRG.)

**Definition 6.** Let  $S \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$ . For  $\mathbf{x} \in H_\beta^n$ , and let  $(x_0, y_0) = S(\mathbf{x})$  where  $\mathbf{x}_0 \in \mathbb{F}_q^n$ ,  $\mathbf{y}_0 \in \mathbb{F}_q^{m-n}$  be the prefix  $n$  elements and the suffix  $m-n$  elements of  $S(\mathbf{x})$  respectively.

Let  $h : \mathbb{F}_q^n \rightarrow H_\beta^n$  be a hash function, and for  $i \in \mathbb{N}$ , we recursively define  $(\mathbf{x}_i, \mathbf{y}_i) = S(h(\mathbf{x}_{i-1}))$  where  $\mathbf{x}_i \in \mathbb{F}_q^n$ ,  $\mathbf{y}_i \in \mathbb{F}_q^{m-n}$  (representing the prefix and suffix of  $S(h(\mathbf{x}_{i-1}))$  respectively). Then we define  $S_h^i(\mathbf{x}) = (\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_i)$ .



Then we are going to argue that for any  $i \leq \text{poly}(k)$ , we have  $(S, S_h^i(\mathbf{x})) \approx_c (S, U_{(m-n) \cdot (i+1)})$ , given  $(S, S(\mathbf{x})) \approx_c (S, U_m)$ . This means, we can get an arbitrarily long (polynomially bounded) pseudorandom string  $S_h^i(\mathbf{x})$  from an initial random seed  $\mathbf{x}$ .

The proof of security follows from a hybrid argument, and it is similar to that of QUAD in the work [4]. We remark that here we need the hash function for some technical reason. The only property we require is that  $h(U_n)$  outputs a (statistically close) uniformly random element in  $H_\beta^n$ . The hash function  $h$  does not need to be collision resistant nor one-way. We can view  $h$  as a reinterpretation from elements in  $\mathbb{F}_q^n$  to elements in  $H_\beta^n$ , and thus there are many simple constructions.

Then we are able to achieve the following theorem.

**Theorem 3.** Let  $k$  be the security parameter. Assuming the MQ problem  $MQ(n, m, q, \Phi_\alpha, H_\beta)$  is hard, and let  $h : \mathbb{F}_q^n \rightarrow H_\beta^n$  be a (randomized) hash function such that  $h(\mathbf{z})$  maps a uniformly random  $\mathbf{z} \in \mathbb{F}_q^n$  to a uniformly random  $\mathbf{y} \in H_\beta^n$ .

Then for any  $i = \text{poly}(k)$ ,  $(S, S_h^i(\mathbf{x}))$  is computationally indistinguishable to  $(S, U_{(m-n) \cdot (i+1)})$ , where  $S \leftarrow MQ(n, m, q, \Phi_\alpha, H_\beta)$ ,  $\mathbf{x} \leftarrow H_\beta^n$ , and  $U_{(m-n) \cdot (i+1)}$  is uniform over  $\mathbb{F}_q^{(m-n) \cdot (i+1)}$ .

The proof will appear in the full version of this paper.

## 5.2 Construction of the KEM Scheme

In previous sections, we have constructed the bit encryption scheme  $\mathcal{E} = (\text{KeyGen}(\cdot), \text{Enc}(\cdot), \text{Dec}(\cdot))$  described in section 3.2, and the pseudorandom generator above. Here we describe a KEM scheme

$\mathcal{E}_{\text{KEM}} = (\text{KeyGen}_{\text{KEM}}(\cdot), \text{Enc}_{\text{KEM}}(\cdot), \text{Dec}_{\text{KEM}}(\cdot))$  that can encrypt messages with un-prespecified lengths (polynomially bounded).

- $\text{KeyGen}_{\text{KEM}}(1^k)$ : run  $\text{KeyGen}(1^k)$ . In particular, the algorithm chooses public parameters  $n, m, q, \Phi_\alpha, H_\beta$  in the range as stated in the MQ assumption, and also a hash function  $h : \mathbb{F}_q^n \rightarrow H_\beta^n$  with the property  $h(U_n)$  being uniform over  $H_\beta^n$  as discussed in the above section. Then it samples a random instance  $(S, S(\mathbf{x})) \leftarrow \text{MQ}(n, m, q, \Phi_\alpha, H_\beta)$ , and deontes  $\mathbf{y} = S(\mathbf{x})$ . Then it sets  $\text{pk} = (S, \mathbf{y})$ ,  $\text{sk} = \mathbf{x}$ .
- For any  $L = \text{poly}(k)$ , and any message  $M \in \mathbb{F}_q^L$ ,  $\text{Enc}_{\text{KEM}}(M)$  does the following: the algorithm samples  $\mathbf{s} \in H_\beta^n$ , and computes  $c_i = \text{Enc}(\text{pk}, \mathbf{s}_i)$  for  $i \in [n]$ . Then let  $t = \lceil L/(m-n) \rceil$ , and compute  $c^* = M \oplus S_h^t(\mathbf{s})$ .<sup>10</sup> The resulting ciphertext will be  $c = (c_1, c_2, \dots, c_n, c^*)$ .
- $\text{Dec}_{\text{KEM}}(c)$ : the algorithm computes  $\mathbf{s}$  by running  $\text{Dec}(\text{sk}, c_i)$  for  $i \in [n]$ . Then it outputs  $M = c^* \oplus S_h^t(\mathbf{s})$ .

Then we are able to obtain the following theorem.

**Theorem 4.** *The scheme above  $\mathcal{E}_{\text{KEM}}$  is a semantically secure encryption scheme.*

### 5.3 Concrete Parameters

Our goal here is to instantiate Theorem 4 with concrete parameters. Here, we exhibit two sets of parameters (for proven security levels  $2^{80}$  and  $2^{128}$ ) based on a conservative estimate of the hardness of MQ systems (i.e., assuming the general applicability of sparse matrix solvers in XL [47]), and no particular effort in optimization.

Our security level aims for time  $2^{80}$  (and  $2^{128}$ ), and  $\varepsilon = 2^{-10}$  for plaintext length  $L = 2^{20}$  (1 Mb), i.e., no adversary within running time  $2^{80}$  (and  $2^{128}$ ) can distinguish two ciphertexts with advantage better than  $2^{-10}$ . Since our construction uses the KEM mechanism, we need parameters for (1)  $(S, S_h^t(\mathbf{x}))$  to be a PRG some length  $L$ , and (2)  $\mathcal{E}$  to be a semantically secure bit-encryption scheme. It follows from a standard argument that the KEM security achieves this level (with a slight loss) once both the underlying PRG and the encryption scheme achieve this level of security. In particular, we instantiate the scheme with the following parameters:

Case	$k$	$n$	$m$	$\alpha$	$\beta$	$q$
1	12	200	400	10	2	$18031317546972632788519 \approx 2^{74}$
2	12	256	512	10	2	$52324402795762678724873 \approx 2^{76}$

And we approximate the hardness in the following table:

Case	Hardness of MQ	Security of Enc	Security of PRG	Security of KEM
1	$2^{156}, 2^{-100}$	$2^{87}, 2^{-11}$	$2^{85}, 2^{-11}$	$2^{85}, 2^{-10}$
2	$2^{205}, 2^{-104}$	$2^{130}, 2^{-11}$	$2^{134}, 2^{-11}$	$2^{130}, 2^{-10}$

We remark the tuple  $(T, \varepsilon)$  in each cell means for any adversary running in time  $T$  has advantage (or success probability) less than  $\varepsilon$ .

In the full version of this paper, we will explain our methodology of the experimental studies, and provide the data. Due to space limit, we omit most details for proofs and experiments.

<sup>10</sup> Here  $\oplus$  means we add two vectors component-wise. That is, let  $\mathbf{a}, \mathbf{b} \in \mathbb{F}_q^L$ , then we say  $\mathbf{a} \oplus \mathbf{b} = [\mathbf{a}_1 + \mathbf{b}_1, \mathbf{a}_2 + \mathbf{b}_2, \dots, \mathbf{a}_L + \mathbf{b}_L]^T$ .

*Acknowledgement.* The authors would like to thank Kai-Min Chung for his valuable comments. Feng-Hao Liu is supported by National Science Foundation for partial support under grant CNS-0347661 and CNS-0831293. Yun-Ju Huang and Bo-Yin Yang thank the Taiwan National Science Council and the Academia Sinica for partial support under grant NSC-100-2218-E-001-002 and the AS Career Award.

## References

1. M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *STOC*, pages 284–293, 1997.
2. B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *STOC*, 2010.
3. G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita. Comparison between xl and gröbner basis algorithms. In *ASIACRYPT*, 2004.
4. C. Berbain, H. Gilbert, and J. Patarin. Quad: A practical stream cipher with provable security. In *EUROCRYPT*, 2006.
5. L. Bettale, J.-C. Faugère, and L. Perret. Cryptanalysis of multivariate and odd-characteristic hfe variants. In *Public Key Cryptography*, 2011.
6. O. Billet, J. Patarin, and Y. Seurin. Analysis of intermediate field systems. In *SCC*, 2008.
7. L. Blum, M. Blum, and M. Shub. A simple unpredictable pseudo-random number generator. *SIAM J. Comput.*, 15, 1986.
8. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 1984.
9. A. Bogdanov, T. Eisenbarth, A. Rupp, and C. Wolf. Time-area optimized public-key engines: Mq cryptosystems as replacement for elliptic curves? Lecture Notes in Computer Science, 2008.
10. D. Boneh and M. K. Franklin. Identity-based encryption from the weil pairing. In *CRYPTO*, 2001.
11. A. I.-T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee, and B.-Y. Yang. Sse implementation of multivariate pkcs on modern x86 cpus. In *CHES*, 2009.
12. C.-H. O. Chen, M.-S. Chen, J. Ding, F. Werner, and B.-Y. Yang. Odd-char multivariate hidden field equations. Cryptology ePrint Archive, Report 2008/543, 2008.
13. Y. Chen and P. Q. Nguyen. Bkz 2.0: Better lattice security estimates. In *ASIACRYPT*, pages 1–20, 2011.
14. N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *EUROCRYPT*, 2000.
15. N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *ASIACRYPT*, 2002.
16. C. Diem. The xl-algorithm and a conjecture from commutative algebra. In *ASIACRYPT*, 2004.
17. W. Diffie and M. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 1976.
18. J. Ding, V. Dubois, B.-Y. Yang, C.-H. O. Chen, and C.-M. Cheng. Could sflash be repaired? In *ICALP (2)*, 2008.
19. Y. Dodis, S. Goldwasser, Y. T. Kalai, C. Peikert, and V. Vaikuntanathan. Public-key encryption schemes with auxiliary inputs. In *TCC*, 2010.

20. V. Dubois, P.-A. Fouque, A. Shamir, and J. Stern. Practical cryptanalysis of sflash. In *CRYPTO*, 2007.
21. V. Dubois, P.-A. Fouque, and J. Stern. Cryptanalysis of sflash with slightly modified parameters. In *EUROCRYPT*, 2007.
22. R. R. Farashahi, B. Schoenmakers, and A. Sidorenko. Efficient pseudorandom generators based on the ddh assumption. In *Public Key Cryptography*, 2007.
23. J. C. Faugère. A new efficient algorithm for computing gröbner bases without reduction to zero (f5). In *ISSAC*, New York, NY, USA, 2002.
24. J.-C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (hfe) cryptosystems using gröbner bases. In *CRYPTO*, 2003.
25. A. S. Fraenkel and Y. Yesha. Complexity of solving algebraic equations. *Inf. Process. Lett.*, 1980.
26. C. Gentry, C. Peikert, and V. Vaikuntanathan. Trapdoors for hard lattices and new cryptographic constructions. In *STOC*, 2008.
27. O. Goldreich. *Foundations of Cryptography. Basic tools*. Cambridge University Press, 2001.
28. O. Goldreich and L. A. Levin. A hard-core predicate for all one-way functions. In *STOC*, pages 25–32, 1989.
29. O. Goldreich, R. Rubinfeld, and M. Sudan. Learning polynomials with queries: The highly noisy case. In *FOCS*, 1995.
30. I. Haitner, O. Reingold, and S. P. Vadhan. Efficiency improvements in constructing pseudorandom generators from one-way functions. In *STOC*, 2010.
31. J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.
32. T. Holenstein. Pseudorandom generators from one-way functions: A simple construction for any hardness. In *TCC*, pages 443–461, 2006.
33. R. Impagliazzo. A personal view of average-case complexity. In *Structure in Complexity Theory Conference*, pages 134–147, 1995.
34. B. S. Kaliski. A pseudo-random bit generator based on elliptic logarithms. In *CRYPTO*, 1986.
35. F.-H. Liu, C.-J. Lu, and B.-Y. Yang. Secure prngs from specialized polynomial maps over any  $\text{gf}(q)$ . In *PQCrypto*, pages 181–202, 2008.
36. F. S. Macaulay. On some formulae in elimination. *Proceedings of the London Mathematical Society*, 1902.
37. T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *EUROCRYPT*, 1988.
38. R. McEliece. A public-key cryptosystem based on algebraic coding theory. *DSN Progress Report*, 1978.
39. J. Patarin. Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt’88. In *CRYPTO*, 1995.
40. J. Patarin. Asymmetric cryptography with a hidden monomial. In *CRYPTO*, 1996.
41. J. Patarin and L. Goubin. Asymmetric cryptography with s-boxes. In *ICICS*, 1997.
42. J. Patarin, L. Goubin, and N. Courtois.  $C^*_{-+}$  and hm: Variations around two schemes of t. matsumoto and h. imai. In *ASIACRYPT*, 1998.
43. C. Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In *STOC*, 2009.
44. C. Peikert, V. Vaikuntanathan, and B. Waters. A framework for efficient and composable oblivious transfer. In *CRYPTO*, 2008.
45. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *J. ACM*, 2009.

46. R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21, 1978.
47. B.-Y. Yang, C.-H. O. Chen, D. J. Bernstein, and J.-M. Chen. Analysis of quad. In *FSE*, 2007.
48. B.-Y. Yang and J.-M. Chen. All in the xl family: Theory and practice. In *ICISC*, 2004.