Cryptography from tensor problems (draft)

Leonard J. Schulman*

May 1, 2012

Abstract

We describe a new proposal for a trap-door one-way function. The new proposal belongs to the "multivariate quadratic" family but the trap-door is different from existing methods, and is simpler.

Known quantum algorithms do not appear to help an adversary attack this trap-door. (Beyond the asymptotic square-root-speedup which applies to all oracle search problems.)

Keywords: Multivariate quadratic cryptosystem, MinRank, tensor rank, post-quantum cryptography.

1 Introduction

The requirement that a one-way function also possess a trap-door appears to impose considerable algebraic structure on the function. Such structure is a security risk, and that risk has increased with the development of quantum algorithms that can efficiently detect specific algebraic structures that are conjecturally opaque to classical algorithms. As is well known, all "abelian" cryptosystems (RSA, Diffie-Hellman, El Gamal, elliptic curve, Buchmann-Williams) have been broken due to the ability of a quantum computer to solve the hidden subgroup problem (HSP) in abelian groups in polynomial time [78, 76, 14, 43, 68].

It is necessary therefore to devise trap-doors (and other cryptographic primitives) with an eye to the particular computational advantages of quantum computers. Honest parties should still require only classical computation. (This is sometimes called "post-quantum cryptography".) There are three main classes of candidates for such cryptosystems. In each case, part of the argument for hardness is the NP-hardness of a closely related problem (a more general case, or nearby parameters); this cannot truly justify a cryptosystem [15] but helps argue first that there may be no sub-exponential time classical algorithm and second that quantum algorithms may have little advantage over classical ones [11]. These three classes of cryptosystems are:

(1) Lattice cryptosystems [2, 54, 70, 52, 69, 55, 53, 56] rely upon the hardness of the HSP in dihedral groups [70]. This assumption is particularly attractive due to the fact that it can be worst-case rather than average-case [1]. However the quantum version of this hardness assumption

*Engineering and Applied Science Division, California Institute of Technology. Email: schulman@caltech.edu. Supported in part by the NSF.

is in question because two positive results are known about dihedral HSP: (a) Single-register coset measurements are information-theoretically sufficient to solve the problem [33]. By contrast this is known to be false for more general HSP problems (e.g., in S_n) [45, 41, 58, 44], and proposed multi-register measurements [34, 6, 7] do not seem to lead toward an efficient algorithm. (b) An elegant and nontrivial algorithm is known for the problem [49] (and see [71]), running in time $\sim 2^{\sqrt{n}}$ on groups of size 2^n. By contrast it is known that this algorithm does not work in S_n [59].

(2) Code-based cryptosystems: McEliece and its relatives [51, 60]. The security of these cryptosystems is based the problem of decoding a random linear error-correcting code. However, the security of the system is known to depend upon the choice of the rapidly-decodable error-correcting code that forms its trap-door, to an extent that was not originally apparent [74, 75, 18, 19]. So far, instantiation of the method with randomly-chosen Goppa codes seems to be secure. However, the originally-proposed parameters have been shown to be insufficient (see [12], based on an attack in [80]). Also, a recent attack [36] has been effective, in a certain parameter range, against the problem of distinguishing a random Goppa code generator matrix from a random matrix; the hardness of this problem had been employed as one of the bases for security of the McEliece cryptosystem.

(3) "Multivariate polynomial" cryptosystems. In these systems the security is based on the hardness of solving systems of polynomial equations in finite fields (even constant-size fields), a problem that is known to be NP-hard (even for "multivariate quadratic" systems, henceforth "MQ"); more importantly this problem appears to be hard on average for a random system of equations. Trap-doors for such systems require designing a system of equations that is (in some helpful way) not fully random, yet is almost as hard for an adversary to solve. There has been a long sequence of work on such proposals (for a survey through 2005 see [90]), beginning with [50], which was cracked by Patarin [66], who then replaced that method by a more flexible generalization [67] called HFE; the most basic form of HFE has been cracked [48, 30, 37, 40], as have some variants [32], but more general forms (HFEv- used in Quartz), appear to remain viable with appropriate parameters [26, 31], although perhaps not as efficient as one might wish [86]. Broadly speaking all MQ cryptosystems are subject to attack by Gröbner basis algorithms for solving systems of polynomial equations, the leading method currently being Faugère's F5 algorithm; the key question is whether the systems of polynomials generated by the cryptosystem have features that enable such an algorithm to run in sub-exponential time.

Quantum algorithms have not been shown to possess any advantage over classical algorithms in solving systems of polynomial equations (other than the slight "Grover search" advantage [42]), despite a few successful attacks (see [20, 21] for some representatives) on problems that lack obvious linear, abelian, or normal group structure. For this reason, MQ cryptosystems are particularly promising for post-quantum cryptography.

Our work. In this paper we present an extremely simple multivariate-quadratic trap-door. As the details are neither long nor difficult, we do not give an overview, and only mention the following points for readers familiar with existing methods:

(a) Honest-party decryption is simpler than in previous methods. The run-time is dominated by inversion of a linear-sized $GF2$ matrix.

(b) Unlike earlier methods, the method is specific to the field $GF2$.

(c) We are aware of three main vulnerabilities of our method: an attack through the "MinRank" problem, an attack through "relinearization" and related methods, and an attack through the "LRTD" problem of finding low-rank tensor decompositions of low-rank tensors. We discuss these attacks in the body of the paper. We present two different versions of our method, "plain" and "+", with differing exposures to an LRTD attack.

2 Definitions

Fix the field $\mathbb{F} = GF2$. Throughout, we use vector spaces $U \cong \mathbb{F}^{n_1}$, $V \cong \mathbb{F}^{n_2}$, and $W \cong \mathbb{F}^{n_3}$, and write $\mathbb{T} = U \otimes V \otimes W$. This is a space of tensors of order 3 or "3-tensors". Taking $U^\dagger = \mathrm{Hom}(U, \mathbb{F})$ (etc.), there is an implied mapping from $\mathbb{T} \times U^\dagger$ to 2-tensors (elements of $V \otimes W$), from $\mathbb{T} \times (U^\dagger \times V^\dagger)$ to 1-tensors, etc. These mappings are restrictions of the *contraction* mappings, for example, the two-index contraction which carries $T \in \mathbb{T}$ and $A \in U^\dagger \otimes V^\dagger$ to a vector $T \cdot A \in W$. As usual "$\cdot$" indicates dot product or contraction of vectors in a space and its dual.

Likewise, the group $G = GL(U) \times GL(V) \times GL(W)$ acts on $\mathbb{T}$. (By definition, on the right.)

If we fix a basis for one of these vector spaces, e.g., $(e_1, \ldots, e_{n_1})$ for U, there is an implied dual basis $(e_1^\dagger, \ldots, e_{n_1}^\dagger)$ for $U^\dagger$ in which $e_i^\dagger e_j = \delta_{ij}$; if e_i are specified in terms of another underlying basis, then the $e_i^\dagger$ are specified (in terms of the dual of the underlying basis) by inverting the matrix of e_i's.

Tensors have a few easily-computed invariants under the action of G, such as the dimensions of the six vector spaces $T \times U^\dagger, T \times V^\dagger, T \times W^\dagger, T \times (V^\dagger \times W^\dagger), T \times (U^\dagger \times W^\dagger), T \times (U^\dagger \times V^\dagger)$.

Another invariant is tensor rank:

Definition 1. *A $(\mathbb{T}; r)$-tensor is one which can be expressed as*

$$T = \sum_{i=1}^{r} u_i \otimes v_i \otimes w_i$$

for some vectors $u_1, \ldots, u_r \in U$, $v_1, \ldots, v_r \in V$, $w_1, \ldots, w_r \in W$.

When we wish to focus on the dimensions of U, V, W, we refer to such a tensor as an $(n_1, n_2, n_3; r)$-tensor.

Definition 2. *The rank of a tensor $T \in \mathbb{T}$ is the least r such that T is a $(\mathbb{T}; r)$-tensor.*

The tensor rank of an $n \times n \times n$ tensor can be, and by a counting argument almost always is, quadratic in n. The maximum attainable rank is unknown, but lies between $n^2/3$ and $\lceil 3n^2/4 \rceil$ [47] (a better upper bound of $n(n+1)/2$ holds for algebraically closed fields [61]). The rank of a tensor is at least as large as the dimensions of the one-tensor and two-tensor spaces mentioned above, but those are usually poor lower bounds.

Unlike the previous invariants, tensor rank is apparently difficult to compute. Håstad [46] showed that the problem "Is rank $T \leq k$" is NP-complete over finite fields.

3 Trap-door one-way functions from rank-planted bilinear mappings

3.1 Bilinear mappings

In what follows we take $n = n_1 = \dim U = n_2 = \dim V$ and $m = n_3 = \dim W$. We now think of $\mathbb{T}$ as the space of bilinear mappings

$$\begin{aligned} T : U^\dagger \times V^\dagger &\to W \\ T(x, y) &= T \cdot (x \otimes y) \end{aligned}$$

or, in coordinates,

$$(T(x,y))_k = \sum_{i,j} T_{i,j,k} x_i y_j \qquad \text{for } k \in \{1,\ldots,m\}.$$

The *bilinear inversion problem* is, given $T \in \mathbb{T}$ and $z \in W$, find x and y such that $T(x,y) = z$.

As is well known, the bilinear inversion problem is NP-complete. (This is true also of inverting the mapping $x \to T(x,x)$.) We go further and conjecture *average-case hardness of bilinear inversion*:

Conjecture 3. *For any $0 < \alpha < 1$, any inversion algorithm A, and any $c > 0$, with $m = n^{1+\alpha}$, there is a $\beta > 0$ such that the probability (over T, x, y selected uniformly, and any internal randomness of A) that A on input $(T, T(x,y))$ outputs (T, x, y) within time 2^{n^β}, is less than n^{-c}.*

Let p_T = the probability (over x, y selected uniformly, and any internal randomness of A) that A on input $(T, T(x,y))$ outputs (T, x, y) within time 2^{n^β}. The above conjecture also implies that: *For any $0 < \alpha < 1$, any inversion algorithm A, and any $c > 0$, with $m = n^{1+\alpha}$, there is a $\beta > 0$ such that the probability (over T selected uniformly) that $p_T > n^{-c}$ is less than c.* (Substitute $2c$ in the conjecture.)

It will be a considerably stronger conjecture that this hardness holds also for the special rank-planted tensors that are used in the trap-door.

3.2 Rank-planted bilinear mappings

For the security of the "plain" variant of our trap-door proposal we need bilinear inversion to remain hard on average even in the following restricted scenario. As above (and throughout) set $m = n^{1+\alpha}$ for $0 < \alpha < 1$.

Definition 4. *The $D_{n,m}$-distribution on tensors T in $\mathbb{T}$ is the following probability distribution: sample independently and uniformly $u_1, \ldots, u_m \in U$, $v_1, \ldots, v_m \in V$, and nonsingular $w \in Hom(W, W)$; let w_i denote the i'th row of w. Form*

$$T = \sum_{i=1}^{m} u_i \otimes v_i \otimes w_i.$$

3.3 Trap-door: "plain" variant

Sample T from the distribution $D_{n,m}$ for $m = n^{1+\alpha}$. The public key is T, and the private key is the decomposition $u_1, \ldots, u_m \in U$, $v_1, \ldots, v_m \in V$, $w_1, \ldots, w_m \in W$. As before, the one-way function is the bilinear map $T : \mathbb{F}^{2n} \to \mathbb{F}^m$, $(x,y) \to T(x,y) = z$.

These bilinear functions seem to be (almost always) hard to invert without the private key, although of course the distribution on instances disables the NP-hardness (this is unavoidable since a trap-door for an NP-hard problem would put NP into P or BPP or BQP, depending on the resources required by the honest inverter [15]). Inversion is no harder than the tensor decomposition problem; this truly seems hard, although we have limited evidence for this claim. We discuss the question at greater length at the end of the paper in the context of the literature on tensor rank bounds.

The tensors in Håstad's reduction have the following superficial similarity to the ones we use: one of the three dimensions is longer than the other two and the rank of the tensors is at most linear in that longer dimension. However, the rank in that reduction could be as much as three times the

longer dimension, rather than exactly equal. The distinction is significant: if one takes $m = O(n)$, a MinRank attack for bilinear inversion is expected to run in polynomial time in the exactly-equal case but not when the rank is larger by a constant factor greater than 1 (see Sec. 4.1).

The trap door. Given the low rank decomposition (the private key), we can invert the mapping T as follows. Write the ℓth bit of z as

$$z_\ell = (T(x,y))_\ell = \sum (w_i)_\ell (x \cdot u_i)(y \cdot v_i)$$

Linearize this quadratic system of equations by letting $s_i = (x \cdot u_i)(y \cdot v_i)$. By construction, the matrix w (whose i'th row is w_i) is invertible. Therefore we can solve for s given z.

The key observation is that when $s_i = 1$, it follows that $x \cdot u_i = 1$ and $y \cdot v_i = 1$. (The $s_i = 0$ values are computationally less useful.)

Now consider any fixed $x \neq 0^n$ and $y \neq 0^n$. We can compute x from s provided that there are n linearly independent vectors u_i for which $s_i = 1$. Likewise, we can compute y from s provided that there are n linearly independent vectors v_i for which $s_i = 1$. In short, we can obtain x and y from s provided the following event A happens: There is a collection of indices $I \subseteq \{1, \ldots, m\}$ such that

1. $x \cdot u_i = 1$ for all $i \in I$,
2. $y \cdot v_i = 1$ for all $i \in I$,
3. $\{u_i\}_{i \in I}$ span U,
4. $\{v_i\}_{i \in I}$ span V.

We argue that this occurs with high probability. Let $c = m/n$; $b > 0$ will be specified below. Picking pairs (u_i, v_i) uniformly iid, there is, by a Chernoff bound [79], probability $\geq 1 - e^{-2(c/4-b)^2 n}$ that there exists a subset of these, I, of cardinality $\geq bn$ satisfying conditions 1, 2.

The vectors $\{u_i\}_{i \in I}$ are uniformly distributed subject to condition 1. For convenience label $I = \{1, \ldots, bn\}$. The probability that condition 3 fails is equal to the probability that the collection $\{u_i - u_1\}_{i \in I}$ fails to span the hyperplane $x^\perp$. For $i \geq 2$ write $u'_i = u_i - u_1$; observe that the vectors $u'_2, \ldots, u'_{bn}$ are iid uniform in $x^\perp$. In order that they not span $x^\perp$, all of $u'_n, \ldots, u'_{bn}$ must lie in $\mathrm{Span}(u'_2, \ldots, u'_{n-1})$; this occurs with probability at most 2^{n-1-bn}.

Likewise the vectors $\{v_i\}_{i \in I}$ are uniformly distributed subject to condition 2 and so the probability that condition 4 fails is also bounded by 2^{n-1-bn}.

In combination,

$$\Pr[x \text{ and } y \text{ can be obtained from } z \text{ by this procedure}] \geq 1 - e^{-2(c/4-b)^2 n} - 2^{-(b-1)n}.$$

Now taking a union bound over all $x \neq 0^n$ and $y \neq 0^n$,

$$\Pr[\text{for all nonzero } x \text{ and } y \text{ the bilinear mapping is invertible by this procedure}] \geq 1 - 2^{2n}(e^{-2(c/4-b)^2 n} + 2^{-(b-1)n}).$$

For any value of c this is maximized with the choice $b - 1 = 2(c/4 - b)^2 \lg e$, or

$$b = \frac{c}{4} + \frac{1}{4 \lg e} - \sqrt{\frac{c}{8 \lg e} + \frac{1}{16 \lg^2 e} - \frac{1}{2 \lg e}}.$$

We have $c = n^\alpha$. Using the above formula for b we have

Theorem 5. *With probability* $\geq 1-2^{2n+1-(n^\alpha/4-n^{\alpha/2}/\sqrt{8\lg e}-O(1))n} = 1-2^{-n^{1+\alpha}/4+n^{1+\alpha/2}/\sqrt{8\lg e}+O(n)}$ *(over the choice of the private key), the bilinear mapping is invertible by the above procedure on the images of all inputs* $x \neq 0, y \neq 0$.

□

We now discuss a few points.

PKE Preprocessing. Obtaining a public-key cryptosystem with semantic security requires some pre-processing in order to make the encryptions random; in an MQ system, as discussed already by Patarin [67], one must also avoid the following attack (a version of a well-known attack on RSA). If the encrypter can be induced to provide not only $T(x, y)$ but also $T(x + \delta_x, y + \delta_y)$ for some known (and generic) vectors $\delta_x \in U$, $\delta_y \in V$, then x and y can be recovered from the resulting system of linear equations. It is easy to protect against this weakness in the usual ways (padding x and y by random bits in each application of T, or—if semantic security is not required—by some nonlinear mapping, e.g., a cryptographic hash function).

Self-reducible alternative: A drawback of the above method is that we do not know a worst-to-average-case reduction. That is, there may be a large (even constant-probability) set of tensors for which the rank-planted bilinear inversion problem is easy, even though on the remaining set of tensors it is hard. This seems unlikely but cannot presently be ruled out, because we do not have a self-reducibility argument. The difficulty in providing such an argument is that the only way we know how to generate from a tensor T another of equivalent rank, is to act on T with some $g \in G$. The orbit of T under G, while large (of size $2^{\Theta(m^2)}$), is yet vanishingly small inside $\mathbb{T}$ (which is of size $2^{\Theta(n^2m)}$). This suggests an alternative method, which is to base the cryptosystem upon a single well-chosen rank-m tensor T_0, and to generate keys by sampling g uniformly in G and setting $T = T_0 g$.

However it seems unlikely that the self-reducibility gained is worth the loss in key size and the increased vulnerability to a solution to the tensor isomorphism problem (i.e., given two tensors T, S in the same orbit under G, find $g \in G$ such that $Tg = S$). (Although we have every reason to think that the tensor isomorphism problem requires exponential time even for quantum algorithms.) Also, at present (as just discussed), we have no guidelines to prefer any particular T_0 over another.

4 Vulnerabilities and variations

4.1 MinRank vulnerability

Like existing MQ cryptosystems, but for different reasons, this one is subject to a MinRank-based attack that will reveal the private key. The MinRank problem [30] is: Given a collection of $n \times n$ matrices (viewed as elements of a vector space) and a parameter $1 \leq r < n$, find a matrix of rank $\leq r$ in their linear span. (Alternatively, decide whether one exists; or, consider r as an output to be minimized.)

A "slice" of a tensor T in this paper will always mean a matrix $T_{**\ell}$ for some value of ℓ. The set of rank 1 linear combinations of the slice matrices $\{T_{**\ell}\}_{\ell=1}^m$ in the "plain" variant includes all the matrices $\{u_i \otimes v_i\}_{i=1}^m$. It is unlikely to include any other rank 1 matrices. (This is slightly delicate and to be formal is stated here as a *Speculation: The probability that the span of* $\{T_{**\ell}\}_{\ell=1}^m$ *contains any rank 1 matrix outside* $\{u_i \otimes v_i\}_{i=1}^m$ *is* $2^{-\Omega(n^2/m)}$. An algorithm which outputs the full listing of rank 1 matrices in the span of the slices can invert the bilinear mapping, unless the length of that

list approaches $m/4$, which appears to be extraordinarily unlikely. Although we are not confident enough of the particular probability bound to call the speculation a conjecture, we do conjecture an exponentially small bound; if this is somehow false, then the MinRank attack might not apply against the trap-door.)

MinRank (as a decision problem) is NP-hard, as is the special case $r = n - 1$ (deciding whether there exists a singular matrix in the span of given matrices). Even an inapproximability result is known for MinRank, and in fact for the following somewhat easier problem, studied by Buss, Frandsen and Shallit [17]: Given a collection of $n \times n$ matrices $M^0, M^1, \ldots, M^k$ which are entry-wise disjoint (for any i, j, at most one $M^\ell_{i,j}$ is nonzero), find the least rank r of any matrix in $M_0 + \mathrm{Span}(M_1, \ldots, M_k)$. They showed that it is NP-hard to approximate r to within a $1+\varepsilon$ factor for any $\varepsilon < 7/520$. (Their problem reduces to MinRank simply by letting M_0 range over the list of matrices.)

The MinRank task which needs to be performed here is a little different from the usual one of exhibiting *some* matrix of minimal rank r. On the one hand, we are in the special case $r = 1$, and this makes the problem much easier than the general case; on the other hand we need the algorithm to output *all* (or close to all) of the matrices of minimal rank, which in principle makes the problem harder. However, the "kernel attack" on MinRank ([22, 23, 48, 39]), which works well for small r, has the property of providing the full listing of minimal rank matrices in no more time than it provides one such matrix. This algorithm runs in time $2^{mr/n}\mathrm{poly}(mn)$. The idea is simple: let $t \in W^\dagger$ be any vector such that $\mathrm{rank}(T \cdot t) \leq r$. Choose $\lceil m/n \rceil$ vectors $x_1, \ldots x_{\lceil m/n \rceil}$ uniformly, independently in $U^\dagger$. If all lie in $\ker(T \cdot t)$ (and are in general position) then the $n\lceil m/n \rceil$ equations $\{(T \cdot t) \cdot x_i\}_i$ suffice to determine t. The probability of this occurring is approximately $2^{-mr/n}$. For recent bounds on MinRank computations (including the kernel attack and another attack less applicable to our parameters) see [35].

For our trap-door this implies an attack running in time $\sim 2^{n^\alpha}$.

4.2 Relinearization / Gröbner basis vulnerability

This vulnerability helps invert the bilinear function without necessarily revealing the private key.

Our bilinear mappings are easily invertible if m is taken to be at least $\binom{n}{2}$, because one may then *linearize* the system: define variables $a_{ij} = x_i y_j$ and solve $T \cdot a = T(x, y)$. This is, of course, simply exploiting the extension of the map T from $U^\dagger \times V^\dagger$ to $U^\dagger \otimes V^\dagger$. An interesting extension of this method, called *relinearization*, was introduced by Kipnis and Shamir [48]; they gave complexity estimates supporting polynomial-time inversion for $m = \varepsilon n^2$ for any fixed $\varepsilon > 0$. An optimization called XL was later introduced [27], and see [29, 28] for variations; later these methods were put into the general framework of Gröbner basis algorithms [5, 85, 3]. There has been particular focus on equations over $GF2$ and other small fields [28, 91, 57, 16, 9]. It is difficult to estimate how fast these methods will work for our system. There has been theoretical and empirical work estimating how fast these methods work on m random quadratic equations over n variables (and in particular for GF2), but these estimates are available only in the limits $m = n + k$ (k constant) and $m = cn$ (c constant) [10] and are not in a form we can see how to extrapolate from. (Encouragingly, in the latter case the conclusion is that such systems are a good source of hardness for cryptography.) These studies have continued apace [38, 9] but even the most recent of these gives estimates only for m up to $O(n)$.

In the absence of a guideline from recent studies, we fall back on an extrapolation of estimates from [27, 28]. Since the estimates predate some developments in Gröbner basis algorithms and

some analyses (empirical and theoretical) of the "degree of regularity" of systems of quadratic equations (this being the key unknown parameter in the runtimes of these algorithms), and since we are extrapolating from estimates that were apparently focused on values of m/n^2 smaller than in our application, the value we provide is quite likely to be inaccurate. In any case, the estimate (see [27] Sec. 6.5 or [28] Sec. 4.1; in the latter, the parameter μ explicitly captures the uncertainty in the estimate, and our numbers make sense if it is larger than any inverse polynomial) is for a runtime of $\sim 2^{n/\sqrt{m}}$. (We have simplified the expressions from the references due to the leaps of extrapolation). Thus, with the setting $m = n^{1+\alpha}$, this gives a very rough time estimate of $2^{n^{(1-\alpha)/2}}$. We hope that subsequent work will give more reliable estimates.

4.3 Structure of slices of T

As has been noted, the rank of the tensor T generated in the "plain" variant is fairly small, $m = n^{1+\alpha}$ for the constant α used in the construction.

One obvious question is whether at this very low rank, it is possible to infer something about the vectors $\{u_i\}_1^m$, $\{v_i\}_1^m$ simply by looking at individual slices of T. For slices we have all the tools of linear algebra at our disposal.

We now show that the answer to this question is negative.

Proposition 6. *For $m = cn$, $c > 2$, the distribution on each slice of T is within variation distance $2^{1-(c-2)n}$ of the uniform distribution on $n \times n$ matrices.*

Proof. Consider the group of $n \times n$ $\mathbb{F}$-matrices under addition; this is isomorphic to $(\mathbb{Z}/2)^{n^2}$. The characters of the group are in bijective correspondence with matrices M,

$$\chi_M(N) = (-1)^{\sum_{1 \le i,j \le n} M_{ij} N_{ij}}.$$

We are implementing a random walk on this group: a single step is an addition of $u \otimes v$ for u, v uniformly distributed. Letting P be this single-step distribution on the group, we are to show that P convolved with itself m times is close to uniform. Write

$$\chi_M(P) = 2^{-2n} \sum_{u,v} (-1)^{\sum_{1 \le i,j \le n} M_{ij} u_i v_j}.$$

For nonzero M, fix any nonzero row i of M. For any v such that $\sum M_{ij} v_j = 1 \bmod 2$ (which is true of half the vectors v), the involution $u \to u + e_i$ (flipping the i'th bit) is a bijection between pairs (u, v) with $\sum_{1 \le i,j \le n} M_{ij} u_i v_j = 0 \bmod 2$ and pairs (u, v) with $\sum_{1 \le i,j \le n} M_{ij} u_i v_j = 1 \bmod 2$. Consequently,

$$-1/2 \le \chi_M(P) \le 1/2.$$

A more careful version of this argument is to consider v according to whether or not its dot product with every row of M is 0. If this is the case, then for every u, $\sum_{1 \le i,j \le n} M_{ij} u_i v_j = 0 \bmod 2$. If this is not the case, then the same involution argument may be applied to the first row of M for which the dot product is 1. Therefore,

$$\chi_M(P) = 2^{-\operatorname{rank} M}.$$

Taking m steps of this walk gives a distribution P^{*m} whose Fourier coefficients are

$$\chi_M(P^{*m}) = 2^{-m \operatorname{rank} M}.$$

The Fourier coefficient of the uniform distribution is also 1 at $M = 0$, and is 0 elsewhere, so the variation distance of our distribution from uniform is equal to

$$2^{-n^2} \sum_N \left| \sum_{M \neq 0} 2^{-m \operatorname{rank} M} (-1)^{\sum_{1 \leq i,j \leq n} M_{ij} N_{ij}} \right|.$$

The number of matrices M of rank r is bounded by 2^{2nr} (this is a wasteful estimate for large r but no matter), so the variation distance is

$$\begin{aligned} &\leq 2^{-n^2} \sum_N \left| \sum_{r=1}^{n} 2^{2nr} 2^{-mr} \right| \\ &= \sum_{r=1}^{n} 2^{(2-c)nr} \\ &\leq 2^{1-(c-2)n} \end{aligned}$$

(Incidentally, observe that almost all of the contribution comes from the rank-one Fourier coefficients.) □

A simple extension of this argument shows that any small number of slices are, for sufficient c, distributed exponentially close to the uniform distribution. It is not clear exactly how quickly c must grow as a function of the number of slices in order to ensure near-uniformity. But the essential question is whether there is any efficient algorithm to reveal something about the collection $\{u_i\}_1^m$, $\{v_i\}_1^m$ by examining several slices in combination (a number growing fast enough in n for the above distribution to be non-uniform). This seems unpromising because so much information has been lost; a small collection of slices is not so nearly "abnormal" (as compared with the uniform distribution) as the full tensor. In any case the "+" variant of the method which we next describe is very well protected from this approach because any collection of $m - 21n$ slices is actually uniformly distributed.

4.4 LRTD vulnerability and the "+" variant

4.4.1 LRTD

Whereas the MinRank attack tries (essentially) to obtain the private key, it would be sufficient, instead, to find a slightly inferior substitute: a decomposition of the tensor T as the sum of m' rank 1 tensors, for m' not too much larger than m. Then an exhaustive search over the values of $m' - m$ "missing equations" would enable us to invert the bilinear mapping. (In a sense this attack is converse to MinRank: instead of trying to write the rank-1 matrices $\{u_i \otimes v_i\}_i$ as combinations of the slices of T, one tries to write the slices T as combinations of some other rank-1 matrices.)

Define the following problem:

Definition 7. *The (n, m, m')-LRTD problem (low-rank tensor decomposition problem, LRTD) is: given T sampled from the distribution $D_{n,m}$, write T as an $(n, n, m; m')$-tensor.*

For the security of the "plain" variant trap-door what we need, then, is that for some $\beta > 0$ (as large as possible), $(n, m, m + n^\beta)$-LRTD cannot be solved in less than time 2^{n^β}.

As far as we are aware, there is no polynomial-time algorithm which has better than a $2^{-n^{\Omega(1)}}$ probability of solving even the $(n, n^{1+\alpha}, n^2/4)$-LRTD-problem, that is, giving a decomposition in $n^2/4$ terms of a tensor known to require just $n^{1+\alpha}$ terms $(0 < \alpha < 1)$.

("LRTD" without parameters will simply mean the problem of finding an approximately-optimal-rank decomposition of a tensor that is, indeed, of low rank.)

In any case however the following trap-door variant is better protected against LRTD attacks.

4.4.2 Trap-door: "+" variant

MQ cryptosystems can be modified in a number of standard ways: see [90]. Of these we draw attention to the "+" variant, which reduces a weakness mentioned in the previous subsection, namely, that to crack the plain variant it is sufficient to have a tensor decomposition algorithm that is effective in the special case that the rank is equal to the longest dimension of the tensor. As we have seen, in that case, due to the connection to MinRank, there is a specialized approach to LRTD. It is of course possible that there are others. Our "+" variant reduces this exposure.

The basic idea is simple: augment the tensor T with additional "noise" slices, each slice being an independent uniformly random $n \times n$ matrix. The simplest option would be to place these slices in a uniformly random subset of the coordinates along the third axis of T (with the identity of this subset being part of the private key), but instead we use the following slightly better method.

Set $m = n^{1+\alpha}$ and $m_1 = m - 21n$. Set $W_0 \cong \mathbb{F}^{21n}$, and let $\{e_i\}(1 \leq i \leq 21n)$ be a basis for W_0. Set $W_1 \cong \mathbb{F}^{m_1}$ and $W = W_0 \times W_1$. Now sample the tensor T as follows: (a) For $1 \leq i \leq 21n$ select u_i, v_i uniformly from U, V respectively, and set $T_0 = \sum u_i \otimes v_i \otimes e_i$. (b) Select $T_1 \in U \otimes V \otimes W_1$ from the uniform distribution. (c) Select $g_3 \in GL(W)$ uniformly at random and set $T = (T_0 \times T_1)(1, 1, g_3)$.

The private key now consists of g_3 and all the u_i, v_i. The honest inverter applies g_3^{-1}, then ignores the information in W_1 and inverts as in the plain variant. Following through the calculation in Sec. 3.3 we have:

Theorem 8. *With probability $\geq 1 - 2^{-n}$ (over the choice of the private key), the bilinear mapping is invertible by the above procedure on the images of all inputs $x \neq 0, y \neq 0$.*

□

Unlike in the "plain" variant, it is no longer obviously sufficient for an adversary to possess an algorithm for low-rank tensor decomposition, because, due to the noise, the public-key tensor has rank almost as high as a uniformly random tensor.

5 Discussion

1. We have used the parametrization $m = n^{1+\alpha}$ which balances the MinRank and relinearization attacks. The former runs in time $\sim 2^{n^\alpha}$; the runtime of the latter is less clear but we have tentatively used the figure $\sim 2^{n^{(1-\alpha)/2}}$. (We are not aware of any really effective LRTD attacks.) We can balance the above figures with the selection $\alpha = 1/3$. This will likely change as the quality of the attacks becomes clearer.

2. A key question which our work motivates is, whether quantum algorithms offer any advantage over classical ones for the problem of solving systems of polynomial equations. This

question should be understood not only in a worst-case complexity sense but more broadly, and especially w.r.t. the types of instance distributions provided by cryptosystems.

3. A desirable argument for the security of the trap-door proposal would be to reduce an apparently hard problem to cracking it—most obviously, of course, MinRank, or perhaps LRTD, in some parameter range. A difficulty in doing this is that the image of the bilinear mapping is exponentially sparse.

4. Even in the absence of such a reduction, it would be encouraging for the security of the trap-door to strengthen Håstad's theorem by showing (even worst-case) hardness of the tensor decomposition problem (or at least its decision version) for r' slightly higher than the true rank r, even $r' = r + \log n$ or $r + \operatorname{polylog} n$. The truth probably extends to large constant factors (e.g., $100r$) and possibly even to powers (e.g., $r^{1.1}$).

   As has been pointed out in [4], good inapproximability results for tensor rank (in this case, just for the value of the rank, not necessarily for finding an actual decomposition) are likely hard to come by, since they would (at least by deterministic reductions) imply construction of explicit tensors with high rank; currently, in spite of substantial efforts, the best known construction (over $\mathbb{F}$) is of $n \times n \times n$ tensors with rank $3n - o(n)$ [77, 4, 88]. However, as noted above, for our purpose even a very weak inapproximability result would provide some support for the security of the trap-door.

5. There is empirical evidence for the hardness of the tensor decomposition problem from work on efficiently computing bilinear problems, including integer multiplication and, most notably, matrix multiplication. The essential target of that work has been, indeed, to obtain low rank decompositions of some specific families of tensors. And, just as for this cracking this cryptosystem, it is not necessary for that goal to find decompositions that achieve the exact rank of the tensor; decompositions which are not too much larger are also useful. It is therefore notable that upper bounds for the matrix multiplication exponent improved only incrementally despite efforts spanning two decades [82, 62, 13, 63, 64, 73, 65, 72, 24, 83, 84, 25], then stalled entirely for another two decades, until two small improvements recently with considerable computational investment [81, 89]. (Perhaps another sign of the hardness of the decomposition problem is that, after the first few papers, efforts ceased to upper bound the rank directly; progress on the exponent came through improved upper bounds on the border rank, a number that can be smaller but still gives an equivalent-exponent algorithm, with some additional constant-factor overhead.)

6. It is worth comparing the (honest) inversion time in our system with that in HFE. In HFE, the dominant step in the runtime is factoring a univariate polynomial over a finite field. The corresponding HFE runtime depends upon assumptions about the parameters in the cryptosystem. One parameter is the degree of a small field extension, and for purpose of comparison we suppose this parameter equals 1. The other, d, is the degree of the underlying univariate polynomial, and is a security parameter; it is therefore fairly large, at least in the thousands. The runtime of the current leading factoring algorithm over small fields [87] (at least asymptotically and in most ranges of the parameters) is then $O(d^{1.5+o(1)} + d^{1+o(1)}n)$. By contrast the runtime of our inversion procedure is $O(n^{2+\alpha} + M(n))$, where $n^{2+\alpha} = mn$ accounts for the matrix-vector multiplication which converts z to s (note that we produce only enough bits of s so that with high probability x and y are determined), and $M(n)$ is the time for computing a selected-inverse of each of the two $n \times bn$ matrices formed from the vectors in $\{u_i\}_{i \in I}$ and $\{v_i\}_{i \in I}$ (see Sec. 3.3). By selected-inverse we mean the inverse of any nonsingular

$n \times n$ minor. Since a randomly chosen $n \times 2n$ minor is likely to have rank n, $M(n)$ is $O(n^\omega)$ where ω as before is the linear algebra computation exponent, currently $\omega < 2.3727$. (The inputs are large enough that at least Strassen's method may be effective, see [8], and we may benefit from the fact that some processors are optimized for linear algebra computations.) Since n is at least in the hundreds, the honest inversion runtimes in the two systems are arguably in the same ballpark. We do not feel confident saying more at present. Better comparison depends on better understanding of the relative security of the two methods, so that the scalings between d, n, m can be adjusted to compare runtimes when controlling for the amount of time required to crack the system. (Of course it is also possible that in time there will be improvements in linear algebra and polynomial factoring algorithms; the ultimate exponent could be 2 for both problems. Such developments would affect runtimes for both honest inversion and attack.)

A similar issue arises in considering the evaluation time, which in our method is $n^{3+\alpha}$ (equal to the size of the public key). Again this cannot be evaluated in isolation because the underlying parameter is not "n", but bits of security (logarithm of attack runtime).

7. The constructions we've described have obvious generalizations to tensors of order higher than 3. ("Multivariate cubic" systems, etc.) It is likely, although difficult to assess, that higher order increases security; unfortunately it also increases encryption time and public key size. Therefore $d = 2$ is the preferred choice unless it transpires that security is significantly better for larger d.

This draft is being posted on the IACR Cryptology ePrint Archive. Comments are welcome to `schulman@caltech.edu`.

Acknowledgments

Thanks to Oded Regev, Yi-Kai Liu and Aram Harrow for helpful comments. Thanks also to the organizers of a 2011 Dagstuhl workshop on post-quantum cryptography, which helped stimulate this work.

References

[1] M. Ajtai. Generating hard instances of lattice problems (extended abstract). In *Proc. 28th Annual ACM STOC*, pages 99–108, 1996.

[2] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proc. 29th Annual ACM STOC*, pages 284–293, 1997.

[3] M. Albrecht, C. Cid, J.-C. Faugère, and L. Perret. On the relation between the MXL family of algorithms and Gröbner basis algorithms. Cryptology ePrint Archive, Report 2011/164, 2011.

[4] B. Alexeev, M. Forbes, and J. Tsimerman. Tensor rank: some lower and upper bounds. (Preprint arXiv:1102.0072v1), 2011.

[5] G. Ars, J.-C. Faugère, H. Imai, M. Kawazoe, and M. Sugita. Comparison Between XL and Gröbner Basis Algorithms. In *ASIACRYPT*, pages 338–353, 2004.

[6] D. Bacon, A. Childs, and W. van Dam. From optimal measurement to efficient quantum algorithms for the hidden subgroup problem over semidirect product groups. In *Proc. 46th Annual IEEE FOCS*, pages 469–478, 2005. (Preprint quant-ph/0504083).

[7] D. Bacon, A. Childs, and W. van Dam. Optimal measurements for the dihedral hidden subgroup problem. *Chicago J. Theoret. Comput. Sci.*, 2:1–25, 2006. (Preprint quant-ph/0501044).

[8] D. H. Bailey, K. Lee, and H. D. Simon. Using Strassen's algorithm to accelerate the solution of linear systems. *The Journal of Supercomputing*, 4:357–371, 1990.

[9] M. Bardet, J.-C. Faugère, B. Salvy, and P.-J. Spaenlehauer. On the complexity of solving quadratic boolean systems. arXiv:1112.6263, 2011.

[10] M. Bardet, J.-C. Faugère, B. Salvy, and B.-Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. In *Eighth International Symposium on Effective Methods in Algebraic Geometry (MEGA)*, 2005.

[11] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, October 1997.

[12] D. J. Bernstein, T. Lange, and C. Peters. Attacking and Defending the McEliece Cryptosystem. In *Proceedings of the 2nd International Workshop on Post-Quantum Cryptography*, PQCrypto '08, pages 31–46, Berlin, Heidelberg, 2008. Springer-Verlag. Cryptology ePrint Archive, Report 2008/318.

[13] D. Bini, M. Capovani, F. Romani, and G. Lotti. O($n^{2.7799}$) complexity for n*n approximate matrix multiplication. *Inf. Process. Lett.*, 8(5):234–235, 1979.

[14] D. Boneh and R. J. Lipton. Quantum Cryptanalysis of Hidden Linear Functions (Extended Abstract). In *CRYPTO*, pages 424–437, 1995.

[15] G. Brassard. A note on the complexity of cryptography. *IEEE Transactions on Information Theory*, 25:232–233, 1979.

[16] J. Buchmann, J. Ding, M. S. E. Mohamed, and W. S. A. E. Mohamed. MutantXL: solving multivariate polynomial equations for cryptanalysis. Dagstuhl seminar proceedings 09031, http://drops.dagstuhl.de/opus/volltexte/2009/1945, 2009.

[17] J. F. Buss, G. S. Frandsen, and J. Shallit. The computational complexity of some problems of linear algebra. *J. Comput. Syst. Sci.*, 58(3):572–596, 1999.

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

[19] A. Canteaut and N. Sendrier. Cryptoanalysis of the Original McEliece Cryptosystem. In *ASIACRYPT*, pages 187–199, 1998.

[20] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proc. 35th Annual ACM STOC*, pages 59–68, New York, NY, USA, 2003. ACM.

[21] A. M. Childs, L. J. Schulman, and U. V. Vazirani. Quantum algorithms for hidden nonlinear structures. In *Proc. 48th Annual IEEE FOCS*, pages 395–404, 2007.

[22] D. Coppersmith, J. Stern, and S. Vaudenay. Attacks on the birational permutation signature schemes. In *CRYPTO*, pages 435–443, 1993.

[23] D. Coppersmith, J. Stern, and S. Vaudenay. The security of the birational permutation signature schemes. *J. Cryptology*, 10(3):207–221, 1997.

[24] D. Coppersmith and S. Winograd. On the asymptotic complexity of matrix multiplication. *SIAM J. Comput.*, 11(3):472–492, 1982.

[25] D. Coppersmith and S. Winograd. Matrix multiplication via arithmetic progressions. *Journal of Symbolic Computation*, 9(3):251–280, 1990.

[26] N. Courtois, M. Daum, and P. Felke. On the Security of HFE, HFEv- and Quartz. In *Proceedings of the 6th International Workshop on Theory and Practice in Public Key Cryptography*, PKC '03, pages 337–350, London, UK, 2003. Springer-Verlag.

[27] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *EUROCRYPT*, pages 392–407, 2000.

[28] N. Courtois and J. Patarin. About the XL Algorithm over GF(2). In *CT-RSA*, pages 141–157, 2003.

[29] N. Courtois and J. Pieprzyk. Cryptanalysis of block ciphers with overdefined systems of equations. In *ASIACRYPT*, pages 267–287, 2002.

[30] N. T. Courtois. The security of Hidden Field Equations (HFE). In D. Naccache, editor, *Cryptographer's Track at RSA Conference 2001, volume 2020 of Lecture Notes in Computer Science*, pages 266–281. Springer, 2001.

[31] N. T. Courtois. Short Signatures, Provable Security, Generic Attacks and Computational Security of Multivariate Polynomial Schemes such as HFE, Quartz and Sflash. *IACR Cryptology ePrint Archive*, page 143, 2004.

[32] V. Dubois, L. Granboulan, and J. Stern. Cryptanalysis of HFE with Internal Perturbation. In *Public Key Cryptography*, pages 249–265, 2007.

[33] M. Ettinger and P. Høyer. On quantum algorithms for noncommutative hidden subgroups. *Advances in Applied Mathematics*, 25(3):239–251, 2000. (LANL preprint quant-ph/9807029, 1998).

[34] M. Ettinger, P. Høyer, and E. Knill. Hidden subgroup states are almost orthogonal. (LANL preprint quant-ph/9901034), 1999.

[35] J.-C. Faugère, M. S. El Din, and P.-J. Spaenlehauer. Computing loci of rank defects of linear matrices using Gröbner bases and applications to cryptology. In *ISSAC*, pages 257–264, 2010.

[36] J.-C. Faugère, V. Gauthier-Umaña, A. Otmani, L. Perret, and J.-P. Tillich. A distinguisher for high rate McEliece cryptosystems. In *Proc. Information Theory Workshop (ITW)*, pages 282–286. IEEE, Oct. 2011.

[37] J.-C. Faugère and A. Joux. Algebraic Cryptanalysis of Hidden Field Equation (HFE) Cryptosystems Using Gröbner Bases. In *CRYPTO*, pages 44–60, 2003.

[38] J.-C. Faugère, F. Levy-Dit-Vehel, and L. Perret. Cryptanalysis of MinRank. In *Proceedings of the 28th Annual conference on Cryptology: Advances in Cryptology*, CRYPTO 2008, pages 280–296, Berlin, Heidelberg, 2008. Springer-Verlag.

[39] L. Goubin and N. Courtois. Cryptanalysis of the ttm cryptosystem. In *ASIACRYPT*, pages 44–57, 2000.

[40] L. Granboulan, A. Joux, and J. Stern. Inverting HFE Is Quasipolynomial. In *CRYPTO*, pages 345–356, 2006.

[41] M. Grigni, L. J. Schulman, M. Vazirani, and U. Vazirani. Quantum mechanical algorithms for the nonabelian hidden subgroup problem. *Combinatorica*, 24(1):137–154, 2004. (STOC '01).

[42] L. Grover. A fast quantum mechanical algorithm for database search. In *Proc. 28th Annual ACM STOC*, pages 212–219, 1996.

[43] S. Hallgren. Polynomial-time quantum algorithms for Pell's equation and the principal ideal problem. *J. ACM*, 54(1), 2007. (STOC '02).

[44] S. Hallgren, C. Moore, M. Rötteler, A. Russell, and P. Sen. Limitations of quantum coset states for graph isomorphism. *J. ACM*, 57:34:1–34:33, November 2010.

[45] S. Hallgren, A. Russell, and A. Ta-Shma. Normal subgroup reconstruction and quantum computation using group representations. In *Proc. 32nd Annual ACM STOC*, pages 627–635, 2000.

[46] J. Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11:644–654, December 1990.

[47] T. D. Howell. Global properties of tensor rank. *Linear Algebra and its Applications*, 22:9 – 23, 1978.

[48] A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem. In M. Wiener, editor, *Advances in Cryptology — CRYPTO 1999, volume 1666 of Lecture Notes in Computer Science*, pages 19–30. Springer, 1999.

[49] G. Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup. *SIAM J. Comput.*, 35(1):170–188, 2005.

[50] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *EUROCRYPT*, pages 419–453, 1988.

[51] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. Technical Report 42-44, Jet Propulsion Lab, Pasadena CA, 1978.

[52] D. Micciancio. Almost perfect lattices, the covering radius problem, and applications to Ajtai's connection factor. *SIAM J. Comput.*, 34:118–169, 2004. (STOC '02 and CCC '02).

[53] D. Micciancio. Generalized compact knapsacks, cyclic lattices, and efficient one-way functions. *Computational Complexity*, 16(4):365–411, 2007. (FOCS '02).

[54] D. Micciancio and S. Goldwasser. *Complexity of Lattice Problems*. Kluwer, 2002.

[55] D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. *SIAM J. Comput.*, 37(1):267–302, 2007. (FOCS '04).

[56] D. Micciancio and O. Regev. Lattice-based cryptography. In D. J. Bernstein, J. Buchmann, and E. Dahmen, editors, *Post-Quantum Cryptography*, pages 147–191. Springer, 2009.

[57] M. S. E. Mohamed, W. S. A. E. Mohamed, J. Ding, and J. Buchmann. MXL2: Solving Polynomial Equations over GF(2) Using an Improved Mutant Strategy. In *PQCrypto*, pages 203–215, 2008.

[58] C. Moore, A. Russell, and L. J. Schulman. The symmetric group defies strong fourier sampling. *SIAM J. Comput.*, 37(6):1842–1864, 2008. (FOCS '05).

[59] C. Moore, A. Russell, and P. Sniady. On the impossibility of a quantum sieve algorithm for graph isomorphism. *SIAM J. Comput.*, 39(6):2377–2396, 2010. (STOC '07).

[60] H. Niederreiter. Knapsack-type cryptosystems and algebraic coding theory. *Problems of Control and Information Theory (Problemy Upravlenija i Teorii Informacii)*, 15(2):159–166, 1986.

[61] P. Pamfilos. On the maximum rank of a tensor product. *Acta Math. Hung.*, 45(1-2):95–97, 1985.

[62] V. Pan. Strassen's algorithm is not optimal. In *Proc. 19th Annual IEEE FOCS*, pages 166–176, 1978.

[63] V. Pan. Field extension and trilinear aggregating, uniting and canceling for the acceleration of matrix multiplications. In *Proc. 20th Annual IEEE FOCS*, pages 28–38, 1979.

[64] V. Pan. New fast algorithms for matrix operations. *SIAM J. Comput.*, 9:321–342, 1980.

[65] V. Pan. New combinations of methods for the acceleration of matrix multiplication. *Comput. Math. with Appl.*, 7:73–125, 1981.

[66] J. Patarin. Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In *CRYPTO*, pages 248–261, 1995.

[67] J. Patarin. Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In *EUROCRYPT*, pages 33–48, 1996.

[68] J. Proos and C. Zalka. Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information & Computation*, 3(4):317–344, 2003. (Preprint arXiv:quant-ph/0301141).

[69] O. Regev. New lattice-based cryptographic constructions. *J. ACM*, 51(6):899–942, November 2004. (STOC '03).

[70] O. Regev. Quantum computation and lattice problems. *SIAM J. Comput.*, 33(3):738–760, June 2004. (FOCS '02).

[71] O. Regev. A subexponential time algorithm for the dihedral hidden subgroup problem with polynomial space. (Preprint arXiv:quant-ph/0406151v1), 2004.

[72] F. Romani. Some properties of disjoint sums of tensors related to matrix multiplication. *SIAM J. Comput.*, 11:263–267, 1982.

[73] A. Schönhage. Partial and total matrix multiplication. *SIAM J. Comput.*, 10(3):434–455, 1981.

[74] N. Sendrier. An algorithm for finding the permutation between two equivalent binary codes. Technical Report RR-2853, INRIA, April 1996.

[75] N. Sendrier. Finding the permutation between equivalent linear codes: The support splitting algorithm. *IEEE Transactions on Information Theory*, 46(4):1193–1203, 2000.

[76] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. (FOCS '94).

[77] A. Shpilka. Lower bounds for matrix product. *SIAM J. Comput.*, 32(5):1185–1200, 2003.

[78] D. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997.

[79] J. Spencer. *Ten Lectures on the Probabilistic Method*. SIAM, 1987. Lecture 4.

[80] J. Stern. A method for finding codewords of small weight. In *Proceedings of the 3rd International Colloquium on Coding Theory and Applications*, pages 106–113, London, UK, 1989. Springer-Verlag.

[81] A. J. Stothers. *On the complexity of matrix multiplication*. PhD thesis, U. Edinburgh, 2010.

[82] V. Strassen. Gaussian elimination is not optimal. *Numer. Math.*, 13:354356, 1969.

[83] V. Strassen. The asymptotic spectrum of tensors and the exponent of matrix multiplication. In *Proc. 27th Annual IEEE FOCS*, pages 49–54, 1986.

[84] V. Strassen. Relative bilinear complexity and matrix multiplication. *J. Reine Angew. Mathe.*, 375:406–443, 1987.

[85] E. Thomae and C. Wolf. Solving systems of multivariate quadratic equations or: from relinearization to MutantXL. Cryptology ePrint Archive 2010/596.

[86] S. Tsujii, M. Gotaishi, K. Tadaki, and R. Fujita. Proposal of a Signature Scheme Based on STS Trapdoor. In *PQCrypto*, pages 201–217, 2010.

[87] C. Umans. Fast polynomial factorization and modular composition in small characteristic. In *Proc. 40th Annual ACM STOC*, pages 481–490, 2008.

[88] B. Weitz. An improvement on rank of explicit tensors. (Preprint arXiv:1102.0580v2), 2011.

[89] V. V. Williams. Breaking the Coppersmith-Winograd barrier. Manuscript, 2011.

[90] C. Wolf and B. Preneel. Taxonomy of public key schemes based on the problem of multivariate quadratic equations. Cryptology ePrint Archive, Report 2005/077, 2005.

[91] B.-Y. Yang and J.-M. Chen. Theoretical Analysis of XL over Small Fields. In *ACISP*, pages 277–288, 2004.