Square, a New Multivariate Encryption Scheme

Crystal Clough[1], John Baena[1,2], Jintai Ding[1,4],
Bo-Yin Yang[3], and Ming-shing Chen[3]

[1] Department of Mathematical Sciences,
University of Cincinnati,
Cincinnati, OH, 45220, USA
{baenagjb,cloughcl}@email.uc.edu,ding@math.uc.edu
http://math.uc.edu
[2] Universidad Nacional de Colombia
Carrera 65, Medellín, Colombia
http://www.unalmed.edu.co/~dirmate/
[3] Institute of Information Science,
Academia Sinica
Taipei, Taiwan
[4] College of Sciences,
South China University of Technology
Guangzhou, China

Abstract. We propose and analyze a multivariate encryption scheme that uses odd characteristic and an embedding in its construction. This system has a very simple core map $F(X) = X^2$, allowing for efficient decryption. We also discuss ways to make this decryption faster with specific parameter choices. We give heuristic arguments along with experimental data to show that this scheme resists all known attacks.

1 Introduction

Multivariate public-key cryptosystems (MPKCs) are considered viable options for post-quantum cryptography. This is because they are based on the problem of solving a system of multivariate polynomial equations, a problem which seems just as hard for a quantum computer to solve as any other computer [12,20]. There are a few MPKCs that are believed secure and practical. We propose and analyze a new encryption scheme that is both efficient and secure.

One tool used in several systems is the "big field" idea. While the public keys of MPKCs are polynomial maps $k^n \to k^n$ for some finite field k, some are constructed using maps over a "big field" $K \cong k^n$, using vector space isomorphisms to go back and forth between the spaces. This approach is a two-edged sword in the sense that the field structure of K can make decryption easier but can also be utilized by attackers.

Until recently, the systems based on the "big field" idea (such as the original MPKC C^* proposed by Matsumoto and Imai, HFE proposed by Patarin, and their many variants) had other commonalities. All used characteristic 2 fields,

M. Fischlin (Ed.): CT-RSA 2009, LNCS 5473, pp. 252–264, 2009.

often $k = \mathbb{F}_2$, and the collection of plaintexts comprised all of K. Both of these conventions have recently been called into question.

Odd-characteristic MPKCs have not been popular, presumably because characteristic 2 is so fast and easy to implement. However, it now appears that even characteristic has a major drawback. The field equations $x_i^2 - x_i = 0$ allow algebraic attacks to be much more successful, as will be discussed below. Recent work shows that odd-characteristic systems can be much simpler than their even-characteristic counterparts while still evading algebraic attacks [2,6].

As for using all of K, the idea of using a "projection" or embedding is not new but has not held significant interest until recently. By using a K larger than k^n, there is hope that the field structure can still be helpful but no longer troublesome. Our data, and that of others, suggests that this is in fact a good idea.

The new system that we propose uses both of these ideas and comes to the surprising conclusion that under specific circumstances, a variant of the original Matsumoto-Imai system (which has been broken for more than 10 years) can be viable. The main idea was also proposed by Patarin [20], but he dismissed it. Also, at the time Patarin's system was published, the powerful algebraic attack tools F_4 and F_5 were not yet invented so he did not have to consider them. The Square system we will describe avoids Patarin's original attack and resists algebraic attacks as well.

This paper is organized as follows. In Section 2, we discuss relevant background material. In Section 3 we describe the new system Square. In Section 4 we analyze the effectiveness of known attacks. In Section 5 we give our parameter suggestions, as well as discuss how the system can be made very fast (see Table 1, [8]). We conclude the paper in Section 6.

Table 1. Speed of Square instances compared to other systems on a Core 2 Duo 2.4GHz

Scheme	q	n	l	PubKey	PrvKey	Encr	Decr
Square-42	31	42	3	28.4 kB	1350 B	9.4 μs	9.6 μs
Square-51	31	51	3	49.6 kB	1944 B	13.6 μs	14.4 μs
NTRU	587	787	n/a	1.5 kB	1854 B	149.2 μs	251.5 μs
McEliece	n/a	n/a	n/a	79.5 kB	137282 B	29.7 μs	444.1 μs

2 MPKCs and Relevant Attacks

2.1 C^* and HFE

Among first MPKCs was an encryption scheme C^* [16]. This system has since been broken [17], but has inspired many new encryption and signature schemes. One of these is HFE (Hidden Field Equations) [18], which can be seen as a generalization of the Matsumoto-Imai idea. Attacks on either of these systems

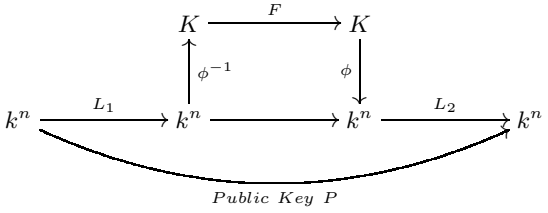

Fig. 1. The MI and HFE systems

or their variants could be relevant to the new system, so we will describe both HFE and C^* here.

Refer to Figure 1. In either system, the plaintext is a vector of length n over k, a field of q elements where q is a power of 2. Since there is a field K of the same size as k^n, we can utilize a nonlinear core map $F\colon K \to K$. In fact, the public key is $P = L_2 \circ \phi \circ F \circ \phi^{-1} \circ L_1$, where L_1 and L_2 are linear maps and ϕ is a vector space isomorphism $K \to k^n$. P is a collection of n polynomials $p_i(x_1, \ldots, x_n)$ in n variables. The decomposition, in particular L_1 and L_2, is the private key.

In the case of C^*, $F(X) = X^{q^\theta+1}$ for an appropriate θ. In the case of HFE, for some D

$$F(X) = \sum_{\substack{0\le i<j<n \\ q^i+q^j\le D}} a_{ij} X^{q^i+q^j} + \sum_{\substack{0\le i<n \\ q^i\le D}} b_i X^{q^i} + c. \tag{1}$$

2.2 Linearization Equations Attack

In the original attack on C^*, Patarin noticed that if $Y = X^{q^\theta+1}$, then $XY^{q^\theta} - X^{q^{2\theta}}Y = 0$ [17]. This equation forces plaintext-ciphertext pairs from C^* systems to satisfy linearization equations

$$\sum_{i,j=1}^{n} a_{ij} x_i y_j + \sum_{j=1}^{n} b_j x_j + \sum_{j=1}^{n} c_j y_j + d = 0,$$

where $(y_1, \ldots, y_n) = P(x_1, \ldots, x_n)$. Such equations are extremely useful for an attacker because given a ciphertext, the linearization equations yield linear equations satisfied by the plaintext. Also, linearization equations can be found easily from the public key, so an attacker has access to them.

Diene et al showed that the space of linearization equations satisfied by a C^* public key has dimension at least n in most cases [4]. Furthermore Patarin showed that for a given nonzero ciphertext, the space of linear equations satisfied by the correspoding plaintext is at least $n - gcd(n, \theta)$ [17].

Note that in the original C^* construction, $\theta = 0$ cannot be chosen since X^2 is a linear map when $q = 2$.

2.3 Algebraic Attack

Algebraic attacks can be employed against any MPKC. Suppose that someone, who does not know the private key, wants to recover the plaintext from a ciphertext $(\tilde{y}_1, \ldots, \tilde{y}_n) \in k^n$. This attacker has access only to the public key $(p_1, p_2, \ldots, p_n) : k^n \to k^n$. The most straightforward way to attack is to solve the system of equations

$$\begin{aligned} p_1(x_1, \ldots, x_n) - \tilde{y}_1 &= 0 \\ p_2(x_1, \ldots, x_n) - \tilde{y}_2 &= 0 \\ &\vdots \\ p_n(x_1, \ldots, x_n) - \tilde{y}_n &= 0. \end{aligned} \tag{2}$$

Solving these equations directly, without the use of the internal structure of the system, is known as the algebraic attack. Currently the most efficient algebraic attacks are the Gröbner basis algorithms F_4 [9] and F_5 [10]. Another algorithm called XL has also been widely discussed but F_4 is seen to be more efficient [1], so we focused our energy on studying algebraic attacks via F_4. Among the best implementations of these algorithms is the F_4 function of MAGMA [15], which represents the state of the art in polynomial solving technology.

Gröbner basis attacks are very fast against the C^* scheme, and also quite effective against HFE as well [11].

2.4 SFlash-Style Attack

SFlash is a C^{*-} signature scheme. The system is constructed in the same way as C^*, except that some number r of the public key polynomials are not published [19]. SFlash was broken using properties of its differential. Recall that for a function f the differential is

$$Df(a, x) = f(a + x) - f(a) - f(x) + f(0).$$

In the case of the C^* core map $F(X) = X^{q^\theta+1}$ for $\xi,\ A,\ X \in K$,

$$DF(\xi A, X) + DF(A, \xi X) = (\xi + \xi^{q^\theta})DF(A, X). \tag{3}$$

This equation leads to conditions which allow us to identify which matrices in $\mathcal{M}_{n\times n}(k)$ correspond to multiplications in K. These matrices N_ξ have the property that for the C^* public key P,

$$P(N_\xi(x_1, \ldots, x_n)) = M(P(x_1, \ldots, x_n))$$

for some linear map M. In other words, the N_ξ mix up the public key polynomials, allowing Dubois et al to complete the collection of public key polynomials thus breaking the system [7].

2.5 Kipnis-Shamir Style Attacks

The Kipnis-Shamir attack against HFE exploits the "big field" structure. It uses the following facts, all found in [14]:

- Let $\mathbb{F}_{q^n}$ be the field of q^n elements. If $G \in \mathbb{F}_{q^n}[X]$ such that the q-Haming weight of all monomials is 2 (ie, $G(X) = \sum a_{ij} X^{q^i+q^j}$), then $\exists \mathcal{G} \in \mathcal{M}_{n\times n}(\mathbb{F}_{q^n})$ such that
$$G(X) = \left(X \; X^q \; \cdots \; X^{q^{n-1}}\right) \mathcal{G} \begin{pmatrix} X \\ X^q \\ \vdots \\ X^{q^{n-1}} \end{pmatrix}.$$
- If $\mathcal{G}$ is such a matrix for G, S is a linear map $\mathbb{F}_{q^n} \to \mathbb{F}_{q^n}$, and $F = S \circ G$, then
$$\mathcal{F} = \sum_{j=0}^{n} s_j \mathcal{G}^{*j},$$
where the s_j are the coefficients of S and the $\mathcal{G}^{*j}$ are obtained from $\mathcal{G}$ via permutations and Frobenius maps ($x \mapsto x^{q^j}$ for $0 \le j \le n-1$).
- If $\mathcal{G}$ is such a matrix for G, S is a linear map $\mathbb{F}_{q^n} \to \mathbb{F}_{q^n}$, and $F = G \circ S$, then
$$\mathcal{F} = W\mathcal{G}W^T,$$
where W is obtained from the coefficients of S.

Kipnis and Shamir noted that in the case of HFE, by "lifting" the public key to an extension $L \cong \mathbb{F}_{q^n}$ via some isomorphism $\psi\colon L \to k^n$ and considering the quadratic part, we can find corresponding matrices whose rank must be no more than D. Though L is not necessarily the K used to construct the public key, we still have a decomposition $P = \psi \circ S \circ F \circ T \circ \psi^{-1}$ where S and T are linear and F is some HFE core map.

Using the facts above, Kipnis and Shamir were able to find such a decomposition [14]. The success depends on solving the MinRank problem: Given a collection of matrices, find a linear combination of them that has minimal rank. In general, this problem is NP-complete [3].

3 Design of Square

Having seen what a multivariate encryption scheme is up against, we now describe a new system Square.

See Figure 2. Let k be a field of size q, where $q \equiv 3 \mod 4$. Plaintexts will be vectors in k^n. Let $K \cong k[y]/\langle g(y)\rangle$ be a degree $n+l$ extension of k, where l is such that $n+l$ is odd. The public key will be built up from the following maps:

- $\varphi\colon K \to k^{n+l}$, the vector space isomorphism given by
$$a_{n+l}y^{n+l-1} + \cdots + a_2 y + a_1 \mapsto (a_{n+l}, \ldots, a_1)$$

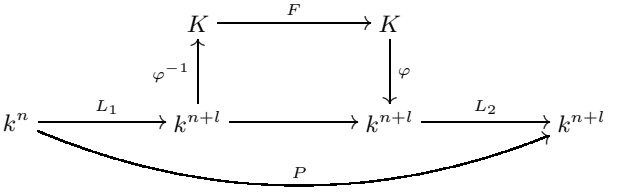

Fig. 2. The Square system

- $F\colon K \to K$, given by $F(X) = X^2$
- $L_1\colon k^n \to k^{n+l}$, an injective affine map
- $L_2\colon k^{n+l} \to k^{n+l}$, an invertible affine map.

From these we construct the public key

$$P = L_2 \circ \varphi \circ F \circ \varphi^{-1} \circ L_1.$$

P will be an $(n+l)$-tuple of quadratic polynomials

$$P(x_1, \ldots, x_n) = \begin{pmatrix} p_1(x_1, \ldots, x_n) \\ p_2(x_1, \ldots, x_n) \\ \vdots \\ p_{n+l}(x_1, \ldots, x_n) \end{pmatrix}.$$

This can be thought of as a C^* system over odd characteristic with $\theta = 0$ and an embedding L_1.

Encryption of a plaintext $(m_1, \ldots, m_n) \in k^n$ is obtained by computing $c_j = p_j(m_1, \ldots, m_n)$ for $j = 1, \ldots, n+l$.

Decryption of a ciphertext $(c_1, \ldots, c_{n+l}) = P(m_1, \ldots, m_n) \in k^{n+l}$ is performed as follows: first, let $Y = \varphi^{-1} \circ L_2^{-1}(c_1, \ldots, c_{n+l})$. Then solve $X^2 = Y$. By choosing $q \equiv 3 \mod 4$ and $n+l$ odd, we ensure that $|K| \equiv 3 \mod 4$. This allows us to use the fact that

$$X = \pm Y^{\frac{q^{n+l}+1}{4}}. \tag{4}$$

This gives two solutions. Since L_1 is affine, in general only one of them will be in the image of $\varphi^{-1} \circ L_1$. The preimage of this solution under $\varphi^{-1} \circ L_1$ will be $(m_1 \ldots, m_n)$.

This simple method to find a preimage under the core map is a major advantage over traditional characteristic-2 HFE systems which require the decryptor to solve a univariate equation of high degree (using Berlekamp's algorithm or its improvements). In fact, encryption and decryption are quite fast. See Table 2 for a summary of times for various choices of q and n. For all experimental data in this paper, we used an Intel(R) Core(TM)2 2.40 GHz processor with 1.99 GB of memory installed.

Table 2. Encryption and decryption times, in seconds, for Square systems. 10 public keys tested and 100 messages encrypted and decrypted per key.

q	n	l	Average Encrypt Time	Average Decrypt Time
31	20	3	0.00022	0.001527
31	32	3	0.00033	0.006781
31	34	3	0.000423	0.003651
43	20	3	0.000234	0.001783
43	34	3	0.000495	0.008135

4 Security Analysis

Let us now present our case for the security of this design. We will explain our motivation and then dig into the specific reasons why each of the aforementioned attacks does not work.

But first, we provide a more thorough comparison to the very similar system proposed by Patarin [20]. His system D^* also uses a square core map, and Patarin broke D^* himself. He did so by finding a way to recover "big field" multiplications without having the big field. As we will see below in Section 4.4, the embedding makes it very hard to recover the multiplicative structure of K. In particular, Patarin's attack on D^* relies on the ability to find pairs of linear maps (C, D) such that for all $\boldsymbol{x_1},\ \boldsymbol{x_2} \in k^n$,

$$C(F(\boldsymbol{x_1} + \boldsymbol{x_2}) - F(\boldsymbol{x_1} - \boldsymbol{x_2})) = F(D(\boldsymbol{x_1}) + \boldsymbol{x_2}) - F(D(\boldsymbol{x_1}) - \boldsymbol{x_2}).$$

When L_1 is invertible, the collection of such pairs forms a vector space of dimension at least n (exactly n according to [20]) which is required for Patarin's attack. In our case, L_1 is a map $k^n \to k^{n+l}$ and thus cannot be invertible.

4.1 Motivation for the Design

All of the ideas used in Square have been seen before; what makes this system novel is that these ideas are combined in such a way that they work.

First, the use of odd characteristic was shown to be a good idea for thwarting algebraic attacks in [6]. This seems to be because the attacker knows that a plaintext $(x_1, \ldots, x_n)$ satisfies not only the public key equations (2) but also the $\mathbb{F}_q$ field equations $x_i^q - x_i = 0$. When q is small, this additional information is very useful, and feeding the field equations into MAGMA along with (2) allows for more efficient solving. However, as discussed in [6], for larger q the field equations do not simplify the algorithm and in fact F_4 runs faster without them.

Secondly, the use of a low-degree core map was inspired by [2], where an odd-characteristic signature scheme with low-degree core map was proposed. It was natural to ask if this idea could be used for encryption as well. However, the signature scheme in [2] uses a vinegar construction, which is not well-suited for encrytion.

This led to the third modification, that of an embedding. Such a tool has been mentioned, but dismissed until recently when a reformulated version of the idea showed promise [5].

Each of these modifications would be weak on their own, but we will make the case below that combined, they are quite strong.

4.2 Linearization Equations Attack

Note that when $\theta = 0$, the equation $XY^{q^\theta} - X^{q^{2\theta}}Y = 0$ that Patarin discovered becomes simply $XY = XY$. So our system should not satisfy any linearization equations other than the trivial one satisfied by any map. In other words, the space of linearization equations should have dimension 0 rather than n. Since the algebraic attack described in 2.3 detects linearization equations, the fact that algebraic attacks are not particularly effective, as descibed below, is an indication that this is in fact the case. To be sure, we did experiments to find the dimension of the space of linearization equations. For each of the 2500 keys we tested, the dimension of this space was 0.

4.3 Algebraic Attack

In order to test the system's resistance to algebraic attacks, we performed the following experiments: We generated a public key and used it to encrypt 50 messages. We then used MAGMA's implementation of F_4 to solve the equations defined by the public key and ciphertext as in 2. We did this for two public keys per choice of parameters q, n, and l. We found that the public key polynomials behave similar to systems of the same size with random polynomials. A sampling of our results are in Table 3.

Plotting this data reveals a clear exponential trend in both time and memory usage as n increases. In fact, linear least-squares approximation on the log of the data has a high correlation. See Figure 3. This trend leads us to believe that $n \geq 33$ is a good choice for a practical system.

Table 3. Average algebraic attack time in seconds and memory usage in MB. $q = 31$

n	l	sec	MB
2	1	0.000	6
3	1	0.000	6
4	1	0.001	6
5	1	0.002	6
6	1	0.006	6
7	1	0.024	6
8	1	0.129	6
9	1	0.696	8
10	1	4.747	13
11	1	27.423	30
12	1	215.678	32
13	1	1330.657	325

n	l	sec.	MB
3	2	0.000	6
4	2	0.001	6
5	2	0.001	6
6	2	0.003	6
7	2	0.008	6
8	2	0.021	6
9	2	0.165	7
10	2	0.818	9
11	2	3.137	14
12	2	17.294	30
13	2	83.516	76
14	2	431.894	262

n	l	sec	MB
4	3	0.000	6
5	3	0.001	6
6	3	0.003	6
7	3	0.007	6
8	3	0.019	6
9	3	0.115	7
10	3	0.406	8
11	3	2.223	12
12	3	15.924	26
13	3	83.433	68
14	3	206.602	137
15	3	2218.500	632

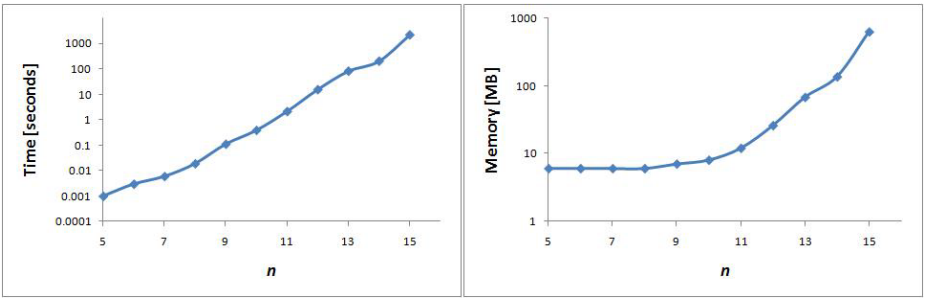

Fig. 3. Algebraic attack for q=31, $l = 3$, varying n

To inform our choice of q, we tested the effect of changing q while fixing n and l. Our results can be found in Table 4.3 and Figure 4.3. These attacks were done without using field equations $x_i^q - x_i = 0$ for the reasons described in Section 4.1. From this data we see that beyond small values of q, the size of the field does not seem to impact F_4's running time or memory usage. This was expected in light of the results of [2] and [6], and justifies the choice of $q = 31$ for a practical system. Another reason to choose $q = 31$ is that such a choice makes good use of memory, in the sense that the elements of k will require 5 bits to be stored and any larger field will require more bits to store an element.

Table 4. Algebraic attack for $n = 12$, $l = 3$ and varying q

q	Average F_4 Running Time	Memory Usage
3	0.011	6
7	7.082	22
11	9.092	24
19	16.502	26
23	16.308	26
31	15.973	26
43	15.685	26
47	15.618	26

4.4 SFlash-Style Attack

Our public keys are maps $k^n \to k^{n+l}$. We have constructed the public key P by emebedding the space of plaintexts into a larger space, but one could imagine P as coming from a larger C^* scheme by setting the last l components of the input to 0. Effectively, P is the public key for a (non-embedded) C^* scheme with all monomials involving $x_{n+1}, \ldots,$ or x_{n+l} deleted.

From this point of view, it is important to study the attack on SFlash since its purpose was to recover missing coefficients of the public key (ie, the coefficients

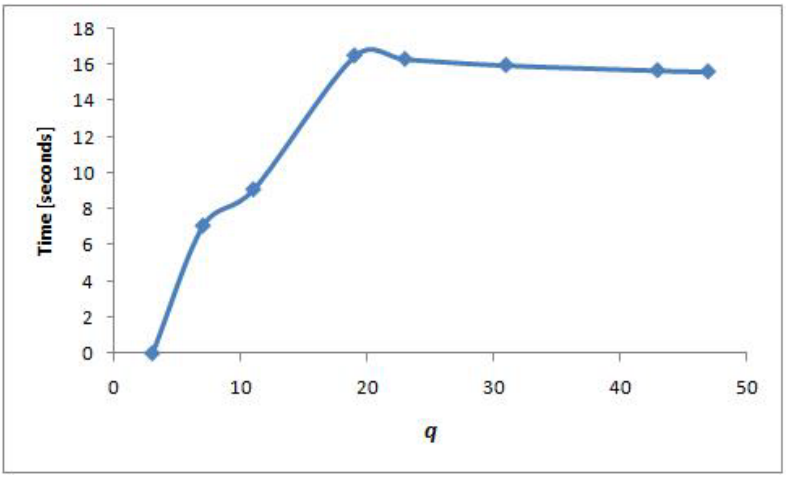

Fig. 4. Algebraic attack for $n = 12$, $l = 3$ and varying q

of the deleted polynomials). Since both SFlash and Square stem from C^*, the differential property (3) exploited by Dubois et al still holds.

While in the case of SFlash this property yields linear equations for the attacker, in our case the property gives us quadratic conditions. In fact, the resulting system of quadratic equations is larger than the system of public key equations. It seemed unlikely that this attack would work. However to be sure, we applied F_4 to these systems of quadratic equations. We quickly realized that the Gröbner basis attack finds no special properties of these systems and takes as long as one might expect. In the "baby" case of $q = 5$, $n = 2$, $l = 1$, this method generates 27 equations in 30 variables which were beyond the abilities of our computer to solve. It stands to reason that with realistic parameters such as $q = 31$, $n = 34$, $l = 3$ these equations will pose no threat.

4.5 Kipnis-Shamir Style Attacks

The attack that Kipnis and Shamir used against HFE depends on finding a combination of matrices derived from the public key which has minimal rank. In our case, we may use the same idea as for the SFlash-style attack and consider the public key as a piece of a C^* public key of $n+l$ variables. In this setting, the rank of the analogous combination of matrices will be 1. As in Kipnis and Shamir's paper [14], we could try to determine the proper combination by finding a basis of the null space of these matrices. The difference between the two systems is that in HFE this yields quadratic equations, while in Square the "missing" coefficients cause the equations to be cubic. One could reduce to quadratic equations, but not without using additional variables.

The HFE attackers claimed that the MinRank problem could be solved in the specific circumstances of HFE [14]. Since that time, doubt has been cast over

the original efficiency claims [13]. We tested this attack in the Square case and found that even with the "baby" case of $q = 5$, $n = 2$, $l = 1$, the system that arises from this attack involves 18 cubic equations in 14 variables and solving it exceeded the memory of our computer. We also tried using 2×2 minors as a way to generate equations from the rank condition, but this yields quartic equations with a savings of only 2 variables. This method also exceeded the memory of our computer with $q = 5$, $n = 2$, $l = 1$. Considering this, it is not plausible that such an attack would be dangerous for realistic parameter choices.

5 Parameter Suggestions and Implementations

Based on the security analysis above, an Square system with the following parameter choices will be viable:

Square-34
- $q = 31$
- $n = 34$
- $l = 3$
- Average encryption time: 0.000423 seconds
- Average decryption time: 0.003651 seconds
- Public key size: 15 KB
- Best known attack: $> 2^{80}$ computations.

A system with these parameters will be secure and have relatively fast decryption using the power forumla mentioned in Section 3. Of course, these numbers are very conservative. If we are concerned for speed and not so much for portability, there are ways to get the implementation *much* faster.

Square Roots. Since we are always dealing with a pre-determined field, pre-computation is not a problem. If (field size) $- 1 = 2^a \times$ (odd number o), taking square roots is always possible via raising to the power of $\frac{o-1}{2}$ using a pre-computed table with 2^a elements (the Tonelli-Shanks method [21]).

Consider the case $n = 51$, $l = 3$. Since $n + l = 54$ is even, during decryption we cannot use the formula $X = \pm Y^{\frac{q^{n+l}+1}{4}}$ as in the Square-34 case. However, via raising to a power of $\frac{1}{128}(31^{54} - 65)$, we see that square roots in 1 (mod 4) and 3 (mod 4) field sizes does not differ much, since the number of "total multiplications" does not increase much (at most 1.5× in all our experiments).

Choice of Field. We should choose a field with a "good" irreducible polynomial. For example, for $k = \mathbb{F}_{31}$, only certain $(x^h - \alpha)$ can be irreducible which makes things very fast. Values of h above 34 that is of interest to us here are 45 and 54. and in fact $\mathbb{F}_{31^{45}} \cong k[x]/(x^{45} - 3)$, $\mathbb{F}_{31^{54}} \cong k[x]/(x^{54} - 3)$. Which in and of itself is very simple. But there is a further trick as in the following:

Tower Fields. Tower fields are very common in MPKCs of characteristic two. Here we may use also this trick and use

$$\mathbb{F}_{31^{15}} \cong k[t]/(t^{15}-3), \quad \mathbb{F}_{31^{45}} \cong \mathbb{F}_{31^{15}}[x]/(x^3-t); \text{ or}$$
$$\mathbb{F}_{31^{18}} \cong k[t]/(t^{18}-3), \quad \mathbb{F}_{31^{54}} \cong \mathbb{F}_{31^{18}}[x]/(x^3-t).$$

Other Techniques. We should delay modulo operations by checking for number sizes appropriately and do not do a modulo-q for as long as we can help it. We should also write the relevant routines in assembly.

Usually the multiplication in an extension of this size would be unwieldy, but due to the tricks mentioned above, "big field" operations have a low computational cost. Our tests show that we can achieve a hundred-fold speed increase.

Hence, we propose alternate parameter choices which can be made even faster than the Square described above as in Tab. 1.

6 Conclusion

In this paper we analyzed a new multivariate encryption scheme that has great promise. In a sense, Square continues the bloodline of the original C^* scheme but our arguments and results above suggest that our system avoids the pitfalls of its predecessors. We showed, via experimental data when possible, that attacks against similar systems are not effective against a reasonably-sized Square system.

We gave parameter choices for a secure system Square-34. We also proposed larger but even more efficient implementations Square-45 and Square-54. Part of our future work will be optimizations of Square systems with other parameters.

References

1. Ars, G., Faugère, J.-C., Imai, H., Kawazoe, M., Sugita, M.: Comparison between XL and gröbner basis algorithms. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 338–353. Springer, Heidelberg (2004)
2. Baena, J., Clough, C., Ding, J.: Square-Vinegar signature scheme. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299. Springer, Heidelberg (2008)
3. Buss, J.F., Frandsen, G.S., Shallit, J.O.: The Computational Complexity of Some Problems of Linear Algebra. Journal of Computer and System Sciences 58(3), 572–596 (1999)
4. Diene, A., Ding, J., Gower, J.E., Hodges, T.J., Yin, Z.: Dimension of the linearization equations of the Matsumoto-Imai cryptosystems. In: Ytrehus, Ø. (ed.) WCC 2005. LNCS, vol. 3969, pp. 242–251. Springer, Heidelberg (2006)
5. Ding, J., Dubois, V., Yang, B.-Y., Chen, O.C.-H., Cheng, C.-M.: Could SFLASH be repaired? In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part II. LNCS, vol. 5126, pp. 691–701. Springer, Heidelberg (2008)
6. Ding, J., Schmidt, D., Werner, F.: Algebraic attack on HFE revisited. In: Wu, T.-C., Lei, C.-L., Rijmen, V., Lee, D.-T. (eds.) ISC 2008. LNCS, vol. 5222, pp. 215–227. Springer, Heidelberg (2008)

7. Dubois, V., Fouque, P.-A., Shamir, A., Stern, J.: Practical cryptanalysis of SFLASH. In: Menezes, A. (ed.) CRYPTO 2007. LNCS, vol. 4622, pp. 1–12. Springer, Heidelberg (2007)
8. eBACS: ECRYPT benchmarking of cryptographic systems, http://bench.cr.yp.to
9. Faugére, J.-C.: A new efficient algorithm for computing Gröbner bases (F_4). J. Pure Appl. Algebra 139(1-3), 61–88 (1999); effective Methods in algebraic geometry (Saint-Malo) (1998)
10. Faugère, J.-C.: A new efficient algorithm for computing Gröbner bases without reduction to zero (F_5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, pp. 75–83. ACM, New York (2002)
11. Faugère, J.-C., Joux, A.: Algebraic cryptanalysis of Hidden Field Equation (HFE) cryptosystems using Gröbner bases. In: Boneh, D. (ed.) CRYPTO 2003. LNCS, vol. 2729, pp. 44–60. Springer, Heidelberg (2003)
12. Garey, M.R., Johnson, D.S., et al.: Computers and Intractability: A Guide to the Theory of NP-completeness. W.H Freeman, San Francisco (1979)
13. Jiang, X., Ding, J., Hu, L.: Public Key Analysis-Kipnis-Shamir Attack on HFE Revisited. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Inscrypt 2007. LNCS, vol. 4990, pp. 399–411. Springer, Heidelberg (2008)
14. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE public key cryptosystem by relinearization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 19–30. Springer, Heidelberg (1999)
15. The MAGMA computational algebra system home page, http://magma.maths.usyd.edu.au/magma
16. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
17. Patarin, J.: Cryptanalysis of the Matsumoto and Imai public key scheme of EUROCRYPT 1988. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 248–261. Springer, Heidelberg (1995)
18. Patarin, J.: Hidden fields equations (HFE) and Isomorphisms of Polynomials (IP): Two new families of asymmetric algorithms. In: Maurer, U.M. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996)
19. Patarin, J., Courtois, N.T., Goubin, L.: FLASH, a fast multivariate signature algorithm. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 298–307. Springer, Heidelberg (2001)
20. Patarin, J., Goubin, L.: Trapdoor one-way permutations and multivariate polynominals. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 356–368. Springer, Heidelberg (1997)
21. Shanks, D.: Five numbertheoretic algorithms. In: Thomas, R.S.D., Williams, H.C. (eds.) Proceedings of the Second Manitoba Conference on Numerical Mathematics, pp. 51–70 (1972)