

# Cryptanalysis of the HFE Public Key Cryptosystem

Aviad Kipnis

NDS Technologies, Jerusalem, Israel

Adi Shamir

Dept. of Applied Math.

The Weizmann Institute of Science

Rehovot 76100, Israel

## Abstract

The RSA public key cryptosystem is based on a single modular equation in one variable. A natural generalization of this approach is to consider systems of several modular equations in several variables. In this paper we consider Patarin's Hidden Field Equations (HFE) scheme, which is believed to be one of the strongest schemes of this type. We represent the published system of multivariate polynomials by a single univariate polynomial of a special form over an extension field, and use it to reduce the cryptanalytic problem to a system of  $\epsilon m^2$  quadratic equations in  $m$  variables over the extension field. Finally, we develop a new relinearization method for solving such systems for any constant  $\epsilon > 0$  in expected polynomial time. The new type of attack is quite general, and in a companion paper we use it to break other multivariate algebraic schemes, such as the Dragon encryption and signature schemes.

## 1 Introduction

The problem of developing new public key encryption and signature schemes had occupied the cryptographic research community for the last 20 years. A particularly active line of research was based on the observation that solving systems of modular multivariate polynomial equations is NP-complete. Consider, for example, a public encryption key consisting of  $n$  random quadratic polynomials in  $n$  variables over the two element field  $\mathbf{F}_2$ . To encrypt an  $n$  bit cleartext, you assign each bit to a variable, and evaluate the  $n$  quadratic polynomials modulo 2. To decrypt this ciphertext, you use it as a right

hand side and solve the resultant system of  $n$  quadratic equations in  $n$  unknowns. When  $n = 100$ , the encryption process is extremely fast, while the decryption process (by an eavesdropper) seems to be completely infeasible.

The system of equations must contain a trapdoor, whose knowledge makes it possible to solve the system of equations efficiently for any right hand side. The main difference between the various multivariate schemes is the type of trapdoor structure they embed into the published polynomials.

An early example of a multivariate signature scheme was developed by Ong Schnorr and Shamir [OSS84], and was broken shortly afterwards by Pollard and Schnorr [PS87]. Fell and Diffie [FD85] published another multivariate scheme, but observed that it was insecure for any practical key size. A different type of trapdoor was developed by Matsumoto and Imai [MI88], but their scheme was shown to be insecure in Patarin [P95]. Shamir [S93] proposed two multivariate schemes modulo large  $n = pq$ , which were shown to be insecure by Coppersmith Stern and Vaudenay [CSV97]. In an attempt to revive the field, Patarin had developed several new types of trapdoors. The simplest of his new constructions was the Oil and Vinegar signature scheme [P97], which was broken by Kipnis and Shamir [KS98]. A more secure construction was the Dragon encryption and signature schemes, described in Patarin [P96b] and Koblitz [K98]. A simplified version of this scheme (called Little Dragon) was broken by Coppersmith and Patarin, but the original Dragon scheme remained unbroken. The Hidden Field Equations (HFE) was published in Patarin [P96a], and conjectured by its author to be the strongest among his various constructions. In spite of extensive cryptanalytic effort, no attacks on the HFE scheme had been published so far.

In this paper we develop a new cryptanalytic approach and use it to attack both the HFE scheme (as shown in this paper) and the Dragon scheme (as shown in a companion paper, due to space limitations). The asymptotic complexity of the attack is polynomial (when some of the parameters grow to infinity while others are kept fixed), but the basic implementation described in this paper may be impractical for sufficiently large keys. Both the scheme and the attack can be enhanced in numerous ways, and thus it is too early to decide whether some variant of the HFE scheme can survive an optimized version of the attack.

The attack is based on the observation that any given system of  $n$  multivariate polynomials in  $n$  variables over a field  $\mathbf{F}$  can be represented by a single univariate polynomial of a special form over  $\mathbf{K}$  which is an extension field of degree  $n$  over  $\mathbf{F}$ . We analyse the effect of the trapdoor hiding op-

erations on this representation, and use it in order to translate the original problem of solving  $n$  quadratic equations in  $n$  variables over the small field  $\mathbf{F}$  into a new problem of solving a system of  $\epsilon m^2$  quadratic equations in  $m$  variables over the large field  $\mathbf{K}$ , where  $m$  is a small multiple of  $n$ . The standard linearization technique for solving such systems is to replace any product of variables  $x_i x_j$  by a new variable  $y_{ij}$ , and to solve the resultant system of  $\epsilon m^2$  linear equations in the  $m^2/2$  new  $y_{ij}$  variables. However, in our attack  $\epsilon < 0.5$ , and thus the linearization technique creates exponentially many parasitic solutions which do not correspond to solutions of the original quadratic equations. We overcome this problem by developing a general new technique (called relinearization) which is expected to solve random systems of equations of this type in polynomial time for any fixed  $\epsilon > 0$ . Since no previously published technique could handle such systems, we expect the relinearization technique to have additional applications in cryptanalysis, algorithm design, and operations research.

## 2 The HFE Scheme

The HFE encryption algorithm was presented by Jacques Patarin at Eurocrypt '96. It uses a small field  $\mathbf{F}$  with  $q$  elements (the recommended choice is  $q = 2$ ), and a large extension field  $\mathbf{K}$  of degree  $n$  over  $\mathbf{F}$  (the recommended choice is  $n = 128$ , yielding a field  $\mathbf{K}$  with  $2^{128}$  elements). The field  $\mathbf{K}$  can be viewed as a vector space of dimension  $n$  over  $\mathbf{F}$ , and the mapping between the two representations is defined by a basis of  $n$  elements  $\omega_0, \dots, \omega_{n-1}$  in  $\mathbf{K}$  via  $\sum_{i=0}^{n-1} x_i \omega_i \leftrightarrow (x_0, \dots, x_{n-1})$ .

To construct his public key, the user picks a random univariate polynomial  $P(x)$  over  $\mathbf{K}$  of the form

$$P(x) = \sum_{i=0}^{r-1} \sum_{j=0}^{r-1} p_{ij} x^{q^i + q^j}$$

where  $r$  is some small constant which guarantees that the degree of  $P(x)$  is bounded by several thousand (the highest recommended value of  $r$  is 13, which for  $q = 2$  gives rise to a polynomial of degree 8192). The bound on the degree is required in order to make it possible to invert  $P(x)$  efficiently (e.g., by using Berlekamp's probabilistic algorithm for solving a univariate polynomial equation over a finite fields).

The univariate polynomial  $P$  over  $\mathbf{K}$  can be expressed as a system of  $n$  multivariate polynomials  $P_1, \dots, P_{n-1}$  in the  $n$  variables  $x_0, \dots, x_{n-1}$  over

**F.** The restricted choice of exponents in  $P$  guarantees that all the  $P'_i$ s are homogeneous quadratic polynomials. The trapdoor consists of two random invertible linear transformations  $S$  and  $T$  over  $n$ -tuples of values in  $\mathbf{F}$ . The user applies  $S$  to the inputs and  $T$  to the outputs of the  $n$  multivariate polynomials, and publishes the evaluated homogeneous quadratic polynomials in  $n$  variables, denoted by  $G_0, \dots, G_{n-1}$ .

To solve the published system of quadratic equations with a given ciphertext as the right hand side, the user applies  $T^{-1}$  to the ciphertext, interprets the result as an element of  $\mathbf{K}$ , solves his secret univariate polynomial with this right hand side, and applies  $S^{-1}$  to the components of the solution. The attacker cannot use this procedure since he does not know the  $S$  and  $T$  transformations. These mixing operations have natural interpretation over  $\mathbf{F}$  but not over  $\mathbf{K}$ , and it is not clear a priori that the  $n$  published polynomials over  $\mathbf{F}$  can be described by a single univariate polynomial  $G$  over  $\mathbf{K}$ . Even if it exists, it may have an exponential number of coefficients, and even if it is sparse, it may have an exponentially large degree which makes it practically unsolvable.

**Remark:** In this extended abstract we simplify the original HFE scheme in several inessential ways. In particular, we consider only homogeneous polynomials (the attacker can ignore lower degree monomials), and assume that the representation of  $\mathbf{K}$  over  $\mathbf{F}$  is fixed (by using a different representation, the attacker obtains a different but equally useful version of the secret key).

### 3 Univariate Representations of Systems of Multivariate Polynomials

The starting point of our attack is the observation that ANY system of  $n$  multivariate polynomials of bounded degree  $d$  in  $n$  variables over a field  $\mathbf{F}$  can be represented as a single sparse univariate polynomial of a special form over an extension field  $\mathbf{K}$  of degree  $n$  over  $\mathbf{F}$ .

We first consider the case of linear multivariate mappings. The mapping  $x \rightarrow x^q$  is a linear function over  $\mathbf{K}$ , and thus any mapping of the form  $x \rightarrow \sum_{i=0}^{n-1} a_i x^{q^i}$  for fixed coefficients  $a_0, \dots, a_{n-1}$  in  $\mathbf{K}$  is also a linear mapping. We need the converse of this result:

**Lemma 3.1** : *Let  $A$  be a linear mapping from  $n$ -tuples to  $n$ -tuples of values in  $\mathbf{F}$ . Then there are coefficients  $a_0, \dots, a_{n-1}$  in  $\mathbf{K}$  such that for any two  $n$*

tuples  $(x_0, \dots, x_{n-1})$  and  $(y_0, \dots, y_{n-1})$  of elements in  $\mathbf{F}$ ,  $(y_0, \dots, y_{n-1}) = A(x_0, \dots, x_{n-1})$  if and only if  $y = \sum_{i=0}^{n-1} a_i x^i$ , where  $x = \sum_{i=0}^{n-1} x_i \omega_i$  and  $y = \sum_{i=0}^{n-1} y_i \omega_i$  are the elements of  $\mathbf{K}$  which correspond to the two  $n$ -tuples over  $\mathbf{F}$ .

**Proof:** There are  $q^{(n^2)}$   $n \times n$  matrices over  $\mathbf{F}$  and  $(q^n)^n$  sums of  $n$  monomials over  $\mathbf{K}$ , and thus the number of linear mappings and the number of polynomials of this form is identical. Each polynomial represents some linear mapping, and two distinct polynomials cannot represent the same mapping since their difference would be a non zero polynomial of degree  $q^{n-1}$  with  $q^n$  roots in a field. Consequently, each linear mapping is represented by some univariate polynomial of this type over the extension field.  $\square$

We now generalize this characterization from linear functions to any system of multivariate polynomials:

**Lemma 3.2** *Let  $P_0(x_0, \dots, x_{n-1}), \dots, P_{n-1}(x_0, \dots, x_{n-1})$  be any set of  $n$  multivariate polynomials in  $n$  variables over  $\mathbf{F}$ . Then there are coefficients  $a_0, \dots, a_{q^n-1}$  in  $\mathbf{K}$  such that for any two  $n$  tuples  $(x_0, \dots, x_{n-1})$  and  $(y_0, \dots, y_{n-1})$  of elements in  $\mathbf{F}$ ,  $y_j = P_j(x_0, \dots, x_{n-1})$  for all  $0 \leq j \leq n-1$  if and only if  $y = \sum_{i=0}^{q^n-1} a_i x^i$ , where  $x = \sum_{i=0}^{n-1} x_i \omega_i$  and  $y = \sum_{i=0}^{n-1} y_i \omega_i$  are the elements of  $\mathbf{K}$  which correspond to the two vectors over  $\mathbf{F}$ .*

**Proof:** We can again use counting arguments, but we prefer a direct proof which better characterizes the generated univariate polynomials. Without loss of generality, we can assume that the first basis element is  $\omega_0 = 1$ . The mapping  $(x_0, \dots, x_{n-1}) \rightarrow (x_i, 0, \dots, 0)$  over  $\mathbf{F}$  is linear, and thus has a univariate polynomial representation over  $\mathbf{K}$ . To represent the mapping  $(x_0, \dots, x_{n-1}) \rightarrow (\prod_{i=0}^{n-1} x_i^{c_i}, 0, \dots, 0)$ , multiply all the univariate polynomials which represent the mappings  $(x_0, \dots, x_{n-1}) \rightarrow (x_i, 0, \dots, 0)$ , with their multiplicities  $c_i$  (note that this can only be done at the first coordinate, which corresponds to the basis element  $\omega_0 = 1$ ; at any other coordinate  $k$  we would get a power of  $\omega_k$  which would spread the resultant monomial all over the vector). By summing the univariate polynomial representations of such monomials with appropriate coefficients we can represent the mapping defined by any multivariate polynomial in the first coordinate of the vector, and zeroes elsewhere. To move the multivariate polynomial to the  $k$ -th coordinate of the vector, we multiply all the coefficients of its univariate representation (which are elements of  $\mathbf{K}$ ) by  $\omega_k$ . Finally, to represent a system of  $n$  (unrelated) multivariate polynomials at the  $n$  coordinates of

the vector, we construct a representation of each polynomial at the first coordinate, shift it to its proper coordinate and add all the resultant univariate polynomials.  $\square$ .

An important corollary of this proof is:

**Lemma 3.3** : *Let  $\mathcal{C}$  be any collection of  $n$  homogeneous multivariate polynomials of degree  $d$  in  $n$  variables over  $\mathbf{F}$ . Then the only powers of  $x$  which can occur with non-zero coefficients in its univariate polynomial representation  $G(x)$  over  $\mathbf{K}$  are sums of exactly  $d$  (not necessarily distinct) powers of  $q$ :  $q^{i_1} + q^{i_2} + \dots + q^{i_d}$ . If  $d$  is a constant, then  $G(x)$  is sparse, and its coefficients can be found in polynomial time.*

**Proof:** Mappings defined by a single variable are linear functions, and thus can be represented as the sum of monomials of the form  $x^{q^i}$ , and each monomial contains a single power of  $q$ . When we multiply  $d$  such polynomials and evaluate the result, we get only powers of  $x$  which are the sums of exactly  $d$  powers of  $q$ . Since  $G(x)$  is the sum of such polynomials (multiplied by constants from  $\mathbf{K}$ ), the same is true for  $G(x)$ .

The degree of  $G(x)$  over  $\mathbf{K}$  can be exponentially large, but at most  $O(n^d)$  of its coefficients can be non-zero, and for any fixed value of  $d$  this is a polynomial number. Once we know that a sparse univariate polynomial representation exists, we can find its coefficients in polynomial time by interpolation based on sufficiently many input/output pairs.  $\square$

The problem of solving a system of multivariate quadratic equations over a finite field is known to be NP complete. This lemma implies that the problem of solving a single univariate polynomial equation over a finite field is also NP complete, if the polynomial is represented by the list of its non zero coefficients. If the polynomial is represented by the list of ALL its coefficients, the problem can be solved in probabilistic polynomial time by Berlekamp's algorithm.

Consider the published system of quadratic polynomials  $G_0, \dots, G_{n-1}$  in  $x_0, \dots, x_{n-1}$ . Each polynomial can be written as the quadratic form  $\bar{x}G_i\bar{x}^t$  where  $G_i$  is an  $n \times n$  matrix of coefficients<sup>1</sup>,  $\bar{x}$  is the row vector of variables  $(x_0, \dots, x_{n-1})$ , and  $\bar{x}^t$  is its transpose. However, our attack does not use this standard representation. Instead, it uses Lemma 3.3 to efficiently find

---

<sup>1</sup>The matrix representation of quadratic forms is not unique, and has to be symmetrized by averaging the matrix and its transpose. In fields of characteristic 2 we just add the matrix and its transpose, (since we cannot divide by 2), and use the result. More details on these fine points will be given in the final version of the paper.

the following representation of the public key:

$$G(x) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g_{ij} x^{q^i + q^j} = \underline{x} G \underline{x}^t \text{ where } G = [g_{ij}] \text{ and } \underline{x} = (x^{q^0}, x^{q^1}, \dots, x^{q^{n-1}})$$

Note that this is an unusual type of quadratic form since the vector  $\underline{x}$  consists of related rather than independent variables, and that  $\bar{x}$  is a vector of elements from  $\mathbf{F}$  whereas  $\underline{x}$  is a vector of elements from  $\mathbf{K}$ . It is this representation which makes it possible to analyse the secret hiding operations in a clean mathematical form.

## 4 The effect of $S$ and $T$ on $P$

Due to their special form, both the original polynomial  $P(x)$  over  $\mathbf{K}$  chosen by the user and the new polynomial  $G(x)$  over  $\mathbf{K}$  derived by the cryptanalyst from the public key can be represented by the (non standard) quadratic forms  $\underline{x} P \underline{x}^t$  and  $\underline{x} G \underline{x}^t$ . The linear mappings  $S$  and  $T$  can be represented as univariate polynomials, and thus the public key is represented by the univariate polynomial composition  $G(x) = T(P(S(x)))$  over  $\mathbf{K}$ . We rewrite this equation as  $T^{-1}(G(x)) = P(S(x))$ , where  $S$  has the form  $S(x) = \sum_{i=0}^{n-1} s_i x^{q^i}$  and  $T^{-1}$  (which is also a linear mapping) has the form  $T^{-1}(x) = \sum_{i=0}^{n-1} t_i x^{q^i}$ . Our goal now is to study the effect of the polynomial compositions  $T^{-1}(G(x))$  and  $P(S(x))$  on the matrices of their (non standard) quadratic form representations.

**Theorem 4.1 :** *The matrix of the quadratic form in  $\underline{x}$  which represents the polynomial composition  $T^{-1}(G(x))$  is  $\sum_{k=0}^{n-1} t_k G^{*k}$  where  $G^{*k}$  is obtained from the  $n \times n$  matrix representation of  $G$  by raising each one of its entries to the power  $q^k$  in  $\mathbf{K}$ , and cyclically rotating forwards by  $k$  steps both the rows and the columns of the result. The matrix of the quadratic form in  $\underline{x}$  which represents the polynomial composition  $P(S(x))$  is  $W P W^t$  in which  $W = [w_{ij}]$  is an  $n \times n$  matrix defined by  $w_{ij} = (s_{j-i})^{q^i}$ , where  $j - i$  is computed modulo  $n$ .*

**Proof (Sketch):** The polynomial representation of  $T^{-1}(x)$  is  $\sum_{k=0}^{n-1} t_k x^{q^k}$  and the polynomial representation of  $G(x)$  is  $\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g_{ij} x^{q^i + q^j}$ . Their polynomial composition can be evaluated by using the fact that raising sums

to the power  $q^i$  is a linear operation:

$$T^{-1}(G(x)) = \sum_{k=0}^{n-1} t_k \left( \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g_{ij} x^{q^i+q^j} \right)^{q^k} = \sum_{k=0}^{n-1} t_k \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (g_{ij})^{q^k} x^{(q^i+q^j)q^k}$$

The exponents of  $q$  can be reduced modulo  $n$  since  $x^{q^n} = x^{q^0} = x$ , and the summation indices can be cyclically rotated if they are computed modulo  $n$ :

$$T^{-1}(G(x)) = \sum_{k=0}^{n-1} t_k \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (g_{ij})^{q^k} x^{q^{i+k}+q^{j+k}} = \sum_{k=0}^{n-1} t_k \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} (g_{i-k,j-k})^{q^k} x^{q^i+q^j}$$

The matrix of the quadratic form representation of this polynomial in terms of  $\underline{x}$  is exactly  $G' = \sum_{k=0}^{n-1} t_k G^{*k}$ , where the  $(i, j)$ -th entry of  $G^{*k}$  is  $g_{i-k,j-k}^{q^k}$ , as specified.

The proof of the other type of composition is similar:

$$P(S(x)) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{ij} \left( \sum_{k=0}^{n-1} s_k x^{q^k} \right)^{(q^i+q^j)} = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{ij} \left( \sum_{u=0}^{n-1} s_u x^{q^u} \right)^{q^i} \left( \sum_{v=0}^{n-1} s_v x^{q^v} \right)^{q^j}$$

Again we use linearity and cyclic index shifting to evaluate  $P(S(x))$  as:

$$\sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{ij} \left( \sum_{u=0}^{n-1} s_u^{q^i} x^{q^{u+i}} \right) \left( \sum_{v=0}^{n-1} s_v^{q^j} x^{q^{v+j}} \right) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p_{ij} \left( \sum_{u=0}^{n-1} s_{u-i}^{q^i} x^{q^u} \right) \left( \sum_{v=0}^{n-1} s_{v-j}^{q^j} x^{q^v} \right)$$

By rearranging the order of the summation and the multiplied terms we get:

$$P(S(x)) = \sum_{u=0}^{n-1} \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \sum_{v=0}^{n-1} x^{q^u} s_{u-i}^{q^i} p_{ij} s_{v-j}^{q^j} x^{q^v} = \underline{x} W P W^t \underline{x}^t$$

where  $W$  is the specified matrix.  $\square$

## 5 Recovering the Secret Key from the Public Key

The attack on a given public key is based on the matrix equation over  $K$ ,  $G' = W P W^t$ , which we call *the fundamental equation*. The matrix  $G$  can be easily computed by representing the public key as a univariate polynomial over  $\mathbf{K}$ , and then representing the univariate polynomial as the quadratic form  $\underline{x} G \underline{x}^t$ . All the  $G^{*k}$  variants of  $G$  can be computed by raising the entries of  $G$  to various powers and cyclically rotating its rows and columns.



We can thus consider  $G' = \sum_{k=0}^{n-1} t_k G^{*k}$  as a matrix whose entries are linear combinations of known values with unknown coefficients  $t_0, \dots, t_{n-1}$  from  $\mathbf{K}$ . The matrix  $P$  is mostly known, since only the top left  $r \times r$  block in the  $n \times n$  matrix can be non zero, and  $r \ll n$ . The matrix  $W$  is unknown, but there are many relations between its  $n^2$  entries since they are all determined by just  $n$  parameters via  $w_{ij} = s_{j-i}^q$ . Our goal is to use all these observations in order to solve the fundamental equation in polynomial time.

## 5.1 Recovering $T$

We first describe the process of recovering  $t_0, \dots, t_{n-1}$  from the fundamental equation  $G' = WPW^t$ , where each entry in  $G'$  is a linear combination of the  $t_k$  variables. The matrix  $P$  contains at most  $r$  non zero rows, and thus both its rank and the rank of  $WPW^t$  cannot exceed  $r$ . For random choices of  $t_k$  values the expected rank of the evaluated  $G'$  matrix is close to  $n$ . What makes the correct choice of  $t_k$  values special is that they force  $G'$  to have the unusually small rank  $r$ . To simplify the asymptotic analysis, we assume that  $r$  is a constant and  $n$  grows to infinity, and argue that the attack should run in expected polynomial time (even though we cannot formally prove this claim).

The basic approach is to express this rank condition as a large number of equations in a small number of variables. Consider the matrix  $G'$  evaluated with the correct choice of  $t_k$  values. Its rank is at most  $r$ , and thus its left kernel (defined as the set of all row vectors  $\tilde{x}$  over  $\mathbf{K}$  satisfying  $\tilde{x}G' = 0$ ) is a  $n - r$  dimensional linear subspace. We thus expect to find in it  $n - r$  linearly independent vectors  $\tilde{x}_1, \dots, \tilde{x}_{n-r}$  even if we force the first  $n - r$  entries in each  $\tilde{x}_k$  to have some arbitrarily specified values. The remaining  $r$  entries in each one of the  $n - r$  vectors  $\tilde{x}_k$  are defined as new variables. Each vector equation  $\tilde{x}G' = 0$  can be viewed as  $n$  scalar equations over  $\mathbf{K}$ , and thus we get a total of  $n(n - r)$  equations in the  $r(n - r) + n$  variables (the original  $t_k$  coefficients in  $G'$  and the new unspecified entries in all the  $\tilde{x}_k$  vectors).

The bad news is that the equations are quadratic, and we don't know how to solve large systems of quadratic equations in polynomial time (in fact, this was the original problem of deriving cleartexts from ciphertexts!). The good news is that instead of a marginally defined system of  $n$  equations in  $n$  variables, we get an overdefined system of about  $n^2$  equations in about  $rn$  variables where  $r \ll n$ .

Consider the general problem of solving  $e$  randomly generated homogeneous quadratic equations in  $m$  variables. The well known linearization

technique for solving such equations is to replace any product of two variables  $x_i x_j$  for  $i \leq j$  by a new variable  $y_{ij}$ . The total number of new variables is  $n(n+1)/2$ . Each quadratic equation in the original  $x$  variables can be rewritten as a linear equation in the new  $y$  variables. If the number of equations satisfies  $e \geq n(n+1)/2$ , we expect the system to be uniquely solvable, but if  $e \ll n(n+1)/2$ , we expect the linear system to have an exponential number of parasitic  $y$  solutions which do not correspond to any real  $x$  solution.

Unfortunately, in our problem we have  $m \approx rn$  variables but only  $e \approx \epsilon m^2$  quadratic equations where  $\epsilon = 1/r^2$  is smaller than  $1/2$ , and thus the linearization method would fail. In the next subsection we describe a novel heuristic technique called *relinearization* which is expected to solve such systems of quadratic equations for any fixed  $\epsilon > 0$  in polynomial time. The technique seems to have many other applications in cryptography, optimization, and computer algebra, and should be studied carefully.

## 5.2 The Relinearization Technique

Consider a system of  $\epsilon m^2$  homogeneous quadratic equations in the  $m$  variables  $x_1, \dots, x_m$ . We rewrite it as a new system of  $\epsilon m^2$  linear equations in the (approximately)  $m^2/2$  new variables  $y_{ij} = x_i x_j$  for  $i \leq j$ . Its solution space is a linear subspace of expected dimension  $(1/2 - \epsilon)m^2$ , and each solution can be expressed as a linear function of  $(1/2 - \epsilon)m^2$  new variables  $z_k$ . Such a parametric solution can be efficiently found by Gauss elimination.

Most of the  $y_{ij}$  solutions found in this way do not correspond to any possible  $x_i$  solutions. We want to add additional constraints which relate the various  $y_{ij}$  variables to each other in the way implied by their definition as  $y_{ij} = x_i x_j$ . To do this, consider any 4-tuple of indices  $1 \leq a \leq b \leq c \leq d \leq m$ . Then  $x_a x_b x_c x_d$  can be parenthesized in three different ways:

$$(x_a x_b)(x_c x_d) = (x_a x_c)(x_b x_d) = (x_a x_d)(x_b x_c) \implies y_{ab} y_{cd} = y_{ac} y_{bd} = y_{ad} y_{bc}$$

There are about  $m^4/4!$  different ways to choose sorted 4-tuples of distinct indices, and each choice gives rise to 2 equations<sup>2</sup>. We thus get about  $m^4/12$  quadratic equations in the  $m^2/2$   $y_{ij}$  variables, and it is not difficult to prove that they are linearly independent (even though they are algebraically dependent). We can lower the number of variables to  $(1/2 - \epsilon)m^2$  by replacing

---

<sup>2</sup>There are additional 4-tuples of non-distinct indices, which give either one or no additional equations. We ignore them in our asymptotic analysis.

each one of the  $y_{ij}$  variables by its parametric representation as a linear combination of the new  $z_k$  variables.

The relinearization technique is based on the observation that the new  $m^4/12$  quadratic equations in the new  $(1/2 - \epsilon)m^2$   $z_i$  variables can be linearized again by replacing each product  $z_i z_j$  for  $i \leq j$  by a new variable  $v_{ij}$ . The new system has  $m^4/12$  linear equations in  $((1/2 - \epsilon)m^2)^2/2$   $v_{ij}$  variables. We expect this linear system to be uniquely solvable when  $m^4/12 \geq ((1/2 - \epsilon)m^2)^2/2$ . This is satisfied whenever  $\epsilon \geq 1/2 - 1/\sqrt{6} \approx 0.1$ , which is one fifth of the number of equations required by simple linearization.

Two small demonstrations of this procedure can be found in the appendix. There are many possible optimizations of the basic technique: we can use relinearization recursively, consider additional constraints, etc. For example, there are about  $m^6/6!$  possible choices of indices in  $x_a x_b x_c x_d x_e x_f$ , and each one gives rise to 14 different equations of degree 3 in the  $(1/2 - \epsilon)m^2$  parameters  $z_i$ . If we relinearize every product of the form  $z_i z_j z_k$  for  $i \leq j \leq k$ , we get about  $14m^6/720$  linear equations in  $((1/2 - \epsilon)m^2)^3/6$  new variables, which can be solved whenever  $\epsilon \geq 0.008$ . In the full version of this paper we show that for any fixed  $\epsilon > 0$  there is a relinearization scheme which is expected to solve in polynomial time random systems of  $\epsilon m^2$  quadratic equations in  $m$  variables.

We now return to the original problem of extracting  $T$  from the fundamental equation  $G^l = WPW^t$ . Since we have about  $n^2$  quadratic equations in about  $rn$  variables, we get  $\epsilon \approx 1/r^2$ . The worst case happens when  $q = 2$  and  $r = 13$ , yielding  $\epsilon \approx 0.006$ , which is marginally smaller than the threshold stated above. We thus have to use the relinearization scheme which considers products of 8  $x_i$  values, and to solve a huge system of  $O(n^8)$  linear equations in  $O(n^8)$  variables, which is polynomial but impractical. For larger fields  $\mathbf{F}$ , both  $r$  and  $n$  drop considerably if we keep fixed both the degree  $q^r$  of the secret polynomial and the size  $q^n$  of the cleartext space (for example, when we replace  $\mathbf{F}_2$  by  $\mathbf{F}_7$ ,  $r$  drops from 13 to 4 and  $n$  drops from 100 to 36). The smaller  $r$  increases  $\epsilon$  and makes it possible to use simpler relinearization schemes which result in smaller systems of  $O(n^6)$  or even  $O(n^4)$  equations, and the smaller  $n$  make their solution more feasible. The practical details are messy, and will be omitted from this extended abstract.

One final complication is the fact that the quadratic equations have multiple solutions (due to two symmetries of the fundamental equation: we can raise all the  $t_i$  to the power  $q$  and cyclically rotate the vector to the right, and we can multiply all of them by a common constant). Any linearized technique to find these solutions will necessarily return the  $n$  dimensional

linear subspace they span. Almost all the points on this subspace are parasitic solutions, which do not solve the original quadratic equations, and cannot be used to break the scheme. To avoid this problem, we want to force the system of quadratic equations to have a unique solution. The standard way to do this is to choose random additional constraints until only one of the original solutions remains. However, the equations are over the large field  $\mathbf{K}$ , and each additional equation kills all but  $1/q^n \approx 2^{-100}$  of the original solutions, which is too severe. Instead, we can reexpress the quadratic equations over the large field  $\mathbf{K}$  as quadratic equations over the small field  $\mathbf{F}$ , and arbitrarily fix the values of some of the new variables in  $\mathbf{F}$ . Each additional choice reduces the number of solutions by the small factor  $q$ , and with reasonable probability the number of solutions will pass through 1. Working over  $\mathbf{F}$  instead of  $\mathbf{K}$  increases the number of variables by another factor of  $n$ , but we can avoid this higher complexity by translating to  $\mathbf{F}$  only the  $O(n)$  parameters of the linear solution space rather than the  $O(n^8)$  variables of the linearized problem. We can then express some of the algebraic relationships between the  $O(n^8)$  linearized variables as quadratic equations in the new  $O(n^2)$  variables over  $\mathbf{F}$ . The number of quadratic equations we get exceeds the square of the number of new variables, and thus we can solve them efficiently by simple linearization.

### 5.3 Recovering $S$

The last part of the attack recovers  $S$  and  $P$  when  $T$  is known. The matrix  $G' = \sum_{k=0}^{n-1} t_k G^{*k}$  in the fundamental equation  $G' = WPW^t$  is now a completely known matrix. The matrix  $P$  contains at most  $r$  non zero rows, and thus both its rank and the rank of  $G' = WPW^t$  cannot exceed  $r$ . Assume without loss of generality that the rank of  $P$  is exactly  $r$ , and that the rank of  $W$  is exactly  $n$ . Let  $v_1, \dots, v_{n-r}$  be a basis for the left kernel of  $G'$ . Since  $W^t$  is invertible the left kernel of  $WPW^t$  is equal to the left kernel of  $WP$ . The left kernel of  $P$  consists of exactly those vectors which are zero in their first  $r$  entries, and thus each  $v_i$  is mapped by  $W$  to a vector of this form. Since  $G'$  is known, its left kernel can be easily computed, and each one of the  $n - r$  basis vectors gives rise to  $r$  equations in the unknown entries of  $W$ .

The problem seems to be underdefined, with  $r(n - r)$  linear equations in  $n^2$  variables. We can reduce the number of variables from  $n^2$  to  $n$  by replacing each  $w_{ij}$  by  $s_{j-i}^q$ , but then we get nonlinear equations. The crucial observation is that these nonlinear equations over  $\mathbf{K}$  become linear if we

reinterpret them as equations over  $\mathbf{F}$ : Replace each  $s_i$  by  $\sum_{j=1}^{n-1} s_{ij}\omega_j$  where the  $s_{ij}$  is a new set of  $n^2$  variables over  $\mathbf{F}$ . Each  $s_{j-i}^q$  becomes a linear combination of the  $s_{uv}$  variables, and each equation over  $\mathbf{K}$  becomes a collection of  $n$  linear equations over  $\mathbf{F}$ . Altogether there are  $r(n-r)n$  equations in the  $n^2$  new variables over  $\mathbf{F}$ , and for any  $r > 1$  the system is greatly overdefined since  $r(n-r)n \gg n^2$ . The solution of the homogeneous equations can be defined at most up to multiplication by a constant, but as explained earlier any solution of this type is satisfactory.

## A Appendix: A Relinearization Example

We demonstrate the complete relinearization technique on a toy example of 5 random quadratic equations in three variables  $x_1, x_2, x_3$  modulo 7:

$$3x_1x_1 + 5x_1x_2 + 5x_1x_3 + 2x_2x_2 + 6x_2x_3 + 4x_3x_3 = 5$$

$$6x_1x_1 + 1x_1x_2 + 4x_1x_3 + 4x_2x_2 + 5x_2x_3 + 1x_3x_3 = 6$$

$$5x_1x_1 + 2x_1x_2 + 6x_1x_3 + 2x_2x_2 + 3x_2x_3 + 2x_3x_3 = 5$$

$$2x_1x_1 + 0x_1x_2 + 1x_1x_3 + 6x_2x_2 + 5x_2x_3 + 5x_3x_3 = 0$$

$$4x_1x_1 + 6x_1x_2 + 2x_1x_3 + 5x_2x_2 + 1x_2x_3 + 4x_3x_3 = 0$$

After replacing each  $x_i x_j$  by  $y_{ij}$ , we solve the system of 5 equations in 6 variables to obtain a parametric solution in a single variable  $z$ :

$$y_{11} = 2 + 5z, \quad y_{12} = z, \quad y_{13} = 3 + 2z, \quad y_{22} = 6 + 4z, \quad y_{23} = 6 + z, \quad y_{33} = 5 + 3z$$

This single parameter family contains 7 possible solutions, but only two of them also solve the original quadratic system. To filter out the parasitic solutions, we impose the additional constraints:  $y_{11}y_{23} = y_{12}y_{13}$ ,  $y_{12}y_{23} = y_{13}y_{22}$ ,  $y_{12}y_{33} = y_{13}y_{23}$ . Substituting the parametric expression for each  $y_{ij}$ , we get:

$$(2+5z)(6+z) = z(3+2z), \quad z(6+z) = (3+2z)(6+4z), \quad z(5+3z) = (3+2z)(6+z)$$

These equations can be simplified to:

$$3z^2 + z + 5 = 0, \quad 0z^2 + 4z + 4 = 0, \quad 1z^2 + 4z + 3 = 0$$

The relinearization step introduces two new variables  $z_1 = z$  and  $z_2 = z^2$ , and treats them as unrelated variables. We have three linear equations in

these two new variables, and their unique solution is  $z_1 = 6, z_2 = 1$ . Working backwards we find that  $y_{11} = 4, y_{22} = 2, y_{33} = 2$ , and by extracting their square roots modulo 7 we find that  $x_1 = \pm 2, x_2 = \pm 3, x_3 = \pm 3$ . Finally, we use the values  $y_{12} = 6$  and  $y_{23} = 5$  to combine these roots in just two possible ways to obtain  $x_1 = 2, x_2 = 3, x_3 = 4$  and  $x_1 = 5, x_2 = 4, x_3 = 3$ , which solve the original quadratic system.

A more interesting example, which we do not describe in full in this extended abstract, consists of 5 randomly generated homogeneous quadratic equations in 4 variables, Note that this is barely larger than the minimum number of equations required to make the solution well defined. The number of linearized variables  $y_{ij} = x_i x_j$  for  $1 \leq i \leq j \leq 4$  is 10, and the solution of the system of 5 linear equations in these 10 variables can be defined by affine expressions in 5 new parameters  $z_i$ . There are 20 equations which can be derived from fundamentally different ways of parenthesizing products of 4  $x_i$  variables:

$$\begin{aligned}
& y_{12}y_{34} = y_{13}y_{24} = y_{14}y_{23} \\
& y_{11}y_{23} = y_{12}y_{13}, \quad y_{11}y_{24} = y_{12}y_{14}, \quad y_{11}y_{34} = y_{13}y_{14} \\
& y_{22}y_{13} = y_{12}y_{23}, \quad y_{22}y_{14} = y_{12}y_{24}, \quad y_{22}y_{34} = y_{23}y_{24} \\
& y_{33}y_{12} = y_{13}y_{23}, \quad y_{33}y_{14} = y_{13}y_{34}, \quad y_{33}y_{24} = y_{23}y_{34} \\
& y_{44}y_{12} = y_{14}y_{24}, \quad y_{44}y_{13} = y_{14}y_{34}, \quad y_{44}y_{23} = y_{24}y_{34} \\
& y_{11}y_{22} = y_{12}y_{12}, \quad y_{11}y_{33} = y_{13}y_{13}, \quad y_{11}y_{44} = y_{14}y_{14} \\
& y_{22}y_{33} = y_{23}y_{23}, \quad y_{22}y_{44} = y_{24}y_{24}, \quad y_{33}y_{44} = y_{34}y_{34}
\end{aligned}$$

When we substitute the affine expressions in the 5 new  $z_i$  parameters and relinearize it, we get 20 linear equations in the 5  $z_i$  and their 15 products  $z_i z_j$  for  $1 \leq i \leq j \leq 5$ , which is just big enough to make the solution unique (up to  $\pm$  sign) with reasonable probability.  $\square$

## References

- [CSV97] D. Coppersmith, J. Stern and S. Vaudenay, *The Security of the Birational Permutation Signature Scheme*, Journal of Cryptology, 1997, pp. 207-221.
- [FD85] H. Fell and W. Diffie, *Analysis of a Public Key Approach Based on Polynomial Substitution*, Crypto 85, Springer Verlag, pp. 340-349.

- [KS98] A. Kipnis and A. Shamir, *Cryptanalysis of the Oil and Vinegar Signature Scheme*, Crypto 98, Springer Verlag, pp. 257-266.
- [K98] N. Koblitz *Algebraic Aspects of Cryptography*, Springer Verlag, 1998.
- [MI88] T. Matsumoto and H. Imai, *Public Quadratic Polynomial Tuples for Efficient Signature Verification and Message Encryption*, Eurocrypt 88, Springer Verlag, pp. 419-453.
- [OSS84] H. Ong, C. P. Schnorr, and A. Shamir *A Fast Signature Scheme Based on Quadratic Equations*, Proc. 16-th ACM Symp. Theory of Computation, 1984, pp. 208-216.
- [P95] J.Patarin, *Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt 88*, Crypto 95, Springer Verlag, pp.248-261.
- [P96a] J.Patarin, *Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms*, Eurocrypt 96, Springer Verlag, pp.33-48.
- [P96b] J. Patarin, *Asymmetric Cryptography with a Hidden Monomial*, Crypto 96, Springer Verlag, pp. 45-60.
- [P97] J. Patarin, *The Oil and Vinegar Algorithm for Signatures*, presented at the Dagstuhl Workshop on Cryptography, September 97.
- [PS87] J. M. Pollard and C. P. Schnorr, *An Efficient Solution of the Congruence  $x^2 + ky^2 = m \pmod n$* , IEEE Trans. Information Theory, vol. IT-33, no. 5, 1987, pp. 702-709.
- [S93] A. Shamir *Efficient Signature Schemes Based on Birational Permutations*, Crypto 93, Springer Verlag, pp.1-12.