SOME NEW HIDDEN IDEAL CRYPTOSYSTEMS

ILIA TOLI

Abstract. We propose public-key cryptosystems with public key a system of polynomial equations and private key an ideal.

1. Introduction

This paper focuses on Hidden Monomial Cryptosystems, a class of public key (PK) cryptosystems first proposed by Imai and Matsumoto [3]. In this class, the PK is a system of polynomial nonlinear equations. The private key is the set of parameters that the user chooses to construct the equations. Before we discuss our variations, we review briefly a simplified version of the original cryptosystem, better described in [5].

Throughout this paper the parties committed to the tasks are:

- Alice who wants to receive secure messages;
- Bob who wants to send her secure messages;
- Eve, the eavesdropper.

Alice takes two finite fields $\mathbb{F}_q \subset \mathbb{K}$, q a power of 2, and $\beta_1, \beta_2, \ldots, \beta_n$ a basis of $\mathbb{K}$ as an $\mathbb{F}_q$-vector space. Next she takes $0 < h < q^n$ such that $h = q^\theta + 1$, and $gcd(h, q^n - 1) = 1$. Then she takes two generic vectors $\mathbf{u} = (u_1, \ldots, u_n)$ and $\mathbf{v} = (v_1, \ldots, v_n)$ upon $\mathbb{F}_q$, and sets:

$$\mathbf{v} = \mathbf{u}^{q^\theta}\mathbf{u}. \tag{1}$$

The condition $gcd(h, q^n - 1) = 1$ is equivalent to requiring that the map $\mathbf{u} \longmapsto \mathbf{u}^h$ on $\mathbb{K}$ is *1↔1*; its inverse is the map $\mathbf{u} \longmapsto \mathbf{u}^{h'}$, where h' is the inverse multiplicative of h modulo $q^n - 1$.

In addition, Alice chooses two secret affine transformations, i.e., two invertible matrices A and B with entries in $\mathbb{F}_q$, and two constant vectors $\mathbf{c} = (c_1, \ldots, c_n)$ and $\mathbf{d} = (d_1, \ldots, d_n)$, and sets:

$$\mathbf{u} = A\mathbf{x} + \mathbf{c} \qquad \text{and} \qquad \mathbf{v} = B\mathbf{y} + \mathbf{d}. \tag{2}$$

Recall that the operation of raising to the q^k-th power in $\mathbb{K}$ is an $\mathbb{F}_q$-linear transformation. Let $P^{(k)} = \{p_{ij}^{(k)}\}$ be the matrix of this linear transformation in the basis $\beta_1, \beta_2, \ldots, \beta_n$, i.e.:

$$\beta_i^{q^k} = \sum_{j=1}^{n} p_{ij}^{(k)} \beta_j, \qquad p_{ij}^{(k)} \in \mathbb{F}_q, \tag{3}$$

for $1 \leq i, k \leq n$. Alice also writes all products of basis elements in terms of the basis, i.e.:

$$\beta_i \beta_j = \sum_{\ell=1}^{n} m_{ij\ell} \beta_\ell, \qquad m_{ij\ell} \in \mathbb{F}_q, \tag{4}$$

for each $1 \leq i, j \leq n$. Now she expands the equation (1). Equalizing to zero the coefficients of the β_i, she obtains a system of n equations, explicit in the v, and quadratic in the u. She uses now her affine relations (2) to replace the u, v by the x, y. She obtains n equations, linear in the y, and of degree 2 in the x. Using linear algebra, she can get n explicit equations, one for each y as polynomials of degree 2 in the x. Alice makes these equations public.

In order to send her a message $(x_1, x_2, \ldots, x_n)$, Bob substitutes it into the public equations. He obtains a linear system of equations in the y. He solves it, and sends $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ to Alice.

To eavesdrop, Eve has to substitute $(y_1, y_2, \ldots, y_n)$ into the public equations, and solve the nonlinear system of equations for the unknowns x.

When Alice receives $\mathbf{y}$, she decrypts:

$$\begin{gathered} y_1, y_2, \ldots, y_n \\ \Downarrow \\ \mathbf{v} = B\mathbf{y} + \mathbf{d} \\ \Downarrow \\ \mathbf{v} = \sum v_i \beta_i \\ \Downarrow \\ \mathbf{u} = \mathbf{v}^{h'} \\ \Downarrow \\ \mathbf{x} = A^{-1}(\mathbf{u} - \mathbf{c}). \end{gathered}$$

In Eurocrypt $'88$ [4], Imai and Matsumoto proposed a digital signature algorithm for their cryptosystem.

At Crypto $'95$, Jacques Patarin [7] showed how to break this cryptosystem. He noticed that if one takes the equation $\mathbf{v} = \mathbf{u}^{q^\theta + 1}$, raises both sides on the $(q^\theta - \ 1)$-th power, and multiplies both sides by $\mathbf{uv}$, he gets the equation $\mathbf{u}\mathbf{v}^{q^\theta} = \mathbf{u}^{q^{2\theta}}\mathbf{v}$ that leads to equations in the x, y, linear in both sets of variables. Essentially the equations may not suffice to identify uniquely the cleartext, but now even an exhaustive search will be feasible.

The system was definitively insecure and breakable, but its ideas inspired a whole class of PK cryptosystems and digital signatures based on structural identities for finite field operations [1, 5, 8, 9, 2]. The security of this class rests on the difficulty of the problem of solving systems of randomly chosen nonlinear polynomial equations.

If a system were indeed random, Alice too would have no priviledge on solving it. Therefore she chooses the system in such a way that she can solve it in virtue of some private information and renders it public resembling a random system as much as possible.

This paper is organized as follows. In the next section we develop an our own, new cryptosystem. Alice builds her PK by manipulations as above, starting from a certain private system of equations.

In the third section we make some due observations, and discuss some security issues. In the fourth one we provide another, new, cryptosystem.

2. A New Cryptosystem

Fix one finite field $\mathbb{F}$, we will not need extensions. It can be anything, to fix ideas take it $\mathbb{F}_2$. We set a limit 2^m to the complexity of Alice, for example, 2^{10} or less. For a row vector $\mathbf{x}$, $\mathbf{x}^t$ denotes the column vector transposed $\mathbf{x}$.

Alice is going to build a cryptosystem with public key a quadratic system of n equations in n variables $x_1, x_2, \dots x_n$. She does the following.

(1) for (int i from 1 to $\lfloor \frac{n}{m} \rfloor$)
  Randomly generate a quadratic system of m equations $v_{m(i-1)+1}$, $\dots$ v_{mi} in the $i \cdot m$ variables $u_1, u_2, \dots u_{i \cdot m}$.

(2) Randomly generate a quadratic system of `n%m` equations $v_{\lfloor \frac{n}{m} \rfloor + 1}$, $\dots$ v_n in n variables $u_1, u_2, \dots u_n$.

(3) Put all n equations together: $\mathbf{v} = (v_1, v_2, \dots v_n)$.

(4) Randomly generate two $n \times n$ invertible matrices A and B.

(5) Set

$$\mathbf{u} = A\mathbf{x}^t \tag{5}$$

$$\mathbf{v} = B\mathbf{y}^t \tag{6}$$

for $\mathbf{x} = (x_1, x_2, \dots x_n)$ and $\mathbf{y} = (y_1, y_2, \dots y_n)$.

(6) Explicitely solve with respect to the y.

The system $\mathbf{y}$ is Alice's public key. Her private key is A, B, $\mathbf{v}$.

Bob to encrypt a string $\mathbf{x} = (x_1, x_2, \dots x_n)$ simply substitutes it to the public key and sends the result $\mathbf{y} = (y_1, y_2, \dots y_n)$ to Alice.

Alice to decrypt does the following.

(1) Substitute the received ciphertext in the righthand of her public key $y_i = p(x_1, x_2, \dots x_n)$.
(2) Recover a system of equations in the u by means of equation 6.
(3) Solve this system by means of Buchberger algorithm.
(4) Recover $\mathbf{x}$ by means of equation 5.

3. Remarks

It is a commonplace in Commutative Computer Algebra community that systems of equations with few solutions are easier to solve, and those with no solutions at all are still easier. Nonetheless, theoretically both tasks remains of exponential complexity in n if the system is randomly generated.

Solving the system $\mathbf{v}$ by Buchberger is easy. We start from calculating the Groebner basis of the block of m equations in m variables, it will take about 2^m bit operations. The m being small, this poses no problem. Then replace the solutions found in the remainder of the system and go forth solving the second block, until the end. Branching is likely when some block has more than one solutions. Some branchess are expected to die out before the end as some blocks will not have solutions for some partial solutions of the system.

The system will have at least one solution, the cleartext. It may have more than one. A random system in n equations in n variables over a finite field has one solution with probability 1. Handling the problem of more than one solution can be done either by adding few more equations to the system $\mathbf{y}$ or by means of Coding Theory, as it is often done at present.

The overall size of the public key is $O(n^3)$ because writing down a random quadratic system of n equations and n variables is cubic. Nonetheless, encryption is extremely fast.

In decryption, the complexity of the Gröbner basis calculation is bounded by $\frac{n}{m} \cdot 2^m$. Therefore it grows linearly with n.

One may choose blocks of various lengths in the system $\mathbf{v}$. This is counterproductive since solving two systems of k equations each will take $2 \cdot 2^k$ bit operations while solving two systems of $k+1$ and $k-1$ equations will take $2^{k-1} + 2^{k+1}$ bit operations.

Taking m very small is inviting. In the limit case $m = 1$ we fall within a (generally bad) case of the cryptosystem in the next Section.

Thus, in general blocklengths of the private system will differ by at most one. In order to have the private system as random as possible, it is better to order the smaller blocks in the beginning (with less variables).

The constant part of the equations in the private system may be erased for being useless. Indeed, from Eve's viewpoint there exists another private system **v'** without constant part.

The main data to Eve is the public key. By brute force she may take $(y_1, y_2, \ldots, y_n)$, substitute it in the public key equations, solve the system within the base field, and take the sensate solution. Supposedly, the solution of the system will pass through the complete calculus of a Gröbner basis. It is known to be an NP-complete problem.

4. Yet Another Cryptosystem

Wishing to make the cryptosystem more deterministic, we have the following choices. Alice fixes a base field $\mathbb{F}$. She builds a random system of equations as follows.

(1) Build the system of equations:

$$\left\{\begin{array}{l} v_1 = p_{1,1}(u_1) \\ v_2 = p_{2,1}(u_2) + p_{2,2}(u_1) \\ v_3 = p_{3,1}(u_3) + p_{3,2}(u_1, u_2) \\ v_4 = p_{4,1}(u_4) + p_{4,2}(u_1, u_2, u_3) \\ \ldots\ldots\ldots \\ v_n = p_{n,1}(u_n) + p_{n,2}(u_1, u_2, u_3, \ldots u_{n-1}) \end{array}\right. \tag{7}$$

(2) Set

$$\mathbf{u} = A\mathbf{x}^t \tag{8}$$

$$\mathbf{v} = B\mathbf{y}^t \tag{9}$$

(3) Explicitely solve the system with respect to the y_i.

Again the system $\mathbf{y}$ is Alice's public key and A, B and the system $\mathbf{v}$ are her private key. Encryption is identic to the previous cryptosystem. Decryption will provide for solution of low-degree univariate polynomials in order to recover $\mathbf{u}$.

All the $p_{i,j}$ polynomials are taken quadratic or pseudoquadratic over $\mathbb{F}$. The $p_{i,1}$ are furthermore chosen to be permutation polynomials. For the rest they are randomly chosen.

Though the public key system can be with coefficients from any field, the system in the u and the v must be with coefficients not from $\mathbb{F}_2$ and $\mathbb{F}_4$.

Consider first the case $\mathbb{F} = \mathbb{F}_2$. There are only linear polynomials with roots in $\mathbb{F}_2$. Therefore Eve knows that the variety generated by the public key system rests on a hyperplane in the space $\mathbb{F}_2^{2n}$. She can calculate this hyperplane by evaluating many enough cleartexts and knowing that the tuples $(\mathbf{x}, \mathbf{y})$ rest there. Having calculated it and added it to the public key, by means of this linear equation Eve can get rid of a variable from the public key.

Now again she knows that the whole variety rests upon a hyperplane. Again she can calculate it and so forth by finite induction she recovers the linear system that the reduced Gröbner basis that the system $\mathbf{v}$ hides.

The case $\mathbb{F} = \mathbb{F}_4$ is similar. In $\mathbb{F}_4$ inversion is elevation in power 2, therefore a linear operation. All permutation polynomials over a finite field are compositions of translations and inversion [6]. Therefore, all permutation polynomials represent affine transformations. Again Eve will know that the public key variety will rest upon some hyperplane and we are brought to the case $\mathbb{F}_2$.

All the other finite fields appear to be good.

These restrictions apply only to the private key system. The public key equations can be over any field. Indeed, once Alice generates the public key system, she may write it as a system over the prime field.

Combinations and variations of the two cryptosystems can be done in many fashions. All of this information can be kept private but the security of the cryptosystem does not rely on it.

References

[1] Nicolas Courtois. `http://www.minrank.org/hfe/` and `http://www.hfe.info/`.

[2] Louis Goubin and Jacques Patarin. Trapdoor one-way permutations and multivariate polynomials. `http://citeseer.nj.nec.com/patarin97trapdoor.html`.

[3] Hideki Imai and Tatsuo Matsumoto. Algebraic methods for constructing asymmetric cryptosystems. In *Algebraic Algorithms and Error-Correcting Codes, Proceedings Third International Conference*, pages 108–119, Grenoble, France, 1985. Springer-Verlag.

[4] Hideki Imai and Tatsuo Matsumoto. Public quadratic polynomial tuples for efficient signature-verification and message-encryption. In *Advances in Cryptology, Eurocrypt '88*, pages 419–453. Springer-Verlag, 1989. `http://link.springer.de/link/service/series/0558/papers/0330/03300419.pdf`.

[5] Neal Koblitz. *Algebraic Aspects of Cryptography.* Springer, 1999.

[6] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley, 1983.

[7] Jacques Patarin. Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt'88. In *Proc. of the 15th Annual International Cryptology Conference on Advances in Cryptology - CRYPTO'95*, pages 248–261, Santa Barbara, California, 1995. `http://link.springer.de/link/service/series/0558/papers/0963/09630248.pdf`.

[8] Jacques Patarin. Asymmetric cryptography with a hidden monomial. In *Advances in Cryptology-CRYPTO'96*, pages 45–60. Springer-Verlag, 1996. `http://link.springer.de/link/service/series/0558/papers/1109/11090045.pdf`.

[9] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. *Lecture Notes in Computer Science*, 1070:33–on, 1996. `http://www.minrank.org/hfe.pdf`.