

HIDDEN POLYNOMIAL CRYPTOSYSTEMS

ILIA TOLI

ABSTRACT. We propose public-key cryptosystems with public key a system of polynomial equations, algebraic or differential, and private key a single polynomial or a small-size ideal. We set up probabilistic encryption, signature, and signcryption protocols.

1. INTRODUCTION

This paper focuses on Hidden Monomial Cryptosystems, a class of public key cryptosystems (PK, PKC) first proposed by Imai and Matsumoto [IM85]. 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 [Kob99]. The parties throughout this paper are:

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

Alice takes two finite fields $\mathbb{F}_q < \mathbb{K}$, q a power of 2, and $\beta_1, \beta_2, \dots, \beta_n$ 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 $\mathbf{u} = (u_1, \dots, u_n)$ and $\mathbf{v} = (v_1, \dots, v_n)$ upon \mathbb{F}_q , and sets¹:

$$(1) \quad \mathbf{v} = \mathbf{u}^{q^\theta} \mathbf{u}.$$

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

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

$$(2) \quad \mathbf{u} = A\mathbf{x} + \mathbf{c} \quad \text{and} \quad \mathbf{v} = B\mathbf{y} + \mathbf{d}.$$

1991 *Mathematics Subject Classification*. Primary: 11T71; Secondary: 12H05.

Key words and phrases. Public key, hidden monomial, HFE, differential algebra, TTM, probabilistic encryption, signcryption, symmetric cryptography.

¹We reserve **boldface** to the elements of \mathbb{K} thought as vectors upon \mathbb{F}_q in the fixed private basis. They are considered vectors or field elements, as convenient, without further notice. This shift in practice takes a Chinese Remainder Theorem.

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 $\beta_1, \beta_2, \dots, \beta_n$, ie:

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

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

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

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 . So 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.

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

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

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

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

In Eurocrypt '88 [IM89], Imai and Matsumoto proposed a digital signature algorithm for their cryptosystem.

At Crypto '95, Jacques Patarin [Pat95] 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 $\mathbf{u}\mathbf{v}$, 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

do not suffice to identify uniquely the message, but now even an exhaustive search will be feasible. The system was definitively insecure and breakable, but its ideas inspired a whole class of PKCs and digital signatures based on structural identities for finite field operations [HFE, Kob99, Pat96a, Pat96b, GP].

The security of this class rests on the difficulty of the problem of solving systems of nonlinear polynomial equations. This problem is hard iff the equations are randomly chosen. If they really were random, the problem is hard to Alice, too. So, all we try to do is to get systems of equations that are not random, but appear to be the most possible.

Our 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 with a single bivariate polynomial. All of Alice's manipulations are meant to hide from Eve this polynomial. It is the most important part of the private key. Its knowledge reduces decryption to the relatively easy problem of solving a single univariate polynomial of a moderate degree. Encryption is probabilistic, in the sense that to a given cleartext correspond zero, one, or more ciphertexts. Decryption is deterministic, in the sense that if encryption succeeds, decryption does succeed, too. Almost any bivariate nonlinear polynomial can give raise to a PK. This plentitude of choices is an important security parameter.

In the Section 3 we discuss some security issues. In Section 4 we compare the our cryptosystem to other known ones. In Section 5 we provide our cryptosystem with a digital signature algorithm. In Section 6 we provide a signcryption protocol, ie, a joint encryption and signature protocol. In Section 7 we discuss some more variations. Essentially, we replace the single bivariate polynomial by a small ideal.

In Section 8 we recall Shannon's unconditionally secure cryptosystems [Sti02]. Nowadays they are considered an exclusive domain of the symmetric cryptography. This is due mostly to the unhappy state of art of PKC. In Section 9 we extend our constructions to differential fields of positive characteristic. They seem the suitable environment for unconditionally secure PK (USPK) cryptosystems.

2. A NEW CRYPTOSYSTEM

2.1. Key Generation. Alice chooses two finite fields $\mathbb{F}_q < \mathbb{K}$, and a basis $\beta_1, \beta_2, \dots, \beta_n$ of \mathbb{K} as an \mathbb{F}_q -vector space. In practice, $q = 2$. However, it can be any p^r , for any p prime, and any $r \in \mathbb{N}$. Next Alice takes a generic (for now) bivariate polynomial:

$$(5) \quad f(X, Y) = \sum_{ij} \mathbf{a}_{ij} X^i Y^j$$

in $\mathbb{K}[X, Y]$, such that she is able to find **all** its roots in \mathbb{K} with respect to X ; $\forall Y \in \mathbb{K}$, if any. For the range of i employed, this is nowadays considered a relatively easy problem. Besides, Alice may appeal to her

own tricks such as polynomial composition, etc. in order to obtain such suitable polynomials. The choice of $f(X, Y)$ is subject to other few constraints, that we make clear at the opportune moment.

In transforming cleartext into ciphertext message, Alice will work with two intermediate vectors, $\mathbf{u} = (u_1, \dots, u_n)$ and $\mathbf{v} = (v_1, \dots, v_n)$; $\mathbf{u}, \mathbf{v} \in \mathbb{K}$. She sets:

$$(6) \quad \sum_{ij} \mathbf{a}_{ij} \mathbf{u}^i \mathbf{v}^j = 0.$$

For $\mathbf{a}_{ij} \neq 0$, she sets somehow:

$$(7) \quad i = \sum_{k=1}^{n_i} q^{\theta_{ik}} \quad \text{and} \quad j = \sum_{k=1}^{n_j} q^{\theta_{jk}},$$

where $\theta_{ik}, \theta_{jk}, n_i, n_j, i, j \in \mathbb{N}_* = \{0, 1, 2, \dots\}$.

Here *somehow* means that (7) may or may not be the q -ary representation of i, j . Taking this freedom, we increase our range of choices, whence the random-looking of the PK. In any fashion, what we are dealing with, are nothing but identities.

Next Alice substitutes the (7) to the exponents in (6), obtaining:

$$(8) \quad \sum_{ij} (\mathbf{a}_{ij} \exp(\mathbf{u}, \sum_{k=1}^{n_i} q^{\theta_{ik}}) \exp(\mathbf{v}, \sum_{k=1}^{n_j} q^{\theta_{jk}})) = 0;$$

that is:

$$(9) \quad \sum_{ij} (\mathbf{a}_{ij} \prod_{k=1}^{n_i} \mathbf{u}^{q^{\theta_{ik}}} \prod_{k=1}^{n_j} \mathbf{v}^{q^{\theta_{jk}}}) = 0.$$

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

$$(10) \quad \beta_i^{q^k} = \sum_{j=1}^n p_{ij}^{(k)} \beta_j, \quad p_{ij}^{(k)} \in \mathbb{F}_q;$$

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

$$(11) \quad \beta_i \beta_j = \sum_{k=1}^n m_{ijk} \beta_k, \quad m_{ijk} \in \mathbb{F}_q;$$

for $1 \leq i, j \leq n$.

Now she substitutes $\mathbf{u} = (u_1, u_2, \dots, u_n)$, $\mathbf{a}_{ij} = (a_{ij1}, a_{ij2}, \dots, a_{ijn})$, $\mathbf{v} = (v_1, v_2, \dots, v_n)$, and the identities (10), (11) to (9), and expands. So she obtains a system of n equations of degree t in the u, v , where:

$$(12) \quad t = \max \{n_i + n_j : \mathbf{a}_{ij} \neq 0\}.$$

So, the total degree of $\mathbf{u}^i \mathbf{v}^j$ becomes $n_i + n_j$.

Here we pause to give some constraints on the range of i, j in (6). The aim of this Section is to generate a set of polynomials; linear in a set of variables, and nonlinear in another one. For that purpose, we relate (6) and (7): $\mathbf{a}_{ij} \neq 0 \Rightarrow \{n_i > 1, n_j = 1\}$.

On the other side, the size of PK is $\mathcal{O}(n^{t+1})$. So, it grows polynomially with n , and exponentially with t . Therefore, we are interested to keep t rather modest, e.g., $t = 2, 3$, or so. So, we have to choose i, j in (5), (7) in order to keep t under a forefixed bound.

Next she takes $\mathbf{c}, \mathbf{d} \in \mathbb{K}$, and sets:

$$(13) \quad \mathbf{u} = \mathbf{x} + \mathbf{c} \quad \text{and} \quad \mathbf{v} = \mathbf{y} + \mathbf{d},$$

where $\mathbf{x} = (x_1, x_2, \dots, x_n)$, $\mathbf{y} = (y_1, y_2, \dots, y_n)$ are vectors of variables.

Now she substitutes (13) to the equations in the u, v above, and expands. So she obtains a system of n equations of degree t in the x, y ; linear in the y , and nonlinear in the x .

After the translations (13) each monomial $X_i Y_j$ expands into polynomials with terms of each degree, from $n_i + n_j$ to zero. So, they shuffle better the terms coming from different monomials of (9). On the other hand, they render the PK very dense, so increase drastically its size.

At this point, we are ready to define the cryptosystem.

2.2. The Protocol. With the notations adopted above, we define the **HPE Cryptosystem** (Hidden Polynomial Equations) as the PKC such that:

- **The PK is:**
 - The system of equations in the x, y as above;
 - The field \mathbb{F}_q ;
 - The alphabet: a set of elements of \mathbb{F}_q , or strings of them.
- **The private key is:**
 - The polynomial (5);
 - The identities (6) to (13);
 - The field \mathbb{K} (up to an isomorphism).
- **Encryption:** Bob substitutes the cleartext $\mathbf{x} = (x_1, x_2, \dots, x_n)$ in the PK, finds one solution $\mathbf{y} = (y_1, y_2, \dots, y_n)$ of it, and sends \mathbf{y} to Alice. Each of solutions if any, are encryptions to the same cleartext. We postpone the case when there are no solutions.
- **Decryption:** Alice substitutes $\mathbf{v} = \mathbf{y} + \mathbf{d} \in \mathbb{K} > \mathbb{F}_q$ in (6), and finds **all** solutions within \mathbb{K} . There is at least one. Indeed, if \mathbf{x} is Bob's cleartext, \mathbf{u} as in (13) is one. She represents all solutions in the basis $\beta_1, \beta_2, \dots, \beta_n$. It takes a Chinese Remainder Theorem. Next she sets: $\mathbf{x} = \mathbf{u} - \mathbf{c}$. With probability ≈ 1 , all results but one, Bob's (x_1, x_2, \dots, x_n) , are gibberish, or even stretch out of the alphabet. We come back later at this point.

2.2.1. The main suspended question is that of the existence of solutions. Well, Bob succeeds to encrypt a certain message \mathbf{x} iff Alice's

equation (6) has solutions for \mathbf{u} as in (13) for that \mathbf{x} . Alice's polynomial (6) in \mathbf{v} for a given \mathbf{u} is a random one. It is a well-known fact from algebra that the probability that a random polynomial with coefficients upon a finite field has a root in it is $1 - \frac{1}{e} \approx 63.2\%$ [Kob99, Mar97]. Now the remedy is probabilistic. Alice's letters are sets of elements of \mathbb{F}_q , or of strings of them. Bob writes down a plaintext, and starts encryption. If he fails, he substitutes a letter or a string of the cleartext with another one of the same set, and retries. So, with the number s of trials, the failure probability approxes to zero acceptably fast: $\frac{1}{e^s}$.

If Bob finds more than one solution, each of them is encryption to the same cleartext. So, the our cryptosystem is probabilistic: to a single cleartext correspond zero, one, or more ciphertexts.

2.2.2. The other problem is that Alice may have to distinguish the right solution among a great number of them. Here is a first remedy. Her number of solution is bounded above by the degree in X of f . So, it is beter to keep it moderate. Later we give other remedies, too.

3. SECURITY ISSUES

The main data to Eve are PK and n . By brute force, she has to take (y_1, y_2, \dots, y_n) , to substitute it in the PK equations, to solve within the base field, and to take the sensate solution. Almost surely, there is only one sensate solution among those that she finds. She has to find it among t^n of them. However, the main difficulty to her is just solving the system. It will probably pass through the complete calculus of a Gröbner basis. It is a well-known hard problem.

So, the complexity of the trapdoor is $\mathcal{O}(t^n)$. On the other hand, the size of the PK is $\mathcal{O}(n^{t+1})$. This fully suggests the values of the parameters. It is better to take n huge. This diminishes the probability that Alice confuses decryption, however close to zero, and, what is most important, it renders Eve's task harder. Bob will have to solve bigger systems of linear equations.

If we take t big, it renders the size of the PK impractical. Actually, $n \geq 100$ and $t = 2, 3, 4$ are quite good sample values. If we only take the monomials of f to be univariate, PK size is roughly the same as HFE , and we have still a plethora of choices. In any case, later in Section 7 we present better settings that all in one: moderate the size of the PK, increase its randomness, and contain better the number of undesired solutions.

There exist facilities [SCPK] to solve overdefined systems of equations. Unlike most of the rest, our PK is irrendundant, so it is not subject to such facilities.

Now, by exhaustive search we mean that Eve substitutes the y in the public equations, and tries to solve it by substituting values to the x . If we have d letters each of them being represented by a single element of

\mathbb{F}_q , the complexity of an exhaustive search is $\mathcal{O}(d^n)$. It is easy for Alice to render exhaustive search more cumbersome than Gröbner attack. The last one seems to be the only choice to Eve.

Affine multiple attack [Pat96b] is of no use in our settings.

4. HPE vs HFE

Patarin [Pat96b, Wol02] uses solving univariate polynomial equations, too. He takes a univariate polynomial:

$$f(x) = \sum_{i,j} \beta_{ij} x^{q^{\theta_{ij}} + q^{\varphi_{ij}}} + \sum_i \alpha_i x^{q^{\xi_i}} + \mu_0,$$

and with manipulations like ours, both the same as Imai-Matsumoto [IM85], he gets his PK; a set of quadratic equations. He uses two affine maps to shuffle the x , ie, two linear maps and two translations. We claim that the linear maps add nothing to the security. Indeed, to each matrix corresponds a linearized polynomial [LN83]. Applying one-to-one linear maps is equivalent to composing f with permutation linearized polynomials. So, they don't increase $\text{card}\{f\}$. After a linear map we obtain a new polynomial that is itself eligible. Applying right translations is equivalent to composing f with linear polynomials. So, Eve knows that there exists a certain univariate polynomial F such that $F(\mathbf{x}) = \mathbf{y}, \forall \mathbf{x}, \mathbf{y}$ cleartext and ciphertext. Given the several restrictions already put on f and n , perhaps even an interpolation is possible to recover such an F . Interpolation is polynomial in the number of monomials of F . The right translation makes F very dense and increase $\text{card}\{f\}$. So, it is very useful. The left translation instead does nothing but change the constant term. So, we skip it. Several right translations are equal to a single one: their compositum. So, we use them only once.

We can defend *HPE* from interpolation attacks by including in f monomials of very high degree in Y . As n grows, we can include higher- and higher-degree monomials containing Y . The upper bound is $Y^{q^{n-1}}$, and we are allowed to reach it. After a translation, Eve's F becomes very dense. So, in HPE the interpolation is exponential in n .

A major use of these classes of cryptosystems are the smartcards. Given their fairly limited power of calculi, we are strongly interested to contain calculi. This is a major reason to skip matrices. Another good reason is that with the matrices we do not know how are the degrees of f and F related, with polynomial composition instead, yes. This late alone fully suffices to defend HFE too from interpolation attacks.

One may apply left compositions to f , and so on. Composing with linearized $p(Y)$ does not seem of particular use. Instead, after each composition one may add to the compositum random linearized univariate polynomials in Y . In decryption s/he will have to subtract

them opportunely. With opportune choices on PK, one can render encryption and decryption one-to-one, . . . , howsomany-to-howsomany.

In HFE the bigger the degree of f is, the more the PK resembles a randomly chosen set of quadratic equations. So, it is a security parameter. On the other side, it slows down decryption, both by adding a lot of undesired solutions, and by having to solve in private univariate polynomials of higher degree. It is well-known that this late problem requires exponential time on degree. To face the first problem, to the PK are added other, randomly chosen, equations. This is its Achilles' heel. It makes the PK overdefined, therefore subject to certain facilities to solve [SCP]. So, it weakens the trapdoor.

We do not add equations to discard undesired solutions. Indeed, we take the degree in X rather modest, so we do not have so many undesired solutions. Thus, we are not subject to attacks exploiting overdefined equations. If in certain variations we ever do, we need to add less equations, however.

What is most important, we have now a practically infinite range of choices of f . This is not Patarin's case. There the choices are bounded below because of being easy to attack cases, and above because of being impractical to the legitimate users. The only few constraints we put on monomials of f aim to:

- keep PK equations linear in the y ;
- have less undesired solutions in decryption process;
- keep the size of PK moderate;
- keep **all** PK equations nonlinear in the x .

This last constraint is the only must. Otherwise we render the Bezout number of the system small. So we weaken the trapdoor problem.

We can take the degree in Y arbitrarily huge. It gives no trouble to us. For now we only require PK to be linear with respect to the y . So, Y^i will be of the form Y^{q^i} . Moreover, $Y^{q^n} = Y$, $X^{q^n} = X$ gives us the upper bound of the values of i, j in (5).

Well, $\text{card}\{PK\} = q^{2n} \cdot \text{card}\{f\}$. Assume now that Alice decides to keep PK quadratic in the x . Then, for $q \neq 2$ we have:

$$\begin{aligned} \text{card}\{X^i \mid n_i \leq 2\} &= \lfloor \frac{n^2+1}{2} \rfloor + n, & \text{card}\{X^i Y^j\} &= n(\lfloor \frac{n^2+1}{2} \rfloor + n), \\ \text{card}\{f\} &= 2^{q^n \cdot n(\lfloor \frac{n^2+1}{2} \rfloor + n)}, & \text{card}\{PK\} &= q^{2n} 2^{q^n n(\lfloor \frac{n^2+1}{2} \rfloor + n)}. \end{aligned}$$

For $q = 2$ we have one less in $\text{card}\{X^i \mid n_i \leq 2\}$, and so on. Several of the f are bad choices, but the good ones are still a plethora.

5. A DIGITAL SIGNATURE ALGORITHM

For Bob to be able to sign messages, he builds a cryptosystem as above with $[\mathbb{K}_B : \mathbb{F}_{q_B}] = n_B$. Assume now that we are publicly given a set of hash functions that send cleartexts to n_B -tuples of \mathbb{F}_{q_B} .

Bob signs a message M as follows:

- calculates $H(M) = (y_1, y_2, \dots, y_{n_B}) = \mathbf{y}_B$, then $\mathbf{v}_B = \mathbf{y}_B + \mathbf{d}_B$;
- finds one solution (if any, otherwise see Section 2.2.1) \mathbf{u}_B of $f_B(\mathbf{u}_B, \mathbf{v}_B) = 0$ in \mathbb{K}_B .
- sets $\mathbf{x} = \mathbf{u}_B - \mathbf{c}_B$;
- appends $\mathbf{x} = (x_1, x_2, \dots, x_{n_B})$ to M , encrypts, and sends it to Alice. $(x_1, x_2, \dots, x_{n_B})$ is a signature to M .

To authenticate, Alice first decrypts, then she calculates $H(M) = (y_1, y_2, \dots, y_{n_B})$. If $(x_1, x_2, \dots, x_{n_B})$, $(y_1, y_2, \dots, y_{n_B})$ is a solution of Bob's PK, she accepts the message; otherwise she knows that Eve has been causing trouble.

If Eve tries to impersonate Bob and send to Alice her own message with hash value $\mathbf{y} = (y_1, y_2, \dots, y_{n_B})$, then to find a signature $(x_1, x_2, \dots, x_{n_B})$, she may try to find one solution of Bob's system of equations for \mathbf{y} . We trust on the hardness of this problem for the security of authentication.

Actually, the hash functions play no role in this class of signatures. They may as well output parts of the cleartext itself.

6. A SIGNCRYPTION PROTOCOL

Here is the shortest possible description. Let F_A and F_B be Alice's and Bob's PK functions respectively. To send a message \mathbf{x} to Alice, Bob sends her a random element of $F_A(F_B^{-1}(\mathbf{x}))$, that she can decrypt by calculating $F_B(F_A^{-1}(F_A(F_B^{-1}(\mathbf{x}))))$. So if $F_A(F_B^{-1}(\mathbf{x})) \neq \emptyset$. Otherwise, the approach is probabilistic, as in Section 2.2.1.

Here is the extended description. Each letter (or some of them, only) is represented by a set of few (two, e.g.) elements of the field, or strings of them. For ease of explanation, assume that $\mathbb{F}_{q_A} = \mathbb{F}_{q_B}$ and $n_A = n_B$.

Bob writes down the cleartext X , calculates $\mathbf{v}_B = X + \mathbf{d}_B$, and finds one solution (if any, otherwise see Section 2.2.1) \mathbf{u}_B of his private polynomial $f_B(X, Y)$. Next he sets $\mathbf{x}_B = \mathbf{u}_B - \mathbf{c}_B$, that he encrypts as above by means of Alice's PK, and sends her the result.

Alice now first decrypts as in Section 2.2. Next, she substitutes the \mathbf{x} -es she finds into Bob's PK variables x , and solves. There is at least one solution, and at most few of them. One of them is Bob's message.

By brute force attack, Eve now has to take the ciphertext, substitute on Alice's PK, find all solutions, substitute them all on Bob's PK, and take the sensate ones. The improvement on authentication matter stands on the fact that we are allowed longer-bits signatures.

Let us assume that the letters are strings of a fixed length. For an exhaustive search Eve now has to run throughout all the n -tuples of all elements of Alice's ground field; not just throughout n -tuples made of letters. She sets up such n -tuples, checks whether they are solutions of Alice's PK for Bob's ciphertext \mathbf{y} substituted to the variables y . If yes, she substitutes to Bob's PK, and takes the sensate ones.

So, Alice now has a full freedom on building alphabet. In decryption she discards a priori the solutions that contain non-letters. Now practically the good solution is unique.

Apart all, we save the space and calculi of the signature.

7. HIDDEN IDEAL EQUATIONS

Instead of a single bivariate polynomial, Alice may employ an ideal of a very modest size. She separates the variables that she employs within two sets, $\{X_i\}$, $\{Y_j\}$; one for encryption, one for decryption. She may leave one of her private equations of higher degree in the Y_j after the manipulations. This renders Bob's task somehow harder, and increases very much Alice's choices. She obtains her PK with manipulations as in Section 2.1 on all variables X_i , Y_j . Her parameters are:

- $n = [\mathbb{K} : \mathbb{F}_q]$;
- the number s_1 , s_2 of variables X_i , Y_j , respectively;
- the number r of private equations.

So, the number of PK equations is $n \cdot r$, the number of the variables x_{ij} is $n \cdot s_1$, and that of the y_{kl} is $n \cdot s_2$.

Alice's number of variables, the X_i , is insignificant so far, so she is supposed to be able to appeal to Gröbner techniques in order to solve her system of equations within the field of coefficients for Bob's ciphertext. What is most important here and throughout, if Bob succeeds to encrypt, Alice does always succeed to decrypt.

Bob fails encryption for a certain cleartext (X_1, \dots, X_{s_1}) iff Alice's private ideal has no solutions in the Y for such an (X_1, \dots, X_{s_1}) . Alice's private ideal is a random one. If she takes $r \leq s_2$, the probability that it has no solutions is ≈ 0 , and ≈ 1 for $r > s_2$. So, it suffices that Alice takes $r \leq s_2$. The rare critical cases that may supervene are faced simply changing alphabet.

The other problem is that the solutions to Alice may be too many, and in any case finitely many, as the base field is finite. The best remedy to that is that Alice takes $r = s_1$. So, the ideal that she obtains after substitution of Bob's ciphertext is zerodimensional (quite easy to cause it happen), and the number of solutions is bounded above by the total degree of the system. So, she can contain the number of solutions by taking the total degree in the X_i modest.

Alice can take all equations of very low degree in the X , and then transform that basis of the ideal they generate to another one of very high degrees in the X . So she has a low Bezout number of the ideal, and higher degrees in the X , and transformations as above can take place. If she takes the first basis linear, the number of solutions of her equations reduce to one: Bob's cleartext. She can use in decryption the most suitable of the bases of her private ideal.

As soon as $r > s_1$, the PK becomes overdefined.

Alice applies a permutation to the equations and a renumeration to the variables before publishing her key, so Eve does not know how are they related.

If $s_1 < s_2$, the size of the ciphertext is bigger than that of cleartext, and nothing else wrong. By this case, encryption is practically always probabilistic. Indeed, even when the equations are linear with respect to the y_{kl} , since there are more variables than equations, the solutions exist, and are not unique.

Actually, Alice can take a big s_2 . She may choose to manipulate some of the Y_j within a subfield of \mathbb{K} , rather than within \mathbb{K} . Doing so, she is allowed a big s_2 , and a contained size of the ciphertext. The number of the variables y_{kl} now is no more $n \cdot s_2$.

One can employ this protocol for signcryption. The sizes of ciphertexts throughout are roughly equal to those of the plaintext ones. So, one can use all the protocols we describe throughout for multiple encryption as well. They seem suitable for symmetric cryptosystems, too.

Now the size of the PK is $\mathcal{O}(s_1(n)^{t+1})$, and the complexity of the trapdoor is $\mathcal{O}(t^{n \cdot s_1})$.

Even though the size of PK throughout grows polynomially with n , before n becomes interesting, the PK is already quite cumbersome. So, opting for the choices of this Section we can employ much smaller n , whence moderate a lot the size of the public key.

Actually, $n \geq 20$ is quite good. We are allowed some more values of t , too. Alice takes s_1 as big as she can handle, e.g., $s_1 = 1, 2, 3, 4, 5, 6, 7$, or more, and $n \cdot s_1 \geq 100$.

For ns_1 fixed, the bigger s_1 is, the exponentially less cumbersome the PK is, and the exponentially harder becomes Eve's task.

Generally speaking, Alice's task becomes exponentially harder with s_1 , too. In practice, it depends very much on whether does she have any good basis of her private ideal, or not. In any case, the speeds of becoming harder of tasks of Alice and Eve are quite different.

7.1. There are practically infinitely many other tricks we can use. In a midstep Alice may leave the equations linear in some of the x_{ij} , then compose with some opportune univariate polynomials, then again a translation, then publish. She can compose with multivariate polynomials as well, and so on.

Another topic of the security is that the PK appears always the same. Eve is not aware on which of the huge amount of the tricks above did Alice use.

7.2. There exist classes of ideals called *with doubly exponential ideal membership property* [Swa]. These are the ideals for which the calculus of a Gröbner basis requires doubly exponential time on the number of variables. It is very interesting to know whether can we employ them

in some fashion in this class of cryptosystems. In any fashion, this is the theoretical limit for employing solving of polynomial systems of equations in PK cryptography.

8. SOME CONSIDERATIONS

The idea of PK was first proposed by Diffie and Hellman [DH76]. Since then, it has seen several vicissitudes [Odl91, Mora, Morb].

A trapdoor function is a map from cleartext units to ciphertext units that can be feasibly computed by anyone having the PK, but whose inverse function cannot be computed without its knowledge:

- either because (at present, publicly) there is no known way;
- or there are, but the amount of calculi is deterring.

Shannon [Sti02] called the cryptosystems with trapdoor functions of the first class *unconditionally secure*.

In PKC one tries to render the trapdoor equivalent to time-honoured hard problems. What does one really often succeed is to render that hard the recovery of the PK. An eavesdropper may attack trying to recover the PK, this is the case of *RSA* [RSA78] and *ECDL* [Kob99], or may try a lot of other ways [Odl91, COU]. It is commonly assumed that Eve has no particular crash on recovering PKs [TTM], and that she is principally interested on recovering cleartexts.

The author is very fond of the idea of the PK, and believes howsoever in new developments that will make it fully suffice for all purposes.

Actually, one tendency is that of investigating poor structures, ie, structures with less operations, like groups, semigroups with cryptosystems upon the word problem [AAFG01, Yam98, Hug02].

William Sit and the author are investigating rich structures. We are investigating among other things effective USPK cryptosystems upon differential fields of positive characteristic. We hope that cryptography will arouse new interests on differential and universal algebra, too, as it did in number theory and arithmetic geometry. One reason of optimism is that in universal algebra one can go on further with new structures and hard or undecidable problems forever. Recall that until now we have appealed to only unary and binary arithmetic operations.

9. GENERALIZATIONS ON DIFFERENTIAL FIELDS

Differential² algebra [Kol73, Sit02, Rit50, Sad, Kap57] owes its existence mostly to the efforts of Ritt [Rit50] to handle differential equations by means of algebra.

A differential field is a field \mathbb{F} endowed with a set of linear maps $\theta : \mathbb{F} \rightarrow \mathbb{F}$ called derivatives, such that: $\theta(ab) = a\theta(b) + \theta(a)b$.

Kaplansky's booklet is perhaps the best introduction in the topic.

²Several considerations given in this Section are suggestions of professor Sit through private communications.

The cryptosystems given throughout work as well in differential settings. Take \mathbb{K} to be a finite differential field extension of a differential field \mathbb{F} of positive characteristic³. Any such \mathbb{K} is defined by a system of linear homogeneous differential equations, and there are structural constants defining the operations for the derivations (one matrix for each derivation), as well for multiplication.

One can now replace (5) with a differential polynomial of higher order and degree. Throughout Section 7, one can replace ideals with small suitable differential ideals, too. The cryptosystems work verbatim.

The techniques given throughout for polynomials, if applied to differential polynomials, will definitely make it much harder to attack any protocol developed. Any affine transformation (by this is meant a linear combination of the differential indeterminates with not-necessarily constant coefficients, and this linear combination is then substituted differentially in place of the differential indeterminates) will not only even out the degrees, but also the orders of the various partials, and making the resulting differential polynomials very dense.

However, there is one thing to caution about: any time one specifies these structural matrices, they have to satisfy compatibility equations. In the algebraic case, it is the relations between $P^k = \{p_{ij}^{(k)}\}$ in (10) and $M_\ell = \{m_{ij\ell}\}$ in (11). The P^k are simply determined uniquely by M_ℓ , given the choices implicitly defined in (11).

It is very interesting to know in the algebraic case whether Alice's PK is invariant under a change of basis, all the other settings being equal. There is probably some group of matrices in $GL(n, q)$ that can do that. Such a knowledge would only weaken all cryptosystems based on equations systems solving.

In the differential case there is a similar action called Loewy action, or the gauge transformation. For ordinary differential equations, two matrices A, B are Loewy similar if there is an invertible matrix K such that $A = \delta K \cdot K^{-1} + KBK^{-1}$. Using this action, one can classify the different differential vector space structures of a finite dimensional vector space. There is also a cyclic vector algorithm to find a special basis, so that the differential linear system defining the vector space becomes equivalent to a single linear *ODE*.

If no other problems arise for the differential algebraic cryptosystems, there is however one caution more for them to be unconditionally secure. We have to avoid the exhaustive search. For that, Alice has to publish a finite alphabet where each letter is represented by an infinite set, disjoint sets for different letters. This is possible in proper differential fields, as they are infinite. Alice renders the sets public parametrically, as differential algebraic functions of elements of the base differential field, and parameters, e.g., in \mathbb{Z} . Bob chooses a

³In zero characteristic numerical analysis tools seriously affect security, or at least constrain us to more careful choices. We shall not dwell on this topic here.

letter, gives random values to parameters, obtains one representant of the letter, and proceeds as above. In any case, if μ is the order of public equations, any two elements $\Xi, \Theta \in \mathbb{F}$ such that $(\Xi - \Theta)^\mu = 0$ must represent the same letter, if any.

The main care for Alice is that the PK equations must not fall into feasible cases by well-known means, such as linear algebra.

Now the size of the PK is $\mathcal{O}(n^{to+1})$, where o is the order of PK equations. Quite explosive!!! One more reason to take $q = 2$, so some more monomials reduce to zero.

Anyway, we do not have to increase parameters for better security. The trapdoor is simply undecidable. Unlike the algebraic case, we can split cleartext into small strings. Actually, quite good sample values are: $n = 20$ and $t, o = 2, 3, 4$, or so. As of now, *HDPE* trapdoor seems undecidable, and the cryptosystem effective.

Acknowledgments. I wish to thank Massimiliano Sala and Christopher Wolf for many suggestions and fruitful discussions. I am particularly indebted to William Sit for several comments and improvements on earlier drafts, and to his advisor, Carlo Traverso.

REFERENCES

- [AAFG01] Iris Anshel, Michael Anshel, Benji Fisher, and Dorian Goldfeld. New Key Agreement Protocol in Braid Group Cryptography. In *Lecture Notes in Comput. Sci.*, 2020, pages 13–27, Topics in cryptology—CT-RSA 2001 (San Francisco, CA), 2001. Springer-Verlag.
<http://www-cs.engr.ccny.cuny.edu/~csmma/MRLpap.pdf>.
- [COU] <http://www.cryptosystem.net/ttm/>.
- [DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. In *IEEE Trans. Information Theory*, pages 644–654, 1976. <http://www.cs.jhu.edu/~rubin/courses/sp03/papers/diffie.hellman.pdf>.
- [GP] Louis Goubin and Jacques Patarin. Trapdoor one-way permutations and multivariate polynomials. <http://citeseer.nj.nec.com/patarin97trapdoor.html>.
- [GvzG99] Jürgen Gerhard and Joachim von zur Gathen. *Modern Computer Algebra*. Cambridge University Press, 1999.
- [HFE] <http://www.minrank.org/hfe/> or <http://www.hfe.info/>.
- [Hug02] James Hughes. A Linear Algebraic Attack on the AAFG1 Braid Group Cryptosystem. In *Lecture Notes in Comput. Sci.*, 2384, pages 176–189. Springer-Verlag, 2002.
<http://www.network.com/hughes/ACISP02.pdf>.
- [IM85] 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.
- [IM89] 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>.

- [Kap57] Irving Kaplansky. *An Introduction to Differential Algebra*. Hermann, Paris, 1957.
- [Kob99] Neal Koblitz. *Algebraic Aspects of Cryptography*. Springer, 1999.
- [Kol73] Ellis R. Kolchin. *Differential Algebra and Algebraic Groups*. Academic Press, 1973. New York.
- [LN83] R. Lidl and H. Niederreiter. *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, 1983.
- [Mar97] Daniel A. Marcus. *Number Fields*. Springer-Verlag New York, 1997.
- [Mora] Teo Mora. De Nugis Groebnerialium 2: Applying Macaulay's Trick in order to easily write a Groebner basis. <ftp://ftp.disi.unige.it/person/MoraF/PUBLICATIONS/NG2y.ps.gz>.
- [Morb] Teo Mora. Why you cannot even hope to use Groebner Bases in Public Key Cryptography. An open letter to a scientist who failed and a challenge to those who have not yet failed. ftp://ftp.disi.unige.it/person/MoraF/PUBLICATIONS/Crypto_Groebner.ps.gz.
- [MvOV96] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996. <http://www.cacr.math.uwaterloo.ca/hac/index.html>.
- [Odl91] Andrew M. Odlyzko. The rise and fall of knapsack cryptosystems. In *PSAM: Proceedings of the 42th Symposium in Applied Mathematics, American Mathematical Society*, 1991. <http://www.dtc.umn.edu/~odlyzko/doc/arch/knapsack.survey.pdf>.
- [Pat95] 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>.
- [Pat96a] 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>.
- [Pat96b] 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>.
- [Rit50] Joseph Fels Ritt. *Differential Algebra*. AMS colloquia, 1950. http://www.ams.org/online_bks/coll133/.
- [RSA78] Ron Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public key cryptosystems. In *Communications of the ACM*, 21, pages 120–126, 1978. <http://theory.lcs.mit.edu/~rivest/rsapaper.pdf>.
- [Sad] Brahim Sadik. <http://www.medicis.polytechnique.fr/~brahim/>.
- [SCPK] Adi Shamir, Nicolas Courtois, Jacques Patarin, and Alexander Klimov. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. <http://www.minrank.org/xlfull.pdf>.
- [Sit92] William Y. Sit. An algorithm for solving parametric linear systems. *J. Symbolic Comput.*, 28:353–394, 1992.
- [Sit02] William Y. Sit. The Ritt-Kolchin Theory for Differential Polynomials. In *DIFFERENTIAL ALGEBRA AND RELATED TOPICS*. World Scientific, 2002.
- [Sti02] Douglas R. Stinson. *CRYPTOGRAPHY, Theory and Practice*. Chapman & Hall/CRC, second edition, 2002.

- [Swa] Irena Swanson. On the embedded primes of the Mayr-Meyer ideals. <http://arxiv.org/pdf/math.AC/0209344>.
- [TTM] <http://www.usdsi.com/TTMcryptosystem.html>.
- [Wol02] Christopher Wolf. “Hidden Field Equations” (HFE) - Variations and Attacks. Master’s thesis, Universität Ulm, December 2002. <http://www.christopher-wolf.de/dpl>.
- [Yam98] Akihiro Yamamura. Public-key cryptosystems using the modular group. In *Lecture Notes in Comput. Sci.*, volume 1431, pages 203–216. Springer, Berlin, 1998. <http://link.springer.de/link/service/series/0558/papers/1431/14310203.pdf>.

DIPARTIMENTO DI MATEMATICA *Leonida Tonelli*, VIA F. BUONARROTI 2,
56127 PISA, ITALY., toli@posso.dm.unipi.it