A Fast Public Key System With Signature And Master Key Functions

T. Moh*, Math Department, Purdue U
Lecture Notes at EE Department of Stanford University

May 15, 1999

Abstract

We present a new and fast public key system. Let $\mathbf{K}$ *be a finite field of* 2^m *elements. Let* $\phi_4, \phi_3, \phi_2, \phi_1$ *be tame automorphisms of the* $n+r$*-dimensional affine space* $\mathbf{K}^{n+r}$*. Let the composition* $\phi_4\phi_3\phi_2\phi_1$ *be* π*. The automorphism* π *and some of the* ϕ_i*'s will be hidden. Let the plaintext space be* $\mathbf{K}^n$ *and the ciphertext space be* $\mathbf{K}^{n+r}$*. Let the restriction of* π *to the plaintext space be* $\hat{\pi}$ *as* $\hat{\pi} = (f_1, \cdots, f_{n+r}) : \mathbf{K}^n \mapsto \mathbf{K}^{n+r}$*. The field* $\mathbf{K}$ *and the polynomial map* $(f_1, \cdots, f_{n+r})$ *will be announced as the public key. The private key will be the set of maps* $\{\phi_1, \phi_2, \phi_3, \phi_4\}$*. The security of the system rests in part on the difficulty of finding the map* π *from the partial information provided by the map* $\hat{\pi}$ *and the factorization of the map* π *into a product (i.e., composition) of tame automorphisms* ϕ_i*'s.*

keywords: tame automorphism, public key system, public key, private key, plaintext, ciphertext, signature, master key, error-detect.

1 Introduction

Thank Prof. D. Allison for inviting me to speak here.

In this talk we will introduce a new public key system, the "Tame Transformation Method" or TTM. One should not view "TTM" as an abbreviation of "Time To Market" nor of "Time To Money". Many years ago, I wrote a text book "Algebra" for graduate students. In there I mentioned RSA ([12]) as a cute example of number theory. In the summer of 1995, Dr. John M. Acken of Intel came to visit my family. He raised the question of a fast public key system. I reviewed my book and realized that classically people glued a big chunk of characters together, say 128 characters with each an 8 bits number, so one got 1024 bits. Then one viewed this huge data as a binary whole number. One played with those 1024 bits numbers in a modular sense. It is naturally slow to manipulate those huge numbers. It is essentially the same for ECC and other group-theoretic ones.

Theoretically, in the past, most encryption systems are based on problems of finite commutative groups (as represented in the number theory); one either asks to find x, if exists, in the following equation $g^x = y$ where g is a given element of the group and y is an arbitrary element of the group as in "discrete log problem", ElGamal system, ECC etc, or asks to find the order, $\phi(n)(= (p-1)(q-1))$, of the group, $(Z/nZ)^*$, as in RSA etc.

Maybe I shall mention that the public key system is not a mathematical problem. Most mathematicians will dismiss any discussion

*Math Department, Purdue University, West Laf ayette, Indiana 47907-1395. tel: (765)-494-1930, e-mail ttm@math.purdue.edu

about public key systems which can be solved in finitely many steps.

Mathematics or not, public key system is interesting in itself. There is no reason to stick to the number-theoretic approachs of the classical encryption systems. We may consider a non-commutative group, $\mathbf{G}$, of *mappings* from a set $\mathbf{S}$ to a set $\mathbf{T}$ such that the group $\mathbf{G}$ is infinite dimensional generated by an infinite set $\{\phi_i\}$ with the property that each generator ϕ_i is easily invertible on a given point in $\mathbf{T}$. If it is hard to factor any element $\pi \in \mathbf{G}$ into a product $\prod_j \phi_j$ of the generators, then we may have a new encryption system if it satisfies further criterion. In this case, the set $\mathbf{S}$ will be the set of plaintext, the set $\mathbf{T}$ will be the set of ciphertext, the public key will be the map π and the private key will be the decomposition $\pi = \prod_j \phi_j$.

Practically, let us reconsider the problem of RSA 1024. Let $a_1, ..., a_{128}$ be 128 characters with each an 8 bits number. The natural way is not to glue them together. We shall treat them as a point $a = (a_1, ..., a_{128})$ (a so called "plaintext") in 128 dimensional space. To scramble it, we simple apply a map π to the 128 dimensional space and get a new point $b = (b_1, ..., b_{128})$ (a so called "ciphertext"). We require that the map $\pi \in \{$ a group generated by some suitable maps $\{\phi_i\}\}$. We require that

(a) the required compositions $\pi = \prod_j \phi_j$ are non-linear to prevent attacks using linear algebra.

(b) both value $\phi_i(a)$ and its inverse value $\phi_i^{-1}(b)$ can be computed easily.

(c) a composition of a few of the maps should be hard to be decompsed and its inverse hard to be recovered.

(d) it should be user-friendly.

There are ready candidates "Tame Automorphisms" (see below). Note that the "Tame Automorphism Group" (the group generated by all tame automorphisms) is non-commutative and infinite dimensional.

The beauty of Tame Automorphisms is that after compositing a few of them, the resulting map loses all appearances of a Tame Automorphism (cf the example below) and is hard to be decomposed back. For a technical reason, we shall select the space $\mathbf{S}$ of plaintext to be a subspace of the space $\mathbf{T}$ (of ciphertext). Furthermore we require the coefficient field of the 128 dimensional space to be the finite field $GF(2^8)$ (see below). Thus all computations will be fast. Then we have a fast public key system.

The concept of the Tame Automorphisms is fundamental to some algebraists and some algebraic geometers. However, it is a surprise that most computer scientists never heard about the concept. The closest topics covered by the researchers in encryption are the Imai-Matsumoto system ([7]), Patarin's dragons ([10]) and the attempts of expressing multivariate polynomial functions as a composition of one polynomial of one variable and another polynomial of several variables ([4], [5]). The last problem is much simpler than the problem of decomposition of maps (see below) and the running time for algorithms to solve the simpler problems tends to grow exponentially with the number of variables.

2 Mathematical background

This is the first time the theory of Tame Automorphisms will be applied to provide a public key system. We shall explain every term used in this lecture.

(a) Finite Field

The finite field $GF(2^m)$ of 2^m elements is the collection of the m bits numbers $(a_1, a_2, ..., a_m)$, where a_i's are zeroes or ones, and the sum of these m bits numbers is bitwise, while the product depends on the defining irreducible polynomial, which can be car-

ried out by a LFSR (linear feedback shift register) or by looking up a table.

(b) Affine Space

Let K be a field, say $GF(2^m)$. Let K^{n+r} be the affine space of dimension n+r over K. Note that an "affine space" K^{n+r} is a vector space without the algebraic structure and the origin, i.e., the "physical space". we prefer an affine space over a vector space because (1) we need to remove the origin, (2) we shall consider non-linear maps such as polynomial maps.

(c) Tame Automorphism

A linear transformation $\psi = (\psi_1, ..., \psi_{n+r})$ is a map of the following form,

$$\psi_i(x_1, ..., x_{n+r}) = \sum_j a_{ij}x_j + b_i$$

where a_{ij} and b_i are elements in K. A linear transformation ψ is said to be *invertible* if the coefficient matrix (a_{ij}) is invertible.

Definition: We define a *tame* automorphism $\phi_i = (\phi_{i,1}, \cdots, \phi_{i,n+r})$ as either an invertible linear transformation, or of the following form in any *order* of variables $x_1, \cdots, x_{n+r}$ with polynomials $h_{i,j}$,

$$\begin{aligned}
&(1) : \phi_{i,1}(x_1, \cdots, x_{n+r}) \\
&\quad = x_1 + h_{i,1}(x_2, \cdots, x_{n+r}) = y_1 \\
&(2) : \phi_{i,2}(x_1, \cdots, x_{n+r}) \\
&\quad = x_2 + h_{i,2}(x_3, \cdots, x_{n+r}) = y_2 \\
&\cdots\cdots\cdots \\
&(j) : \phi_{i,j}(x_1, \cdots, x_{n+r}) \\
&\quad = x_j + h_{i,j}(x_{j+1}, \cdots, x_{n+r}) = y_j \\
&\cdots\cdots\cdots \\
&(n+r) : \phi_{i,n+r}(x_1, \cdots, x_{n+r}) \\
&\quad = x_{n+r} = y_{n+r}
\end{aligned}$$

Example:

Let $K = GF(2), \psi(x_1, x_2, x_3) = (x_1 + x_2x_3, x_2 + x_3^2, x_3), \eta(x_1, x_2, x_3) = (x_1, x_2, x_3 + x_1^2)$ be two tame automorphisms. Then it is easy to see that $\psi^2(x_1, x_2, x_3) = (x_1 + x_3^3, x_2, x_3)$ and $\psi\eta(x_1, x_2, x_3) = (x_1 + x_2x_3 + x_1^2x_2, x_2 + x_3^2 + x_1^4, x_3 + x_1^2)$

The group generated by all tame automorphisms is called the *tame automorphism group*. Note that the group product is the composition of maps, i.e., **substitution**, which is different from the product of polynomials. The following proposition and its corollaries will be given without proofs.

Proposition 1 *Let a tame automorphism ϕ_i be defined as in the preceding paragraph. We have the inverse $\phi_i^{-1} = (\phi_{i,1}^{-1}, \cdots, \phi_{i,n+r}^{-1})$ with $x_{n+r} = \phi_{i,n+r}^{-1}(y_1, \cdots, y_{n+r}) = y_{n+r}$ and $x_j = \phi_{i,j}^{-1}(y_1, \cdots, y_{n+r}) = y_j - h_{i,j}(\phi_{i,j+1}^{-1}(y_1, \cdots, y_{n+r}), \cdots, \phi_{i,n+r}^{-1}(y_1, \cdots, y_{n+r}))$, for $j = n + r - 1, \cdots, 1$.*

For instance, in the case of four variables, we have the inverse polynomial map ϕ_i^{-1} in the following abstract **general** form in terms of variables,

$$\begin{aligned}
&\phi_{i,4}^{-1}(y_1, \cdots, y_4) = y_4 \\
&\phi_{i,3}^{-1}(y_1, \cdots, y_4) = y_3 - h_{i,3}(y_4) \\
&\phi_{i,2}^{-1}(y_1, \cdots, y_4) = y_2 - h_{i,2}(y_3 - h_{i,3}(y_4), \\
&\qquad y_4) \\
&\phi_{i,1}^{-1}(y_1, \cdots, y_4) = y_1 - h_{i,1}(y_2 - h_{i,2}(y_3 - \\
&\qquad h_{i,3}(y_4), y_4), y_3 - h_{i,3}(y_4), y_4)
\end{aligned}$$

In general, the total degree of $\phi_{i,j}^{-1}(y_1, \cdots, y_{n+r})$ increases very fast and the number of terms can be quite large as indicated by our later discussions. Therefore it is impractical to actually write down the polynomials $\phi_{i,j}^{-1}(y_1, \cdots, y_{n+r})$. However, if a point $(y'_1, \cdots, y'_{n+r})$ is given, the value of the inverse map can be readily computed in the following **special** form in terms of numbers.

Corollary 2

Given a set of values $(y'_1, \cdots, y'_{n+r}) \in \mathbf{K}^{n+r}$ and a tame automorphism ϕ_i as in the

Definition of this section, then the values $(x'_1, \cdots, x'_{n+r}) = (\phi_{i,1}^{-1}(y'_1, \cdots, y'_{n+r}), \cdots, \phi_{i,n+r}(y'_1, \cdots, y'_{n+r})) \in \mathbf{K}^{n+r}$ *can be found by induction; first, we have* $x'_{n+r} = \phi_{i,n+r}^{-1}(y'_1, \cdots, y'_{n+r}) = y'_{n+r}$, *inductively we have* $x'_{j+1}, \cdots, x'_{n+r} \in \mathbf{K}$, *then we have* $x'_j = \phi_{i,j}^{-1}(y'_1, \cdots, y'_{n+r}) = y'_j - h_{i,j}(x'_{j+1}, \cdots, x'_{n+r})$ *for* $j = n+r-1, \cdots, 1$.

Corollary 3 *Given the decomposition* $\pi = \prod_{i=1}^{i=n} \phi_i$ *where* ϕ_i *are tame automorphisms, then we have* $\pi^{-1} = \prod_{i=n}^{i=1} \phi_i^{-1}$. *Furthermore, if a set of values* $\{y'_j\}$ *is given, then we have* $\pi^{-1}(y'_1, \cdots, y'_{n+r}) = \prod_{i=n}^{i=1} \phi_i^{-1}(y'_1, \cdots, y'_{n+r})$.

3 Theory of automorphisms groups

There is a long history of studying 'automorphism groups' for affine spaces $\mathbf{K}^{n+r}$ and 'embedding theory' in mathematics. There are thousands of papers on those subjects. The theory of automorphism groups for $\mathbf{K}^2$ was established by W. Van der Kulk in 1953 in [13] which stated that the automorphism group for $\mathbf{K}^2$ is the tame automorphism group, i.e, any automorphism of $\mathbf{K}^2$ can be written as a canonical product of tame automorphisms. The most famous problem in this area is the fifty eight year old Jacobian Conjecture ([2]) for 2-dimensional space. For embedding theory ([1], [8], [9]), the simplest case, i.e., the (algebraic) embedding of affine line to affine plane in characteristic 0, had been an open problem for forty years. It was solved in [1] using long and difficult arguments. In the case of finite fields, the embedding problem is open for $n = 1$ and $n + r = 2$, see [8]. They are beyond the scope of the present article.

There is an abyss between our knowledge of the automorphism group of $\mathbf{K}^2$ and the automorphism group of $\mathbf{K}^{n+r}$ for $n + r \geq 3$. For $n + r \geq 3$, every element π in the tame automorphism group has a factorization $\pi = \prod_i \phi_i$ by its definition, however, there is no known way to find it. In [9], Nagata constructed an automorphism as follows; $\sigma(x_1, x_2, x_3) = (x_1, x_2 + x_1(x_1x_3 + x_2^2), x_3 - x_2(x_1x_3 + x_2^2) - x_1(x_1x_3 + x_2^2)^2)$ for $n+r = 3$. One can not decide whether σ is in the tame automorphism group since there is no theorem for the above factorization (Note that one can show that $\sqrt{2}$ is not rational since we know the factorization theorem for integers).

4 Principle or Algorithm

Principle: Let m, n, r, s be positive integers. Let $n + r \geq 3$, and $\mathbf{K}$ a field of 2^m elements. Let the user select k tame automorphism $\phi_k, \cdots, \phi_2, \phi_1$ of $\mathbf{K}^{n+r}$. Let $\pi = \phi_k \cdots \phi_2\phi_1 = (\pi_1, \cdots, \pi_{n+r})$. Let $\hat{\pi} = (\pi_1(x_1, \cdots, x_n, 0, \cdots, 0), \cdots, \pi_{n+r}(x_1, \cdots, x_n, 0 \cdots, 0))$, and $f_i(x_1, \cdots, x_n) = \pi_i(x_1, \cdots, x_n, 0, \cdots, 0)$ for $i = 1, \cdots, n + r$.

The user will announce the map $\hat{\pi} = (f_1, \cdots, f_{n+r})$: $\mathbf{K}^n \mapsto \mathbf{K}^{n+r}$ and the field $\mathbf{K}$ of 2^m elements as the public key.

Given a plaintext $(x'_1, \cdots, x'_n) \in \mathbf{K}^n$. The sender evaluates $y'_i = f_i(x'_1, \cdots, x'_n)$. Then the ciphertext will be $(y'_1, \cdots, y'_{n+r}) \in \mathbf{K}^{n+r}$.

The legitimate receiver (i.e., the user) recovers the plaintext by $(x'_1, \cdots, x'_n, 0, \cdots, 0) = \phi_1^{-1} \cdots \phi_k^{-1}(y'_1, \cdots, y'_{n+r})$ (see **Corollaries** 2 & 3). The private key is the set of maps $\{\phi_1, \cdots, \phi_k\}$.

5 Component

We will give a report of an implementation (for a complete detail for TTM 1.9, see http://www.usdsi.com/ttm.html) for the case that n=64, n+r=100. In our implementation, let the field K be $GF(2^8)$, the finite field of 2^8 elements. We will build four tame automorphisms $\phi_4, \phi_3, \phi_2, \phi_1$ which will be decided by user's input. The maps ϕ_4, ϕ_1 are invertible linear transformations. The composition $\phi_3\phi_2 = (q{*}_1, ..., q{*}_{100})$, which is provided by the software and the user's input,

will have the following properties,

(1) all componenets, $q*_i$, are polynomials in 64 variables of degree 2.

(2) the degree 2 homogeneous parts of $q*_i$'s are linear independent.

(3) no polynomial in $q*_i$'s of degree less than 8 will generate a power of any polynomial of degree 1.

Furthermore, we require that the linear transformation ϕ_1 to move the origin (0,...,0) to a point $(b_1, ..., b_{64})$ where all b_i's are non-zeroes, and the linear transformation ϕ_4 to make the composition $\phi_4\phi_3\phi_2\phi_1$ fixes the origin. The reason is that then all linear forms of $q*_i$'s will not form a linear transformation of the vector space K^{64}. The purpose of the above requirement is to safeguard the linear terms from an attack using linear algebra.

To implement the above principle, we use a Component $\mathbf{Q}_8$ (see below). To indicate the influence of the user on the selection of this component, we insert only one parameter (which could be many), a_1, in the formulation of it. The user will select more functions (see section 6) to make individual scheme non-traceable. The component in this section is example by nature, it is selected due to the theoretical clearness. Similar ones can be constructed.

The following definition will be used in the discussion of Component $\mathbf{Q}_8$,

Definition Let $q_1, \cdots, q_s$ be polynomials in variables $x_1, \cdots, x_t$. Let $\ell(x_1, \cdots, x_t)$ be a polynomial. If

$$Q(q_1(x_1, \cdots, x_t) \cdots, q_s(x_1, \cdots, x_t)) = \ell(x_1, \ldots, x_t)$$

Then Q is called a *generating polynomial* of ℓ (over $q_1, \cdots, q_s$) . Furthermore, if Q is of the minimal degree among all possible generating polynomials of ℓ, then it is called a *minimal generating polynomial* of ℓ, and its degree is called the *generating degree* of ℓ, in symbol $gendeg(\ell)$. If there is no such polynomial Q, then we define $gendeg(\ell)= \infty$.

Now, let us define the Component $\mathbf{Q}_8$ as follows,

Component $\mathbf{Q}_8$: Let the field $\mathbf{K}$ be of 2^8 elements, $t = 19$, $s = 30$. and a_1 is any element in $\mathbf{K}$. Let

$$\begin{aligned}
&q_1(x_1, \cdots, x_{19}) = x_1 + a_1x_2 + x_2x_6;\\
&q_2(x_1, \cdots, x_{19}) = x_2^2 + x_3x_7;\\
&q_3(x_1, \cdots, x_{19}) = x_3^2 + x_4x_{10};\\
&q_4(x_1, \cdots, x_{19}) = x_3x_5;\\
&q_5(x_1, \cdots, x_{19}) = x_3x_{11};\\
&q_6(x_1, \cdots, x_{19}) = x_4x_7;\\
&q_7(x_1, \cdots, x_{19}) = x_4x_5;\\
&q_8(x_1, \cdots, x_{19}) = x_7^2 + x_5x_{11};\\
&q_9(x_1, \cdots, x_{19}) = x_6^2 + x_8x_9;\\
&q_{10}(x_1, \cdots, x_{19}) = x_8^2 + x_{12}x_{13};\\
&q_{11}(x_1, \cdots, x_{19}) = x_9^2 + x_{14}x_{15};\\
&q_{12}(x_1, \cdots, x_{19}) = x_7x_{10};\\
&q_{13}(x_1, \cdots, x_{19}) = x_{10}x_{11};\\
&q_{14}(x_1, \cdots, x_{19}) = x_{12}^2 + x_7x_8;\\
&q_{15}(x_1, \cdots, x_{19}) = x_{13}^2 + x_{11}x_{16};\\
&q_{16}(x_1, \cdots, x_{19}) = x_{14}^2 + x_{10}x_{12};\\
&q_{17}(x_1, \cdots, x_{19}) = x_{15}^2 + x_{11}x_{17};\\
&q_{18}(x_1, \cdots, x_{19}) = x_{12}x_{16};\\
&q_{19}(x_1, \cdots, x_{19}) = x_{11}x_{12};\\
&q_{20}(x_1, \cdots, x_{19}) = x_8x_{13};\\
&q_{21}(x_1, \cdots, x_{19}) = x_7x_{13};\\
&q_{22}(x_1, \cdots, x_{19}) = x_8x_{16};\\
&q_{23}(x_1, \cdots, x_{19}) = x_{14}x_{17};\\
&q_{24}(x_1, \cdots, x_{19}) = x_7x_{11};\\
&q_{25}(x_1, \cdots, x_{19}) = x_{12}x_{15};\\
&q_{26}(x_1, \cdots, x_{19}) = x_{10}x_{15};\\
&q_{27}(x_1, \cdots, x_{19}) = x_{12}x_{17};\\
&q_{28}(x_1. \cdots, x_{19}) = x_{11}x_{14};\\
&q_{29}(x_1, \cdots, x_{19}) = x_{18} + x_1^2 + a_1^2x_2^2;\\
&q_{30}(x_1, \cdots, x_{19}) = x_{19} + x_{18}^2;
\end{aligned}$$

Then the following Q_8 is a minimal generating polynomial of x_{19}^2 of degree 8 in q_i,

$$\begin{aligned}
Q_8 = {} & q_1^8 + [q_2^4 + q_3^2q_8^2 + q_4^2q_5^2 + q_6^2q_{12}^2\\
& + q_7^2q_{13}^2][q_9^4 + (q_{10}^2 + q_{14}q_{15} + q_{18}q_{19}\\
& + q_{20}q_{21} + q_{22}q_{24})(q_{11}^2 + q_{16}q_{17}\\
& + q_{23}q_{28} + q_{25}q_{26} + q_{13}q_{27})] + q_{29}^4 + q_{30}^2
\end{aligned}$$

Proof. (sketch) It is easy to see that Q_8 is a generating polynomial of x_{19}^2 by substitution. Clearly any generating polynomial R of x_{19}^2 must involve q_{30}^2 which produces x_{18}^4. Therefore the polynomial R must involve q_{29}^4 which produces x_1^8. Its degree will be at least 8. Henceforth the above polynomial Q_8 is a minimal generating polynomial of x_{19}^2. ■

Remark : The Component $\mathbf{Q}_8$ will be used to construct a public key scheme in the next section. The security of the scheme depends partially on the degrees of polynomials q_i, Q_8 and their complexities. With the degree 8 in the present implementation, an attacker is forced to consider $3.52(10^{11})$ vectors in a vector space of dimension $2.69(10^{16})$ in our scheme. The dimension is too high to be handled by the present technology. The degrees of polynomials q_i, Q_8 can be increased if necessary. ■

For the convenience of discussion in the next section, let us define

Definition An invertible linear transformation $\phi_1 = (\phi_{1,1}, \cdots, \phi_{1,n+r})$ is said to be of type A if

$$\phi_{1,i} = \sum_{j=1}^{n+r} a_{i,j}x_j + b_i$$

then

(1)For $i = 1, \cdots, n$, we always have $b_i \neq 0$, $a_{i,j} = 0$, for $j = n+1, \cdots, n+r$ and at least half of the remainning $a_{i,j}$ are non-zero.

(2)For $i = n+1, \cdots, n+r$, we always have $\phi_{1,i} = x_i$.

6 Implementation Scheme

Let $n = 64$, $r = 36$, $n + r = 100$. Let the field $\mathbf{K}$ be $\mathbf{GF}(2^8)$, the finite field of 2^8 elements. We will build four tame automorphisms $\phi_4, \phi_3, \phi_2, \phi_1$. The maps ϕ_1, ϕ_4, provided by the user are invertible **linear** transformations with minor restrictions (see (A), (C) below). The **non-linear** maps ϕ_2, ϕ_3 are built essentially with Component $\mathbf{Q}_8$ of the last section with minor supplements from the user (see (B) below).

Notations:

Let us use the same notations as in the last section; polynomials $q_1, \cdots, q_{30}$, the generating polynomial $Q_8(q_1, \cdots, q_{30})$. Let $[j] = j \ mod\ 8$ and $1 \leq [j] \leq 8$.

User's selection:

(A) The user selects $\phi_1 = (\phi_{1,1}, \cdots, \phi_{1,100})$ to be any invertible linear transformation of type A (see the last Definition of section 5).

(B) The tame automorphisms $\phi_2 = (\phi_{2,1}, \cdots, \phi_{2,100})$ and $\phi_3 = (\phi_{3,1}, \cdots, \phi_{3,100})$ are defined according to the following rules $(1)^* - -(11)^*$,

$(1)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i$, for $i = 1, 2$.

$(2)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{i-1}x_{i-2}$, for $i = 3, \cdots, 9$.

$(3)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{[i-1]}^2 + x_{[i]}$ $x_{[i-5]} + x_{[i+1]}x_{[i+6]}$, for $i = 10, \cdots, 17$,

$(3)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{[i-1]}x_{[i+1]}$ $+x_{[i]}x_{[i+4]}$, for $i = 18, \cdots, 25$,

$(3)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{[i-1]}x_{[i+1]}$ $+x_{[i+2]}x_{[i+5]}$, for $i = 26, \cdots, 30$,

$(4)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{i-10}^2$, $for\ i = 31, \cdots, 60$

$(5)^* : \phi_{2,61}(x_1, \cdots, x_{100}) = x_{61} + a_1^2x_{11}^2 + x_9^2$,

$(5)^* : \phi_{2,62}(x_1, \cdots, x_{100}) = x_{62} + x_{61}^2$,

$(5)^* : \phi_{2,63}(x_1, \cdots, x_{100}) = x_{63} + a_1^2x_{17}^2 + x_{10}^2$,

$(5)^* : \phi_{2,64}(x_1, \cdots, x_{100}) = x_{64} + x_{63}^2$,

$(6)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i +$ $q_{i-64}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{62})$, $for\ i = 65, \cdots, 92$

$(7)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i +$ $q_{i-92}(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51},$

$\cdots, x_{60}, x_{63}, x_{64}),$

$for\ i = 93, \cdots, 100$

$(8)^* : \phi_{3,i}(x_1, \cdots, x_{100}) = x_i,$

$for\ i = 3, \cdots, 100$

$(9)^* : \phi_{3,2}(x_1, \cdots, x_{100}) = x_2 +$

$Q_8(x_{93}, \cdots, x_{100}, x_{73}, \cdots, x_{92}, x_{63}, x_{64})$

$(10)^* : \phi_{3,1}(x_1, \cdots, x_{100}) = x_1 +$

$Q_8(x_{65}, \cdots, x_{92}, x_{61}, x_{62})$

(C) The user selects ϕ_4 to be an invertible linear transformation satisfying condition $(11)^*$ in the following way, where $\pi = (\pi_1, \cdots, \pi_{100})$,

$$(11)^* : \pi = \phi_4\phi_3\phi_2\phi_1, \text{ and } \pi_i(0, \cdots, 0) = 0$$

The field $\mathbf{K}$ and polynomials $f_i(x_1, \cdots, x_{64}) = \pi_i(x_1, \cdots, x_{64}, 0, \cdots, 0)$ for $i = 1, \cdots, 100$ will be announced as the public key. The private key is the set of maps $\{\phi_1, \phi_2, \phi_3, \phi_4\}$.

Let $(x'_1, \cdots, x'_{64}) \in \mathbf{K}^{64}$ be the plaintext. The sender evaluates $y'_i = f_i(x'_1, \cdots, x'_{64})$. Then the resulting $(y'_1, \cdots, y'_{100}) \in \mathbf{K}^{100}$ will be the ciphertext.

The legitimate receiver (i.e., the user) recovers the plaintext by $(x'_1, \cdots, x'_{64}, 0, \cdots, 0) = \phi_1^{-1} \cdots \phi_4^{-1}(y'_1, \cdots, y'_{100})$ which can be done easily according to **Corollaries** 2 &3. ■

It is easy to see that the above constructed maps satisfying the conditions 1), 2), and 3) of the section 5.

7 Plaintext, Users and Compactness

Let us count the possible **number of plaintext**. Since the number of plaintext is the number of choices for $x'_1, \cdots, x'_{64}$, we see that there are 2^{512} such plaintext.

Of equal importance to having a large number of possible plaintext is having lots of possible **users**. A simple count of terms of ϕ_1, ϕ_4 results in the total possible number of users $> 2^{61424}$.

Now let us look at the **compactness** of the scheme. It is easy to see that the number of terms of the public key is 214, 400 (for another software implementation TTM 3.2, the number 8,512). On the receiver's side, the total number of terms is 17, 000 (the corresponding size of the private key for TTM 3.2 is 3,502).

8 Technique Report

Following the principle of this article, there are several software implementations. For the convenience of discussions, the method will be called "tame transformation method" (TTM). There are versions TTM 1.9 (of this article), TTM 2.1, TTM 2.3, TTM 2.5, TTM 3.0 and TTM 3.2 available. They use C Language. For m=8, the rates of expansion of data are 1.4, 1.56, 1.63, 1.5, 2.66 and 3.5 respectively. They have been used on commonly available machines as 266 Mhz PowerPC 750 etc. The speed of encryption is 94 k bit/sec for TTM 1.9, 1,001 k bit/sec for TTM 3.0 and 1,626 k bit/sec for TTM 3.2. In comparison, the speed is 16 k bit/sec for RSA 1024 Bsafe 4.0 and 21 k bit/sec for ECC 160.

The decrypting speed is in general 4 to 15 times faster than the encrypting speed (the decrypting speed for TTM 3.2 is 8,271 k bit/sec). For the user who has the private key, the speed of encoding can be increased to 8,271 k bit/sec.

9 Useful Properties of the Scheme

Error-Detect Function

Upon receiving the ciphertext $(y'_1, \cdots, y'_{100})$, the user applies **Corollaries** 2 & 3 to evaluate $\phi_1^{-1}\phi_2^{-1}\phi_3^{-1}\phi_4^{-1}(y'_1, \cdots, y'_{100})$ to decode it and get $(\bar{x}_1, \cdots, \bar{x}_{100})$. If one of

$\bar{x}_{73}, \cdots, \bar{x}_{100}$ is not zero, then there must be an error.

Master Key Function

Let a group of indices, S , be a few extra indices $101, 102, \cdots$ added. Select ϕ_4 such that the corresponding subspace generated by x_i with $i \in S$ and the subspace generated by x_j with $j \notin S$ are both invariant. The original public key scheme gives a master key. A subordinate key can be produced by deleting all f_i's with $i \in S$.

The 'master key-subordinate key' relation can be broken by alternating any one of the linear transformations ϕ_1, ϕ_4 involved.

Signatures

The map $\hat{\pi}$ is not an onto map. However, we may restrict the map to a suitable subspace. Let $V = \{(d_1, \cdots, d_j, 0, \cdots, 0) : d_i \in \mathbf{K}\} \subset \mathbf{K}^{64}$ where j is a fixed integer less than or equal to 62. Let $\bar{V} = \phi_1^{-1}(V)$. We will require that ϕ_4 induces a linear transformation on $W = \{(e_1, \cdots, e_j, 0, \cdots, 0) : e_i \in \mathbf{K}\} \subset \mathbf{K}^{100}$. Let $\tau : (c_1, \cdots, c_j, \cdots, c_{100}) \mapsto (c_1, \cdots, c_j)$ be a projection. Clearly $\tau\hat{\pi}$ is a one to one and onto map from $\bar{V}$ to the j-dimensional affine space. Moreover, the map is *tame*, and its inverse at a point $(y'_1, \cdots, y'_j)$ can be found. The inverse at $(y'_1, \cdots, y'_j)$ forms a *signature*.

10 Cryptanalysis for the Scheme

I. Direct Methods

There is no known way to recover the private key $\{\phi_4, \cdots, \phi_1\}$ from the public key $\hat{\pi}$ and the field $\mathbf{K}$. There are three other direct ways to attack the scheme: (1) use 'inverse formula' for power series to find polynomial expressions of π^{-1} ([3]). Note that only $\hat{\pi}$ is given, since $\hat{\pi}^{-1}$ does not exist theoretically, there is no way to find it, (2) let x_i be a polynomial, g_i, of $\{y_j\}$ with indeterminate coefficients for all i. Do enough experiments using $\{x_i\}$ to determine $\{y_j\}$ and then solve the system of linear equations in indeterminate coefficients to find polynomials g_i, or (3) using 'resultant' to the expression $y'_i = f_i(x'_1, \cdots, x'_{64})$ to eliminate all x'_i except one, say x'_j, and recover the expressions of x'_j in terms of $y'_1, \cdots, y'_{100}$.

At this moment, the above three direct methods can be shown to be ineffective. Due to the space limitation, the detail discussion is omited. The other possible way of attacking is to recover ϕ_i's or their equivalent forms which can be shown to be ineffective too.

II. Other methods

There are two other methods: (1) search for polynomial relations, (2) identify the highest homogeneous parts. They all involve huge memory space and can be shown to be ineffective.

There is another method using the technique of [11] and [12]. The complexity of the problem as shown is higher than 2^{1024} (the cryptanalysis there can not be carried out directly due to the problems of finding $\hat{\pi}^{-1}$, of finding the individualized $\phi_3\phi_2$ and the inequality of two dimensions $64, 100$. The estimates of the other two attacks are disregarded due to the large requirements of memory spaces).

Lately, there is an attack by the method of "relinearization" (cf [6], page 10) of Kipnis and Shamir. This attack can be made impractical either involving large number of variables or jack up the degrees of q_i to three or higher. In our present case, the "relinearization" methid involves solving $1.4(10^9)$ linear equations in $1.2(10^9)$ variables. Or if we increase the degrees of q_i to three, the attack will result in $O(10^{16})$ equations in $O(10^{16})$ variables.

As for the brute force method, **Assuming** it only take one clock cycle to test if a set of 64 random numbers is correct, the attacker still need 3×10^{139} mips (one million instruction per second) years to crack the scheme. In comparison, it requires 3×10^{20} mips years to

cracked RSA 2048.

11 Summary

The present implementation scheme can withstand all known attacks. By its nature, the algorithm is less cumbersome to use than methods that are number theory based. Furthermore, it has the novel functions of error-detect and master key. We wish that this algorithm will provide a new direction of research.

References

[1] Abhyankar, S.S. and Moh, T. *Embeddings of the line in the plane.*Journal für die reine und angewandte Mathematik., pp 148-166, vol 276, 1975.

[2] Bass, H. Connell, E.H. Wright, D.L. *The Jacobian conjecture: reduction of degree and formal expansion of the inverse.* Bull. Amer. Math. Soc. pp 287-330 no. 2 (N.S.) 7, 1983.

[3] Dickerson, Mathew *The inverse of an Automorphism in Polynomial Time.* J. Symbolic Computation, vol 13. 209-220, 1992.

[4] van der Gathen, J. *Functional decomposition of polynomials: the tame case.* J. Symbolic Comp. 9, 281-299, 1990.

[5] van der Gathen, J. *Functional decomposition of polynomials: the wild case.* J. Symbolic Comp. 10, 437-452, 1990.

[6] Kipnis, A. Shamir, A. *Cryptanalysis of the HFE public Key Cryptosystem.* preprint, 1999.

[7] Imai, H. Matsumoto *Algebraic methods for constructing asymmetric cryptosystems, Algebraic and Error -Correcting Codes.* Prod. Third Intern. Conf., Grenoble, France, Springer-Verlag, 108-119, 1985.

[8] Moh, T. *On the Classification Problem of Embedded Lines in Characteristic p.* Algebraic Geometry and Commutative Algebra *in honor of M. Nagata*, vol I, pp 267-280, Kinokuniya, Kyoto, Japan, 1988.

[9] Nagata, M *On the automorphism group of* $\mathbf{K}[X, Y]$., vol 5, Kinokuniya, Tokyo, Japan, 1972.

[10] Patarin, J *Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms.* Advances in Cryptology- Eurocrypt '96, Springer-Verlag, 33-48, 1996.

[11] Patarin, J. Goubin L. and Courtois N. *Improved Algorithms for Isomorphisms of Polynomials.* Advances in Cryptology- Eurocrypt '98, Springer-Verlag, 184-200, 1998.

[12] Rivest, R.L. Shamir, A. and Adleman, L.M. *A Method for Obtaining Digital Signatures and Public Key Cryptosystems.* Communications of the ACM 21(2), 120-126, Feb 1978.

[13] van der Kulk, W. *On polynomial rings in two variables,* Nieuw Archief voor Wiskunde. vol 3, I(1953).