A Public Key System With Signature And Master Key Functions

T. Moh[*]

received April 1997, revised May 1997 and Oct 1997

Abstract

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 component expression of π be $(\pi_1(x_1, \cdots, x_{n+r}), \cdots, \pi_{n+r}(x_1, \cdots, x_{n+r}))$. Let the restriction of π to a subspace be $\hat{\pi}$ as $\hat{\pi} = (\pi_1(x_1, \cdots, x_n, 0, \cdots, 0), \cdots, \pi_{n+r}(x_1, \cdots, x_n, 0, \cdots, 0)) = (f_1, \cdots, f_{n+r}) : \mathbf{K}^n \mapsto \mathbf{K}^{n+r}$. The field $\mathbf{K}$ and the polynomial map $(f_1, \cdots, f_{n+r})$ will be announced as the public key. Given a plaintext $(x'_1, \cdots, x'_n) \in \mathbf{K}^n$, let $y'_i = f_i(x'_1, \cdots, x'_n)$, then the ciphertext will be $(y'_1, \cdots, y'_{n+r}) \in \mathbf{K}^{n+r}$. Given ϕ_i and $(y'_1, \cdots, y'_{n+r})$, it is easy to find $\phi_i^{-1}(y'_1, \cdots, y'_{n+r})$. Therefore the plaintext can be recovered by $(x'_1, \ldots, x'_n, 0, \cdots, 0) = \phi_1^{-1}\phi_2^{-1}\phi_3^{-1}\phi_4^{-1}\hat{\pi}(x'_1, \cdots, x'_n) = \phi_1^{-1}\phi_2^{-1}\phi_3^{-1}\phi_4^{-1}(y'_1, \cdots, y'_{n+r})$. 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 informations 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

In this article we will introduce a public key system. This is the first time that the theory of *tame* automorphism groups (see section 2) of the affine spaces $\mathbf{K}^{n+r}$ is applied to formulate a public key system. For readers' convenience, the necessary materials about tame automorphism groups are compiled in section 2. The theory of automorphism groups is well studied and difficult. Its short history is outlined in section 3. Section 3 is non-essential to our discussions, and it can be skipped.

Let π be an element in the tame automorphism group. Although the inverse map π^{-1} exists mathematically, it is impractical to write down the **polynomial expression** of π^{-1}, which gives the formula of π^{-1} for all points in $\mathbf{K}^{n+r}$, due to high degrees and the large number of terms (in our scheme, the number of terms $> 10^{92}$, see section 10). **Corollary 2**

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

and **Corollary 3** of section 2 point out that at one known point $(y_1', \cdots, y_{n+r}')$, the **value** (as a numerical point in $\mathbf{K}^{n+r}$) of the map π^{-1} can be readily computed if the decomposition $\pi = \prod \ \phi_i$ is given, where ϕ_i's are tame automorphisms.

In our public key system, we further employ an 'embedding map' $\hat{\pi}$:$\mathbf{K}^n \mapsto \mathbf{K}^{n+r}$ as the restriction of an automorphism map π. Note that there is no inverse map for an embedding map. To illustrate the feasibility of our system, we give an explicit scheme for $n = 64$, $n + r = 100$. Let $\pi = \phi_4\phi_3\phi_2\phi_1$, where ϕ_i's are tame automorphisms, with component expression $\pi = (\pi_1(x_1, \cdots, x_{100}), \cdots, \pi_{100}(x_1, \cdots, x_{100}))$, and $\hat{\pi} = (\pi_1(x_1, \cdots, x_{64}, 0, \cdots, 0), \cdots, \pi_{100}(x_1, \cdots, x_{64},\ 0, \cdots, 0))$. Let $f_i(x_1, \cdots, x_{64}) = \pi_i(x_1, \cdots, x_{64}, 0 \cdots, 0)$. Then we have $\hat{\pi} = (f_1, \cdots, f_{100})$:$\mathbf{K}^{64} \mapsto \mathbf{K}^{100}$, where the 100 polynomials $\{f_1, \cdots, f_{100}\}$ are of degree 2 in 64 variables. The polynomial map $\hat{\pi} = (f_1, \cdots, f_{100})$ and the finite field $\mathbf{K}$ will be announced as the public key. Let $(x_1', \cdots, x_{64}') \in \mathbf{K}^{64}$ be a plaintext. Let $y_i' = f_i(x_1', \cdots, x_{64}')$ for $i = 1, \cdots, 100$. Then the ciphertext is $(y_1', \cdots, y_{100}') \in \mathbf{K}^{100}$. Given a ciphertext $(y_1', \cdots, y_{100}')$ and the set of maps $\{\phi_1, \phi_2, \phi_3, \phi_4\}$, the plaintext can be recovered by $(x_1', \cdots, x_{64}', 0 \cdots, 0) = \phi_1^{-1}\phi_2^{-1}\phi_3^{-1}\phi_4^{-1}\ (y_1', \cdots, y_{100}')$ (see **Corollaries** 2 & 3, section 2). The set of maps $\{\phi_1, \phi_2, \phi_3, \phi_4\}$ is the private key. It is easy for users to select individual schemes, and there are plenty of them (see sections 6 and 7). The computations are over the finite field $\mathbf{K}$, and hence fast. They can also be carried out in a parallel way. The scheme has the functions of error-detect, master key and signature (see section 9).

The attacker faces a difficult job. Mathematically, there is no known way to **recover** the private key $\{\phi_1, \phi_2, \phi_3, \phi_4\}$ from the public key $\hat{\pi}$ and the field $\mathbf{K}$. The **inverse** of the map $\hat{\pi}$ can not be found since it does not exist. Due to the selection of ϕ_i in our scheme (see section 6), for each individual scheme, the highest homogeneous parts (which are of degree 2) of polynomials $\{f_1, \cdots, f_{100}\}$ are all **linearly independent** (see section 6), hence any linear combination of $\{f_i\}$ will not produce a polynomial of smaller degree. A search for **polynomial relations** among $\{f_i\}$ will involve vector space of dimension $3.2(10^{11})$ which is beyond the present technology (see section 10). The attacker may try to construct $\phi_1^{-1}\phi_2^{-1}\phi_3^{-1}\phi_4^{-1}\hat{\pi}$ step by step. The time needed to **find** $\phi_4^{-1}\hat{\pi}$, according to the analysis of Part III of section 10, is at least 10^{317} years. The **random attack** is ineffective (see Part IV of section 10). The present scheme can withstand known attacks (see section 10).

2 Mathematical background

Let $\mathbf{K}$ be a field. Let $\mathbf{K}^{n+r}$ be the affine space of dimension $n+r$ over $\mathbf{K}$. Let $\mathbf{K}[x_1, \cdots, x_{n+r}]$ be the polynomial ring of $n + r$ variables, $x_1, \cdots, x_{n+r}$, over $\mathbf{K}$. Let σ be a polynomial map, $\sigma : \mathbf{K}^{n+r} \mapsto \mathbf{K}^{n+r}$. The *induced map* $\sigma^* : \mathbf{K}[x_1, \cdots, x_{n+r}] \mapsto \mathbf{K}[x_1, \cdots, x_{n+r}]$ defined as

$$\sigma^*(p(x_1, \cdots, x_{n+r})) = p(\sigma(x_1, \cdots, x_{n+r}))$$

where $p(x_1, \cdots, x_{n+r})$ is any polynomial in $\mathbf{K}[x_1, \cdots, x_{n+r}]$. A map σ is said to be an '*invertible linear transformation*' if σ^* sends linear polynomials to linear polynomials and is invertible. For example, translations and rotations are invertible linear transformations.

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 $h_{i,j}$,

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

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 $x_j = \phi_{i,j}^{-1}(y_1, \cdots, y_{n+r}) = y_j - h_{ij}(\phi_{i,j+1}^{-1}(y_1, \cdots, y_{n+r}), \cdots, \phi_{i,n+r}^{-1}(y_1, \cdots, y_{n+r}))$, 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 term of variables,

$$\begin{aligned}
&\phi_{i,4}^{-1}(y_1, \cdots, y_4) = y_4\\
&\phi_{i,3}^{-1}(y_1, \cdots, y_4) = y_3 - h_{i3}(y_4)\\
&\phi_{i,2}^{-1}(y_1, \cdots, y_4) = y_2 - h_{i2}(y_3 - h_{i3}(y_4), y_4)\\
&\phi_{i,1}^{-1}(y_1, \cdots, y_4) = y_1 - h_{i1}(y_2 - h_{i2}(y_3 - h_{i3}(y_4), y_4), y_3 - h_{i3}(y_4), y_4)
\end{aligned}$$

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. As shown in section 9, the number of terms in π^{-1} in our scheme is greater than 10^{254}. 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 term 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 $x_j' = \phi_{i,j}^{-1}(y_1', \cdots, y_{n+r}') = y_j' - h_{ij}(x_{j+1}', \cdots, x_{n+r}')$ 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 in [16] 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 six year old Jacobian Conjecture ([3]) for 2-dimensional space. For embedding theory ([1], [12], [13]), 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 difficult arguments. In the case of finite fields, the embedding problem is open for $n = 1$ and $n + r = 2$, see [12]. 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 [13], Nagata constructed an automorphism σ 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 (Some unproved conjecture about factorization for $n + r = 3$ is listed in [13]. 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 $n + r \geq 3$, m a positive integer 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

To implement the above principle, we use a Component $\mathbf{Q}_8$ (see below). This component does **not** need to vary according to users. It can be made as part of the hardware. The user will select some other 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*(ℓ). If there is no such polynomial Q, then we define *gendeg*(ℓ)$= \infty$.

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

Component $\mathbf{Q}_8$: Let the field $\mathbf{K}$ be of 2^m elements, $t = 19$ and $s = 30$. Let

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

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_{29}^2 which produces x_{18}^4. Therefore the polynomial R must involve q_{28}^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 degree of polynomials Q_8 and its complexity. In part **II** of section 9, we will have a detailed discussion. With the degree 8, an attacker is forced to consider a vector space of dimension $3.2(10^{11})$ in our scheme (see section 10, **II**). The dimension is too high to be handled by the present technology. The degree of polynomials 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

$$\begin{aligned} &(1)\text{For } i = 1, \cdots, n, \quad \text{we always have } b_i \neq 0, a_{i,j} = 0, \text{for } j = n+1, \cdots, n+r \\ &\qquad\qquad \text{and at least half of the remainning } a_{i,j} \text{ are non-zero.} \\ &(2)\text{For } i = n+1, \cdots, n+r, \quad \text{we always have } \phi_{1,i} = x_i. \end{aligned}$$

6 Implementation Scheme

Let $n = 64,\ r = 36,\ n{+}r = 100$. Let the field $\mathbf{K}$ be $\mathbf{F}_{2^m}$, the finite field of 2^m elements, where m is a positive integer ≥ 8. 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 $q_1, \cdots, q_{29}$, the generating polynomial $Q_8(q_1, \cdots, q_{29})$. 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 $(1)^* - -(11)^*$,

$$\begin{aligned}
&(1)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i, \ \text{for } i = 1, 2. \\
&(2)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{i-1}x_{i-2}, \ \text{for } i = 3, \cdots, 9. \\
&(3)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{[i-1]}^2 + x_{[i]}x_{[i-5]} + x_{[i+1]}x_{[i+6]}, \ \text{for } i = 10, \cdots, 17, \\
&(3)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{[i-1]}x_{[i+1]} + x_{[i]}x_{[i+4]}, \ \text{for } i = 18, \cdots, 25, \\
&(3)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{[i-1]}x_{[i+1]} + x_{[i+2]}x_{[i+5]}, \ \text{for } i = 26, \cdots, 30, \\
&(4)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{i-10}^2, \ for\ i = 31, \cdots, 60 \\
&(5)^* : \phi_{2,61}(x_1, \cdots, x_{100}) = x_{61} + x_9^2, \\
&(5)^* : \phi_{2,62}(x_1, \cdots, x_{100}) = x_{62} + x_{61}^2, \\
&(5)^* : \phi_{2,63}(x_1, \cdots, x_{100}) = x_{63} + x_{10}^2, \\
&(5)^* : \phi_{2,64}(x_1, \cdots, x_{100}) = x_{64} + x_{63}^2, \\
&(6)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + q_{i-64}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{62}), \ for\ i = 65, \cdots, 92 \\
&(7)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + q_{i-92}(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}), \\
&\qquad\qquad for\ i = 93, \cdots, 100
\end{aligned}$$

$(8)^* : \phi_{3,i}(x_1, \cdots, x_{100}) = x_i, \ for \ i = 3, \cdots, 100$
$(9)^* : \phi_{3,2}(x_1, \cdots, x_{100}) = x_2 + Q_8(x_{93}, \cdots, x_{100}, x_{73}, \cdots, x_{92}x_{63}, x_{64})$
$(10)^* : \phi_{3,1}(x_1, \cdots, x_{100}) = x_1 + Q_8(x_{65}, \cdots, x_{92}, x_{61}, x_{62})$

(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. ■

Detailed description of ϕ_2

The invertible linear transformations ϕ_1 and ϕ_4 selected by the user are not complicated. The tame automorphism ϕ_2 needs further explanation. We will write down a concrete example for ϕ_2 as follows,

$(1)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i, \ for \ i = 1, 2$
$(2)^* : \phi_{2,i}(x_1, \cdots, x_{100}) = x_i + x_{i-1}x_{i-2}, \text{ for } i = 3, \cdots, 9,$
$(3)^* : \phi_{2,10}(x_1, \cdots, x_{100}) = x_{10} + x_1^2 + x_2x_5 + x_3x_8$
$(3)^* : \phi_{2,11}(x_1, \cdots, x_{100}) = x_{11} + x_2^2 + x_3x_6 + x_4x_1$
$(3)^* : \phi_{2,12}(x_1, \cdots, x_{100}) = x_{12} + x_3^2 + x_4x_7 + x_5x_2$
$(3)^* : \phi_{2,13}(x_1, \cdots, x_{100}) = x_{13} + x_4^2 + x_5x_8 + x_6x_3$
$(3)^* : \phi_{2,14}(x_1, \cdots, x_{100}) = x_{14} + x_5^2 + x_6x_1 + x_7x_4$
$(3)^* : \phi_{2,15}(x_1, \cdots, x_{100}) = x_{15} + x_6^2 + x_7x_2 + x_8x_5$
$(3)^* : \phi_{2,16}(x_1, \cdots, x_{100}) = x_{16} + x_7^2 + x_8x_3 + x_1x_6$
$(3)^* : \phi_{2,17}(x_1, \cdots, x_{100}) = x_{17} + x_8^2 + x_1x_4 + x_2x_7$
$(3)^* : \phi_{2,18}(x_1, \cdots, x_{100}) = x_{18} + x_1x_3 + x_2x_6$
$(3)^* : \phi_{2,19}(x_1, \cdots, x_{100}) = x_{19} + x_2x_4 + x_3x_7$
$(3)^* : \phi_{2,20}(x_1, \cdots, x_{100}) = x_{20} + x_3x_5 + x_4x_8$
$(3)^* : \phi_{2,21}(x_1, \cdots, x_{100}) = x_{21} + x_4x_6 + x_5x_1$
$(3)^* : \phi_{2,22}(x_1, \cdots, x_{100}) = x_{22} + x_5x_7 + x_6x_2$
$(3)^* : \phi_{2,23}(x_1, \cdots, x_{100}) = x_{23} + x_6x_8 + x_7x_3$
$(3)^* : \phi_{2,24}(x_1, \cdots, x_{100}) = x_{24} + x_7x_1 + x_8x_4$
$(3)^* : \phi_{2,25}(x_1, \cdots, x_{100}) = x_{25} + x_8x_2 + x_1x_5$
$(3)^* : \phi_{2,26}(x_1, \cdots, x_{100}) = x_{26} + x_1x_3 + x_4x_7$

$(3)^* : \phi_{2,27}(x_1, \cdots, x_{100}) = x_{27} + x_2x_4 + x_5x_8$

$(3)^* : \phi_{2,28}(x_1, \cdots, x_{100}) = x_{28} + x_3x_5 + x_6x_1$

$(3)^* : \phi_{2,29}(x_1, \cdots, x_{100}) = x_{29} + x_4x_6 + x_7x_2$

$(3)^* : \phi_{2,30}(x_1, \cdots, x_{100}) = x_{30} + x_5x_7 + x_8x_3$

$(4)^* : \phi_{2,31}(x_1, \cdots, x_{100}) = x_{31} + x_{21}^2,$

$(4)^* : \phi_{2,32}(x_1, \cdots, x_{100}) = x_{32} + x_{22}^2,$

$(4)^* : \phi_{2,33}(x_1, \cdots, x_{100}) = x_{33} + x_{23}^2,$

$(4)^* : \phi_{2,34}(x_1, \cdots, x_{100}) = x_{34} + x_{24}^2,$

$(4)^* : \phi_{2,35}(x_1, \cdots, x_{100}) = x_{35} + x_{25}^2,$

$(4)^* : \phi_{2,36}(x_1, \cdots, x_{100}) = x_{36} + x_{26}^2,$

$(4)^* : \phi_{2,37}(x_1, \cdots, x_{100}) = x_{37} + x_{27}^2,$

$(4)^* : \phi_{2,38}(x_1, \cdots, x_{100}) = x_{38} + x_{28}^2,$

$(4)^* : \phi_{2,39}(x_1, \cdots, x_{100}) = x_{39} + x_{29}^2,$

$(4)^* : \phi_{2,40}(x_1, \cdots, x_{100}) = x_{40} + x_{30}^2,$

$(4)^* : \phi_{2,41}(x_1, \cdots, x_{100}) = x_{41} + x_{31}^2,$

$(4)^* : \phi_{2,42}(x_1, \cdots, x_{100}) = x_{42} + x_{32}^2,$

$(4)^* : \phi_{2,43}(x_1, \cdots, x_{100}) = x_{43} + x_{33}^2,$

$(4)^* : \phi_{2,44}(x_1, \cdots, x_{100}) = x_{44} + x_{34}^2,$

$(4)^* : \phi_{2,45}(x_1, \cdots, x_{100}) = x_{45} + x_{35}^2,$

$(4)^* : \phi_{2,46}(x_1, \cdots, x_{100}) = x_{46} + x_{36}^2,$

$(4)^* : \phi_{2,47}(x_1, \cdots, x_{100}) = x_{47} + x_{37}^2,$

$(4)^* : \phi_{2,48}(x_1, \cdots, x_{100}) = x_{48} + x_{38}^2,$

$(4)^* : \phi_{2,49}(x_1, \cdots, x_{100}) = x_{49} + x_{39}^2,$

$(4)^* : \phi_{2,50}(x_1, \cdots, x_{100}) = x_{50} + x_{40}^2,$

$(4)^* : \phi_{2,51}(x_1, \cdots, x_{100}) = x_{51} + x_{41}^2,$

$(4)^* : \phi_{2,52}(x_1, \cdots, x_{100}) = x_{52} + x_{42}^2,$

$(4)^* : \phi_{2,53}(x_1, \cdots, x_{100}) = x_{53} + x_{43}^2,$

$(4)^* : \phi_{2,54}(x_1, \cdots, x_{100}) = x_{54} + x_{44}^2,$

$(4)^* : \phi_{2,55}(x_1, \cdots, x_{100}) = x_{55} + x_{45}^2,$

$(4)^* : \phi_{2,56}(x_1, \cdots, x_{100}) = x_{56} + x_{46}^2,$

$(4)^* : \phi_{2,57}(x_1, \cdots, x_{100}) = x_{57} + x_{47}^2,$

$(4)^* : \phi_{2,58}(x_1, \cdots, x_{100}) = x_{58} + x_{48}^2,$

$(4)^* : \phi_{2,59}(x_1, \cdots, x_{100}) = x_{59} + x_{49}^2,$

$(4)^* : \phi_{2,60}(x_1, \cdots, x_{100}) = x_{60} + x_{50}^2,$

$(5)^* : \phi_{2,61}(x_1, \cdots, x_{100}) = x_{61} + x_{9}^2,$

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

$$
\begin{aligned}
&(5)^* : \phi_{2,63}(x_1, \cdots, x_{100}) = x_{63} + x_{10}^2,\\
&(5)^* : \phi_{2,64}(x_1, \cdots, x_{100}) = x_{64} + x_{63}^2,\\
&(6)^* : \phi_{2,65}(x_1, \cdots, x_{100}) = x_{65} + q_1(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{65} + x_9 + x_{11}x_{15},\\
&(6)^* : \phi_{2,66}(x_1, \cdots, x_{100}) = x_{66} + q_2(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{66} + x_{11}^2 + x_{12}x_{16},\\
&(6)^* : \phi_{2,67}(x_1, \cdots, x_{100}) = x_{67} + q_3(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{67} + x_{12}^2 + x_{13}x_{53},\\
&(6)^* : \phi_{2,68}(x_1, \cdots, x_{100}) = x_{68} + q_4(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{68} + x_{12}x_{14},\\
&(6)^* : \phi_{2,69}(x_1, \cdots, x_{100}) = x_{69} + q_5(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{69} + x_{12}x_{54},\\
&(6)^* : \phi_{2,70}(x_1, \cdots, x_{100}) = x_{70} + q_6(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{70} + x_{13}x_{16},\\
&(6)^* : \phi_{2,71}(x_1, \cdots, x_{100}) = x_{71} + q_7(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{71} + x_{13}x_{14},\\
&(6)^* : \phi_{2,72}(x_1, \cdots, x_{100}) = x_{72} + q_8(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{72} + x_{16}^2 + x_{14}x_{54},\\
&(6)^* : \phi_{2,73}(x_1, \cdots, x_{100}) = x_{73} + q_9(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{73} + x_{15}^2 + x_{51}x_{52},\\
&(6)^* : \phi_{2,74}(x_1, \cdots, x_{100}) = x_{74} + q_{10}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{74} + x_{51}^2 + x_{55}x_{56},\\
&(6)^* : \phi_{2,75}(x_1, \cdots, x_{100}) = x_{75} + q_{11}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{75} + x_{52}^2 + x_{57}x_{58},\\
&(6)^* : \phi_{2,76}(x_1, \cdots, x_{100}) = x_{76} + q_{12}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{76} + x_{16}x_{53},\\
&(6)^* : \phi_{2,77}(x_1, \cdots, x_{100}) = x_{77} + q_{13}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{77} + x_{53}x_{54},\\
&(6)^* : \phi_{2,78}(x_1, \cdots, x_{100}) = x_{78} + q_{14}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{78} + x_{55}^2 + x_{16}x_{51},\\
&(6)^* : \phi_{2,79}(x_1, \cdots, x_{100}) = x_{79} + q_{15}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{79} + x_{56}^2 + x_{54}x_{59},\\
&(6)^* : \phi_{2,80}(x_1, \cdots, x_{100}) = x_{80} + q_{16}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{80} + x_{57}^2 + x_{53}x_{55},\\
&(6)^* : \phi_{2,81}(x_1, \cdots, x_{100}) = x_{81} + q_{17}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62})\\
&\qquad\qquad\quad = x_{81} + x_{58}^2 + x_{54}x_{60},
\end{aligned}
$$

$$\begin{aligned}
(6)^* : \phi_{2,82}(x_1, \cdots, x_{100}) &= x_{82} + q_{18}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{82} + x_{55}x_{59}, \\
(6)^* : \phi_{2,83}(x_1, \cdots, x_{100}) &= x_{83} + q_{19}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{83} + x_{54}x_{55}, \\
(6)^* : \phi_{2,84}(x_1, \cdots, x_{100}) &= x_{84} + q_{20}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{82} + x_{51}x_{56}, \\
(6)^* : \phi_{2,85}(x_1, \cdots, x_{100}) &= x_{85} + q_{21}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{85} + x_{16}x_{56}, \\
(6)^* : \phi_{2,86}(x_1, \cdots, x_{100}) &= x_{86} + q_{22}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{86} + x_{51}x_{59}, \\
(6)^* : \phi_{2,87}(x_1, \cdots, x_{100}) &= x_{87} + q_{23}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{87} + x_{57}x_{60}, \\
(6)^* : \phi_{2,88}(x_1, \cdots, x_{100}) &= x_{88} + q_{24}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{88} + x_{15}x_{54}, \\
(6)^* : \phi_{2,89}(x_1, \cdots, x_{100}) &= x_{89} + q_{25}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{89} + x_{55}x_{58}, \\
(6)^* : \phi_{2,90}(x_1, \cdots, x_{100}) &= x_{90} + q_{26}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{90} + x_{53}x_{58}, \\
(6)^* : \phi_{2,91}(x_1, \cdots, x_{100}) &= x_{91} + q_{27}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{91} + x_{55}x_{60}, \\
(6)^* : \phi_{2,92}(x_1, \cdots, x_{100}) &= x_{92} + q_{28}(x_9, x_{11}, \cdots, x_{16}, x_{51}, \cdots, x_{60}, x_{61}, x_{62}) \\
&= x_{92} + x_{54}x_{57}, \\
(7)^* : \phi_{2,93}(x_1, \cdots, x_{100}) &= x_{93} + q_1(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{93} + x_{10} + x_{17}x_{15}, \\
(7)^* : \phi_{2,94}(x_1, \cdots, x_{100}) &= x_{94} + q_2(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{94} + x_{17}^2 + x_{18}x_{16}, \\
(7)^* : \phi_{2,95}(x_1, \cdots, x_{100}) &= x_{95} + q_3(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{95} + x_{18}^2 + x_{19}x_{53}, \\
(7)^* : \phi_{2,96}(x_1, \cdots, x_{100}) &= x_{96} + q_4(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{96} + x_{18}x_{20}, \\
(7)^* : \phi_{2,97}(x_1, \cdots, x_{100}) &= x_{97} + q_5(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{97} + x_{18}x_{54}, \\
(7)^* : \phi_{2,98}(x_1, \cdots, x_{100}) &= x_{98} + q_6(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{98} + x_{19}x_{16}, \\
(7)^* : \phi_{2,99}(x_1, \cdots, x_{100}) &= x_{99} + q_7(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\
&= x_{99} + x_{19}x_{20},
\end{aligned}$$

$$\begin{aligned}(7)^* : \phi_{2,100}(x_1, \cdots, x_{100}) &= x_{100} + q_8(x_{10}, x_{17}, \cdots, x_{20}, x_{15}, x_{16}, x_{51}, \cdots, x_{60}, x_{63}, x_{64}) \\ &= x_{100} + x_{16}^2 + x_{20}x_{54},\end{aligned}$$

Detailed description of $\phi_3\phi_2$

Let $(x'_1, \cdots, x'_{64})$ be the plaintext. The machine will read it as $(x'_1, \cdots, x'_{64}, 0, \cdots, 0)$ with thirty six extra zeroes. Applying ϕ_1, we have $(z'_1, \cdots, z'_{64}, 0, \cdots, 0)$. Then applying ϕ_2 to the previous results, we have $(u'_1, \cdots, u'_{100})$, where

$$\begin{aligned}&(1)^{**} : u'_1 = z'_1, \\ &(2)^{**} : u'_2 = z'_2, \\ &\cdots\cdots \\ &(6)^{**} : u'_{65} = z'_9 + z'_{11}z'_{16}, \\ &(6)^{**} : u'_{66} = {z'_{11}}^2 + z'_{12}z'_{16}, \\ &\cdots\cdots\end{aligned}$$

Further applying ϕ_3 to the above results, we have $(v'_1, \cdots, v'_{100})$, where

$$\begin{aligned}v'_i &= u'_i, \qquad \text{for} \quad i = 100, 99, , \cdots, 3 \\ v'_2 &= u'_2 + Q_8(u'_{93}, \cdots, u'_{100}, u'_{73}, \cdots, u'_{92}, u'_{63}, u'_{64}) \\ &= z'_2 + (z'_{64})^2 \\ v'_1 = u'_1 + Q_8(u'_{65}, \cdots, u'_{92}, u'_{61}, u'_{62}) & \\ &= z'_1 + (z'_{62})^2\end{aligned}$$

The net effect of $\phi_3\phi_2$ is that the point $(z'_1, \cdots, z'_{64}, 0, \cdots, 0)$ is sent to a point $(v'_1, \cdots, v'_{100})$ with all v'_i as quadratic polynomials of $z'_1, \cdots, z'_{64}$. Moreover, if we treat $z'_1, \cdots, z'_{64}$ as variables, then the parts of degree 2 of v'_i as polynomials in $z'_1, \cdots, z'_{64}$ are all linearly independent. Note that all v'_i's are square-free in the sense that none of them is a square of a linear polynomial. The above facts can be verified by direct checking. Taking the invertible linear transformations ϕ_1, ϕ_4 into consideration, it is easy to see that all the polynomials f_i are polynomials of degree 2, with degree 2 homogeneous parts generating a vector space of dimension 100. Furthermore, it can be shown that all elements except zero in the vector space generated by the set $\{f_1, \cdots, f_{100}\}$ are square-free.

7 Plaintexts, Users and Compactness

Let us count the possible **number of plaintexts**. Since the number of plaintexts is the number of choices for $x'_1, \cdots, x'_{64}$, we see that there are 2^{64m} such plaintexts. To have a rich scheme and to prevent attackers from forming tables of plaintext-ciphertext, and to avoid the usage of the following identities over the finite field $\mathbf{F}_{2^m}$ to cut down the degrees,

$$x_i^{2^m} - x_i = 0$$

it is suggested to use $m \geq 8$ as stated in our scheme.

Of equal importance to having a large number of possible plaintexts is having lots of possible **users**. In order to allow for many such users, we first get an expression for this number in terms of m and 64. This amounts to counting the number of automorphisms π of the form $\pi = \phi_1\phi_2\phi_3\phi_4$. Assuming that a negligible proportion of these automorphisms π have more than one representation $\pi = \phi_1\phi_2\phi_3\phi_4 = \phi_1'\phi_2'\phi_3'\phi_4'$, the number of users is asymptotic to (choices for ϕ_4) × (choices for ϕ_3) × (choices for ϕ_2) × (choices for ϕ_1). The condition of type A for ϕ_1 is not a big restriction. We may use the possible numbers for the invertible linear transformations as an estimation for ϕ_1, ϕ_4. The number of invertible linear transformations ϕ_1 is $\prod_{j=0}^{n-1}(2^{mn} - 2^{mj}) = 2^{mn(n-1)/2}\prod_{j=1}^{n}(2^{mj} - 1)$. For our selection of $n = 64, n + r = 100$, it is easy to see that the number for $\phi_1 > 2^{2628m}$. A similar count of terms of ϕ_4 results in the total possible number of users $> 2^{7678m}$.

Now let us look at the **compactness** of the scheme. Note that Component $\mathbf{Q}_4$ are fixed for all users. To have an individual scheme, the user needs to select ϕ_1, ϕ_4 which can be generated by certain computer programs. The end results are 100 quadratic polynomials $(f_1, \cdots, f_{100})$ in 64 variables $x_1, \cdots, x_{64}$. It is easy to see that the number of terms of polynomials of degree 2 is (67)(64)/2! and we have 100 polynomials, therefore the total number of terms is $214,400$ (for another aoftware implementation TTM 2.3, the number is 61,920). We believe that the number can be further reduced. The expense to the sender is mainly in evaluating polynomials $y_i' = f_i(x_1', \cdots, x_{64}')$. Note that the computations are carried over the finite field $\mathbf{K}$, hence fast. They can also be computed in a parallel way. Thus the process can be sped up several hundred fold. On the receiver's side, the total number of terms for $\phi_1, \phi_2, \phi_3, \phi_4$ is $17,000$. The legitimate receiver needs to evaluate $\phi_1^{-1}\phi_2^{-1}\phi_3^{-1}\phi_4^{-1}(y_1', \cdots, y_{100}')$ (according to **Corollaries** 2 & 3) which is not expensive.

The number of terms will be reduced, and the efficiency will be improved, as the technique is improving, and new Components $\mathbf{Q}_4$ are being discovered.

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 available. They use C Language. For m=8, the rates of expansion of data are 1.4, 1.56, 1.63 and 1.5 respectively. They have been used on various machines listed below,

$$200MhzPowerPC604ew/1024Kcache.$$
$$225MhzPowerPC603ew/256Kcache,$$
$$167MhzUltrasparcw/512Kcache.$$
$$167MhzPentiumw/512Kcache.$$

Their encrypting speeds on 200 Mhz PowerPC 604e (w/1024K cache: virtual memory off) are listed in the following

The decrypting speed is in general 10 to 20 times faster than the encrypting speed. The PC software TTM 2.5 is faster than a possible hardware implementation for RSA 1024. According

	speeds of software implementations
TTM 1.9,	94,939b/s
TTM 2.1,	106.224b/s.
TTM 2.3,	207,000b/s
TTM 2.5,	300,000b/s.

Table 1:

to the opinion of certain expert, a couple added instruction about finite field multiplication in the chip architecture would increase the speed of software implementations at least 10–16 times. If it is done, then our software implementations would reach a few million bits per second for the PC and match the speed of the software for secret-key encryptions (DES etc.). A software in assembly language should be faster.

The newly announced "Motorola's high-performance vector parallel processing expansion to the PowerPC architecture" (http://www.mot.com/SPS/PowerPC/AltiVec/) will increase the speed of our algorithm conservatively by a factor of five; perhaps by 10, depending on what kind of memory subsystem it's attached to. Although this doesn't have *exactly* the instructions we need, this will make the speeds of our softwares several million bits per second. It will be a while before this technology will be available, but it is *very* exciting.

It is possible to encrypt voice communications(64,000 bits/sec) and video phone (1,250,000 bits/sec for Motorola's new chip) by those softwares on an ordinary PC. Note that in comparison, RSA toolkit BSAFE 3.0 for 1024 is 7 Kb/s. It is conceivable that a hardware implementation, using finite field multiplication and parallel computing, would approximate the speed of the fastest hardware implementation of the triple DES 128.

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

Select a group of indices, S, from $\{94, \cdots, 100\}$ as in our scheme (in general, a few extra indices $101, 102, \cdots$ may be 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 $i \in S$.

A different way to produce a master key is to find a polynomial $Q'(h_1, \cdots, h_{16}, \cdots, h_{16+s})$, such that both Q' and its specialization $Q'(h_1, \cdots, h_{16}, 0, \cdots, 0)$ can be used to construct a public key scheme. We require that ϕ_1 to keep space $\{(c_1, \cdots, c_{100}, 0, \cdots, 0) : c_i \in \mathbf{K}\}$ $=\mathbf{K}^{100} \times 0 \times \cdots \times 0$ invariant and use the specialization $x_i \mapsto 0$ for $i = 101, \cdots, 100 + s$ to create a subordinate key from the original key (i.e., the master key).

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 an 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} ([9]). 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 $y'_1, \cdots, y'_{100}$.

For the method (1), note that the form of the map π is not given to the public, the attacker has to guess π correctly. Furthermore, it follows from **Corollary 2** and the explicit expression of ϕ_2 in the scheme that we have,

$$max\{deg_{y_1}\ \phi_{2,j}^{-1}(y_1, \cdots, y_{100})\} \geq 2^8$$

Since ϕ_1, ϕ_4 are linear transformations, the **theoretic total number of terms** in π^{-1} is $100(\prod_{i=1}^{100}(2^8 + i))/100! > 10^{92}$. It is too large to be practical. To use the method (2), the attacker has to give an estimation of the degrees of g_i. According to our previous analysis, there are too many terms in g_i's for the method to be useful. As for the method (3), the resultant is only practical for polynomials of very few variables, it is impractical in our scheme.

At this moment, the above three direct methods are ineffective. The only possible way of attacking is to recover ϕ_i's or their equivalent forms.

II. Search for Polynomial Relations

Although polynomials $\{f_1, \cdots, f_{100}\}$ are linearly independent, the attacker may search for polynomial relations, which are linear relations of monomials, among them. Knowing the recipe of the construction of the public key scheme, the attacker may launch a '**step by step search**' to search for useful polynomials in the ring $\mathbf{K}[f_1, \cdots, f_{100}]$ as follows. The attacker considers all monomials $\prod\ f_i^{n_i}$ with degrees $\sum n_i$ less than or equal to some fixed number d (i.e., $\sum n_i \leq d$). Then the attacker shall find some relation among those monomials to produce a power of a linear polynomial $\ell^s(x_1, \cdots, x_{64}) = r(f_1, \cdots, f_{100})$. Note that the linear polynomial ℓ may be included in a set of variables, say $\{\ell, x_2, \cdots, x_{64}\}$. Let f_i take the value y'_i for all i, then the value of the linear polynomial ℓ can be computed, which is a s-th root of $r(y'_1, \cdots, y'_{100})$. Once the value of the linear polynomial ℓ is substituted in f_i which is expressed in the new set of variables $\{\ell, x_2, \cdots, x_{64}\}$, the total number of variables will be reduced from 64 to 63. The attacker would achieve a reduction in this way. However, in our scheme, one can show that, with elementary analysis and tedious computations, the attacker has to consider the vector space of all polynomials of degree 8 in $f_1, \cdots, f_{100}$. The dimension of homogeneous polynomials of degree 8 is $C_8^{107} \approx 3.2(10^{11})$. The dimension is too high to be handled by present technology. We can select polynomials Q_8 with higher degrees to defend the scheme if necessary.

III. Identify the Highest Homogeneous Parts

The attacker may try to find $\phi_4^{-1}\hat{\pi}$ first. Let us introduce a new number, $diffdim$. Let h_i be a polynomial of $(x_1, \cdots, x_t)$ with the homogeneous part q_i of the highest degree. Let us define the *diffdim* (h_i)=dim (the vector space generated by $\{\frac{\partial q_i}{\partial x_j} : j = 1, \cdots, t\}$). Note that *diffdim* h_i= $0, 2, 4$ in ϕ_2, ϕ_3.

Let $V = \{$ the 100 dimensional vector space generated by $f_i\}$. The attacker shall try to pick up the i-th component, $(\phi_4^{-1}\hat{\pi})_i$, of the polynomial map $\phi_4^{-1}\hat{\pi}$ from the vector space V. Let us study the highest homogeneous parts of elements in V.

Let the highest homogeneous part of f_i be r_i. Let $U = \{$the vector space generated by $r_i\}$. Let q_i=the highest homogeneous part of $(\phi_4^{-1}\hat{\pi})_i$. It is easy to see that for some k, $diffdim(q_k) = 4$. The attacker wants to find suitable numbers $(z_1, \cdots, z_{100})$ such that

$$w = \sum_{i=1}^{100} z_i r_i = q_k$$

Let us consider a way to find the above w. A necessary condition is that all partial derivatives, w_j, of w with respect to x_j span a vector space of dimension 4. Let $r_{ij} = \sum a_{ijk} x_k$ be the partial derivative of r_i with respect to x_j.

The attacker attempts to list all elements in U with $diffdim$ 4. The attacker may use the above information as follows.

For a fixed i, let A_i be the 64×64 coefficient matrix (a_{ijk}), and $A = \sum_{i=1}^{100} z_i A_i$. Let us assume that A is of rank 4 with coefficient linear homogeneous polynomials in variables $z_1, \cdots, z_{100}$. It produces 100 homogeneous equations in 100 variables of degree 5. It follows from pg 75 of [8] that the upper bound of time required to solve the equations is $O(m^2(100)^2 5^{500}) \approx m^2 10^{337}$. (For the discussion of $difdim(h_i) = 2$, the same arguments give $O(m^2(100)^2 3^{500})$)

Note that there are $3(10^7)$ seconds in a year. Let us use a futuristic computer which operates 10^{12} shift operations per second. It will take up to $m^2 10^{317}$ years to list all elements in U with $diffdim$ 4. Then the attacker can list all elements in V with $diffdim$ 4. Similarly, the attacker can list all elements in V with $diffdim$ $0, 2$.

Only after the above elements of V are listed, can the attacker consider the task of picking up (non-homogeneous) polynomials $(\phi_4^{-1}\hat{\pi})_i$, and thus undo the effect of ϕ_4.

IV. Brute Force Attack

The attacker may use linearly independent linear polynomials $\{v_1, \cdots, v_{64}\}$ to express $\{f_1, \cdots, f_{100}\}$. Then the attacker assign random values from the field K for a subsets of $\{v_1, \cdots, v_{64}\}$ to see if the assigned values are correct. It is easy to see that the said subset should have at least 15 elements to have a chance of testing to find out if the assigned values are correct. **Assuming** it only take one clock cycle to test if a set of 15 random numbers is correct, the attacker still need 3×10^{22} misp (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.

APPENDIX

Acknowledgement

We wish to express our thanks to J. M. Acken, A. Odlyzko, H. Lenstra, P. Montgomery, E. Croot, C. Bajaj, S. Wagstaff, F. Chao, R. Osawa, P. Huang for discussions.

After we sent out our original draft in 1995, P. Montgomery responded in showing us a successive attack on the example of four variables in that draft. We produced another version which defended against the ‘analysis of the highest homogeneous parts’ to stand the attack proposed by P. Montgomery. Then A. Sathaye launched a ‘step by step search’, i.e., searching for relations among all monomials of $\{f_i\}$ with some fixed degree, which could theoretically crack our second version. Only then did we understand that the attack of P. Montgomery was the beginning of a ‘step by step search’. The point is that the final polynomials $(f_1, \cdots, f_{n+r})$ are of various degrees, and they can be grouped and analyzed according to their degrees. In our previous versions, for a particular degree, there are only a few polynomials (or it is the same to say, a small dimensional vector space). Those polynomials can be discovered by a ‘step by step search’ even though they were covered up at the beginning by a linear transformation of the vector space generated by $\{f_i\}$. Therefore, the previous public key system would dissolve step by step theoretically.

Due to our intention of including a ‘master key function’ (see section 8) in the system, we had considered a *specialization* $x_{n+r} \mapsto x_i$. Independently, from the point of view of *embedding theory* (cf [1], [12]), A. Sathaye suggested that we should consider $x_{n+r} \mapsto 0$. These two approaches were identical up to a linear transformation. Very soon we solved the technical problem involved (see section 5). In our present public key scheme, the resulting polynomials are of degree two uniformly and their degree two homogeneous parts are linearly independent (see the end of section 6).

Applying ‘step by step search’ to the present public key scheme, an attacker will have to consider vector spaces of dimensions 3.2(10^{11} see section 10, **II**). The dimension is too high to be handled by the present technology. Moreover, since the encoding scheme uses 'embedding maps' without inverses, it is impossible to crack this scheme by looking for inverses. The present scheme can withstand known attacks. We are especially grateful to A. Sathaye for the enlightening discussions and for checking our computations in section 5.

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] Bajaj, C. Garrity, T. Warren, J. *On the Application of Multi-Equational Resultants.* Purdue University, Dept of C. S. Technical Report CSD-TR-826, 1988.

[3] 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.

[4] Berlekamp, E.R. *Factoring polynomials over finite fields.*Bell System Tech. J. pp 1853-1859, vol 46, 1967.

[5] Brandstrom, H. *A public-key cryptosystem based upon equations over a finite field.* Cryptologia, pp 347-358, vol 7, 1983.

[6] Brent, R., and Kung, H. *Fast Algorithms for Manipulating Formal Power Series.*Journal of the ACM, pp 581-595, vol 25 no 4, 1978.

[7] Cohen, Henri *A Course in Computational Algebraic Number Theory.* Springer-Verlag. Berlin, 1993.

[8] Canny, John F. *Complexity of Robot Motion Planning.* The MIT Press, Cambridge, Massachusetts, 1988.

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

[10] Lidl, R *Finite fields.* Addison-Wesley, Reading, Massachusetts, 1983.

[11] Lidl, R *On Cryptosystems Based on Polynomials and Finite Fields.* Advances in Cryptology (Proceedings of EUROCRYPT 84), pp 10-15, 1983.

[12] 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.

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

[14] Niederreiter, H. *New Deterministic Factorization Algorithms for Polynomials over Finite Fields.* Contemporary Mathematics (Finite Fields) (AMS), vol 168, 1993.

[15] 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.

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