Proposal of PPS Multivariate Public Key Cryptosystems

Shigeo Tsujii[†] Kohtaro Tadaki[†‡] Masahito Gotaishi[†]
Ryo Fujita[†] Masao Kasahara[†§]

[†] Research and Development Initiative, Chuo University
1–13–27 Kasuga, Bunkyo-ku, Tokyo 112–8551, Japan
[‡] JST, CREST
[§] Faculty of Informatics, Osaka Gakuin University
2–36–1 Kishibe-minami, Suita-shi, Osaka 564–8511, Japan

Abstract. In this paper we propose a new MPKC, called PPS, based on (i) the 2-layer nonlinear piece in hand method, (ii) PMI, and (iii) STS. The PPS is a specific MPKC obtained by applying the 2-layer nonlinear piece in hand method to STS, in the manner that the rank and randomness of the lower rank steps in the original secret polynomial vector of STS are enhanced by adding a perturbation polynomial vector and moreover PMI is used in the auxiliary part. The PPS overcomes the drawbacks of the three schemes by the advantage of the three schemes themself. Thus, PPS can be thought to be immune simultaneously from the algebraic attacks, such as the Gröbner bases attacks, from the rank attacks, and from the differential attacks.

Key words: post-quantum cryptography, public key cryptosystem, multivariate public key cryptosystem, Piece In Hand, PMI, STS, multivariate polynomial, signature scheme

1 Introduction

There are two mainstreams in multivariate public key cryptosystems (MPKCs, for short). One is *the Matsumoto-Imai cryptosystem* (MI, for short), proposed by Matsumoto and Imai [19, 20], and another is an MPKC based on *the sequential solution method*, invented and developed by Tsujii, et al. [28, 29, 30], and independently invented by Shamir [26].

MI, which is regarded as the origin of the practical MPKCs at present, was invented around 1983. After it was presented within Japan [19], it was presented at EUROCRYPT '88 [20]. Thereafter, MI was successfully broken by Patarin in 1995 [22]. In the next year, Patarin proposed an MPKC, called *hidden field equation* (HFE, for short), by generalizing the trapdoor of MI [23]. HFE is more secure than MI, as discussed by various researchers so far. However, it is not bijective. In addition, its security against algebraic attack is limited. Although its security can be increased by increasing the HFE degree d, the computational complexity of decryption also increases in that case.

In 1985, the MPKC based on the sequential solution method was proposed by Tsujii, et al. [28]. After it was attacked by Kaneko, et al. [15], its improved version was proposed by Tsujii, et al. [30] in 1989. A signature scheme using almost the same trapdoor as the sequential solution method was proposed by Shamir [26] at CRYPTO '93, and was attacked by Coppersmith, et al. [1]. Recently, an English translation of the Japanese paper [30] was uploaded in the Cryptology ePrint Archive [33]. Thereafter, the MPKC proposed by [30] was attacked by Ding, et al. [8] in 2008.

The original sequential solution method appends variables one by one to polynomial components of the trapdoor secret polynomial vector. Kasahara, et al. [16, 17] proposed MPKCs by generalizing the sequential solution method, where variables are appended r by r, instead of one by one, to polynomial components of the trapdoor secret polynomial vector. Later on, this type of MPKCs was called *stepwise triangular scheme* (STS, for short) by [41]. Several attacks including Gröbner bases and rank attacks have been applied to STS (see e.g. [41]).

It was discovered in 1994 that RSA cryptosystems, which rely on the difficulty of prime factorization, and elliptic-curve cryptosystems, which rely on the difficulty to compute discrete logarithm, would not remain secure in case where sufficiently large quantum computers are made practical [27]. Afterwards various types of MPKCs have been proposed. This is because solving quadratic equations over finite fields, which MPKCs rely on in general, is NP-complete and it is believed that any quantum computer cannot solve any NP-complete problem in polynomial time. However, almost all MPKCs proposed so far have been solved by appropriate attacks. The major attacks against MPKCs are classified as follows.

(i) **Algebraic attacks,** which are applicable to any type of MPKCs and include the computation of Gröbner bases and XL algorithm. These attacks try to invert the encryption procedure to obtain a plaintext by directly solving the system of polynomial equations obtained from the public key and the ciphertext [2, 3, 10].

(ii) **Rank attacks,** which are applicable to STS-type MPKCs [1, 14, 42, 41].

(iii) **Differential attacks,** which are applicable to MI-type MPKCs [13, 9] (see below).

Tsujii, et al. have developed *the piece in hand method* (PH method, for short) which can be applicable to various types of MPKCs for the purpose of enhancing their security against the above attacks [31, 32, 33, 34, 35, 36, 37, 38, 39]. The latest one is *the 2-layer nonlinear piece in hand method* proposed in 2008 [37, 39]. During the decryption of this PH method, the division is needed to calculate the random perturbation polynomial vector in the first layer using the information obtained from the second layer, called the auxiliary part. Thus, the size q of the ground finite field has to be large in order to keep sufficiently large success probability of the decryption. However, this results in the increase of the computational complexity of the decryption also.

On the other hand, Ding, et al. [4] proposed a different type of security enhancement method, called *Internal Perturbation*. They applied the Internal Perturbation to MI to propose a MPKC, called *Perturbed Matsumoto-Imai* (PMI, for short), and discussed the security enhancement by the Internal Perturbation based on computer experiments. Subsequently, PMI was cryptanalyzed by the differential attack proposed by Fouque, et al. [13]. In order to inoculate PMI against the differential attack, Ding, et al. [6] then modified PMI to PMI+ using the Plus method [25]. Note that the Internal Perturbation is not only applicable to MI but also to any type of MPKC. However, its drawback is that the perturbation dimension has to be increased for enhancing the security, which leads to the exponential increase of computational complexity in decryption.

With the background above, in this paper we propose an MPKC, called *PPS*, as a specific MPKC obtained by the application of the 2-layer nonlinear piece in hand method.[1] We choose STS as the original MPKC to be enhanced. We increase the rank and randomness of the lower rank steps in the secret polynomial vector of STS by adding a perturbation polynomial vector

[1]The name of PPS comes from (i) piece in hand, (ii) PMI, and (iii) STS.

based on the 2-layer nonlinear piece in hand method. In addition, PMI with a slight modification is used in the auxiliary part with comparatively small number of variables. This modification on PMI enhances the security more, compared with the simple use of PMI. Each advantage of the three constituents: (i) the 2-layer nonlinear piece in hand method, (ii) PMI, and (iii) STS, works effectively to overcome the drawbacks of these three constituents (see Figure 1).

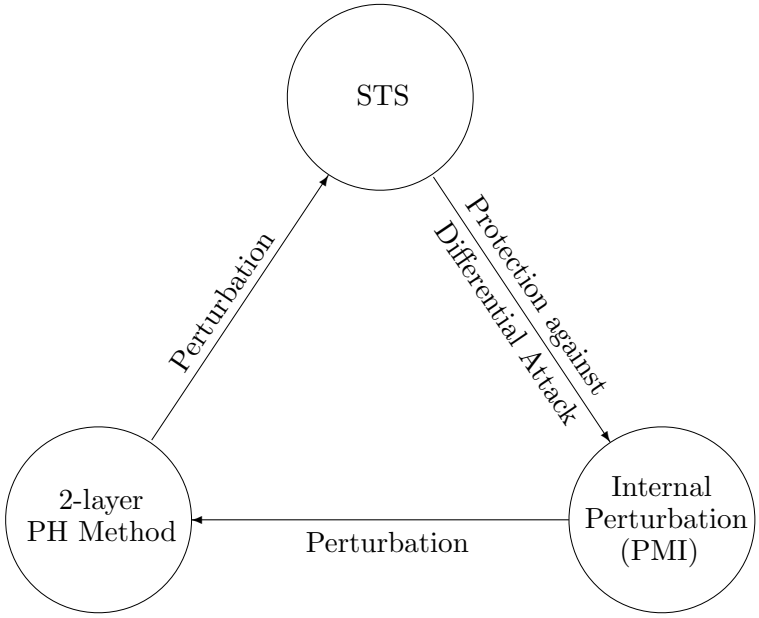

Figure 1: Security Enhanced MPKC Integrated by STS, 2-layer PH Method, and Internal Perturbation

This paper is organized as follows. We begin in Section 2 with a review of the general scheme of MPKCs. We then describe the details and the security features of STS, PMI, and the 2-layer nonlinear piece in hand method, in particular. Based on these three schemes, in Section 3 we propose PPS, which overcomes the drawbacks of the three schemes by the advantage of the three schemes themself. We then consider the security of PPS against various attacks in Section 4, partially based on computer experiments. We conclude this paper with Section 5. Due to the 12-page limit, supplementary materials are attached in the Optional Appendix for clarity.

2 The existing MPKCs and their features

2.1 General scheme of MPKCs

We first review the general scheme of MPKCs. We begin with some basic notation and definitions, which will be used in this paper. $\mathbf{F}_q$ is a finite field which has q elements with $q \geq 2$. $\mathbf{F}_q[x_1, \ldots, x_k]$ is the set of all polynomials in variables $x_1, x_2, \ldots, x_k$ with coefficients in $\mathbf{F}_q$. For every nonempty set S and every positive integers n and ℓ, $S^{n\times\ell}$ denotes the set of all $n \times \ell$ matrices whose entries are in S, and S^n denotes the set of all column vectors consisting n components in S. Therefore

$S^{n\times 1} = S^n$. We represent a column vector in general by bold face symbols such as $\boldsymbol{p}$, $\boldsymbol{v}$, and $\boldsymbol{G}$. For every matrix $A \in S^{n\times\ell}$, $A^T \in S^{\ell\times n}$ denotes the transpose of A.

In general, MPKC can be used both as encryption and as signature scheme. In this section, we explain encryption in particular. Most MPKCs are constituted in the way shown in Figure 2. The meaning of Figure 2 is explained in what follows.

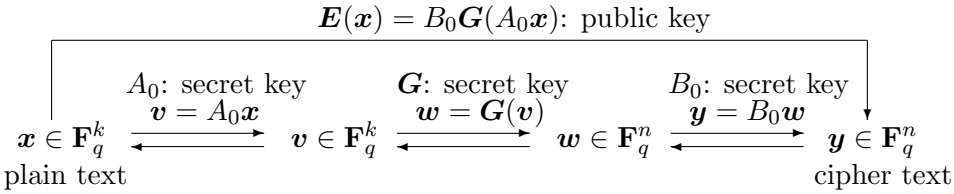

Figure 2: Scheme of Multivariate Public Key Cryptosystem

A plain text is represented by a column vector $\boldsymbol{p} = (p_1, \ldots, p_k)^T \in \mathbf{F}_q^k$, and a cipher text is represented by a column vector $\boldsymbol{c} = (c_1, \ldots, c_n)^T \in \mathbf{F}_q^n$. Then, q, k, and a polynomial column vector $\boldsymbol{E}(\boldsymbol{x}) \in \mathbf{F}_q[x_1, \ldots, x_k]^n$ form the public key of the cryptosystem. The encryption is given by the transformation $\boldsymbol{c} = \boldsymbol{E}(\boldsymbol{p})$ from $\boldsymbol{p}$ to $\boldsymbol{c}$. The secret key of the cryptosystem gives an efficient method to solve the system $\boldsymbol{E}(\boldsymbol{x}) = \boldsymbol{c}$ of polynomial equations on $\boldsymbol{x} = (x_1, \ldots, x_k)^T$ for any given $\boldsymbol{c} \in \mathbf{F}_q^n$. Thus, $\boldsymbol{E}(\boldsymbol{x})$ has to be constructed so that, without the knowledge about this method, it is difficult to find $\boldsymbol{p}$ for any given $\boldsymbol{c}$ in polynomial-time.

Normally, the public key $\boldsymbol{E}(\boldsymbol{x}) \in \mathbf{F}_q[x_1, \ldots, x_k]^n$ has the following form:

$$\boldsymbol{E}(\boldsymbol{x}) = B_0\, \boldsymbol{G}(A_0\boldsymbol{x}). \tag{1}$$

Here A_0 and B_0 are invertible matrices in $\mathbf{F}_q^{k\times k}$ and $\mathbf{F}_q^{n\times n}$, and $\boldsymbol{G}(\boldsymbol{v})$ is a polynomial column vector in $\mathbf{F}_q[v_1, \ldots, v_k]^n$, called the *secret polynomial vector*. In (1), each component of the polynomial vector $A_0\boldsymbol{x} \in \mathbf{F}_q[x_1, \ldots, x_k]^k$ is substituted for the corresponding variables $v_1, \ldots, v_k$ in $\boldsymbol{G}(\boldsymbol{v})$. While keeping A_0, B_0, and $\boldsymbol{G}(\boldsymbol{v})$ secret from anyone else, the legitimate receiver publishes the public key $\boldsymbol{E}(\boldsymbol{x})$ in the form of a system of trimmed multivariate polynomials obtained by simplifying the right-hand side of (1). Normally, $\boldsymbol{G}(\boldsymbol{v})$ consists only of polynomials in $\mathbf{F}_q[v_1, \ldots, v_k]$ of total degree at most two in order to avoid the blowup of the size of the public key $\boldsymbol{E}(\boldsymbol{x})$.

Numerous varieties of MPKCs have been proposed so far by devising the structure of $\boldsymbol{G}(\boldsymbol{v})$, which affects the security directly. MPKCs can be categorized as shown in the Table 1, which is a slight modification of the list in [7] by Ding, et al. Note that, since the unbalanced oil and vinegar signature scheme (UOV, for short) is specialized in signature scheme, MPKCs fall into two major classifications: (i) MI-type cryptosystems and (ii) STS-type cryptosystems. In the subsequent subsections, we review STS and PMI in particular.

2.2 STS

We first note that $k = n$ in STS. The secret polynomial vector $\boldsymbol{w} = \boldsymbol{G}(\boldsymbol{v}) = (g_1, \ldots, g_n)^T \in \mathbf{F}_q[v_1, \ldots, v_n]^n$ of STS is given as shown in Figure 3, where r and L are positive integers with $Lr = n$.

The components of the polynomial column vector $\boldsymbol{G}(\boldsymbol{v})$ form L steps. For each $i = 1, 2, \ldots, L$, the polynomial components $g_{(i-1)r+1}, \ldots, g_{ir}$ of $\boldsymbol{G}(\boldsymbol{v})$ in the step i only contain the variables

Table 1: Taxonomy of the MPKCs based on [7]

System		Authors/Paper
Mixed-Field (or "Big Field")	MIA	MI Scheme A or C^* [20] Matsumoto and Imai
	HFE	Hidden Field Equation [23] Patarin Generalization of MIA
Single-Field (or "True")	UOV	Unbalanced Oil and Vinegar [18] Patarin et al.
	STS	Stepwise Triangular System [14, 41] Tsujii, et al. [29] Shamir [26], Moh [21] Kasahara and Sakai [16, 17]

$$
\text{Step } 1 \left\{ \begin{array}{rcl} w_1 & = & g_1(v_1, \ldots, v_r, \ldots, v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}, \ldots, v_{(L-1)r+1}, \ldots, v_{Lr}) \\ \vdots & & \\ w_r & = & g_r(v_1, \ldots, v_r, \ldots, v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}, \ldots, v_{(L-1)r+1}, \ldots, v_{Lr}) \end{array} \right.
$$

$$
\vdots
$$

$$
\text{Step } i \left\{ \begin{array}{rcl} w_{(i-1)r+1} & = & g_{(i-1)r+1}(v_1, \ldots, v_r, \ldots, v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}) \\ \vdots & & \\ w_{ir} & = & g_{ir}(v_1, \ldots, v_r, \ldots, v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}) \end{array} \right.
$$

$$
\vdots
$$

$$
\text{Step } L \left\{ \begin{array}{rcl} w_{(L-1)r+1} & = & g_{(L-1)r+1}(v_1, \ldots, v_r) \\ \vdots & & \\ w_{Lr} & = & g_{Lr}(v_1, \ldots, v_r) \end{array} \right.
$$

Figure 3: The step-structure of $\boldsymbol{G}(\boldsymbol{v})$ in STS

$v_1, \ldots, v_{(L-i+1)r}$, and chosen randomly during the key-generation. On the decryption, the calculation from $\boldsymbol{w} = (w_1, \ldots, w_n)^T$ to $\boldsymbol{v} = (v_1, \ldots, v_n)^T$ with $\boldsymbol{w} = \boldsymbol{G}(\boldsymbol{v})$ is performed as follows, based on the step-structure of $\boldsymbol{G}$:

First, find $v_1, \ldots, v_r$ with $w_{(L-1)r+1} = g_{(L-1)r+1}(v_1, \ldots, v_r), \ldots, w_{Lr} = g_{Lr}(v_1, \ldots, v_r)$ in Step L by exhaustive search. In general, after substituting $v_1, \ldots, v_{(L-i)r}$ obtained already, find $v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}$ with

$$
\begin{array}{c} w_{(i-1)r+1} = g_{(i-1)r+1}(v_1, \ldots, v_{(L-i)r}, v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}), \\ \vdots \\ w_{ir} = g_{ir}(v_1, \ldots, v_{(L-i)r}, v_{(L-i)r+1}, \ldots, v_{(L-i+1)r}) \end{array}
$$

in Step i by exhaustive search. In this manner, based on the step-structure of $\boldsymbol{G}(\boldsymbol{v})$, the system $\boldsymbol{G}(\boldsymbol{v}) = \boldsymbol{w}$ of the polynomial equations on $\boldsymbol{v}$ can be solved for any given $\boldsymbol{w}$.

The security features of STS against the known attacks are described as follows.

The strength against the Gröbner bases attacks and XL algorithm: As the number L of steps decreases, the strength against the attack increases. To the extent to which computer experiments can be applicable, the strength against the Gröbner bases attack is comparable level between MI and STS.

The strength against the rank attacks: Wolf, et al. [41] proposed the rank attack against STS. The attack first calculates an equivalent matrix to B_0, based on the difference of the ranks of the quadratic forms of constituent polynomials between the steps shown in Figure 3. The equivalent matrix is then used to recover the plain text based on the step-structure of $\boldsymbol{G}(\boldsymbol{v})$ again. The computational complexity of the rank attack is almost the same as the decryption by the legitimate receiver. It would seem necessary to reconsider the validity of the attack from a mathematical point of view. However, STS would seem to involve the weakness against the rank attack inherently.

The strength against the differential attacks: The differential attack was proposed against PMI, and it cannot be applicable to STS, as explained in Section 2.3.

As L decreases, the secret polynomial vector $\boldsymbol{G}$ becomes random. In the extreme case of $L = 1$, it consists of completely random quadratic polynomials in $\boldsymbol{v}$. For example, consider the case of $n = 160$ and $r = 4$. Then $L = 40$ and each polynomial in Step i with $i \leq 24$ contains more than 64 variables. If we can regard each polynomial in Step i with $i \leq 24$ as a random polynomial in all 160 variables, it would be expected that the secret polynomial vector $\boldsymbol{G}(\boldsymbol{v})$ can be made sufficiently random as a whole by adding a highly random perturbation polynomial vector to Steps i with $i \geq 25$. In this paper, we propose a 2-layer nonlinear PH method, called PPS, where STS is used as the original MPKC and is enhanced by the perturbation polynomial vector in this manner.

2.3 PMI and PMI+

In order to enhance the security of MI against the algebraic attacks such as the Gröbner bases attack, PMI is constructed by applying the Internal Perturbation to MI as shown in Figure 4.

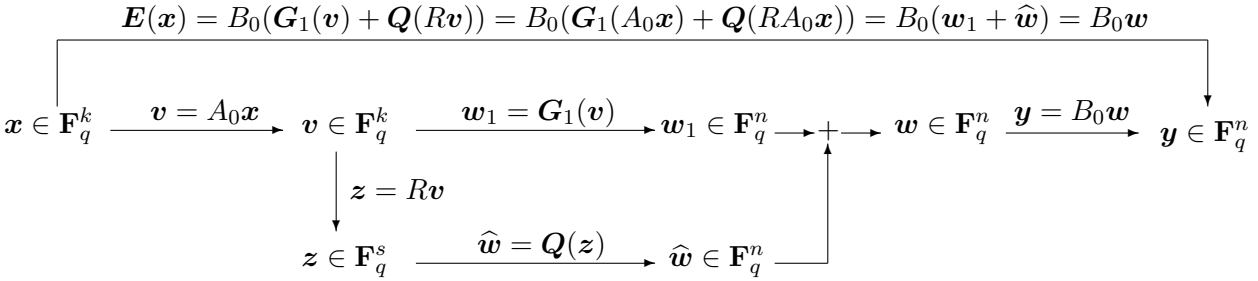

Figure 4: MPKC enhanced by Internal Perturbation

We first note that $k = n$ in PMI. The secret polynomial vector $\boldsymbol{G}(\boldsymbol{v})$ of PMI is constructed by adding a perturbation polynomial vector $\boldsymbol{Q}(R\boldsymbol{v})$ to the secret polynomial vector $\boldsymbol{G}_1(\boldsymbol{v})$ of MI:

$$\boldsymbol{G}(\boldsymbol{v}) \underset{\text{def}}{=} \boldsymbol{G}_1(\boldsymbol{v}) + \boldsymbol{Q}(R\boldsymbol{v}).$$

Here $\boldsymbol{Q}(\boldsymbol{z})$ is a randomly chosen polynomial vector in $\mathbf{F}_q[z_1, \ldots, z_s]^n$, called a *perturbation polynomial vector*. R is a matrix in $\mathbf{F}_q^{s\times n}$ of full row rank. The positive integer s is called *the perturbation dimension*.

On the decryption of PMI, the system $\boldsymbol{G}(\boldsymbol{v}) = \boldsymbol{w}$ of the polynomial equations on $\boldsymbol{v}$ can be solved as follows for any given $\boldsymbol{w}$. For each $\boldsymbol{\lambda} \in \mathbf{F}_q^s$, calculate $\boldsymbol{v}_{\boldsymbol{\lambda}} \in \mathbf{F}_q^n$ which satisfies $\boldsymbol{G}_1(\boldsymbol{v}_{\boldsymbol{\lambda}}) + \boldsymbol{Q}(\boldsymbol{\lambda}) = \boldsymbol{w}$, using the secret key of MI for $\boldsymbol{G}_1(\boldsymbol{v})$. By the exhaustive search for all $\boldsymbol{\lambda} \in \mathbf{F}_q^s$, the legitimate receiver can eventually find $\boldsymbol{v}_{\boldsymbol{\lambda}} \in \mathbf{F}_q^n$ such that $\boldsymbol{\lambda} = R\boldsymbol{v}_{\boldsymbol{\lambda}}$. This $\boldsymbol{v}_{\boldsymbol{\lambda}}$ satisfies $\boldsymbol{G}(\boldsymbol{v}) = \boldsymbol{w}$, obviously. Note that the computational complexity of the decryption is proportional to q^s, which comes from the exhaustive search for all $\boldsymbol{\lambda} \in \mathbf{F}_q^s$.

PMI was cryptanalyzed by Fouque, et al. [13] using the differential attack, and thereby improved into PMI+ by Ding, et al. [6]. The differential attack is based on the fact that the differential of the public key polynomial vector $\boldsymbol{E}(\boldsymbol{x})$ forms an affine transformation. The attack calculates the dimension of the kernel of the linear part of this affine transformation in order to determine the kernel of R, which results in the elimination of $\boldsymbol{Q}(R\boldsymbol{v})$ from $\boldsymbol{G}(\boldsymbol{v})$ in the public key $\boldsymbol{E}(\boldsymbol{x})$.

Subsequently, in order to inoculate PMI against the differential attack, Ding et al. modified PMI to PMI+ by using the Plus method [25], i.e., by adding random polynomials to the secret polynomial vector $\boldsymbol{G}(\boldsymbol{v})$ of PMI in a parallel fashion. However, PMI+ still seems to have the following drawbacks.

In order to enhance the randomness of the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{z})$, it is necessary to increase s. In PMI+ (and PMI), the n polynomial components of $\boldsymbol{Q}(\boldsymbol{z})$ have to be linearly independent at least for keeping the randomness of $\boldsymbol{Q}(\boldsymbol{z})$. Thus the following has to hold for the relation between s and n:

$$\frac{s(s+1)}{2} \geq n.$$

Note that $s(s-1)/2 \geq n$ has to hold instead in the case of $q = 2$. For example, $s \geq 18$ has to hold in the case of $n = 160$. During the decryption of PMI, q^s trials are needed at most. Thus, in the case of $q = 2$ and $s = 18$, 2^{18} trials (i.e., about 2.6×10^5 trials) are needed. Ding, et al. [5] performed the computer experiments for the parameters $q = 2$, $0 \leq s \leq 10$, and $14 \leq n \leq 59$. They then reported that, in the case of $21 \leq n \leq 26$, PMI with $s = 6$ acquires a substantial immunity from the Gröbner bases attack in comparison with PMI with $s = 5$. This phenomenon can be thought to result from the fact that $s(s+1)/2$ equals to 21 and is close to 26 in the case of $s = 6$.

2.4 The 2-layer nonlinear PH method

The 2-layer nonlinear PH method, shown in Figure 5, was proposed for the aim of enhancing the security of various reasonable MPKCs mainly against the algebraic attacks such as the Gröbner bases attacks [37, 39]. In the method, the public key polynomial vector $\boldsymbol{E}$ of the original MPKC is made more random by adding a random perturbation polynomial vector $D_0\boldsymbol{J}$, which can be eliminated during the decryption using the information obtained from the auxiliary part $C\boldsymbol{h}_{\boldsymbol{0}}$. For the detail, see [37, 39].

Unlike Internal Perturbation, the 2-layer nonlinear PH method does not cause the increase of the computational complexity during the decryption in exchange for the security against algebraic attacks. However, it is thought to have the following two drawbacks: (i) During the decryption, the division is needed to calculate the random perturbation polynomial vector $D_0\boldsymbol{J}$ using the information obtained from the auxiliary part. (ii) The randomness of the auxiliary part is not so high.

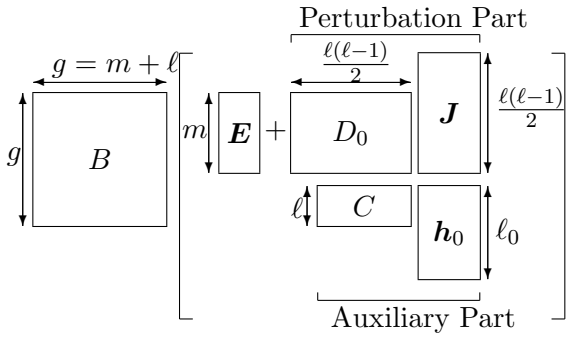

Figure 5: Constitution of the 2-layer nonlinear PH method

3 A new MPKC: PPS

In this section, based on the three schemes described in the above three subsections we propose a new MPKC, called *PPS*, which overcomes the drawbacks of the three schemes by the advantage of the three schemes themself. PPS can be thought to be immune simultaneously from the algebraic attacks, such as the Gröbner bases attack, from the rank attacks, and from the differential attacks. PPS is constructed based on the following design guide, and is depicted in Figure 6.

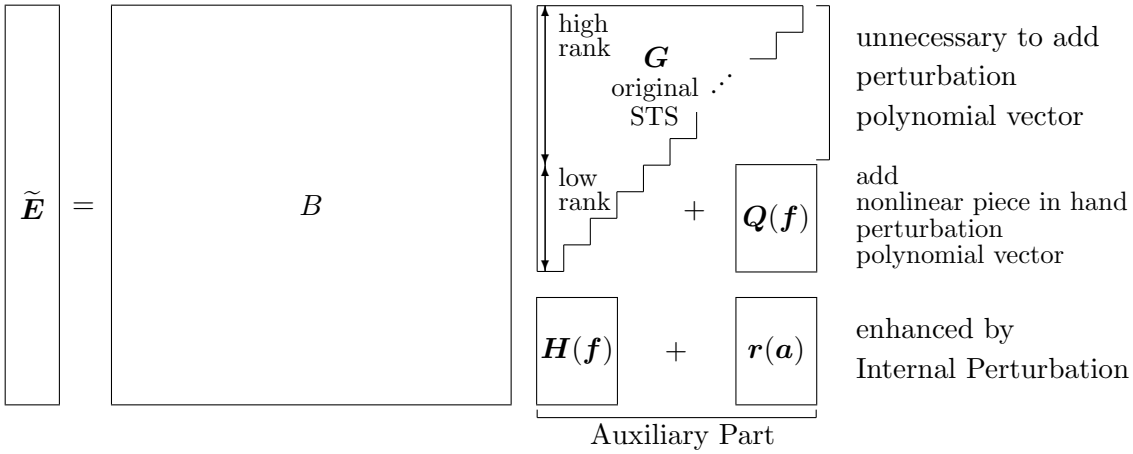

Figure 6: Constitution of PPS

(i) The low rank part of the secret polynomial vector $\boldsymbol{G}$ of the original STS is randomized and strengthened by the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$ constructed based on the auxiliary part.

(ii) The randomness of the auxiliary part is enhanced by adding another perturbation polynomial vector $\boldsymbol{r}(\boldsymbol{a})$ to it.

(iii) The perturbation polynomial vector $\boldsymbol{r}(\boldsymbol{a})$ is added to the auxiliary part based on the Internal Perturbation in a similar manner to PMI. Since the number of polynomial components of the auxiliary part is less than one-tenth of the number of polynomial components of $\boldsymbol{G}$, the perturbation dimension s does not have to be chosen large, unlike PMI.

(iv) Unlike the 2-layer nonlinear PH method, during the decryption the division is not needed to calculate $\boldsymbol{Q}(\boldsymbol{f})$ using the information obtained from the auxiliary part.

Note that additional variables, called *the random variables*, can be introduced in a series of the PH methods [34, 35, 36, 37, 38, 39] in addition to the latest one, the 2-layer nonlinear PH method [37, 39], described in Subsection 2.4 above. This is a crucial property of the PH method in general, and we can introduce the random variables in PPS as well. The aim of introducing the random variables is

(i) to enhance the security against the algebraic attacks, such as the Gröbner bases attacks, and

(ii) to make the PH method available as a signature scheme.

However, a naive introduction of the random variables only in the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$ or the auxiliary part might allow another type of attack based on the distinction between the plaintext variables and the random variables. In order to fend off such an attack we have to insert linear combinations of the random variables among plaintext variables in the public key polynomial vector of the original MPKC. However, the introduction of the random variables results in the decrease of the number of real plaintext variables. Thus, we have to determine appropriately the sizes of the random variables to keep the efficiency of the PH method.

Due to the 12-page limit and also for clarity, we only describe a simplification of PPS, which does not have the random variables, in the main part of this paper as follows. For the complete specification of PPS and its application to signature scheme, see Optional Appendix.

3.1 The specification of primitive PPS

The MPKC: PPS without the random variables, called *primitive PPS*, is specified as follows. We first describe the parameters in PPS.

(i) Parameters on the original STS

- q: the size of the finite field $\mathbf{F}_q$.
- k: the size of the plain text.
- n: the number of components of the public key. Normally, $n = k$.

(ii) Parameters on the auxiliary part

- f: the number of variables in the secret polynomial vector $\boldsymbol{H}$ in the auxiliary part.
- ℓ: the number of polynomial components of the auxiliary part. Normally, $\ell = f$.

(iii) Parameters on the whole MPKC

- $g = n + \ell$: the number of components of the public key.
- n_0: the number of components of the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$ in the layer of the original STS with $n_0 \leq n$.
- s: the perturbation dimension for Internal Perturbation in the auxiliary part.

A plain text vector is represented by $\boldsymbol{p} = (p_1, \ldots, p_k)^T \in \mathbf{F}_q^k$ while a cipher text vector is represented by $\widetilde{\boldsymbol{c}} = (\widetilde{c}_1, \ldots, \widetilde{c}_g)^T \in \mathbf{F}_q^g$.

Secret Key. The following seven form the secret key: (i) A secret polynomial vector $\boldsymbol{G} \in \mathbf{F}_q[x_1, \ldots, x_k]^n$ of the original STS, (ii) A secret polynomial vector $\boldsymbol{H} \in \mathbf{F}_q[x_1, \ldots, x_f]^\ell$ of the MPKC in the auxiliary part, (iii) An invertible matrix $B \in {\mathbf{F}_q}^{g \times g}$, (iv) A matrix $C \in {\mathbf{F}_q}^{f \times k}$, (v) A matrix $D \in {\mathbf{F}_q}^{s \times f}$, (vi) A perturbation polynomial vector $\boldsymbol{Q} \in \mathbf{F}_q[x_1, \ldots, x_f]^{n_0}$ in the layer of the original MPKC, and (vii) A perturbation polynomial vector $\boldsymbol{r} \in \mathbf{F}_q[x_1, \ldots, x_s]^\ell$ in the auxiliary part.

Public Key. The following forms the public key:

$$\widetilde{\boldsymbol{E}} = B \begin{pmatrix} \boldsymbol{G}(\boldsymbol{x}) + \begin{pmatrix} \mathbf{0}_{n-n_0} \\ \boldsymbol{Q}(\boldsymbol{f}) \end{pmatrix} \\ \boldsymbol{H}(\boldsymbol{f}) + \boldsymbol{r}(\boldsymbol{a}) \end{pmatrix},$$

where $\boldsymbol{x} = (x_1, \ldots, x_k)^T \in \mathbf{F}_q[x_1, \ldots, x_k]^k$, $\boldsymbol{f} = (f_1, \ldots, f_f)^T = C\boldsymbol{x} \in \mathbf{F}_q[x_1, \ldots, x_k]^f$, $\boldsymbol{a} = (a_1, \ldots, a_s)^T = D\boldsymbol{f} \in \mathbf{F}_q[x_1, \ldots, x_k]^s$. Finally, $\mathbf{0}_{n-n_0}$ denotes the zero vector in $\mathbf{F}_q^{n-n_0}$.

Encryption. Given a plain text vector $\boldsymbol{p} \in \mathbf{F}_q^k$, calculate the cipher text vector $\widetilde{\boldsymbol{c}} \in \mathbf{F}_q^g$ by $\widetilde{\boldsymbol{c}} = \widetilde{\boldsymbol{E}}(\boldsymbol{p})$ using the public key $\widetilde{\boldsymbol{E}}$.

Decryption. The decryption is performed as follows.

(i) First, calculate $\boldsymbol{w} = (w_1, \ldots, w_g)^T = B^{-1}\widetilde{\boldsymbol{c}}$ and then set $\boldsymbol{w}_1 := (w_1, \ldots, w_n)^T \in \mathbf{F}_q^n$ and $\boldsymbol{w}_2 := (w_{n+1}, \ldots, w_{n+\ell})^T \in \mathbf{F}_q^\ell$.

(ii) For each $\boldsymbol{p}'' \in \mathbf{F}_q^s$, calculate $\widetilde{\boldsymbol{p}}' = \boldsymbol{H}^{-1}(\boldsymbol{w}_2 - \boldsymbol{r}(\boldsymbol{p}''))$, and then check whether $D\widetilde{\boldsymbol{p}}' = \boldsymbol{p}''$ holds or not. If this holds, set $\overline{\boldsymbol{p}}' := \widetilde{\boldsymbol{p}}'$.

(iii) Calculate $\boldsymbol{c} = \boldsymbol{w}_1 - \begin{pmatrix} \mathbf{0}_{n-n_0} \\ \boldsymbol{Q}(\overline{\boldsymbol{p}}') \end{pmatrix}$.

(iv) Since $\boldsymbol{c} = \boldsymbol{G}(\boldsymbol{p})$, the plain text $\boldsymbol{p}$ is recovered from $\boldsymbol{c}$ based on the decryption of the original STS.

Note that the legitimate receiver may fail to recover the plain text uniquely. This is because $\overline{\boldsymbol{p}}'$ obtained in the stage (ii) of the decryption may not be unique. However, it is possible to eliminate this nonuniqueness, using the familiar method based on a hash function H, i.e., we use a part $(p_1, \ldots, p_u)^T$ of the whole plain text vector $\boldsymbol{p} = (p_1, \ldots, p_k)^T$ to represent a real plain text and substitute $H(p_1, \ldots, p_u)$ to the remaining part $(p_{u+1}, \ldots, p_k)^T$ of $\boldsymbol{p}$ on the encryption.

4 Security of PPS

4.1 Security against the Gröbner bases attack

As discussed in the previous works e.g. [39], the security of various MPKCs including STS can be enhanced by the PH methods. For example, MI with $n = 25$ and $q = 256$ is enhanced against the

Gröbner bases attack by the order of 10^3 times in the 2-layer nonlinear PH method with $\ell = 15$. The purpose of the PH methods is to enhance the security of MPKCs, and therefore it is desirable that the computational complexity of the Gröbner bases attack increases exponentially with the parameter n in the enhanced cryptosystem (see also [5]). In PPS proposed in this paper, it is expected that the computational complexity of the Gröbner bases attack is exponential to the n while the size n_0 of the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$ is comparable to the dimension of the linear space spanned by the quadratic polynomials $f_i f_j$, where f_i and f_j are components of $\boldsymbol{f}$ and $i \neq j$, with the parameters $k = 60$ and $\ell = 12$. Although, the perturbation effect would decline when the size n_0 becomes far more than the dimension mentioned above. This is tested by the computer experiments reported in what follows.

4.1.1 Experiment of computing Gröbner bases

We perform the experiments of the Gröbner bases attacks against PPS whose parameters are set as follows.

PMI Auxiliary Part: In the auxiliary part, $f = \ell = 10$ or 12. The perturbation dimension s is 5 for all experiments.

Original STS: The number n of the polynomial components in the secret polynomial vector $\boldsymbol{G}$ of the original STS varies from 24 to 64. Note that $k = n$ in general due to the specification of STS. The number of variables and polynomials are adjusted to equal in the public key $\widetilde{\boldsymbol{E}}$ of PPS by introducing random variables to the auxiliary part and perturbation polynomials. To be specific, ℓ random variables are introduced to make the number of variables and polynomial components equal in the public key $\widetilde{\boldsymbol{E}}$.) For example, when $n = 32$ and $\ell = 12$, 12 random variables are introduced to make $z = 32 + 12 = n + \ell$. See Appendix A for the detailed description of the random variables and their effects. The step height and width r of the original STS is 4. $n_0 = n - 8$, i.e., the two steps from the top of the secret polynomial vector $\boldsymbol{G}$ of the original STS are left unperturbed.

Computing Environment: We perform the experiments using the computational algebra system Magma. Gröbner bases are computed by F_4 algorithm implemented in Magma as the function `GroebnerBasis()`. The attack is repeated 5 times for each condition. All computer experiments are performed with the following environment: (i) Computer: Japan Computing System (JCS) VC98220WSA-4U/T workstation, with CPU AMD Opteron 8220 (2.80 GHz) quadcore and 128 Gbyte Memory (ii) Magma ver. 2.15-15 running on Red Hat Enterprise Linux Advanced Platform Standard. The computation time is counted by the function `Cputime()` in Magma.

In PPS, the effect of the PH method in enhancing the security of the STS against the Gröbner bases attacks is shown in Figures 7 and 8. The relationship between the F_4 computation time and the number n of the polynomial components in the secret polynomial vector $\boldsymbol{G}$ of the original STS in PPS is shown in Figure 7. On the other hand, the binary logarithm of the F_4 computation time vs. n is plotted in Figure 8. The time of the Gröbner basis computation by F_4 algorithm against the corresponding single STS is also shown in Figure 7. In these experiments, it is observed that the time of the Gröbner basis computation by F_4 algorithm is increased by at least 200 times in the PPS, compared with the corresponding single STS. We may observe, in Figure 8, that the linear relationship is maintained until $n = 48$ (i.e., $n_0 = 40$), but this linear relationship begins to decline at around $n = 56$ and deviates rather significantly when $n = 64$ (i.e., $n_0 = 56$).

The above experimental results suggest that the computational complexity of the Gröbner bases attack increases exponentially as n_0 increases, as long as n_0 lies within the dimension $\ell(\ell - 1)/2$ of the linear space spanned by the quadratic polynomials $f_i f_j$ with $i \neq j$. Therefore, in the case of

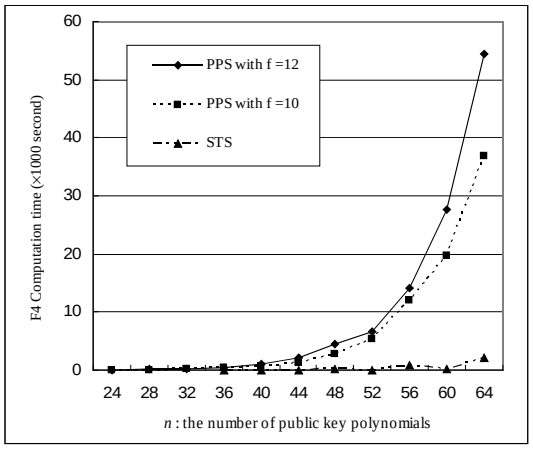

Figure 7: F_4 computation time vs. the number n of polynomials of the original STS part of PPS (and the corresponding single STS)

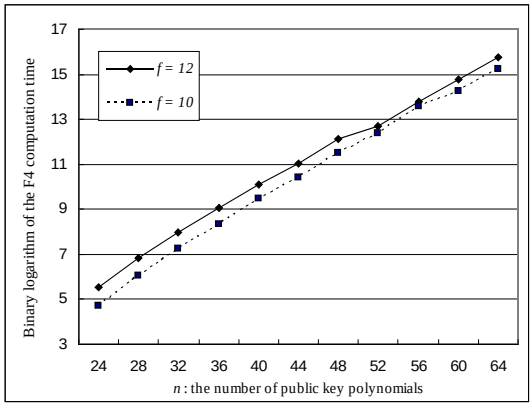

Figure 8: Binary logarithm of the F_4 computation time vs. the number n of polynomials of the original STS part of PPS

$\ell = 25$, the computational complexity would be expected to increase exponentially until n_0 reaches 300 ($= 25 \times (25-1)/2$). Moreover, it is possible to increase the computational complexity further by making the public key $\widetilde{\boldsymbol{E}}$ underdefined based on the introduction of more random variables. See Appendix A for the effects of the random variables.

4.1.2 Recommended parameter setting

We suggest a secure parameter setting of PPS in Table 2, where p is the size of the plain text vector, z is the total number of variables including both the plain text variables and the random variables, and t is the number of the random variables which are included in the secret polynomial vector $\boldsymbol{G}$ of the original STS. See Appendix A for the meaning of these parameters. Since the exhaustive

Table 2: Recommended parameter setting

	Parameters											Public Key
	q	p	k	n	z	g	f	ℓ	n_0	s	t	Size
The original STS	2		260	260								1.10 MB
PPS	2	256	260	260	340	280	20	20	190	6	42	2.03 MB

search among 2^{80} candidates is thought to be impossible at present, the complexity 2^{80} seems to be selected as the standard security level in present cryptographic community. Thus, in suggesting the parameters in Table 2, we assume that the security level is greater than the complexity 2^{80}. In the parameters in Table 2, the information transmission rate (i.e., the size p of plain text divided by the size g of cipher text) of PPS is $256/280 \approx 0.914$. The public key size of PPS is about 1.8 times as large as that of the original STS. In the original STS, the number of both plain text variables and cipher text variables are 260.

While measuring the F_4 computation time at the experiments above, we also measured the maximal degree d of polynomials during the Gröbner basis computation by F_4 algorithm. We observed that $d \leq 3$ in all cases of STS and $d \geq 4$ in our PPS with $f \geq 10$ and $s \geq 5$. Note that, in our PPS, the observed value of d is not constant but takes various values independently of the

number of public key polynomials. Based on the consideration on the complexity of Gröbner basis computation in [11, 12], we estimate that the complexity of Gröbner bases attack against our PPS with the parameters in Table 2 is at least 2^{100}, though that of the corresponding single STS is at most 2^{72}. It is expected with certainty that the complexity is far greater than 2^{100} in practice.

4.2 Security against the rank attack

Wolf, et al. [41] proposed the rank attack against STS. In our PPS, the secret polynomial vector $\boldsymbol{G}$ of STS is used in the 1st layer. Thus, at first glance, the rank attack might seem effective to the 1st layer of PPS. However, the low rank part of the secret polynomial vector $\boldsymbol{G}$ of the original STS in the 1st layer is covered by the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$ and therefore the stepwise triangular structure of the original STS is broken. Thus it is expected that the rank attack does not work properly against PPS.

4.3 Security against the differential attack

The differential attack was proposed by Fouque, et al. [13] against PMI, which is used in the auxiliary part of our PPS. In order to inoculate PMI against the differential attack, Ding, et al. [6] then modified PMI to PMI+ using the Plus method [25]. This method is the countermeasure which adds some random polynomial vector to the public key polynomial vector. In the 1st layer (i.e., the layer of the original STS) of PPS, the polynomials in the higher steps of the secret polynomial vector $\boldsymbol{G}$ of the original STS are virtually random and the polynomials in the lower steps of $\boldsymbol{G}$ is randomized by the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$. Thus, all the polynomials in the 1st layer are considered to be virtually random and are therefore expected to work as the plus part to protect the auxiliary part, PMI, from the differential attack.

5 Conclusion

In this paper we have proposed a new MPKC, called PPS, based on (i) the 2-layer nonlinear piece in hand method, (ii) PMI, and (iii) STS. We have shown that PPS overcomes the drawbacks of these three schemes by the advantage of the three schemes themself. In particular, based on the computer experiments we have shown that PPS can be immune from the Gröbner bases attacks.

Acknowledgments

We appreciate Professor Jintai Ding of University of Cincinnati for the various active discussion with us.
This work was supported by SCOPE (Strategic Information and Communications R&D Promotion Programme) from the Ministry of Internal Affairs and Communications of Japan. The second author was also supported by Japan Science and Technology Agency, CREST.

References

[1] D. Coppersmith, J. Stern, and S. Vaudenay. Attacks on the birational permutation signature schemes. *Proc. CRYPTO '93*, Lecture Notes in Computer Science, Vol.773, pp.435–443, Springer, 1994.

[2] N. Courtois, A. Klimov, J. Patarin, and A. Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. *Proc. EUROCRYPT 2000*, Lecture Notes in Computer Science, Vol.1807, pp.392–407, Springer, 2000.

[3] N. Courtois, M. Daum, and P. Felke. On the security of HFE, HFEv- and Quartz. *Proc. PKC 2003*, Lecture Notes in Computer Science, Vol.2567, pp.337–350, Springer, 2003.

[4] J. Ding. A new variant of the Matsumoto-Imai cryptosystem through perturbation. *Proc. PKC 2004*, Lecture Notes in Computer Science, Vol.2947, pp.305–318, Springer, 2004.

[5] J. Ding, J. E. Gower, D. Schmidt, C. Wolf, and Z. Yin. Complexity estimates for the F4 attack on the perturbed Matsumoto-Imai cryptosystem. *Proc. IMA Int. Conf. 2005*, Lecture Notes in Computer Science, Vol.3796, pp.262–277, Springer, 2005.

[6] J. Ding and J. E. Gower. Inoculating multivariate schemes against differential attacks. *Proc. PKC 2006*, Lecture Notes in Computer Science, Vol.3958, pp.290–301, Springer, 2006.

[7] J. Ding, C. Wolf, and B. Y. Yang. ℓ-Invertible Cycles for $\mathcal{M}$ultivariate $\mathcal{Q}$uadratic ($\mathcal{MQ}$) public key cryptography. *Proc. PKC 2007*, Lecture Notes in Computer Science, Vol.4450, pp.266–281, Springer, 2007.

[8] J. Ding and J. Wagner. Cryptanalysis of rational multivariate public key cryptosystems. *Proc. PQCrypto 2008*, Lecture Notes in Computer Science, Vol.5299, pp.124–136, Springer, 2008.

[9] V. Dubois, P. A. Fouque, A. Shamir, and J. Stern. Practical cryptanalysis of SFLASH. *Proc. CRYPTO 2007*, Lecture Notes in Computer Science, Vol.4622, pp.1–12, Springer, 2007.

[10] J. C. Faugère and A. Joux. Algebraic cryptanalysis of hidden field equation (HFE) cryptosystems using Gröbner bases. *Proc. CRYPTO 2003*, Lecture Notes in Computer Science, Vol.2729, pp.44–60, Springer, 2003.

[11] M. Bardet, J. C. Faugère, and B. Salvy. On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. Proceedings of the International Conference on Polynomial System Solving (ICPSS 2004), pp.71–75, November 2004.

[12] M. Bardet, J. C. Faugère, B. Salvy, and B. Y. Yang. Asymptotic behaviour of the degree of regularity of semi-regular polynomial systems. Proceedings of MEGA 2005, May 2005.

[13] P. A. Fouque, L. Granboulan, and J. Stern. Differential cryptanalysis for multivariate schemes. *Proc. EUROCRYPT 2005*, Lecture Notes in Computer Science, Vol.3494, pp.341–353, Springer, 2005.

[14] L. Goubin and N. Courtois. Cryptanalysis of the TTM cryptosystem. *Proc. ASIACRYPT 2000*, Lecture Notes in Computer Science, Vol.1976, pp.44–57, Springer, 2000.

[15] S. Hasegawa and T. Kaneko. An attacking method for a public-key cryptosystem based on the difficulty of solving a system of non-linear equations. *Proc. 10th SITA*, JA5-3, November 1987. In Japanese.

[16] M. Kasahara and R. Sakai. A construction of public key cryptosystem for realizing ciphertext of size 100 bit and digital signature scheme. *IEICE Transactions on Fundamentals*, E87-A, No.1 (2004), 102–109.

[17] M. Kasahara and R. Sakai. A construction of public-key cryptosystem based on singular simultaneous equations. *IEICE Transactions on Fundamentals*, E88-A, No.1 (2005), 74–80.

[18] A. Kipnis, J. Patarin, and L. Goubin. Unbalanced Oil and Vinegar signature schemes. *Proc. EUROCRYPT '99*, Lecture Notes in Computer Science, Vol.1592, pp.206–222, Springer, 1999.

[19] T. Matsumoto, H. Imai, H. Harashima, and H. Miyakawa. A class of asymmetric cryptosystems using obscure representations of enciphering functions. *1983 National Convention Record on Information Systems, IECE Japan*, S8-5, 1983. In Japanese.

[20] T. Matsumoto and H. Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. *Proc. EUROCRYPT '88*, Lecture Notes in Computer Science, Vol.330, pp.419–453, Springer, 1988.

[21] T. T. Moh. A public key system with signature and master key functions. Communications in Algebra, 27(5), pp.2207–2222, 1999.

[22] J. Patarin. Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88. *Proc. CRYPTO '95*, Lecture Notes in Computer Science, Vol.963, pp.248–261, Springer, 1995.

[23] J. Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): two new families of asymmetric algorithms. *Proc. EUROCRYPT '96*, Lecture Notes in Computer Science, Vol.1070, pp.33–48, Springer, 1996.

[24] J. Patarin. The oil and vinegar signature scheme. Presented at the Dagstuhl Workshop on Cryptography, September 1997. transparencies.

[25] J. Patarin, L. Goubin, and N. Courtois. C^*_{-+} and HM: Variations around two schemes of T. Matsumoto and H. Imai. *Proc. ASIACRYPT '98*, Lecture Notes in Computer Science, Vol.1514, pp.35–49, Springer, 1998.

[26] A. Shamir. Efficient signature schemes based on birational permutations. *Proc. CRYPTO '93*, Lecture Notes in Computer Science, Vol.773, pp.1–12, Springer, 1994.

[27] P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. *Proc. FOCS '94*, pp.124–134, November 1994.

[28] S. Tsujii. Public key cryptosystem using nonlinear equations. *Proc. 8th SITA*, pp.156–157, December 1985. In Japanese.

[29] S. Tsujii, K. Kurosawa, T. Itoh, A. Fujioka, and T. Matsumoto. A public-key cryptosystem based on the difficulty of solving a system of non-linear equations. *IECE Transactions* (D), J69-D, No.12 (1986), 1963–1970. In Japanese.

[30] S. Tsujii, A. Fujioka, and Y. Hirayama. Generalization of the public-key cryptosystem based on the difficulty of solving a system of non-linear equations. *IEICE Transactions* (A), J72-A, No.2 (1989), 390–397. In Japanese.

[31] S. Tsujii. A new structure of primitive public key cryptosystem based on soldiers in hand matrix. Technical Report TRISE 02-03, Chuo University, July 2003.

[32] S. Tsujii, R. Fujita, and K. Tadaki. Proposal of MOCHIGOMA (piece in hand) concept for multivariate type public key cryptosystem. Technical Report of IEICE, ISEC2004-74 (2004-09), September 2004.

[33] S. Tsujii, K. Tadaki, and R. Fujita. Piece in hand concept for enhancing the security of multivariate type public key cryptosystems: public key without containing all the information of secret key. Cryptology ePrint Archive, Report 2004/366, December 2004. Available at URL: `http://eprint.iacr.org/2004/366` .

[34] S. Tsujii, K. Tadaki, and R. Fujita. Proposal for piece in hand matrix ver.2: general concept for enhancing security of multivariate public key cryptosystems. Workshop Record of the International Workshop on Post-Quantum Cryptography (PQCrypto 2006), pp.103–117, 2006. Available at URL: `http://postquantum.cr.yp.to/pqcrypto2006record.pdf` .

[35] S. Tsujii, K. Tadaki, and R. Fujita. Proposal for piece in hand matrix: general concept for enhancing security of multivariate public key cryptosystems. *IEICE Transactions on Fundamentals*, E90-A, No.5 (2007), 992–999. Available at URL: `http://lab.iisec.ac.jp/~tsujii/TTF07.pdf` .

[36] S. Tsujii, K. Tadaki, and R. Fujita. Nonlinear piece in hand matrix method for enhancing security of multivariate public key cryptosystems. Proceedings of the First International Conference on Symbolic Computation and Cryptography (SCC 2008), pp.124–144, 2008.

[37] S. Tsujii, T. Kaneko, K. Tadaki, and M. Gotaishi. Design Policy of MPKC based on Piece in Hand Concept. Technical Report of IEICE, ISEC2008-18, SITE2008-12 (2008-07), July 2008. In Japanese.

[38] R. Fujita, K. Tadaki, and S. Tsujii. Nonlinear piece in hand perturbation vector method for enhancing security of multivariate public key cryptosystems. *Proc. PQCrypto 2008*, Lecture Notes in Computer Science, Vol.5299, pp.148–164, Springer, 2008.

[39] S. Tsujii, K. Tadaki, R. Fujita, M. Gotaishi, and T. Kaneko. Security enhancement of various MPKCs by 2-layer nonlinear piece in hand method. Cryptology ePrint Archive, Report 2009/061, February 2009. Available at URL: `http://eprint.iacr.org/2009/061` .

[40] S. Tsujii, K. Tadaki, R. Fujita, M. Gotaishi, and T. Kaneko. Security enhancement of various MPKCs by 2-layer nonlinear piece in hand method. To appear in *IEICE Transactions on Fundamentals*, E92-A, No.10 (2009).

[41] C. Wolf, A. Braeken, and B. Preneel. Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. *Proc. SCN 2004*, Lecture Notes in Computer Science, Vol.3352, pp.294–309, Springer, 2004.

[42] B. Y. Yang and J. M. Chen. TTS: Rank attacks in tame-like multivariate PKCs. Cryptology ePrint Archive, Report 2004/061, February 2004. Available at URL: `http://eprint.iacr.org/2004/061` .

Optional Appendix

In what follows, we present supplementary discussion about random variables and signature scheme.

A PPS in a complete form

In this section, we give the full specification of PPS by describing the random variables explicitly. This full PPS is depicted in Figure 6 again.

A.1 The specification of PPS

The MPKC: PPS in its complete form is specified as follows. We first describe the parameters in the PPS.

(i) Parameters on the original STS

- q: the size of the finite field $\mathbf{F}_q$.
- k: the size of the plain text.
- n: the number of components of the public key. Normally, $n = k$.

(ii) Parameters on the auxiliary part

- f: the number of variables in the secret polynomial vector $\boldsymbol{H}$ of the MPKC in the auxiliary part.
- ℓ: the number of polynomial components of the auxiliary part. Normally, $\ell = f$.

(iii) Parameters on the whole MPKC

- p: the size of the plain text with $p \leq k$.
- z: the total number of variables including both the plain text variables and the random variables with $z \geq k$, where $z - p$ is the number of the random variables.
- t: the number of the random variables which, in particular, are included in the secret polynomial vector $\boldsymbol{G}$ of the original STS with $0 \leq t \leq z - p$. This t is introduced for generality, and it is possible to set $t = z - p$.
- $g = n + \ell$: the number of components of the public key.
- n_0: the number of components of the perturbation polynomial vector $\boldsymbol{Q}(\boldsymbol{f})$ in the layer of the original STS with $n_0 \leq n$.
- s: the perturbation dimension for Internal Perturbation in the auxiliary part.

A plain text vector is represented by $\boldsymbol{p} = (p_1, \ldots, p_p)^T \in \mathbf{F}_q^p$ while a cipher text vector is represented by $\widetilde{\boldsymbol{c}} = (\widetilde{c}_1, \ldots, \widetilde{c}_g)^T \in \mathbf{F}_q^g$.

Secret Key. The following eight form the secret key: (i) A secret polynomial vector $\boldsymbol{G} \in \mathbf{F}_q[x_1, \ldots, x_k]^n$ of the original STS, (ii) A secret polynomial vector $\boldsymbol{H} \in \mathbf{F}_q[x_1, \ldots, x_f]^\ell$ of the

MPKC in the auxiliary part, (iii) A matrix $A \in \mathbf{F}_q^{(k-p)\times t}$, (iv) An invertible matrix $B \in \mathbf{F}_q^{g\times g}$, (v) A matrix $C \in {\mathbf{F}_q}^{f\times z}$, (vi) A matrix $D \in {\mathbf{F}_q}^{s\times f}$, (vii) A perturbation polynomial vector $\boldsymbol{Q} \in \mathbf{F}_q[x_1,\ldots,x_f]^{n_0}$ in the layer of the original MPKC, and (viii) A perturbation polynomial vector $\boldsymbol{r} \in \mathbf{F}_q[x_1,\ldots,x_s]^{\ell}$ in the auxiliary part.

Public Key. The following forms the public key:

$$\widetilde{\boldsymbol{E}} = B \begin{pmatrix} \boldsymbol{G}\begin{pmatrix} \boldsymbol{x} \\ A\boldsymbol{\mu} \end{pmatrix} + \begin{pmatrix} \mathbf{0}_{n-n_0} \\ \boldsymbol{Q}(\boldsymbol{f}) \end{pmatrix} \\ \boldsymbol{H}(\boldsymbol{f}) + \boldsymbol{r}(\boldsymbol{a}) \end{pmatrix},$$

where $\boldsymbol{x} = (x_1,\ldots,x_p)^T \in \mathbf{F}_q[x_1,\ldots,x_p]^p$, $\boldsymbol{\mu} = (x_{p+1},\ldots,x_{p+t})^T \in \mathbf{F}_q[x_{p+1},\ldots,x_{p+t}]^t$, $\boldsymbol{f} = (f_1,\ldots,f_f)^T = C\begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{\lambda} \end{pmatrix} \in \mathbf{F}_q[x_1,\ldots,x_z]^f$, $\boldsymbol{\lambda} = (x_{p+1},\ldots,x_z)^T \in \mathbf{F}_q[x_{p+1},\ldots,x_z]^{z-p}$, $\boldsymbol{a} = (a_1,\ldots,a_s)^T = D\boldsymbol{f} \in \mathbf{F}_q[x_1,\ldots,x_z]^s$. Finally, $\mathbf{0}_{n-n_0}$ denotes the zero vector in $\mathbf{F}_q^{n-n_0}$.

Encryption. Given a plain text vector $\boldsymbol{p} \in \mathbf{F}_q^p$, first choose a random vector $\boldsymbol{u} = (u_1,\ldots,u_{z-p})^T \in {\mathbf{F}_q}^{z-p}$ which is substituted for the random variables, and then calculate the cipher text vector $\widetilde{\boldsymbol{c}} \in \mathbf{F}_q^g$ by $\widetilde{\boldsymbol{c}} = \widetilde{\boldsymbol{E}}(\boldsymbol{z})$ using the public key $\widetilde{\boldsymbol{E}}$, where

$$\boldsymbol{z} = \begin{pmatrix} \boldsymbol{p} \\ \boldsymbol{u} \end{pmatrix} \in \mathbf{F}_q^z.$$

Decryption. Note that the vector $\boldsymbol{u}$ is decomposed into $\boldsymbol{u}_1 \in \mathbf{F}_q^t$ and $\boldsymbol{u}_2 \in \mathbf{F}_q^{z-p-t}$ by

$$\boldsymbol{u} = \begin{pmatrix} \boldsymbol{u}_1 \\ \boldsymbol{u}_2 \end{pmatrix} \in \mathbf{F}_q^{z-p}.$$

The decryption is performed as follows.

(i) First, calculate $\boldsymbol{w} = (w_1,\ldots,w_g)^T = B^{-1}\widetilde{\boldsymbol{c}}$ and then set $\boldsymbol{w}_1 := (w_1,\ldots,w_n)^T \in \mathbf{F}_q^n$ and $\boldsymbol{w}_2 := (w_{n+1},\ldots,w_{n+\ell})^T \in \mathbf{F}_q^{\ell}$.

(ii) For each $\boldsymbol{z}'' \in \mathbf{F}_q^s$, calculate $\widetilde{\boldsymbol{z}}' = \boldsymbol{H}^{-1}(\boldsymbol{w}_2 - \boldsymbol{r}(\boldsymbol{z}''))$, and then check whether $D\widetilde{\boldsymbol{z}}' = \boldsymbol{z}''$ holds or not. If this holds, set $\overline{\boldsymbol{z}}' := \widetilde{\boldsymbol{z}}'$.

(iii) Calculate $\boldsymbol{c} = \boldsymbol{w}_1 - \begin{pmatrix} \mathbf{0}_{n-n_0} \\ \boldsymbol{Q}(\overline{\boldsymbol{z}}') \end{pmatrix}$.

(iv) Since $\boldsymbol{c} = \boldsymbol{G}\begin{pmatrix} \boldsymbol{p} \\ A\boldsymbol{u}_1 \end{pmatrix}$, the plain text $\boldsymbol{p}$ is recovered from $\boldsymbol{c}$ based on the decryption of the original STS.

Note that the legitimate receiver may fail to recover the plain text uniquely. This is because $\overline{\boldsymbol{z}}'$ obtained in the stage (ii) of the decryption may not be unique. However, it is possible to eliminate this nonuniqueness, using the familiar method based on a hash function H, i.e., we use a part $(p_1,\ldots,p_u)^T$ of the whole plain text vector $\boldsymbol{p} = (p_1,\ldots,p_k)^T$ to represent a real plain text and substitute $H(p_1,\ldots,p_u)$ to the remaining part $(p_{u+1},\ldots,p_k)^T$ of $\boldsymbol{p}$ on the encryption.

A.2 Effects of random variables to the security against the Gröbner bases attack

In PPS described in Subsection A.1, it is possible to make the public key $\widetilde{\boldsymbol{E}}$ underdetermined, i.e., to make the parameter z greater than the parameter g. The computational complexity of the Gröbner bases attack against PPS can be increased by appropriately adjusting the number of the random variables in the public key. This is because the strength against algebraic attacks increases as the total number of variables increases, in general. Note that the inclusion of the random variables to the public key does not increase the size of the cipher text although the one of the public key increases somewhat. Therefore the inclusion of the random variables in the public key does not detract the efficiency of information transmission. We made experiments to observe the relationship between the number of the random variables and the computational complexity of the Gröbner bases attack against PPS with other parameters fixed. Figures 9 and 10 show the relationship between the F_4 computation time and the number of the random variables for the PPS with $n = 32$ and $f = \ell = 12$ fixed. The number of the random variables are increased from $12(z = n + \ell)$ to 30. We may see that the complexity increases exponentially, in particular, from Figure 10 where the binary logarithm of the F_4 computation time vs. the number of the random variables is plotted.

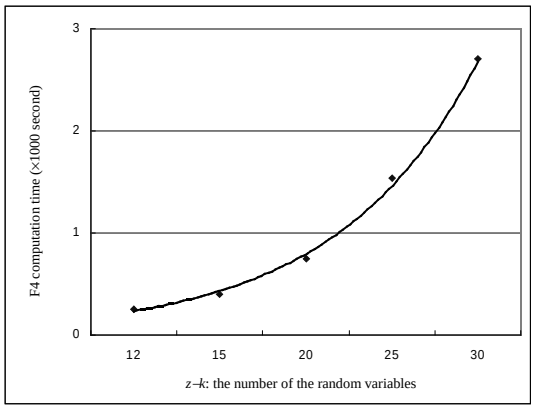

Figure 9: F_4 computation time vs. the length of the random variables

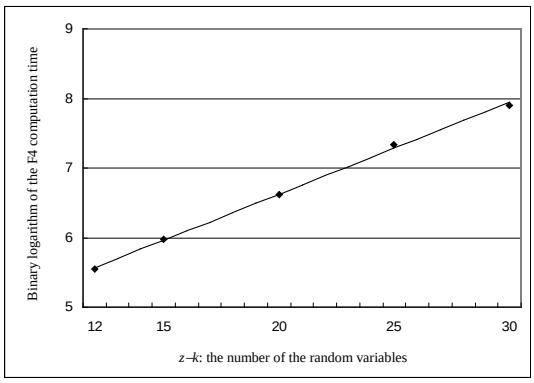

Figure 10: Binary logarithm of the F_4 computation time vs. the length of the random variables

B PPS as signature scheme

In PPS described in Appendix A, the decryption can be performed without calculating the random variables. The PPS can be converted into a signature scheme by calculating the random variables in addition to the plain text variables as follows. The difficulty to overcome for converting PPS into a signature scheme is that the public key $\widetilde{\boldsymbol{E}}$ might not be a surjection onto $\mathbf{F}_q^g$ in general. Since the secret polynomial vector $\boldsymbol{G}$ in the main part (i.e., the 1st layer of $\widetilde{\boldsymbol{E}}$) is chosen to be STS in PPS, this part can be a surjection onto $\mathbf{F}_q^n$. On the other hand, the auxiliary part might not be a surjection onto $\mathbf{F}_q^\ell$ due to the existence of the perturbation polynomial vector $\boldsymbol{r}(\boldsymbol{a})$. To overcome this difficulty, it is effective to append to the auxiliary part a polynomial vector $\boldsymbol{OV} \in \mathbf{F}_q[x_1, \ldots, x_z]^\ell$, where a certain part of the random variables appears only as a variable of degree one in each monomial in $\boldsymbol{OV}$. When we generate the signature $\boldsymbol{z}$ given a message $\widetilde{\boldsymbol{c}}$, by decomposing $\boldsymbol{w_2}$ into $\boldsymbol{w_2} = \boldsymbol{d_1} + \boldsymbol{d_2}$

appropriately and then solving $\boldsymbol{H}(\boldsymbol{f}) + \boldsymbol{r}(\boldsymbol{a}) = \boldsymbol{d_1}$ and $\boldsymbol{OV} = \boldsymbol{d_2}$ with respect to $\boldsymbol{f}$ and the part of the random variables, the decryption in the auxiliary part can be performed. Note here that we can use the technique of the oil and vinegar signature scheme [24] in order to solve $\boldsymbol{OV} = \boldsymbol{d_2}$. The detail will be described in a sequel to this paper.