Public-Key Identification Schemes Based on Multivariate Cubic Polynomials

Koichi Sakumoto

Sony Corporation 5-1-12 Kitashinagawa Shinagawa-ku,
Tokyo 141-0001, Japan
Koichi.Sakumoto@jp.sony.com

Abstract. Solving a system of multivariate polynomials over a finite field is a promising problem in cryptography. Recently, Sakumoto et al. proposed public-key identification schemes based on the quadratic version of the problem, which is called the MQ problem. However, it is still an open question whether or not it is able to build efficient constructions of public-key identification based on multivariate polynomials of degree greater than two. In this paper, we tackle the *cubic* case of this question and construct public-key identification schemes based on the *cubic* version of the problem, which is called the MC problem. The MQ problem is a special case of the MC problem. Our schemes consist of a protocol which is zero-knowledge argument of knowledge for the MC problem under the assumption of the existence of a non-interactive commitment scheme. For a practical parameter choice, the efficiency of our scheme is highly comparable to that of the schemes based on the MQ problem. Furthermore, the parallel version of our scheme also achieves the security under active attack with some additional cost.

Keywords: public-key identification scheme, zero knowledge, MQ problem, MC problem.

1 Introduction

Diversity of underlying mathematical problems is important for cryptography. Although the ones widely used today are factorization and a discrete logarithm problem, there are other various problems which are used for cryptography. Among them, a problem of solving a system of multivariate polynomials over a finite field is a promising problem. In particular the quadratic case of the problem is called the MQ problem. Even in the quadratic case, the associated decision problem is known to be NP-complete [14, 23], and a random instance of the problem is widely believed to be intractable. Naturally, the problem of degree greater than two is expected to be equally or more intractable than the quadratic one. The generic attacks on the MQ problem using Gröbner basis are known to have exponential complexity in time and space [3, 11], and there is no known polynomial-time quantum algorithm to solve the MQ problem in contrast to factorization or a discrete logarithm problem.

M. Fischlin, J. Buchmann, and M. Manulis (Eds.): PKC 2012, LNCS 7293, pp. 172–189, 2012.

Over the past few decades, many studies have been made on cryptographic primitives based on multivariate polynomials. Most of them deal with quadratic polynomials [5, 18, 19, 21, 26], and some of them deal with polynomials of degree greater than two [6, 10, 12, 22, 32]. In symmetric cryptography, Berbain et al. proposed QUAD, which is a stream cipher with provable security based on the MQ problem [5]. In asymmetric cryptography, several public-key schemes have been proposed, which are known as multivariate public-key cryptography (MPKC) [18, 19, 21].

Recently, Sakumoto et al. proposed public-key identification schemes based on the MQ problem [26]. A remarkable advantage of their schemes is that they have provable security based on the conjectured intractability of the MQ problem under the assumption of the existence of a non-interactive commitment scheme. In fact, their schemes do *not* depend either on the Isomorphism of Polynomials (IP) problem or on the Functional Decomposition (FD) problem, while the other schemes in MPKC depend on the IP problem [18, 19, 21] or the FD problem [22]. Their new cut-and-choose techniques are specialized for the quadratic case and are based on the bilinearity of the map $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{F}_2(\mathbf{x} + \mathbf{y}) - \mathbf{F}_2(\mathbf{x}) - \mathbf{F}_2(\mathbf{y})$, where $\mathbf{F}_d$ is a function consisting of multivariate polynomials of degree d. In fact, their techniques do not work in the case of degree $d > 2$, because the map $(\mathbf{x}, \mathbf{y}) \mapsto \mathbf{F}_d(\mathbf{x} + \mathbf{y}) - \mathbf{F}_d(\mathbf{x}) - \mathbf{F}_d(\mathbf{y})$ where $d > 2$ is not linear either in $\mathbf{x}$ or in $\mathbf{y}$. Thus it is an interesting question whether or not it is able to build efficient constructions of public-key identification based on multivariate polynomials of degree greater than two.

In this paper, we tackle the *cubic* case of this question and construct public-key identification schemes based on the MC problem, which is a problem of solving a system of multivariate *cubic* polynomials over a finite field. The MQ problem is a special case of the MC problem, and we have less perspective on solving the MC problem compared to the MQ problem even considering the state-of-the-art algorithms [7, 8, 11]. It is important for higher security to be based on such a more intractable problem even though the MQ problem is very hard. A function consisting of multivariate cubic polynomials is also called an MC function.

We present two concrete protocols, a three-pass protocol and a five-pass one, which are statistical zero-knowledge argument of knowledge for the MC problem. Our schemes consisting of the protocol have provable security based on the conjectured intractability of the MC problem under the assumption of the existence of a non-interactive commitment scheme. Concretely, the identification schemes consisting of the *sequential* composition and the *parallel* composition of our protocol are secure against impersonation under *active* attack and under *passive* attack, respectively. Moreover, the parallel version of our scheme is also secure under active attack if its underlying MC function is substantially compressing (e.g., mapping 160 bits to 80 bits). These levels of provable security are the same as those of the identification schemes based on the MQ problem. Of course, our schemes do *not* depend either on the IP problem or on the FD problem.

Efficiency of our five-pass protocol is highly comparable to that of the MQ-based schemes for a practical parameter choice. The size of communication data

in our five-pass protocol is 26,697 bits when the impersonation probability is less than 2^{-30}, while those in the three-pass protocol and in the five-pass protocol of [26] are 29,640 bits and 26,565 bits, respectively. Our five-pass protocol also has the small sizes of a public key and a secret key, 88 bits and 132 bits for 80-bit security, respectively. Although our schemes have the relatively large size of the system parameter, it can be reduced to a small seed, e.g., 128 bits, by employing a pseudo-random number generator. The technique is also used in the implementation of QUAD [2]. We note that cubic systems with only 33 variables and 22 equations over $\mathbb{F}_{2^4}$ achieve 80-bit security, while quadratic systems over $\mathbb{F}_{2^4}$ require 45 variables and 30 equations. This evaluation is derived from the way of [7] of selecting the minimum parameters for 80-bit security and contributes to the efficiency of our five-pass scheme.

Techniques for our constructions. First, we briefly review the techniques for the MQ-based construction. They employ the cut-and-choose approach, where a prover first divides her secret into shares and then proves the correctness of some shares depending on the choice of a verifier without revealing the secret itself.

Let $\mathbf{F}_{\mathtt{MQ}}$ be a function $(x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$ where $y_l = \sum_{i,j} a_{l,i,j} x_i x_j + \sum_i b_{l,i} x_i$. The function $\mathbf{F}_{\mathtt{MQ}}$ is called an MQ function. The associated bilinear form of $\mathbf{F}_{\mathtt{MQ}}$ is defined as $\mathbf{G}_{\mathtt{MQ}}(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbf{F}_{\mathtt{MQ}}(\mathbf{x}+\tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MQ}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MQ}}(\tilde{\mathbf{x}})$. It is easy to see the bilinearity of the function $\mathbf{G}_{\mathtt{MQ}}(\mathbf{x}, \tilde{\mathbf{x}})$, since it maps $(x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_n) \mapsto (z_1, \ldots, z_m)$ where $z_l = \sum_{i,j} a_{l,i,j}(x_i \tilde{x}_j + \tilde{x}_i x_j)$. Let $\mathbf{s}$ be a secret key and $\mathbf{v} = \mathbf{F}_{\mathtt{MQ}}(\mathbf{s})$ the corresponding public key. When the secret key is divided as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$, the public key $\mathbf{v} = \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0 + \mathbf{r}_1)$ can be represented as $\mathbf{v} = \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) + \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_1) + \mathbf{G}_{\mathtt{MQ}}(\mathbf{r}_0, \mathbf{r}_1)$. This representation still contains the term $\mathbf{G}_{\mathtt{MQ}}(\mathbf{r}_0, \mathbf{r}_1)$ which depends on both $\mathbf{r}_0$ and $\mathbf{r}_1$. Then, the two vectors $\mathbf{r}_0$ and $\mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0)$ are also divided as $\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{t}_1$ and $\mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$. Accordingly, the public key can also be represented as $\mathbf{v} = \mathbf{e}_0 + \mathbf{e}_1 + \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_1) + \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_1, \mathbf{r}_1)$, due to the bilinearity of $\mathbf{G}_{\mathtt{MQ}}$. As a result, it yields the following equations:

$$\mathbf{r}_0 - \mathbf{t}_0 = \mathbf{t}_1, \quad \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) - \mathbf{e}_0 = \mathbf{e}_1, \quad \text{and}$$
$$\mathbf{v} - \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_1, \mathbf{r}_1) - \mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_1) - \mathbf{e}_1 = \mathbf{e}_0 + \mathbf{G}_{\mathtt{MQ}}(\mathbf{t}_0, \mathbf{r}_1).$$

Each side of each of the three equations can be checked by using some one of three tuples $(\mathbf{r}_0, \mathbf{t}_0, \mathbf{e}_0)$, $(\mathbf{r}_1, \mathbf{t}_1, \mathbf{e}_1)$, and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$, while no information on the secret key $\mathbf{s}$ can be obtained from one out of the three tuples. As described above the bilinearity of $\mathbf{G}_{\mathtt{MQ}}$ plays an important role in their dividing technique.

Then we consider the case of the MC function $\mathbf{F}_{\mathtt{MC}} : (x_1, \ldots, x_n) \mapsto (y_1, \ldots, y_m)$ where $y_l = \sum_{i,j,k} a_{l,i,j,k} x_i x_j x_k + \sum_{i,j} b_{l,i,j} x_i x_j + \sum_i c_{l,i} x_i$. Unfortunately, the mapping $(\mathbf{x}, \tilde{\mathbf{x}}) \mapsto \mathbf{F}_{\mathtt{MC}}(\mathbf{x} + \tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MC}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MC}}(\tilde{\mathbf{x}})$ is *not* bilinear, since it maps $(x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_n) \mapsto (z_1, \ldots, z_m)$ where $z_l = \sum_{i,j,k} a_{l,i,j,k}(x_i x_j \tilde{x}_k + x_i \tilde{x}_j x_k + x_i \tilde{x}_j \tilde{x}_k + \tilde{x}_i x_j x_k + \tilde{x}_i x_j \tilde{x}_k + \tilde{x}_i \tilde{x}_j x_k) + \sum_{i,j} b_{l,i,j}(x_i \tilde{x}_j + \tilde{x}_i x_j)$. Thus the dividing technique using the mapping does not work in the cubic case. We also note that there is a trivial construction derived from the MQ-based scheme, because it is always possible to express degree three terms $x_i x_j x_k$ as degree two terms $w_{i,j} x_k$

by introducing auxiliary variables $w_{i,j}$ and equations $w_{i,j} - x_i x_j = 0$. However, this reduction makes the numbers of variables and equations much larger, and the construction becomes inefficient.

Therefore, in the cubic case, we introduce another function which is associated with $\mathbf{F}_{\mathtt{MC}}$. Concretely, we define a function $\mathbf{G}_{\mathtt{MC}} : (x_1, \ldots, x_n, \tilde{x}_1, \ldots, \tilde{x}_n) \mapsto (z_1, \ldots, z_m)$ where $z_l = \sum_{i,j,k} a_{l,i,j,k}(x_i\tilde{x}_j\tilde{x}_k + \tilde{x}_i x_j \tilde{x}_k + \tilde{x}_i\tilde{x}_j x_k) + \sum_{i,j} b_{l,i,j} x_i \tilde{x}_j$. The function $\mathbf{G}_{\mathtt{MC}}(\mathbf{x}, \tilde{\mathbf{x}})$ is linear in one argument $\mathbf{x}$. In this paper we call $\mathbf{G}_{\mathtt{MC}}$ the associated linear-in-one-argument (LOA) form of $\mathbf{F}_{\mathtt{MC}}$. By using the function $\mathbf{G}_{\mathtt{MC}}$, it is able to divide $\mathbf{F}_{\mathtt{MC}}(\mathbf{x} + \tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MC}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MC}}(\tilde{\mathbf{x}})$ into two parts $\mathbf{G}_{\mathtt{MC}}(\mathbf{x}, \tilde{\mathbf{x}})$ and $\mathbf{G}_{\mathtt{MC}}(\tilde{\mathbf{x}}, \mathbf{x})$ which are linear in $\mathbf{x}$ and in $\tilde{\mathbf{x}}$, respectively. In fact, it is seen that $\mathbf{G}_{\mathtt{MC}}(\mathbf{x}, \tilde{\mathbf{x}}) + \mathbf{G}_{\mathtt{MC}}(\tilde{\mathbf{x}}, \mathbf{x}) = \mathbf{F}_{\mathtt{MC}}(\mathbf{x} + \tilde{\mathbf{x}}) - \mathbf{F}_{\mathtt{MC}}(\mathbf{x}) - \mathbf{F}_{\mathtt{MC}}(\tilde{\mathbf{x}})$.

With this associated LOA form $\mathbf{G}_{\mathtt{MC}}$, our new dividing techniques for $\mathbf{F}_{\mathtt{MC}}$ are briefly described as follows. Let $\mathbf{s}$ be a secret key and $\mathbf{v} = \mathbf{F}_{\mathtt{MC}}(\mathbf{s})$ the corresponding public key. When the secret key is divided as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$, the public key $\mathbf{v} = \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0 + \mathbf{r}_1)$ can be represented as $\mathbf{v} = \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) + \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_1) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$. This representation still contains the terms $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_0, \mathbf{r}_1)$ and $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$ which depend on both $\mathbf{r}_0$ and $\mathbf{r}_1$. Then, the two vectors $\mathbf{r}_0$ and $\mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$ are also divided as $\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{u}$ and $\mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) + \mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$ similarly to the quadratic case. However, the latter equation also contains the term $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0)$ depending on both $\mathbf{r}_0$ and $\mathbf{r}_1$ in contrast to the case of $\mathbf{F}_{\mathtt{MQ}}(\mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$. Thus $\mathbf{r}_1$ is further divided as $\mathbf{r}_1 = \mathbf{t}_1 + \mathbf{u}$. Accordingly, the terms depending on both $\mathbf{r}_0$ and $\mathbf{r}_1$ are divided as $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_0, \mathbf{r}_1) = \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_0, \mathbf{r}_1) + \mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_1)$ and $\mathbf{G}_{\mathtt{MC}}(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_1, \mathbf{r}_0) + \mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_0)$, due to the linearity in one argument. As a result, it yields the following equations:

$$\begin{gathered}
\mathbf{r}_0 - \mathbf{u} = \mathbf{t}_0, \quad \mathbf{r}_1 - \mathbf{u} = \mathbf{t}_1, \\
\mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_1) + \mathbf{e}_1 = \mathbf{v} - \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_1) - \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_0, \mathbf{r}_1) - \mathbf{e}_0, \quad \text{and} \\
\mathbf{G}_{\mathtt{MC}}(\mathbf{u}, \mathbf{r}_0) - \mathbf{e}_0 = \mathbf{e}_1 - \mathbf{F}_{\mathtt{MC}}(\mathbf{r}_0) - \mathbf{G}_{\mathtt{MC}}(\mathbf{t}_1, \mathbf{r}_0).
\end{gathered}$$

Each side of each of the four equations can be checked by using some one of four tuples $(\mathbf{r}_0, \mathbf{u}, \mathbf{e}_0)$, $(\mathbf{r}_0, \mathbf{t}_1, \mathbf{e}_1)$, $(\mathbf{r}_1, \mathbf{u}, \mathbf{e}_1)$, and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$, while no information on the secret key $\mathbf{s}$ can be obtained from one out of the four tuples. We note that using the common $\mathbf{u}$ in dividing $\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{u}$ and $\mathbf{r}_1 = \mathbf{t}_1 + \mathbf{u}$ does not damage the zero-knowledge property, since each of the four tuples contains only one out of $\mathbf{t}_0$, $\mathbf{t}_1$, and $\mathbf{u}$.

Related work. Identification schemes based on Permuted Kernels (PK) [27], binary Syndrome Decoding (SD) [28, 30], Constrained Linear Equations (CLE) [29], Permuted Perceptrons (PP) [24, 25], and q-ary SD [9] have some features similar to the MQ-based schemes [26] and ours as follows. First, these schemes depend on the hardness of a random instance of each of the problems whose associated decision version is known to be NP-complete. Second, their protocols have perfect correctness. Finally, assuming the existence of a non-interactive commitment scheme, the sequential version and the parallel version of the schemes are secure against impersonation under active attack and passive attack, respectively. However, it is not explicitly known that the parallel versions of these

schemes achieve the security under active attack. The efficiency of our scheme is highly comparable to that of these schemes. Indeed, the data sizes of a public key of the schemes of [9, 24, 25, 27–30] are between 245 bits and 2,450 bits, and those of communication are between 27,234 bits and 120,652 bits.

Paper Organization. The remainder of this paper is organized as follows. In Section 2 we define some notions related to the MC function and evaluate the intractability of the function. In Section 3 and Section 4, our 3-pass construction and 5-pass one are presented, respectively. In Section 5 we discuss their security and efficiency for a practical parameter choice. In Section 6 we study the security of the parallel composition of our scheme at the expense of the efficiency. Finally, we close with some extensions, open problems, and conclusion.

2 Multivariate Cubic Functions

In this section we define a family of MC functions $\mathcal{MC}(n, m, \mathbb{F}_q)$ and study its parameters achieving 80-bit security.

Definition 1. *We denote by $\mathcal{MC}(n, m, \mathbb{F}_q)$ a family of functions $\{\mathbf{F} = (f_1, \ldots, f_m)\}$ such that, for $l = 1, \ldots, m$, $f_l(x_1, \ldots, x_n) = \sum_{i,j,k} a_{l,i,j,k} x_i x_j x_k + \sum_{i,j} b_{l,i,j} x_i x_j + \sum_i c_{l,i} x_i$ where $a_{l,i,j,k}, b_{l,i,j}, c_{l,i} \in \mathbb{F}_q$. We call $\mathbf{F} \in \mathcal{MC}(n, m, \mathbb{F}_q)$ an MC function.*

For the simplicity, constant terms are omitted without any security loss. The MQ function is a special case of the MC function, where the coefficients $a_{l,i,j,k}$ are all zero. For the MC function $\mathbf{F}$, we define a binary relation $R_{\mathbf{F}} = \{(\mathbf{v}, \mathbf{x}) \in \mathbb{F}_q^m \times \mathbb{F}_q^n : \mathbf{v} = \mathbf{F}(\mathbf{x})\}$. and a set $R_{\mathbf{F}}(\mathbf{v}) = \{\mathbf{x} : (\mathbf{v}, \mathbf{x}) \in R_{\mathbf{F}}\}$. Given an instance $\mathbf{F} \in \mathcal{MC}(n, m, \mathbb{F}_q)$ and a vector $\mathbf{v} \in \mathbb{F}_q^m$, the MC problem is finding a solution $\mathbf{s} \in R_{\mathbf{F}}(\mathbf{v})$. The associated linear-in-one-argument (LOA) form of the MC function is defined as follows.

Definition 2. *Let $\mathbf{F} = (f_1, \ldots, f_m) \in \mathcal{MC}(n, m, \mathbb{F}_q)$ and $f_l(x_1, \ldots, x_n) = \sum_{i,j,k} a_{l,i,j,k} x_i x_j x_k + \sum_{i,j} b_{l,i,j} x_i x_j + \sum_i c_{l,i} x_i$. Then a function $\mathbf{G} = (g_1, \ldots, g_m)$ is called the associated linear-in-one-argument (LOA) form of $\mathbf{F}$ if, for $l = 1, \ldots, m$, $g_l(x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i,j,k} a_{l,i,j,k}(x_i y_j y_k + y_i x_j y_k + y_i y_j x_k) + \sum_{i,j} b_{l,i,j} x_i y_j$.*

When $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$ are vectors of n variables, the associated LOA form $\mathbf{G}(\mathbf{x}, \mathbf{y})$ is linear with respect to the first argument $\mathbf{x}$. Moreover, it satisfies that $\mathbf{F}(\mathbf{x} + \mathbf{y}) = \mathbf{F}(\mathbf{x}) + \mathbf{G}(\mathbf{x}, \mathbf{y}) + \mathbf{G}(\mathbf{y}, \mathbf{x}) + \mathbf{F}(\mathbf{y})$.

Then, we study the intractability of the MC function. An intractability assumption for a random instance of $\mathcal{MC}(n, m, \mathbb{F}_q)$ is defined as follows.

Definition 3. *For polynomially bounded functions $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$, it is said that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is intractable if there is no polynomial-time algorithm that takes $(\mathbf{F}, \mathbf{v})$ generated via $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$, $\mathbf{s} \in_R \mathbb{F}_q^n$, and $\mathbf{v} \leftarrow \mathbf{F}(\mathbf{s})$ and finds a preimage $\mathbf{s}' \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{s}') = \mathbf{v}$ with non-negligible probability $\epsilon(\lambda)$.*

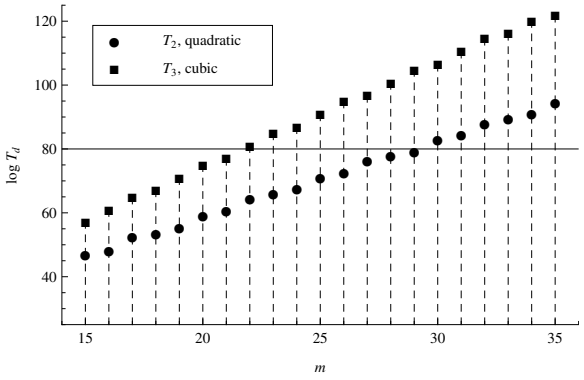

Fig. 1. The complexity of the hybrid approach where $n = m$, $q = 2^4$, and $w = 2$

All the state-of-the-art solving techniques have exponential complexity to break the intractability [7, 8, 11]. In particular, it is known that complexity of generic attacks using Gröbner basis is exponential in time and space [3, 11]. In this paper we use two sets of parameters $\mathcal{MC}(84, 80, \mathbb{F}_2)$ and $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ for 80-bit security. It is easy to see that the former achieves 80-bit security, because even a *quadratic* system with 84 variables and 80 equations over $\mathbb{F}_2$ satisfies 80-bit security [26]. In fact, the complexity of the improved exhaustive search algorithm [8] and the F_5 algorithm [11] to break $\mathcal{MC}(84, 80, \mathbb{F}_2)$ is more than 2^{80}. On the other hand, the latter requires more detailed analysis. The hybrid approach which is proposed by Bettale et al. [7] is the best known algorithm for solving multivariate cubic systems over $\mathbb{F}_{2^4}$. We follow their evaluation method of [7] to select the minimal parameters for 80-bit security and obtain the parameter set $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ as follows.

Let $D(n, m, d)$ be the degree of regularity of a semi-regular system with m equations of degree d in n variables. The complexity of solving a semi-regular system with n variables and m equations of degree d over $\mathbb{F}_q$ is estimated as $\min_{0 \le k \le n}(q^k \cdot (m \cdot \binom{n-k-1+D(n-k,m,d)}{D(n-k,m,d)})^w)$ where $2 \le w \le 3$. They stated that $D(n, m, d)$ corresponds to the index i of the first non-positive coefficient c_i of the series $\sum_{i>0} c_i \cdot z^i = \frac{(1-z^d)^m}{(1-z)^n}$. Let $T_d(m)$ be the complexity of the hybrid approach where $n = m$, $q = 2^4$, and $w = 2$. Figure 1 shows the comparison of $T_2(m)$ and $T_3(m)$. The complexity $T_3(m)$ increases faster than $T_2(m)$. In particular, $\min\{m | T_3(m) > 2^{80}\} = 22$ and $T_3(22) \approx 2^{81}$. Finally, the number of variables n is conservatively chosen as $n = \frac{3}{2}m$. Thus we can see that $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ achieves 80-bit security.

3 A 3-Pass Protocol

This section describes our 3-pass protocol which is statistical zero-knowledge argument of knowledge for $R_{\mathbf{F}}$ with knowledge error 3/4, assuming the existence

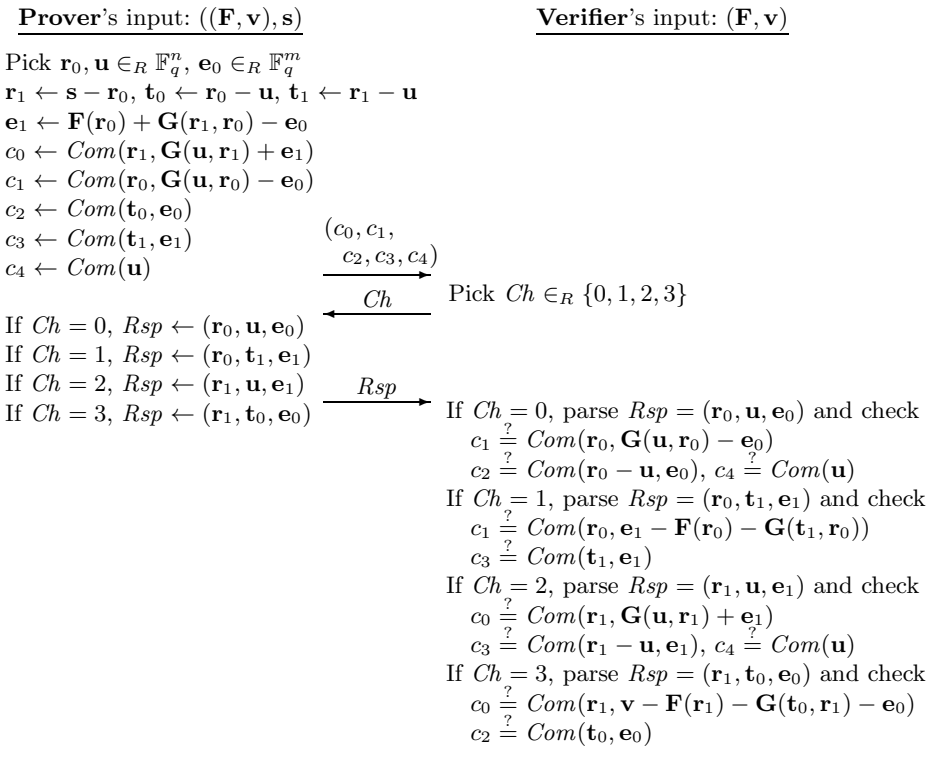

Fig. 2. Our 3-pass protocol

of a non-interactive commitment scheme Com which is statistically hiding and computationally binding.

We begin with describing a setup algorithm and a key-generation algorithm. Let λ be a security parameter. Let $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$ be polynomially bounded functions. The setup algorithm `Setup` takes 1^λ and outputs a system parameter $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ which consists of m-tuple of random multivariate cubic polynomials. The key-generation algorithm `Gen` takes $\mathbf{F}$. After choosing a random vector $\mathbf{s} \in_R \mathbb{F}_q^n$, `Gen` computes $\mathbf{v} \leftarrow \mathbf{F}(\mathbf{s})$, then outputs $(pk, sk) = (\mathbf{v}, \mathbf{s})$.

The basic idea for our 3-pass construction is that a prover proves that she has a tuple $(\mathbf{r}_0, \mathbf{r}_1, \mathbf{u}, \mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_0, \mathbf{e}_1)$ satisfying

$$\mathbf{G}(\mathbf{u}, \mathbf{r}_1) + \mathbf{e}_1 = \mathbf{v} - \mathbf{F}(\mathbf{r}_1) - \mathbf{G}(\mathbf{t}_0, \mathbf{r}_1) - \mathbf{e}_0, \tag{1}$$

$$\mathbf{t}_0 = \mathbf{r}_0 - \mathbf{u}, \tag{2}$$

$$\mathbf{t}_1 = \mathbf{r}_1 - \mathbf{u}, \tag{3}$$

$$\text{and} \quad \mathbf{G}(\mathbf{u}, \mathbf{r}_0) - \mathbf{e}_0 = \mathbf{e}_1 - \mathbf{F}(\mathbf{r}_0) - \mathbf{G}(\mathbf{t}_1, \mathbf{r}_0), \tag{4}$$

since if the tuple satisfies (1), (2), (3), and (4) then $\mathbf{v} = \mathbf{F}(\mathbf{r}_0 + \mathbf{r}_1)$. Note that $\mathbf{G}$ is the associated LOA form of $\mathbf{F}$. Then, corresponding to a challenge

$Ch \in \{0,1,2,3\}$ of a verifier, the prover reveals one out of four tuples $(\mathbf{r}_0, \mathbf{u}, \mathbf{e}_0)$, $(\mathbf{r}_0, \mathbf{t}_1, \mathbf{e}_1)$, $(\mathbf{r}_1, \mathbf{u}, \mathbf{e}_1)$, and $(\mathbf{r}_1, \mathbf{t}_0, \mathbf{e}_0)$. The verifier can check each side of each of the equations (1), (2), (3), and (4) by using some one of the four tuples. Such vectors $\mathbf{r}_0, \mathbf{r}_1, \mathbf{u}, \mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_0, \mathbf{e}_1$ are produced by using the dividing techniques described in Section 1. Thus, when $\mathbf{r}_0$, $\mathbf{u}$, and $\mathbf{e}_0$ are randomly chosen, the verifier can obtain no information on the secret key $\mathbf{s}$ from only one out of the four tuples.

The 3-pass protocol is described in Figure 2. For the simplicity, a random string ρ in Com is not written explicitly. The verifier finally outputs 1 if all of the checks "$\stackrel{?}{=}$" are passed, otherwise outputs 0. This is denoted by $0/1 \leftarrow Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1, c_2, c_3, c_4), Ch, Rsp)$. It is easy to see that the verifier always accepts an interaction with the honest prover. Thus the 3-pass protocol has perfect correctness.

Now we show two properties of the protocol in Theorem 4 and Theorem 5 as follows.

Theorem 4. *The 3-pass protocol is statistically zero knowledge when the commitment scheme Com is statistically hiding.*

Proof sketch. Let $\mathcal{S}$ be a simulator which takes $\mathbf{F}$ and $\mathbf{v}$ without knowing $\mathbf{s}$, and interacts with a cheating verifier $\mathcal{CV}$. We show that the simulator $\mathcal{S}$ can impersonate the honest prover with probability 3/4.

The simulator $\mathcal{S}$ randomly chooses a value $Ch^* \in_R \{0,1,2,3\}$ and vectors $\mathbf{s}', \mathbf{r}'_0, \mathbf{u}' \in_R \mathbb{F}_q^n$, $\mathbf{e}'_0 \in_R \mathbb{F}_q^m$, where Ch^* is a prediction of what value the cheating verifier $\mathcal{CV}$ will *not* choose. Then, it computes $\mathbf{r}'_1 \leftarrow \mathbf{s}' - \mathbf{r}'_0$, $\mathbf{t}'_0 \leftarrow \mathbf{r}'_0 - \mathbf{u}'$, and $\mathbf{t}'_1 \leftarrow \mathbf{r}'_1 - \mathbf{u}'$. If $Ch^* \in \{0,1\}$ then it computes $\mathbf{e}'_1 \leftarrow \mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{r}'_0, \mathbf{r}'_1) - \mathbf{e}'_0$, else $\mathbf{e}'_1 \leftarrow \mathbf{F}(\mathbf{r}'_0) + \mathbf{G}(\mathbf{r}'_1, \mathbf{r}'_0) - \mathbf{e}'_0$. If $Ch^* = 2$ then it computes $c'_0 \leftarrow Com(\mathbf{r}'_1, \mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{t}'_0, \mathbf{r}'_1) - \mathbf{e}'_0)$, else $c'_0 \leftarrow Com(\mathbf{r}'_1, \mathbf{G}(\mathbf{u}', \mathbf{r}'_1) + \mathbf{e}'_1)$. If $Ch^* = 0$ then it computes $c'_1 \leftarrow Com(\mathbf{r}'_0, \mathbf{e}'_1 - \mathbf{F}(\mathbf{r}'_0) - \mathbf{G}(\mathbf{t}'_1, \mathbf{r}'_0))$, else $c'_1 \leftarrow Com(\mathbf{r}'_0, \mathbf{G}(\mathbf{u}', \mathbf{r}'_0) - \mathbf{e}'_0)$. It computes $c'_2 \leftarrow Com(\mathbf{t}'_0, \mathbf{e}'_0)$, $c'_3 \leftarrow Com(\mathbf{t}'_1, \mathbf{e}'_1)$, and $c'_4 \leftarrow Com(\mathbf{u}')$ and sends $(c'_0, c'_1, c'_2, c'_3, c'_4)$ to $\mathcal{CV}$.

Due to the statistically hiding property of Com, a challenge Ch from $\mathcal{CV}$ is different from Ch^* with probability 3/4. If $Ch \neq Ch^*$ then $(\mathbf{r}'_0, \mathbf{u}', \mathbf{e}'_0)$, $(\mathbf{r}'_0, \mathbf{t}'_1, \mathbf{e}'_1)$, $(\mathbf{r}'_1, \mathbf{u}', \mathbf{e}'_1)$, and $(\mathbf{r}'_1, \mathbf{t}'_0, \mathbf{e}'_0)$ are accepted responses to $Ch = 0$, 1, 2, and 3, respectively. Note that if $Ch^* \in \{0,1\}$ and $Ch = 3$ then it is seen that $\mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{t}'_0, \mathbf{r}'_1) - \mathbf{e}'_0 = \mathbf{e}'_1 + \mathbf{G}(\mathbf{r}'_0 - \mathbf{t}'_0, \mathbf{r}'_1) = \mathbf{e}'_1 + \mathbf{G}(\mathbf{u}', \mathbf{r}'_1)$, since $\mathbf{e}'_1 = \mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{r}'_0, \mathbf{r}'_1) - \mathbf{e}'_0$ and $\mathbf{t}'_0 = \mathbf{r}'_0 - \mathbf{u}'$. Note that if $Ch^* \in \{2,3\}$ and $Ch = 1$ then it is seen that $\mathbf{e}'_1 - \mathbf{F}(\mathbf{r}'_0) - \mathbf{G}(\mathbf{t}'_1, \mathbf{r}'_0) = \mathbf{G}(\mathbf{r}'_1 - \mathbf{t}'_1, \mathbf{r}'_0) - \mathbf{e}'_0 = \mathbf{G}(\mathbf{u}', \mathbf{r}'_0) - \mathbf{e}'_0$, since $\mathbf{e}'_1 = \mathbf{F}(\mathbf{r}'_0) + \mathbf{G}(\mathbf{r}'_1, \mathbf{r}'_0) - \mathbf{e}'_0$ and $\mathbf{t}'_1 = \mathbf{r}'_1 - \mathbf{u}'$. □

Theorem 5. *The 3-pass protocol is argument of knowledge for $R_{\mathbf{F}}$ with knowledge error 3/4 when the commitment scheme Com is computationally binding.*

Proof sketch. For $i \in \{0,1,2,3\}$, let $((c_0, c_1, c_2, c_3, c_4), Ch_i, Rsp_i)$ be a transcript such that $Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1, c_2, c_3, c_4), Ch_i, Rsp_i) = 1$ and $Ch_i = i$. Then, by using the four transcripts, it is shown to be able to either break the binding property of Com or extract a solution for $\mathbf{v}$. Consider the situation where the responses are parsed as $Rsp_0 = (\tilde{\mathbf{r}}_0^{(0)}, \tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{e}}_0^{(0)})$, $Rsp_1 = (\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{e}}_1^{(1)})$,

$Rsp_2 = (\tilde{\mathbf{r}}_1^{(2)}, \tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{e}}_1^{(2)})$, and $Rsp_3 = (\tilde{\mathbf{r}}_1^{(3)}, \tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{e}}_0^{(3)})$. Then, it is seen that

$$\begin{aligned} c_0 &= Com(\tilde{\mathbf{r}}_1^{(2)}, \mathbf{G}(\tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{r}}_1^{(2)}) + \tilde{\mathbf{e}}_1^{(2)}) \\ &= Com(\tilde{\mathbf{r}}_1^{(3)}, \mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(3)}) - \mathbf{G}(\tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{r}}_1^{(3)}) - \tilde{\mathbf{e}}_0^{(3)}), \quad (5) \\ c_1 &= Com(\tilde{\mathbf{r}}_0^{(0)}, \mathbf{G}(\tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{r}}_0^{(0)}) - \tilde{\mathbf{e}}_0^{(0)}) \\ &= Com(\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{e}}_1^{(1)} - \mathbf{F}(\tilde{\mathbf{r}}_0^{(1)}) - \mathbf{G}(\tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{r}}_0^{(1)})), \quad (6) \\ c_2 &= Com(\tilde{\mathbf{r}}_0^{(0)} - \tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{e}}_0^{(0)}) = Com(\tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{e}}_0^{(3)}), \quad (7) \\ c_3 &= Com(\tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{e}}_1^{(1)}) = Com(\tilde{\mathbf{r}}_1^{(2)} - \tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{e}}_1^{(2)}), \quad \text{and} \quad (8) \\ c_4 &= Com(\tilde{\mathbf{u}}^{(0)}) = Com(\tilde{\mathbf{u}}^{(2)}). \quad (9) \end{aligned}$$

If the two pairs of the arguments of *Com* are distinct on any one of the above equations, the binding property of *Com* is broken. Otherwise, the equation (5) yields $\mathbf{v} = \mathbf{G}(\tilde{\mathbf{u}}^{(2)} + \tilde{\mathbf{t}}_0^{(3)}, \tilde{\mathbf{r}}_1^{(2)}) + \mathbf{F}(\tilde{\mathbf{r}}_1^{(2)}) + \tilde{\mathbf{e}}_1^{(2)} + \tilde{\mathbf{e}}_0^{(3)}$. By combining it with the equations (6), (7), and (8), it is seen that $\mathbf{v} = \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_0^{(0)} + \tilde{\mathbf{u}}^{(2)} - \tilde{\mathbf{u}}^{(0)}, \tilde{\mathbf{r}}_1^{(2)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(2)} + \tilde{\mathbf{u}}^{(0)} - \tilde{\mathbf{u}}^{(2)}, \tilde{\mathbf{r}}_0^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_1^{(2)})$. Finally, putting it together with the equation (9), we obtain $\mathbf{v} = \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_0^{(0)}, \tilde{\mathbf{r}}_1^{(2)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(2)}, \tilde{\mathbf{r}}_0^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_1^{(2)}) = \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)} + \tilde{\mathbf{r}}_1^{(2)})$. It means that a solution $\tilde{\mathbf{r}}_0^{(0)} + \tilde{\mathbf{r}}_1^{(2)}$ for $\mathbf{v}$ is extracted. □

Extension. A standard trick for saving the communication data size can be applied to our 3-pass protocol. The trick employs a collision resistant hash function H. Let $c_a = H(c_0, c_2)$ and $c_b = H(c_1, c_3)$ be hash values. In the first pass, a prover sends one hash value $c = H(c_a, c_b, c_4)$ instead of five commitments $(c_0, c_1, c_2, c_3, c_4)$. In the third pass, the pairs of the hash values (c_0, c_3), (c_a, c_4), (c_1, c_2), and (c_b, c_4) are appended to prover's responses *Rsp* for $Ch = 0, 1, 2,$ and 3, respectively. Finally, a verifier checks $c = H(c_a, c_b, c_4)$. We note that the hash values c_a, c_b, and c_4 can be obtained from the prover's response *Rsp* in every case of $Ch = 0, 1, 2,$ and 3. As a result, the number of hash values sent is reduced from 5 to 3. The modified version of our 3-pass protocol is also shown to be zero-knowledge argument of knowledge with knowledge error 3/4.

4 A 5-Pass Protocol

This section describes our 5-pass protocol which is statistical zero-knowledge argument of knowledge for $R_{\mathbf{F}}$ with knowledge error $1/2 + 1/2q$, assuming the existence of a non-interactive commitment scheme *Com* which is statistically hiding and computationally binding. The knowledge error of the 5-pass protocol is smaller than that of the 3-pass protocol when $q \geq 3$. The setup algorithm and the key-generation algorithm for the 5-pass protocol are identical to those for the 3-pass protocol.

In the 5-pass protocol, a prover also divides the secret key $\mathbf{s}$ and the public key $\mathbf{F}(\mathbf{s})$ as $\mathbf{s} = \mathbf{r}_0 + \mathbf{r}_1$ and $\mathbf{F}(\mathbf{s}) = \mathbf{F}(\mathbf{r}_0 + \mathbf{r}_1) = \mathbf{F}(\mathbf{r}_0) + \mathbf{F}(\mathbf{r}_1) + \mathbf{G}(\mathbf{r}_0, \mathbf{r}_1) + \mathbf{G}(\mathbf{r}_1, \mathbf{r}_0)$,

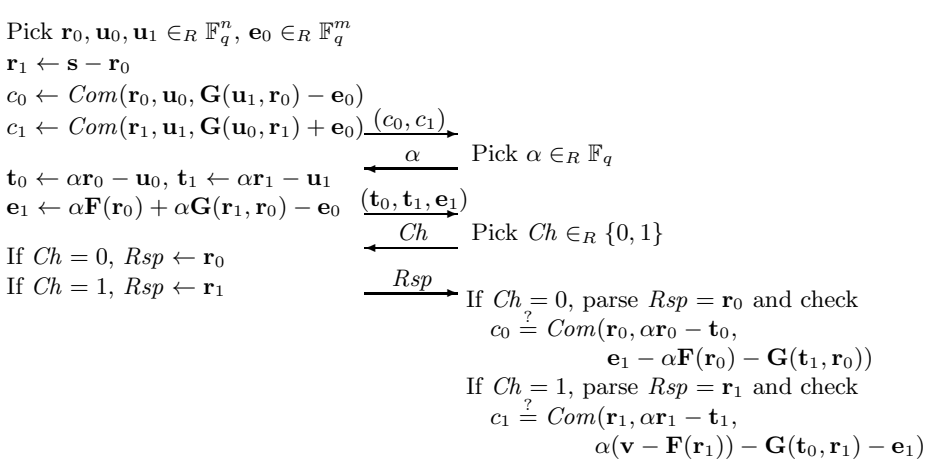

Fig. 3. Our 5-pass protocol

respectively. The difference from the 3-pass protocol is that $\mathbf{r}_0$, $\mathbf{r}_1$, and $\mathbf{F}(\mathbf{r}_0) + \mathbf{G}(\mathbf{r}_1, \mathbf{r}_0)$ are divided as $\alpha\mathbf{r}_0 = \mathbf{t}_0 + \mathbf{u}_0$, $\alpha\mathbf{r}_1 = \mathbf{t}_1 + \mathbf{u}_1$, and $\alpha\mathbf{F}(\mathbf{r}_0) + \alpha\mathbf{G}(\mathbf{r}_1, \mathbf{r}_0) = \mathbf{e}_0 + \mathbf{e}_1$ where $\alpha \in \mathbb{F}_q$ is a choice of a verifier. In particular, we note that $\mathbf{r}_0$ and $\mathbf{r}_1$ are divided by using two independent vectors $\mathbf{u}_0$ and $\mathbf{u}_1$. The reason is that the prover of the 5-pass protocol sends *both* $\mathbf{t}_0$ and $\mathbf{t}_1$, while that of the 3-pass protocol sends *either* $\mathbf{t}_0$ or $\mathbf{t}_1$. After sending $(\mathbf{t}_0, \mathbf{t}_1, \mathbf{e}_1)$ to the verifier, corresponding to a challenge $Ch \in \{0, 1\}$ of the verifier, the prover reveals one out of two vectors $\mathbf{r}_0$ and $\mathbf{r}_1$. When $\mathbf{r}_0$, $\mathbf{u}_0$, $\mathbf{u}_1$, and $\mathbf{e}_0$ are randomly chosen, the verifier can obtain no information on the secret key $\mathbf{s}$ from only one out of the two vectors $\mathbf{r}_0$ and $\mathbf{r}_1$. On the other hand, the argument-of-knowledge property comes from that, for more than one choice of $\alpha \in \mathbb{F}_q$, an impersonator cannot response both of verifier's challenges $Ch = 0$ and $Ch = 1$ unless the impersonator has a solution $\mathbf{s}$ for $\mathbf{v}$.

The 5-pass protocol is described in Figure 3 where $\mathbf{G}$ is the associated LOA form of $\mathbf{F}$. The verifier finally outputs 1 if the check of "$\stackrel{?}{=}$" is passed, otherwise outputs 0. This is denoted by $0/1 \leftarrow Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1), \alpha, (\mathbf{t}_1, \mathbf{e}_1), Ch, Rsp)$. It is easy to see that the verifier always accepts an interaction with the honest prover. Thus the 5-pass protocol has perfect correctness.

Now we show two properties of the protocol in Theorem 6 and Theorem 7 as follows.

Theorem 6. *The 5-pass protocol is statistically zero knowledge when the commitment scheme Com is statistically hiding.*

Proof sketch. Let $\mathcal{S}$ be a simulator which takes $\mathbf{F}$ and $\mathbf{v}$ without knowing $\mathbf{s}$, and interacts with a cheating verifier $\mathcal{CV}$. We show that the simulator $\mathcal{S}$ can impersonate the honest prover with probability 1/2. The simulator $\mathcal{S}$ randomly chooses a value $Ch^* \in_R \{0, 1\}$ and vectors $\mathbf{s}', \mathbf{r}'_0, \mathbf{u}'_0, \mathbf{u}'_1 \in_R \mathbb{F}_q^n$, $\mathbf{e}'_0 \in_R \mathbb{F}_q^m$,

where Ch^* is a prediction of what value the cheating verifier $\mathcal{CV}$ *will* choose. Then, it computes $\mathbf{r}'_1 \leftarrow \mathbf{s}' - \mathbf{r}'_0$, $c'_0 \leftarrow Com(\mathbf{r}'_0, \mathbf{u}'_0, \mathbf{G}(\mathbf{u}'_1, \mathbf{r}'_0) - \mathbf{e}'_0)$, and $c'_1 \leftarrow Com(\mathbf{r}'_1, \mathbf{u}'_1, \mathbf{G}(\mathbf{u}'_0, \mathbf{r}'_1) + \mathbf{e}'_0)$. It sends (c'_0, c'_1) to $\mathcal{CV}$. Receiving a challenge α from $\mathcal{CV}$, it computes $\mathbf{t}'_0 \leftarrow \alpha\mathbf{r}'_0 - \mathbf{u}'_0$ and $\mathbf{t}'_1 \leftarrow \alpha\mathbf{r}'_1 - \mathbf{u}'_1$. If $Ch^* = 0$ then it computes $\mathbf{e}'_1 \leftarrow \alpha\mathbf{F}(\mathbf{r}'_0) + \alpha\mathbf{G}(\mathbf{r}'_1, \mathbf{r}'_0) - \mathbf{e}'_0$, else $\mathbf{e}'_1 \leftarrow \alpha(\mathbf{v} - \mathbf{F}(\mathbf{r}'_1) - \mathbf{G}(\mathbf{r}'_0, \mathbf{r}'_1)) - \mathbf{e}'_0$. It sends $(\mathbf{t}'_0, \mathbf{t}'_1, \mathbf{e}'_1)$ to $\mathcal{CV}$. Due to the statistically hiding property of Com, a challenge Ch from $\mathcal{CV}$ is equal to Ch^* with probability $1/2$. If $Ch = Ch^*$ then $\mathbf{r}'_0$ and $\mathbf{r}'_1$ are accepted responses to $Ch = 0$ and 1, respectively. Note that the case of $\alpha = 0$ does not spoil the zero-knowledge property. □

Theorem 7. *The 5-pass protocol is argument of knowledge for $R_{\mathbf{F}}$ with knowledge error $1/2 + 1/2q$ when the commitment scheme Com is computationally binding.*

Proof sketch. Let $\alpha_0, \alpha_1 \in \mathbb{F}_q$ such that $\alpha_0 \neq \alpha_1$. For $(i, j) \in \{(0,0), (0,1), (1,0), (1,1)\}$, let $((c_0, c_1), \alpha_i, (\tilde{\mathbf{t}}_0^{(i)}, \tilde{\mathbf{t}}_1^{(i)}, \tilde{\mathbf{e}}_1^{(i)}), Ch_j, Rsp^{(i,j)})$ be a transcript such that $Dec(\mathbf{F}, \mathbf{v}; (c_0, c_1), \alpha_i, (\tilde{\mathbf{t}}_0^{(i)}, \tilde{\mathbf{t}}_1^{(i)}, \tilde{\mathbf{e}}_1^{(i)}), Ch_j, Rsp^{(i,j)}) = 1$ and $Ch_j = j$. By using the four transcripts, it is shown to be able to either break the binding property of Com or extract a solution for $\mathbf{v}$. Consider that the responses are parsed as $Rsp^{(0,0)} = \tilde{\mathbf{r}}_0^{(0)}$, $Rsp^{(0,1)} = \tilde{\mathbf{r}}_1^{(0)}$, $Rsp^{(1,0)} = \tilde{\mathbf{r}}_0^{(1)}$, and $Rsp^{(1,1)} = \tilde{\mathbf{r}}_1^{(1)}$. Then, it is seen that

$$\begin{aligned} c_0 &= Com(\tilde{\mathbf{r}}_0^{(0)}, \alpha_0\tilde{\mathbf{r}}_0^{(0)} - \tilde{\mathbf{t}}_0^{(0)}, \tilde{\mathbf{e}}_1^{(0)} - \alpha_0\mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) - \mathbf{G}(\tilde{\mathbf{t}}_1^{(0)}, \tilde{\mathbf{r}}_0^{(0)})) \\ &= Com(\tilde{\mathbf{r}}_0^{(1)}, \alpha_1\tilde{\mathbf{r}}_0^{(1)} - \tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{e}}_1^{(1)} - \alpha_1\mathbf{F}(\tilde{\mathbf{r}}_0^{(1)}) - \mathbf{G}(\tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{r}}_0^{(1)})) \quad \text{and} \qquad (10) \\ c_1 &= Com(\tilde{\mathbf{r}}_1^{(0)}, \alpha_0\tilde{\mathbf{r}}_1^{(0)} - \tilde{\mathbf{t}}_1^{(0)}, \alpha_0(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)})) - \mathbf{G}(\tilde{\mathbf{t}}_0^{(0)}, \tilde{\mathbf{r}}_1^{(0)}) - \tilde{\mathbf{e}}_1^{(0)}) \\ &= Com(\tilde{\mathbf{r}}_1^{(1)}, \alpha_1\tilde{\mathbf{r}}_1^{(1)} - \tilde{\mathbf{t}}_1^{(1)}, \alpha_1(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(1)})) - \mathbf{G}(\tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(1)}) - \tilde{\mathbf{e}}_1^{(1)}). \qquad (11) \end{aligned}$$

If the two tuples of the arguments of Com are distinct on either of the above equations, the binding property of Com is broken. Otherwise, it is seen that $(\alpha_0 - \alpha_1)(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)})) = \mathbf{G}(\tilde{\mathbf{t}}_0^{(0)} - \tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + \tilde{\mathbf{e}}_1^{(0)} - \tilde{\mathbf{e}}_1^{(1)}$ and $\tilde{\mathbf{t}}_1^{(0)} - \tilde{\mathbf{t}}_1^{(1)} = (\alpha_0 - \alpha_1)\tilde{\mathbf{r}}_1^{(0)}$ from the equation (11). Combining them with the equation (10) yields $(\alpha_0 - \alpha_1)(\mathbf{v} - \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)})) = \mathbf{G}(\tilde{\mathbf{t}}_0^{(0)} - \tilde{\mathbf{t}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + (\alpha_0 - \alpha_1)\mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{t}}_1^{(0)} - \tilde{\mathbf{t}}_1^{(1)}, \tilde{\mathbf{r}}_0^{(0)}) = (\alpha_0 - \alpha_1)(\mathbf{G}(\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(0)}, \tilde{\mathbf{r}}_0^{(0)}))$. Thus, we obtain $\mathbf{v} = \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_0^{(1)}, \tilde{\mathbf{r}}_1^{(0)}) + \mathbf{G}(\tilde{\mathbf{r}}_1^{(0)}, \tilde{\mathbf{r}}_0^{(0)}) + \mathbf{F}(\tilde{\mathbf{r}}_0^{(0)}) = \mathbf{F}(\tilde{\mathbf{r}}_1^{(0)} + \tilde{\mathbf{r}}_0^{(0)})$, since $\alpha_0 \neq \alpha_1$. It means that a solution $\tilde{\mathbf{r}}_1^{(0)} + \tilde{\mathbf{r}}_0^{(0)}$ for $\mathbf{v}$ is extracted. □

5 Security and Efficiency

This section we summarize the security of our identification schemes which is easily obtained from results in Section 3 and 4, and evaluate the efficiency of our schemes for a practical parameter choice.

5.1 Security

Here we briefly mention the security of each of the sequential and the parallel compositions in the same way as [26]. Let $(\mathtt{P}, \mathtt{V})$ be our 3-pass protocol or 5-pass protocol and ϵ its knowledge error. Let $N = \omega(\log \lambda)$. Then identification protocols which consist of repeating $(\mathtt{P}, \mathtt{V})$ N-times in sequential and in parallel are denoted by $(\mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ and $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$, respectively. When $\mathcal{MC}(n, m, \mathbb{F}_q)$ is intractable and the commitment scheme *Com* is statistically hiding and computationally binding, the security of our identification schemes $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ and $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is evaluated as follows.

The former $(\mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ is statistically zero-knowledge argument of knowledge with knowledge error ϵ^N due to the sequential composition lemma [15] and Stern's proof techniques of [29, 30]. Thus the identification scheme $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(s)}, \mathtt{V}_N^{(s)})$ is secure against impersonation under *active* attack. On the other hand, the parallel repetition of $(\mathtt{P}, \mathtt{V})$ reserves zero-knowledge with respect to an *honest verifier*. By combining it with Pass and Venkitasubramaniam's result [20], the latter $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is also honest-verifier zero-knowledge argument of knowledge with a negligible knowledge error. Therefore, the identification scheme $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is secure against impersonation under *passive* attack. In addition, for a certain parameter choice, the parallel version of our scheme is also secure under *active* attack as shown in Section 6.

5.2 Efficiency

The efficiency of the schemes consisting of our 5-pass protocol is highly comparable to that of the schemes based on binary SD, q-ary SD, CLE, PP, PK, and MQ, even though our 3-pass protocol is not so efficient. Here we evaluate the data sizes of system parameters, a public key, a secret key, and a transcript of our schemes. The numbers of arithmetic operations, computing permutations, and computing hash functions are also estimated as computational cost. These are evaluated according to [9, 26]. In this paper $\mathcal{MC}(84, 80, \mathbb{F}_2)$ and $\mathcal{MC}(33, 22, \mathbb{F}_{2^4})$ are used for our 3-pass protocol and for our 5-pass one, respectively.

First, we consider the schemes consisting of each of the 3-pass protocols. Table 1 compares our scheme with the schemes based on binary SD, CLE, PP, and MQ when each protocol is sequentially repeated until impersonation probability is less than 2^{-30}. In this comparison we consider the case where each scheme uses techniques for saving the communication data size such as the trick mentioned at the end of Section 3.

Second, consider the 5-pass protocols. Table 2 compares our scheme with the schemes based on binary SD, q-ary SD, CLE, PK, PP, and MQ when each protocol is sequentially repeated until impersonation probability is less than 2^{-30}. The data sizes of a public key and a secret key of our scheme are smaller than those of the other schemes. The communication data size is almost the smallest in Table 2. Although the size of system parameter of our scheme is relatively large, it can be reduced to some small seed, e.g. 128 bits, if a pseudo-random

Table 1. Comparison of 3-pass schemes on 80-bit security against key-recovery attack when the impersonation probability is less than 2^{-30}

	SD [30]	CLE [29]	PP [25]	MQ [26]	Our
round	52	52	73	52	73
system parameter (bit)	122,500	4,608	28,497	285,600	7,908,320
public key (bit)	350	288	245	80	80
secret key (bit)	700	192	177	84	84
communication (bit)	59,800	45,517	100,925	29,640	53,290
arithmetic ops. (times/field)	2^{24} / $\mathbb{F}_2$	2^{16} / $\mathbb{F}_{257}$	2^{22} / $\mathbb{F}_{127}$	2^{26} / $\mathbb{F}_2$	2^{32} / $\mathbb{F}_2$
permutations[*1] (times/size)	$2/S_{700}$	$4/S_{24}$	$2/S_{161}$,S_{177}	NO	NO
hash function (times)	4	4	8	4	8
best known key-recovery attack	2^{87}	2^{84}	$> 2^{74}$	2^{80}	2^{80}

Table 2. Comparison of 5-pass schemes on 80-bit security against key-recovery attack when the impersonation probability is less than 2^{-30}

	SD [30]	SD [9]	PK [27]	CLE [29]	PP [24, 25]	MQ [26]	Our
round	31	31	31	31	52	33	33
system parameter (bit)	122,500	32,768	4,608	4,608	28,497	129,600[*2]	581,768
public key (bit)	2450	512	384	288	245	120	88
secret key (bit)	4900	1024	203	192	177	180	132
communication (bit)	120,652	61,783	27,234	27,528	105,060	26,565	26,697
arithmetic ops. (times/field)	$2^{23}/\mathbb{F}_2$	$2^{18}/\mathbb{F}_{256}$	$2^{15}/\mathbb{F}_{251}$	$2^{15}/\mathbb{F}_{257}$	$2^{21}/\mathbb{F}_{127}$	$2^{22}/\mathbb{F}_{2^4}$	$2^{27}/\mathbb{F}_{2^4}$
permutations[*1] (times/size)	$8/S_{700}$	$2/S_{128}$	$3/S_{48}$	$4/S_{24}$	$2/S_{161}$,S_{177}	NO	NO
hash function (times)	2	2	2	2	5	2	2
best known key-recovery attack	2^{87}	2^{87}	2^{85}	2^{84}	$> 2^{74}$	2^{83}	2^{81}

[*1] This shows the number of times of computing permutations and the size of the permutation, where S_n means a permutation over $\{1, \ldots, n\}$.

[*2] This is the correct size of the system parameters, although it is stated as 259,200 bits in the original paper [26].

number generator is used as the implementation of QUAD [2]. Although the cost of arithmetic operations of our scheme is relatively high, it is still reasonable.

6 On the Security against Active Attack in Parallel Repetition

In this section we focus on the case of $n = m + k$ and $k = \omega(\log \lambda)$. For example, the MC function $\mathbf{F} \in \mathcal{MC}(2m, m, \mathbb{F}_q)$ satisfies the requirement where $m = \omega(\log \lambda)$. In this case, $(\mathtt{Setup}, \mathtt{Gen}, \mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ is shown to be secure against impersonation under *active* attack, although the data sizes of the secret key and the communication increase at most double compared to those of Section 5.2. The security can be shown in almost the same way as that of the MQ-based scheme. Although we consider the scheme consisting of our 3-pass protocol in this section, the same argument can also be applied to that of our 5-pass protocol.

We begin with defining the preimage resistance and the second-preimage resistance of the MC function. Note that the difference between the preimage

resistance and the intractability of Definition 3 is only in the distribution of the challenge $\mathbf{v}$, and both of them are widely believed.

Definition 8. *For polynomially bounded functions $n = n(\lambda)$, $m = m(\lambda)$, and $q = q(\lambda)$, it is said that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is preimage resistant if there is no polynomial-time algorithm that takes $(\mathbf{F}, \mathbf{v})$ generated via $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ and $\mathbf{v} \in_R \mathbb{F}_q^m$ and finds a preimage $\mathbf{s} \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{s}) = \mathbf{v}$ with non-negligible probability $\epsilon(\lambda)$. On the other hand, it is said that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is second-preimage resistant if there is no polynomial-time algorithm that takes $(\mathbf{F}, \mathbf{x})$ generated via $\mathbf{F} \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ and $\mathbf{x} \in_R \mathbb{F}_q^n$ and finds a second preimage $\mathbf{x}' \in \mathbb{F}_q^n$ such that $\mathbf{F}(\mathbf{x}') = \mathbf{F}(\mathbf{x})$ and $\mathbf{x}' \neq \mathbf{x}$ with non-negligible probability $\epsilon(\lambda)$.*

Then we present the following lemma.

Lemma 9. *If there exists an algorithm that breaks the second-preimage resistance of $\mathcal{MC}(n+1, m, \mathbb{F}_q)$ with advantage ϵ, then there exists an algorithm that breaks the preimage resistance of $\mathcal{MC}(n, m, \mathbb{F}_q)$ with advantage $\epsilon/(q-1)(n+1)$. That is, if $\mathcal{MC}(n, m, \mathbb{F}_q)$ is preimage resistant, then $\mathcal{MC}(n+1, m, \mathbb{F}_q)$ is second-preimage resistant.*

Proof sketch. Let $\mathcal{A}$ be an algorithm that breaks the second-preimage resistance of $\mathcal{MC}(n+1, m, \mathbb{F}_q)$. Let $\mathbf{F} = (f_1, \ldots, f_m) \in_R \mathcal{MC}(n, m, \mathbb{F}_q)$ and $\mathbf{v} = (v_1, \ldots, v_m) \in_R \mathbb{F}_q^m$, where $f_l(x_1, \ldots, x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} a_{l,i,j,k}x_ix_jx_k + \sum_{i=1}^{n}\sum_{j=1}^{n} b_{l,i,j}x_ix_j + \sum_{i=1}^{n} c_{l,i}x_i$. We show that, given $\mathbf{F}$ and $\mathbf{v}$, a preimage $\mathbf{x}$ satisfying $\mathbf{v} = \mathbf{F}(\mathbf{x})$ can be found by using the algorithm $\mathcal{A}$. For the simplicity, suppose that the algorithm $\mathcal{A}$ takes $\tilde{\mathbf{F}} = (\tilde{f}_1, \ldots, \tilde{f}_m) \in \mathcal{MC}(n+1, m, \mathbb{F}_q)$ and $\mathbf{t} = (t_1, \ldots, t_{n+1}) \in \mathbb{F}_q^{n+1}$ and outputs a second preimage $\mathbf{t} + \Delta$ such that $\tilde{\mathbf{F}}(\mathbf{t} + \Delta) = \tilde{\mathbf{F}}(\mathbf{t})$ and $\Delta = (d_1, \ldots, d_n, 1)$, where $\tilde{f}_l(x_1, \ldots, x_{n+1}) = \sum_{i=1}^{n+1}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1} \tilde{a}_{l,i,j,k}x_ix_jx_k + \sum_{i=1}^{n+1}\sum_{j=1}^{n+1} \tilde{b}_{l,i,j}x_ix_j + \sum_{i=1}^{n+1} \tilde{c}_{l,i}x_i$. Note that in the full proof it is necessary to guess an index ξ and a value d_ξ of a non-zero element in Δ, but in this proof sketch we suppose $\xi = n+1$ and $d_\xi = 1$. In this case, the equation $\tilde{\mathbf{F}}(\mathbf{t} + \Delta) - \tilde{\mathbf{F}}(\mathbf{t}) = \mathbf{0}$ is expanded as follows:

$$
\begin{aligned}
&\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{n} \tilde{a}_{l,i,j,k}d_id_jd_k \\
&+ \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\begin{array}{l}\displaystyle\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,i,k,j} + \tilde{a}_{l,j,i,k})t_k \\ \qquad +\tilde{b}_{l,i,j} + \tilde{a}_{l,i,j,n+1} + \tilde{a}_{l,n+1,i,j} + \tilde{a}_{l,i,n+1,j}\end{array}\right)d_id_j \\
&+ \sum_{i=1}^{n}\left(\begin{array}{l}\displaystyle\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,j,i,k} + \tilde{a}_{l,i,k,j})t_jt_k \\ \displaystyle+\sum_{k=1}^{n+1}\left(\begin{array}{l}\tilde{a}_{l,k,i,n+1} + \tilde{a}_{l,n+1,k,i} + \tilde{a}_{l,n+1,i,k} \\ +\tilde{a}_{l,k,n+1,i} + \tilde{a}_{l,i,k,n+1} + \tilde{a}_{l,i,n+1,k} + \tilde{b}_{l,k,i} + \tilde{b}_{l,i,k}\end{array}\right)t_k \\ +\tilde{a}_{l,n+1,i,n+1} + \tilde{a}_{l,i,n+1,n+1} + \tilde{a}_{l,n+1,n+1,i} + \tilde{b}_{l,n+1,i} + \tilde{b}_{l,i,n+1} + \tilde{c}_{l,i}\end{array}\right)d_i
\end{aligned}
$$

$$+\begin{pmatrix}\sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,n+1}+\tilde{a}_{l,j,n+1,k}+\tilde{a}_{l,n+1,k,j})t_jt_k\\ +\sum_{k=1}^{n+1}(\tilde{a}_{l,k,n+1,n+1}+\tilde{a}_{l,n+1,k,n+1}+\tilde{a}_{l,n+1,n+1,k}+\tilde{b}_{l,k,n+1}+\tilde{b}_{l,n+1,k})t_k\\ +\tilde{a}_{l,n+1,n+1,n+1}+\tilde{b}_{l,n+1,n+1}+\tilde{c}_{l,n+1}\end{pmatrix}=0$$

for $l = 1, \ldots, m$. From the above equation, we can see that the output $\mathbf{t} + \Delta$ of $\mathcal{A}$ satisfies $\mathbf{v} = \mathbf{F}(d_1, \ldots, d_n)$ if the input $(\tilde{\mathbf{F}}, \mathbf{t})$ of $\mathcal{A}$ is produced as follows.

- The vector $\mathbf{t}$ is generated via $\mathbf{t} \in_R \mathbb{F}_q^{n+1}$.
- For $1 \le i, j, k \le n$ do $\tilde{a}_{l,i,j,k} \leftarrow a_{l,i,j,k}$, otherwise $\tilde{a}_{l,i,j,k} \in_R \mathbb{F}_q$.
- For $1 \le i, j \le n$ do $\tilde{b}_{l,i,j} \leftarrow b_{l,i,j} - \sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,i,k,j} + \tilde{a}_{l,j,i,k})t_k - (\tilde{a}_{l,i,j,n+1} + \tilde{a}_{l,n+1,i,j} + \tilde{a}_{l,i,n+1,j})$, otherwise $\tilde{b}_{l,i,j} \in_R \mathbb{F}_q$.
- For $1 \le i \le n$ do $\tilde{c}_{l,i} \leftarrow c_{l,i} - \sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,i} + \tilde{a}_{l,j,i,k} + \tilde{a}_{l,i,k,j})t_jt_k - \sum_{k=1}^{n+1}(\tilde{a}_{l,k,i,n+1} + \tilde{a}_{l,n+1,k,i} + \tilde{a}_{l,n+1,i,k} + \tilde{a}_{l,k,n+1,i} + \tilde{a}_{l,i,k,n+1} + \tilde{a}_{l,i,n+1,k} + \tilde{b}_{l,k,i} + \tilde{b}_{l,i,k})t_k - (\tilde{a}_{l,n+1,i,n+1} + \tilde{a}_{l,i,n+1,n+1} + \tilde{a}_{l,n+1,n+1,i} + \tilde{b}_{l,n+1,i} + \tilde{b}_{l,i,n+1})$.
- $\tilde{c}_{l,n+1} \leftarrow -v_l - \sum_{j=1}^{n+1}\sum_{k=1}^{n+1}(\tilde{a}_{l,k,j,n+1} + \tilde{a}_{l,j,n+1,k} + \tilde{a}_{l,n+1,k,j})t_jt_k - \sum_{k=1}^{n+1}(\tilde{a}_{l,k,n+1,n+1} + \tilde{a}_{l,n+1,k,n+1} + \tilde{a}_{l,n+1,n+1,k} + \tilde{b}_{l,k,n+1} + \tilde{b}_{l,n+1,k})t_k - (\tilde{a}_{l,n+1,n+1,n+1} + \tilde{b}_{l,n+1,n+1})$.

The details of the proof of Lemma 9 are described in the full paper. □

Moreover, the following lemma is shown.

Lemma 10. *Let $n = m + k$, $k = \omega(\log \lambda)$, and $N = \omega(\log \lambda)$. Suppose that $\mathcal{MC}(n, m, \mathbb{F}_q)$ is second-preimage resistant. Then, $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ achieves the security against impersonation under* active *attack when Com is statistically hiding and computationally binding.*

Proof sketch. The proof of this lemma is described in the full paper, since it is similar to that of Lemma 8 of [26]. □

Finally, combining Lemma 9 and Lemma 10 yields the following theorem.

Theorem 11. *Let $n = m + k$, $k = \omega(\log \lambda)$, and $N = \omega(\log \lambda)$. Suppose that $\mathcal{MC}(n-1, m, \mathbb{F}_q)$ is preimage resistant. Then, $(\mathtt{P}_N^{(p)}, \mathtt{V}_N^{(p)})$ achieves the security against impersonation under* active *attack when Com is statistically hiding and computationally binding.*

7 Concluding Remarks

In this section we mention some extensions and an open problem.

Extensions. The Fiat-Shamir method transforms an identification scheme into a signature scheme which is secure against chosen-message attack in the random oracle model, if the underlying identification scheme is secure against impersonation under passive attack [1, 13]. According to it, a signature scheme based on the conjectured intractability of the MC problem can be obtained from the parallel composition of our 3-pass protocol. Using the signature scheme, we can also extend our identification/signature scheme to an identity-based one in a natural way [4].

An open problem. Efficient constructions based on multivariate polynomials of degree $d \geq 4$ remain as an open problem. However, it might be difficult to construct them by using techniques similar to those of [26] or of ours. This is because, for a multivariate polynomial $f(\mathbf{x})$ of degree $d \geq 4$, the polynomial $f(\mathbf{x}+\mathbf{y}) - f(\mathbf{x}) - f(\mathbf{y})$ contains terms which are not linear either in $\mathbf{x}$ or in $\mathbf{y}$.

8 Conclusion

We proposed an efficient construction of zero-knowledge argument of knowledge for the MC problem, and showed that the MC function is useful for public-key identification as well as the MQ function. In particular the efficiency of our scheme is highly comparable to the identification schemes based on another problem including PK, SD, CLE, PP, and MQ.

Acknowledgements. We thank Taizo Shirai and Harunaga Hiwatari for their generous support, and Marc Fischlin and the anonymous reviewers for useful comments.

References

1. Abdalla, M., An, J.H., Bellare, M., Namprempre, C.: From Identification to Signatures via the Fiat-Shamir Transform: Minimizing Assumptions for Security and Forward-Security. In: Knudsen, L.R. (ed.) EUROCRYPT 2002. LNCS, vol. 2332, pp. 418–433. Springer, Heidelberg (2002)
2. Arditti, D., Berbain, C., Billet, O., Gilbert, H.: Compact FPGA Implementations of QUAD. In: Bao, F., Miller, S. (eds.) ASIACCS, pp. 347–349. ACM (2007)
3. Bardet, M., Faugère, J.-C., Salvy, B.: Complexity of Gröbner Basis Computation for Semi-regular Overdetermined Sequences over F_2 with Solutions in F_2. Research Report RR-5049, INRIA (2003)
4. Bellare, M., Namprempre, C., Neven, G.: Security Proofs for Identity-Based Identification and Signature Schemes. J. Cryptology 22(1), 1–61 (2009)
5. Berbain, C., Gilbert, H., Patarin, J.: QUAD: A Practical Stream Cipher with Provable Security. In: Vaudenay (ed.) [31], pp. 109–128

6. Bettale, L., Faugère, J.-C., Perret, L.: Security Analysis of Multivariate Polynomials for Hashing. In: Yung, M., Liu, P., Lin, D. (eds.) Inscrypt 2008. LNCS, vol. 5487, pp. 115–124. Springer, Heidelberg (2009)
7. Bettale, L., Faugère, J.-C., Perret, L.: Hybrid Approach for Solving Multivariate Systems over Finite Fields. Journal of Mathematical Cryptology 3(3), 177–197 (2009)
8. Bouillaguet, C., Chen, H.-C., Cheng, C.-M., Chou, T., Niederhagen, R., Shamir, A., Yang, B.-Y.: Fast Exhaustive Search for Polynomial Systems in F_2. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 203–218. Springer, Heidelberg (2010)
9. Cayrel, P.-L., Véron, P., El Yousfi Alaoui, S.M.: A Zero-Knowledge Identification Scheme Based on the q-ary Syndrome Decoding Problem. In: Biryukov, A., Gong, G., Stinson, D.R. (eds.) SAC 2010. LNCS, vol. 6544, pp. 171–186. Springer, Heidelberg (2011)
10. Ding, J., Yang, B.-Y.: Multivariates Polynomials for Hashing. In: Pei, D., Yung, M., Lin, D., Wu, C. (eds.) Inscrypt 2007. LNCS, vol. 4990, pp. 358–371. Springer, Heidelberg (2008)
11. Faugère, J.C.: A New Efficient Algorithm for Computing Gröbner Bases without Reduction to Zero (F5). In: Proceedings of the 2002 International Symposium on Symbolic and Algebraic Computation, ISSAC 2002, pp. 75–83. ACM, New York (2002)
12. Faugère, J.-C., Perret, L.: Cryptanalysis of 2R$^-$ Schemes. In: Dwork, C. (ed.) CRYPTO 2006. LNCS, vol. 4117, pp. 357–372. Springer, Heidelberg (2006)
13. Fiat, A., Shamir, A.: How to Prove Yourself: Practical Solutions to Identification and Signature Problems. In: Odlyzko, A.M. (ed.) CRYPTO 1986. LNCS, vol. 263, pp. 186–194. Springer, Heidelberg (1987)
14. Garey, M.R., Johnson, D.S.: Computers and Intractability; A Guide to the Theory of NP-Completeness. W.H. Freeman & Co., New York (1979)
15. Goldreich, O.: Foundations of Cryptography. Basic Tools, vol. I. Cambridge University Press, Cambridge (2001)
16. Han, Y., Okamoto, T., Qing, S. (eds.): ICICS 1997. LNCS, vol. 1334. Springer, Heidelberg (1997)
17. Johnson, D.S., Feige, U. (eds.): Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13. ACM (2007)
18. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar Signature Schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999)
19. Matsumoto, T., Imai, H.: Public Quadratic Polynomial-Tuples for Efficient Signature-Verification and Message-Encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
20. Pass, R., Venkitasubramaniam, M.: An Efficient Parallel Repetition Theorem for Arthur-Merlin Games. In: Johnson, Feige (eds.) [17], pp. 420–429
21. Patarin, J.: Hidden Fields Equations (HFE) and Isomorphisms of Polynomials (IP): Two New Families of Asymmetric Algorithms. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 33–48. Springer, Heidelberg (1996)
22. Patarin, J., Goubin, L.: Asymmetric Cryptography with S-Boxes. In: Han, et al. (eds.) [16], pp. 369–380
23. Patarin, J., Goubin, L.: Trapdoor One-Way Permutations and Multi- variate Polynominals. In: Han, et al. (eds.) [16], pp. 356–368

24. Pointcheval, D.: A New Identification Scheme Based on the Perceptrons Problem. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 319–328. Springer, Heidelberg (1995)
25. Pointcheval, D., Poupard, G.: A New NP-Complete Problem and Public-key Identification. Des. Codes Cryptography 28(1), 5–31 (2003)
26. Sakumoto, K., Shirai, T., Hiwatari, H.: Public-Key Identification Schemes Based on Multivariate Quadratic Polynomials. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 706–723. Springer, Heidelberg (2011)
27. Shamir, A.: An Efficient Identification Scheme Based on Permuted Kernels (Extended Abstract). In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 606–609. Springer, Heidelberg (1990)
28. Stern, J.: A New Identification Scheme Based on Syndrome Decoding. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 13–21. Springer, Heidelberg (1994)
29. Stern, J.: Designing Identification Schemes with Keys of Short Size. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 164–173. Springer, Heidelberg (1994)
30. Stern, J.: A New Paradigm for Public Key Identification. IEEE Transactions on Information Theory, 13–21 (1996)
31. Vaudenay, S. (ed.): EUROCRYPT 2006. LNCS, vol. 4004. Springer, Heidelberg (2006)
32. Ye, D.-F., Lam, K.-Y., Dai, Z.-D.: Cryptanalysis of "2R" Schemes. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 315–325. Springer, Heidelberg (1999)