Advances in Mathematics of Communications doi:10.3934/amc.2021016

# DELEGATING SIGNING RIGHTS IN A MULTIVARIATE PROXY SIGNATURE SCHEME

Sumit Kumar Debnath

Department of Mathematics
National Institute of Technology Jamshedpur, Jamshedpur-831014, India

Tanmay Choudhury

Department of Mathematics
National Institute of Technology Jamshedpur, Jamshedpur-831014, India

Pantelimon Stănică

Department of Applied Mathematics
Naval Postgraduate School, Monterey, CA 93943, USA

Kunal Dey

Department of Mathematics
National Institute of Technology Jamshedpur, Jamshedpur-831014, India

Nibedita Kundu

Department of Mathematics
The LNM Institute of Information Technology, Jaipur-302031, India

(Communicated by Subhamoy Maitra)

Abstract. In the context of digital signatures, the proxy signature holds a significant role of enabling an original signer to delegate its signing ability to another party (i.e., proxy signer). It has significant practical applications. Particularly it is useful in distributed systems, where delegation of authentication rights is quite common. For example, key sharing protocol, grid computing, and mobile communications. Currently, a large portion of existing proxy signature schemes are based on the hardness of problems like integer factoring, discrete logarithms, and/or elliptic curve discrete logarithms. However, with the rising of quantum computers, the problem of prime factorization and discrete logarithm will be solvable in polynomial-time, due to Shor's algorithm, which dilutes the security features of existing ElGamal, RSA, ECC, and the proxy signature schemes based on these problems. As a consequence, construction of secure and efficient post-quantum proxy signature becomes necessary. In this work, we develop a post-quantum proxy signature scheme **Mult-proxy**, relying on multivariate public key cryptography (MPKC), which is one of the most promising candidates of post-quantum cryptography. We employ a 5-pass identification protocol to design our proxy signature scheme. Our work attains the usual proxy criterion and a one-more-unforgeability criterion under the hardness of the Multivariate Quadratic polynomial (MQ) problem. It produces optimal size proxy signatures and optimal size proxy shares in the field of MPKC.

2020 *Mathematics Subject Classification:* Primary: 94A60, 94A62, 68M12; Secondary: 68P30.

*Key words and phrases:* Multivariate public key cryptography, post-quantum cryptography, proxy signature, provable secure proxy signature, security.

The first author is supported by DRDO, India (ERIP/ER/202005001/M/01/1775).

* Corresponding author: nknkundu@gmail.com.

## 1. Introduction

A digital signature is one of the widely used cryptographic tool, where each user generates his own public key-secret key pair and the user is uniquely identified by his public key. A proxy signature scheme enables a party, namely designator (original signer) to delegate the power of signing messages to another party (namely proxy signer) on its behalf. Requirement of proxy signature arises in case of temporal absence or lack of time or lack of computational power of original signer. Due to the delegation of signing power to the proxy signer, one would be able to derive a proxy signature, which is verifiable by anyone with access to the original signer's authorized public key. In 1996, Mambo et al. [14, 15] introduced the idea of a proxy signature. A proxy signature scheme involves original signer, proxy signer, and verifier throughout the whole protocol. A proxy share, generated by the original signer using his signing key, is sent to the proxy signer in order to delegate the signing capability. In the following, the proxy signer may utilize that proxy share for signing messages on behalf of the original signer. The verification of a proxy signature is modified in such a way that the verifier can agree on involvement of the original signer in the generation of proxy signature by the proxy signer. In a proxy signature scheme, delegation type may be broadly categorized into the following: full delegation, partial delegation, and delegation by warrant. According to [4], for each of these delegation types, several number of proxy signature schemes have been proposed. Proxy signature captures a wide variety of applications, such as distributed shared object systems, electronic commerce, electronic cash, to name just a few [8, 23]. Further, the basic proxy signature has been modified to possess various features, for instance, anonymous proxy signatures [10], blind proxy signatures [1], threshold proxy signatures [25], etc. As the study on proxy signature expanded, its covetable security properties have also been evolved from the first introduction. The widely accepted and required security properties of an ideal proxy signature are listed below:

- ***Unforgeability***: On behalf of the original signer, only an authorized proxy signer is able to generate a valid proxy signature. None of the original signer and any third party would be able to generate a valid proxy signature.
- ***Identifiability:*** Anyone with access to the proxy signature should be able to identify the corresponding proxy signer.
- ***Undeniability:*** Once a valid proxy signature is generated by a proxy signer on behalf of the original signer, it is not possible for the proxy signer to deny the generation of the signature.
- ***Verifiability:*** It is possible to convince the verifier about the signers' agreement from the proxy signature.
- ***Distinguishability:*** A proxy signature can be distinguished from the normal signature by everyone.
- ***Secrecy:*** It is not possible to extract the original signer's private key from any information, like the proxy signature, proxy share, etc.
- ***Prevention of misuse:*** The proxy signer is not allowed to use the proxy share for other purposes than it is intended. In other words, he will not be able to sign a message with the proxy key that is undefined in the warrant. If he does so, it will be possible to explicitly identify him from the warrant.
- ***Revocability:*** Once the time period $t$ in the warrant $w$ is expired, the designated signing privilege is revoked automatically. Moreover, the original signer

can also broadcast a signed message to revoke the warrant $w$. Then the proxy signature created by that particular proxy signer will become invalid.

A proxy signature protocol is a tuple of algorithms: Setup, Proxy Share Generation, Proxy Share Verification, Proxy Signature Generation and Proxy Signature Verification. It can be instantiated with other special signatures to get new types of proxy signatures. Almost all of the existing proxy signature schemes are based on the assumption that the problems of integer factoring, discrete logarithm, elliptic curve discrete logarithm are difficult. However, with the development of quantum computers, the problem of prime factorization and discrete logarithm will be solvable in polynomial-time due to Shor's algorithm [21]. This algorithm dilutes the security features of existing ElGamal, RSA, ECC, and the proxy signature schemes based on these problems. The post-quantum cryptography (PQC) [2] has received cryptographers' considerable attention for resisting attacks of quantum computers. Apart from hash-based cryptography, code-based cryptography, lattice-based cryptography, and isogeny-based cryptography, multivariate cryptography is one of the main candidates for PQC. Multivariate cryptographic constructions are very fast in computation as they use simple mathematical operations such as addition and multiplications over finite fields. Moreover, multivariate cryptographic constructions require only modest computational resources. These useful characteristics of multivariate cryptography can be utilized to build low-cost devices such as smart cards, RFID chips [5, 3], just to name a few applications.

In multivariate public key cryptography (MPKC), a system of multivariate polynomials of the following form behaves as public key.

$$\begin{aligned}
p^{(1)}(x_1,\ldots,x_\eta) &= \sum_{1\le i\le j\le \eta} \alpha_{ij}^{(1)} x_i x_j + \sum_{1\le i\le \eta} \beta_i^{(1)} x_i + \gamma^{(1)} \\
p^{(2)}(x_1,\ldots,x_\eta) &= \sum_{1\le i\le j\le \eta} \alpha_{ij}^{(2)} x_i x_j + \sum_{1\le i\le \eta} \beta_i^{(2)} x_i + \gamma^{(2)} \\
&\vdots \\
p^{(\lambda)}(x_1,\ldots,x_\eta) &= \sum_{1\le i\le j\le \eta} \alpha_{ij}^{(\lambda)} x_i x_j + \sum_{1\le i\le \eta} \beta_i^{(\lambda)} x_i + \gamma^{(\lambda)}.
\end{aligned}$$

Its security depends on the hardness of solving a system of multivariate polynomials over a finite field. The quadratic case of this hard problem is known as MQ problem. It has been shown that the MQ problem is NP-hard, even over the small binary prime field $\mathbb{F}_2$ [11, 19]. In the context of MPKC, there are several constructions of encryption and signature schemes such as MI [16], HFE [17], UOV [12], Rainbow [9], Gui [18], etc. However, there was a lack of secure and efficient multivariate proxy signature scheme, and so it became essential to develop such a scheme.

Tang and Xu [22] came up with the first multivariate proxy signature based on isomorphism of polynomials (IP) problem. Recently, Chen et al. [6] constructed an MPKC based proxy signature utilizing isomorphism of polynomials and identification protocol of [20]. Apart from these, there are several constructions of post-quantum proxy signatures [26, 24, 13] based on other candidates of PQC.

### 1.1. Our contribution.

Our aim in this work is to build a multivariate proxy signature scheme by utilizing the 5-pass identification protocol of [20] and employing a multivariate digital signature scheme. We use the technique of [7] for transforming the identification protocol of Sakumoto et al. [20] to a digital signature scheme.

In this transformation, the identification protocol is coupled with a multivariate signature scheme. Our proxy signature scheme attains the security properties like strong unforgeability, strong identifiability, strong undeniability, distinguishability, secrecy, revocability, etc., under the hardness of the MQ problem. For simplicity, the underlying multivariate signature scheme in our construction is considered as a Rainbow signature [9]. However, any MQ problem based multivariate signature scheme can be instantiated with our design. Let us consider $m$ multivariate quadratic equations in $n$ variables as our central map for Rainbow and $p$ as the length of maximum possible underlying field element size in bits. Then the public key sizes of the original signer (O.S.) and of the proxy signer (P.S.) are $\frac{(mn^2+3mn+2m)p}{2}$ bits and $\frac{(mn^2+3mn+2m)p}{2}$ bits respectively. Moreover, proxy share and proxy signature sizes are $2np$ bits and $2k\omega + (k(m+2n)+n)p$ bits respectively. Here, $\omega$ represents the size of commitment scheme in bits and $k$ stands for the round of underlying identification scheme. Our scheme **Mult-proxy** performs better than Tang and Xu's scheme [22] in terms of sizes of O.S's public key, P.S's public key, proxy share and proxy share and the same as that of Proxy Rainbow [6] in terms of sizes of proxy share and proxy signature. Moreover, our scheme requires much less computational power in terms of proxy share generation and warrant publication than both schemes of [22, 6].

1.2. Organization. This paper is organized as follows. Section 2 contains the preliminaries. Then our proposed proxy signature **Mult-proxy** scheme is presented in Section 3. In Section 4, we analyze the security of our scheme. A comparison with existing multivariate proxy signature schemes is presented in Section 5 followed by the conclusions in Section 6.

## 2. Preliminaries

We first give some basic notations. In this work, $\kappa$ represents a "security parameter", $x \in_R S$ stands for "$x$ is chosen uniformly at random from a set $S$", the finite field of order $q$ is denoted by $\mathbb{F}_q$, an $\eta$ degree extension field of $\mathbb{F}_q$ is represented by $\mathbb{F}_{q^\eta}$ and $\mathbb{F}_q^\eta = (\mathbb{F}_q)^\eta = \{\mathbf{x} = (x_1, \ldots, x_\eta) | x_i \in \mathbb{F}_q \text{ for } i = 1, \ldots, \eta\}$ is a vector space.

2.1. Hardness assumption. **MQ Problem** [17]**:** Given a system of multivariate polynomials $\{p^{(1)}(x_1, \ldots, x_\eta), \ldots, p^{(\lambda)}(x_1, \ldots, x_\eta)\}$ of degree 2 over $\mathbb{F}_q$, find a solution $\mathbf{x} = (x_1, \ldots, x_\eta)$ of the system of equations $p^{(1)}(\mathbf{x}) = \cdots = p^{(\lambda)}(\mathbf{x}) = 0$.

It is known that the problem is NP-hard even for polynomials of degree 2 over $\mathbb{F}_2$ [11]. Furthermore, solving random instances with $\lambda \approx \eta$ of the MQ problem is hard [11].

2.2. Multivariate signature. An MPKC signature scheme is a tuple of algorithms – Key Gen, Sign Gen and Sign Ver which are described below. The communication flow of a multivariate signature scheme is depicted in Figure 1.

1. Key Gen : Choose a quadratic map $\mathcal{F} : (\mathbb{F}_q)^n \to (\mathbb{F}_q)^m$ (central map) whose pre-image can be easily computed. Note that $\mathcal{F}$ is a system of $m = m(\kappa)$ multivariate quadratic polynomial in $n = n(\kappa)$ variables, $\kappa$ being the security parameter. In general, $n \geq m$ for a signature scheme. To hide the structure of $\mathcal{F}$ in the public key, we choose invertible affine transformations $\mathcal{S} : (\mathbb{F}_q)^m \to (\mathbb{F}_q)^m$ and $\mathcal{T} : (\mathbb{F}_q)^n \to (\mathbb{F}_q)^n$. The signing key and the verification key are set as $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ and $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$, respectively.

2. Sign Gen : To sign a message $\mathbf{x} \in (\mathbb{F}_q)^m$, the signer recursively computes $\mathbf{y} = \mathcal{S}^{-1}(\mathbf{x}) \in (\mathbb{F}_q)^m$, $\mathbf{z} = \mathcal{F}^{-1}(\mathbf{y}) \in (\mathbb{F}_q)^n$ and $\mathbf{w} = \mathcal{T}^{-1}(\mathbf{z}) \in (\mathbb{F}_q)^n$. Finally, it outputs $\mathbf{w}$ as the signature of $\mathbf{x}$. Note that $\mathcal{F}^{-1}(\mathbf{y})$ denotes one pre-image (of possibly many) of $\mathbf{y}$ under $\mathcal{F}$.
3. Sign Ver : The verifier accepts a message-signature pair $(\mathbf{x}, \mathbf{w})$ if the equality $\mathbf{x} \stackrel{?}{=} \mathcal{P}(\mathbf{w})$ holds, otherwise rejects.

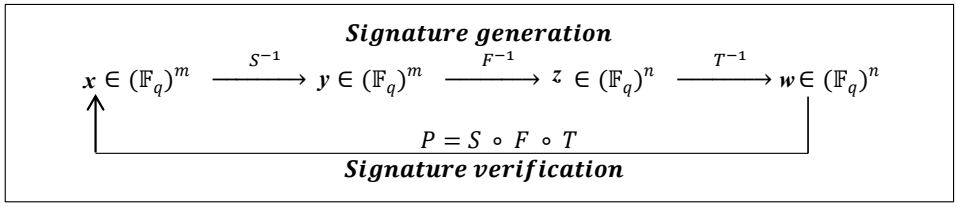


FIGURE 1. Communication flow in signature scheme.

2.3. THE RAINBOW SIGNATURE SCHEME. The Rainbow signature scheme proposed by Ding et al. [9] is one of the most promising and best studied MPKC signature scheme (Subsection 2.2) and can be recognized as a multilayered UOV [12] signature scheme. By some modification in UOV signature scheme Ding and Schmidt had reduced key and signature sizes of UOV while improving its execution. The scheme can be briefly described as below:

Let $\mathbb{L} = \mathbb{F}_q$ be a finite field with cardinality $q$, $n \in \mathbb{N}$ and $v_1 < v_2 < \ldots < v_t < v_{t+1} = n$ be a sequence of natural numbers. Set $m = n - v_1$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$, used for oil variables' indexation and $V_i = \{1, \ldots, v_i\}$, used for vinegar variables' indexation $(i = 1, \ldots, t)$.

- *Key Generation*: The Rainbow signature scheme's central map $\mathcal{W}(\mathbf{x}) = (g^{(v_1+1)}(\mathbf{x}), \ldots, g^{(n)}(\mathbf{x})) : \mathbb{L}^n \to \mathbb{L}^m$ consists of $m = n - v_1$ quadratic polynomial $g^{(i)}(\mathbf{x})$ $(i = v_1 + 1, \ldots, n)$ of the form

$$g^{(l)}(\mathbf{x}) = \sum_{i,j \in V_k} \alpha_{ij}^{(l)} \cdot x_i \cdot x_j + \sum_{i \in V_k, l \in O_k} \beta_{ij}^{(l)} \cdot x_i \cdot x_j + \sum_{i \in V_k \cup O_k} \gamma_i^{(l)} \cdot x_i + \delta^{(l)}, \tag{1}$$

  where $\mathbf{x} = (x_1, \ldots, x_n)$ and the coefficients of the polynomials are randomly chosen from $\mathbb{L}$. Here, $k \in \{1, \ldots, t\}$ is the only integer such that $l \in O_k$. To hide the underlying structure of $\mathcal{W}$ in the *public key*, it is composed with two invertible affine transformations $\mathcal{A} : \mathbb{L}^m \to \mathbb{L}^m$ and $\mathcal{B} : \mathbb{L}^n \to \mathbb{L}^n$ to form the *public key* $\mathcal{Q} = \mathcal{A} \circ \mathcal{W} \circ \mathcal{B} : \mathbb{L}^n \to \mathbb{L}^m$.
- *Signature Generation*: To generate a signature for a message $\mathbf{w} \in \mathbb{L}^m$, we compute successively $\mathbf{y} = \mathcal{A}^{(-1)}(\mathbf{w}) \in \mathbb{L}^m$, $\mathbf{x} = \mathcal{W}^{(-1)}(\mathbf{y}) \in \mathbb{L}^n$ and $\mathbf{z} = \mathcal{B}^{(-1)}(\mathbf{x}) \in \mathbb{L}^n$. Here, $\mathcal{W}^{(-1)}(\mathbf{y})$ means finding one of the (approximately $p^{v_1}$) pre-images of $\mathbf{y}$ under the central map $\mathcal{G}$, which is achieved inverting recursively each single UOV layer as described in the following steps:
  1: The first layer vinegar variables $x_1, \ldots, x_{v_1}$ are assigned random values from $\mathbb{L}$ and then substituted into the multivariate quadratic polynomials $g^{(l)}$ $(l = v_1 + 1, \ldots, n)$ to obtain a reduced system of multivariate polynomial equations.
  2: Set $k = 1$.

3: If $k \leq t$, then we will perform the following steps.
4: Use Gaussian elimination on the polynomials $g^{(l)}$ $(l \in O_k)$ to obtain the values of the variables $x_l$ $(l \in O_k)$.
5: Replace the values of $x_l$ $(l \in O_k)$ obtained in the previous step into the polynomials $g^{(l)}$ $(l \in \{v_k + 1, \ldots, n\})$.
6: Increase the value of $k$ by 1 and go to step 3.

It may happen that the linear systems in consideration in step 4 does not have a solution. In this case we can choose other values for $x_1, \ldots, x_{v_1}$ in step 1 and start again. Thus, the signature of the document $\mathbf{w}$ is $\mathbf{z} \in \mathbb{L}^n$.

- *Signature Verification*: To check whether a signature $\mathbf{z} \in \mathbb{L}^n$ is valid, one simply checks whether $\mathbf{w} \stackrel{?}{=} \mathcal{Q}(\mathbf{z}) \in \mathbb{L}^m$. If it holds, the signature is accepted, otherwise it is invalid.

| Prover$(\mathcal{R}, \mathbf{w}, \mathbf{x})$ | | Verifier$(\mathcal{R}, \mathbf{w})$ |
|---|---|---|
| $\mathbf{s}_0, \mathbf{t}_0 \in_R (\mathbb{F}_q)^n, \mathbf{g}_0 \in_R (\mathbb{F}_q)^m$ | | |
| $\mathbf{s}_1 = \mathbf{x} - \mathbf{s}_0$ | | |
| $c_0 = \mathsf{Comm}(\mathbf{s}_0, \mathbf{t}_0, \mathbf{g}_0)$ | | |
| $c_1 = \mathsf{Comm}(\mathbf{s}_1, \mathcal{G}(\mathbf{t}_0, \mathbf{s}_1) + \mathbf{g}_0)$ | $\xrightarrow{c_0,\, c_1}$ | |
| | $\xleftarrow{a}$ | $a \in_R \mathbb{F}_q$ |
| $\mathbf{t}_1 = a\mathbf{s}_0 - \mathbf{t}_0$ | | |
| $\mathbf{g}_1 = a\mathcal{R}(\mathbf{s}_0) - \mathbf{g}_0$ | $\xrightarrow{\mathbf{t}_1,\, \mathbf{g}_1}$ | |
| | $\xleftarrow{char}$ | $char \in_R \{0, 1\}$ |
| If $char = 0$, $\mathsf{Resp} = \mathbf{s}_0$ | | |
| If $char = 1$, $\mathsf{Resp} = \mathbf{s}_1$ | $\xrightarrow{\mathsf{Resp}}$ | |
| | | If $char = 0$ check |
| | | $c_0 \stackrel{?}{=} \mathsf{Comm}(\mathbf{s}_0, a\mathbf{s}_0 - \mathbf{t}_1, a\mathcal{R}(\mathbf{s}_0) - \mathbf{g}_1)$ |
| | | If $char = 1$ check |
| | | $c_1 \stackrel{?}{=} \mathsf{Comm}(\mathbf{s}_1, a(\mathbf{v} - \mathcal{R}(\mathbf{s}_1) + \mathcal{R}(\mathbf{0})) - \mathcal{G}(\mathbf{t}_1, \mathbf{s}_1) - \mathbf{g}_1)$ |

Figure 2. 5-pass identification protocol.

2.4. 5-pass identification protocol [20]. The 5-pass identification protocol of [20] is a zero-knowledge scheme between the prover Prov and the verifier Ver, where Prov has knowledge of a solution $\mathbf{x}$ of $\mathcal{R}(\mathbf{t}) = \mathbf{w}$, without revealing the solution $\mathbf{x}$ to Ver for a system $\mathcal{R} : (\mathbb{F}_q)^n \to (\mathbb{F}_q)^m$ of $m = m(\kappa)$ quadratic multivariate polynomials in $n = n(\kappa)$ variables. The bilinear polar form $\mathcal{G}$ of $\mathcal{R}$ is represented by $\mathcal{G}(\mathbf{x}, \mathbf{y}) = \mathcal{R}(\mathbf{x} + \mathbf{y}) - \mathcal{R}(\mathbf{x}) - \mathcal{R}(\mathbf{y}) + \mathcal{R}(\mathbf{0})$. The public parameters are $\mathcal{R}$ and $\mathbf{w}$, which are available to Ver. In order to prove its knowledge $\mathbf{x}$, the prover Prov interacts with Ver in five successive rounds (see Figure 2). Given the underlying commitment scheme *Comm* is computationally binding and statistically-hiding, the security of the 5-pass protocol depends on the hardness of the MQ problem. The knowledge error per round of the protocol is given by $\mathsf{ERR} = \frac{1}{2} + \frac{1}{2q}$. One has to execute $r = \left\lceil \frac{-\eta}{\log_2(\mathsf{ERR})} \right\rceil$ rounds, in order to reduce the impersonation probability lower than $2^{-\eta}$. Note that $\eta \approx 30$ may be sufficient for identification purposes. However, in the case of a signature scheme, $\eta$ has to be at least as large as the security level.

2.5. General construction of proxy signature scheme. A proxy signature scheme is comprised of five algorithms Setup, Proxy Share Generation, Proxy Share Verification, Proxy Sign, Proxy Signature Verification, which run as follows.

1. Setup $(1^\kappa)$ : On input of the security parameter $\kappa$, the original signer A and a proxy signer B generate public key-secret key pair $(\mathsf{pk_A}, \mathsf{sk_A})$ and $(\mathsf{pk_B}, \mathsf{sk_B})$ respectively. The corresponding public keys and information related to the signature scheme are published on the bulletin board.
2. Proxy Share Generation $(\mathsf{sk_A}, \mathsf{d_B})$ : On input $\mathsf{sk_A}$ and attribute $\mathsf{d_B}$ related to identity and other necessary information of a particular proxy signer B, the original signer A runs Proxy Share Generation to generate a proxy share or warrant $\mathsf{z_B}$, which is delivered to B and published on the bulletin board. The warrant $\mathsf{z_B}$ enables A to delegate his/her signing authority to B.
3. Proxy Share Verification $(\mathsf{d_B}, \mathsf{z_B}, \mathsf{pk_A})$ : On input $(\mathsf{d_B}, \mathsf{z_B}, \mathsf{pk_A})$, the proxy signer B runs the algorithm Proxy Share Verification to check whether the warrant is valid or not. If $\mathsf{w_B}$ is valid corresponding to $\mathsf{z_B}$, then Proxy Share Verification outputs 1 and the proxy signer B get assured of its signing authority on behalf of the original signer A; otherwise, it outputs 0.
4. Proxy Sign $(\mathsf{m}, \mathsf{sk_B}, \mathsf{z_B})$ : On input of a message m, $\mathsf{sk_B}$, and $\mathsf{z_B}$, B runs the algorithm Proxy Sign to generate a proxy signature $\sigma$ for m.
5. Proxy Signature Verification $(\mathsf{m}, \sigma, \mathsf{pk_B}, \mathsf{d_B}, \mathsf{z_B})$ : On input $(\mathsf{m}, \sigma, \mathsf{pk_B}, \mathsf{d_B}, \mathsf{z_B})$, the verifier runs Proxy Signature Verify to check the correctness of the message-proxy signature pair m, $\sigma$. For a valid pair with respect to $\mathsf{d_B}$, $\mathsf{z_B}$, Proxy Signature Verification outputs 1; otherwise, it outputs 0.

2.6. Existential unforgeability under chosen-message attack (uf-cma). The notion of *uf-cma* can be considered as the security notion for a proxy signature. It is defined by a "game", or an experiment, run between a forger FRG and a challenger CHL for a particular delegated signer with attribute $d$, which, once chosen it is fixed. Let us consider a proxy signature that consists of the algorithms Setup, Proxy Share Generation, Proxy Share Verification, Proxy Sign, Proxy Signature Verification. The experiment $\mathsf{Ex}^{\text{uf-cma}}_{proxy(1^\kappa)}$ is described below:

- The CHL runs Setup $(1^\kappa)$ to generate public key-secret key pairs (pk, sk) for original signer and $(\mathsf{pk_D}, \mathsf{sk_D})$ for proxy signer.
- The FRG adaptively queries the following to CHL:
  - Sign-query : For a message msg, the FRG can ask the CHL for the corresponding proxy signature associated with the same proxy signer with attribute $d$. In order to give a response, the CHL first recognizes the corresponding warrant $(d, z)$ from the bulletin board and runs Proxy Sign on inputs of the message msg, the warrant and the secret key $sk_D$ to get a signature $\sigma = Sign(\mathsf{msg})$, which is sent to the FRG.
- The FRG outputs a message-signature pair $(\mathsf{msg}^*, \sigma^*)$ for the delegate $D$ with attribute $d$. The FRG will win the game, i.e., output of the experiment $\mathsf{Ex}^{\text{uf-cma}}_{proxy(1^\kappa)}$ will be 1, if Proxy Signature Verification$(\mathsf{msg}^*, \sigma^*) = 1$ and FRG has not made any Sign-query on $(\mathsf{msg}^*, d)$. We denote the advantage or the success probability of FRG by $\mathsf{Adv}^{\mathsf{Ex}^{\text{uf-cma}}_{proxy(1^\kappa)}}_{\mathsf{FRG}}$, which is defined as $\mathsf{Adv}^{\mathsf{Ex}^{\text{uf-cma}}_{proxy(1^\kappa)}}_{\mathsf{FRG}}$ $= \mathsf{Prob}[\mathsf{Ex}^{\text{uf-cma}}_{proxy(1^\kappa)} = 1] = \mathsf{Prob}[\mathsf{FRG}\ wins\ the\ game]$.

**Definition 2.1.** An proxy signature is said to be uf-cma secure if $\mathsf{Adv}_{\mathsf{FRG}}^{\mathsf{Ex}^{uf-cma}_{proxy(1^\kappa)}}$ is negligible in the security parameter $\kappa$ for any probabilistic polynomial time (PPT) forger FRG who is allowed to make at most $Q_s$ number of $\mathsf{Sign-query}$.

## 3. Proposed proxy signature scheme (mult-proxy)

This section describes the design of the proposed scheme **Mult-proxy**. We use the 5-pass identification protocol of Sakumoto et al. [20] and a secure MQ [17] based signature scheme (say Rainbow [9]) as the building blocks of **Mult-proxy**. In order to convert the 5-pass identification protocol into a signature scheme, the technique of Hülsing et al. [7] is utilized. The **Mult-proxy** is a sequence of five steps: (1) Setup, (2) Proxy Share Generation, (3) Proxy Share Verification, (4) Proxy Sign and (5) Proxy Signature Verification. The universal public key of our protocol consists of two multivariate quadratic systems $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ and $\mathcal{R} = \mathcal{M} \circ \mathcal{E} \circ \mathcal{N} : \mathbb{F}_q^n \to \mathbb{F}_q^m$, where $\mathcal{P}$ and $\mathcal{R}$ are Rainbow public keys of the original signer and a proxy signer, respectively. The private keys of the original signer and the proxy signer allow them to invert the corresponding system, as discussed in the Section 2.3. Furthermore, the bilinear polar form $\mathcal{G}(\mathbf{u}, \mathbf{v})$ of the quadratic system $\mathcal{R}$ is $\mathcal{G}(\mathbf{u}, \mathbf{v}) = \mathcal{R}(\mathbf{u} + \mathbf{v}) - \mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{v}) + \mathcal{R}(\mathbf{0}) = \mathcal{R}(\mathbf{u} + \mathbf{v}) - \mathcal{R}(\mathbf{u}) - \mathcal{R}(\mathbf{v})$ as $\mathcal{R}(\mathbf{0}) = \mathbf{0}$ for the Rainbow public key. In order to obtain a proxy signature for a message $msg$ by the delegated signer, we proceed with a computationally binding and statistically hiding commitment scheme $Comm$. We describe below the five steps of our signature scheme in detail. A high level overview of **Mult-proxy** is depicted in Figure 3.

**Protocol 1. Mult-proxy**

- Setup $(1^\kappa)$ : On input $1^\kappa$, the original signer generates a secret key-public key pair $(\mathcal{S}, \mathcal{F}, \mathcal{T})$ and $\mathcal{P}(= \mathcal{S} \circ \mathcal{F} \circ \mathcal{T})$ for the underlying Rainbow signature scheme. A proxy signer generates a secret key-public key pair $(\mathcal{M}, \mathcal{E}, \mathcal{N})$ and $\mathcal{R}(= \mathcal{M} \circ \mathcal{E} \circ \mathcal{N})$ for the underlying Rainbow signature scheme. The corresponding public keys $\mathcal{P}$ and $\mathcal{R}$ and the information related to the signature scheme are published on the bulletin board.
- Proxy Share Generation $(\mathcal{S}, \mathcal{F}, \mathcal{T}, d)$ : To generate a proxy share for a proxy signer, the original signer computes $\mathbf{w} = \mathcal{H}_1(\mathcal{R}||d) \in \mathbb{F}_q^m$, using some publicly known collision free hash function $\mathcal{H}_1$. Here, $\mathcal{R}$ and $d$ are, respectively, the public key and the attribute (consisting identity and information that includes name, d.o.b, period of proxy validity, etc.) of the delegated signer, extracted from predesigned format published on the bulletin board. The original signer further uses his private key to compute a proxy share $\mathbf{z} = \mathcal{P}^{-1}(\mathbf{w}) = \mathcal{T}^{-1} \circ \mathcal{F}^{-1} \circ \mathcal{S}^{-1}(\mathbf{w})$ for that particular designated signer. The original signer then publishes $(d,\ \mathbf{z})$ on the bulletin board, which works as warrant as well as proxy share for the proxy signer. It enables the original signer to delegate his signing authority to that designated proxy signer.
- Proxy Share Verification $(\mathsf{d}, \mathbf{z}, \mathcal{P})$ : The proxy signer finds $\mathbf{w} = \mathcal{H}_1(\mathcal{R}||d)$ and checks whether $\mathcal{P}(\mathbf{z}) = \mathbf{w}$. If equality holds, he accepts it as a valid proxy share for further proxy signature generation. Otherwise, he rejects it and requests the original signer for a valid one, or abort.
- Proxy Sign $(msg,\ \mathcal{M}, \mathcal{E}, \mathcal{N},\ \mathbf{z})$ : On receiving a valid proxy share from the bulletin board broadcasted by the original signer, the proxy signer further uses his private key to compute a knowledge $\mathbf{x} = \mathcal{R}^{-1}(\mathbf{w}) = \mathcal{N}^{-1} \circ \mathcal{E}^{-1} \circ \mathcal{M}^{-1}(\mathbf{w})$ of a solution of $\mathcal{R}(\mathbf{t}) = \mathbf{w}$. To generate a signature for a message $msg$, the designated signer (proxy signer) uses his knowledge $\mathbf{x}$ of a solution to the verifier in a zero knowledge way for the system $\mathcal{R}(\mathbf{t}) = \mathbf{w}$, which is a system of $m$ quadratic equations in $n$ variables. In particular, the delegated signer performs the following steps:

1. Uses the hash function $\mathcal{H}_1$ to compute $\mathcal{C} = \mathcal{H}_1(\mathcal{R}||msg)$.
2. Randomly chooses $\mathbf{s}_{0,1}, \ldots, \mathbf{s}_{0,k}, \mathbf{t}_{0,1}, \ldots, \mathbf{t}_{0,k} \in_R \mathbb{F}_q^n$, $\mathbf{g}_{0,1}, \ldots, \mathbf{g}_{0,k} \in_R \mathbb{F}_q^m$.
3. For $i = 1, \ldots, k$, sets $\mathbf{s}_{1,i} = \mathbf{x} - \mathbf{s}_{0,i}$ and computes the commitments
   - $c_{0,i} = Comm(\mathbf{s}_{0,i}, \mathbf{t}_{0,i}, \mathbf{g}_{0,i})$
   - $c_{1,i} = Comm(\mathbf{s}_{1,i}, \mathcal{G}(\mathbf{t}_{0,i}, \mathbf{s}_{1,i}) + \mathbf{g}_{0,i})$ $(i = 1, \ldots, k)$.
4. Sets $COM = (c_{0,1}, c_{1,1}, c_{0,2}, c_{1,2}, \ldots, c_{0,k}, c_{1,k})$.
5. Derives challenge $\mathcal{D} = \beta_1\beta_2\cdots\beta_k = \mathcal{H}_2(\mathcal{C}||COM) \in \mathbb{F}_q^k$, where $\mathcal{H}_2$ is a publicly known collision free hash function.
6. Computes $\mathbf{t}_{1,i} = \beta_i \cdot \mathbf{s}_{0,i} - \mathbf{t}_{0,i} \in \mathbb{F}_q^n$ and $\mathbf{g}_{1,i} = \beta_i \cdot \mathcal{R}(\mathbf{s}_{0,i}) - \mathbf{g}_{0,i}$ for $(i = 1, \ldots, k)$ and set $Rspn_1 = (\mathbf{t}_{1,1}, \mathbf{g}_{1,1}, \ldots, \mathbf{t}_{1,k}, \mathbf{g}_{1,k})$.
7. Derives challenge $\Gamma = \gamma_1\gamma_2\cdots\gamma_k = \mathcal{H}_3(\mathcal{D}||COM||Rspn_1) \in \{0,1\}^k$, where $\mathcal{H}_3$ is also a publicly known collision free hash function.
8. Sets $Rspn_2 = (\mathbf{s}_{\gamma_1,1}, \ldots, \mathbf{s}_{\gamma_k,k})$.
9. The proxy signature $\sigma$ for the message $msg$ is given by

$$\sigma = (\mathcal{C}, COM, Rspn_1, Rspn_2).$$

- Proxy Signature Verification $(msg,\ \sigma,\ \mathcal{R},\ d,\ \mathbf{z})$ : To check whether a proxy signature $\sigma$ for a message $msg$ is valid, the verifier first construct $d$ from the information available on the bulletin board for that particular proxy signer and accordingly identifies the warrant $(d,\ \mathbf{z})$ from the bulletin board. The verifier first evaluates the involvement of the original signer in the proxy signing process by finding $\mathbf{w} = \mathcal{H}_1(\mathcal{R}||d)$ and checks whether $\mathcal{P}(\mathbf{z}) \stackrel{?}{=} \mathbf{w}$. If it fails, then the verifier treat $\sigma$ as an invalid signature, otherwise it breaks $\sigma$ into its parts and finds $\mathcal{D} = \mathcal{H}_2(C||COM)$. It obtains the corresponding challenges $\beta_i$ from $\mathcal{D}$ and $\gamma_i$ from $(\mathcal{D}, COM, Rspn_1)$ by using the publicly known hash function $\mathcal{H}_3$ for each $i \in \{1, 2, \ldots, k\}$. Further, the verifier parses $COM$ into $(c_{0,1}, c_{1,1}, \ldots, c_{0,k}, c_{1,k})$, $Rspn_1$ into $\mathbf{t}_{1,1}, \mathbf{g}_{1,1}, \ldots, \mathbf{t}_{1,k}, \mathbf{g}_{1,k}$ and $Rspn_2$ into $\mathbf{s}_{\gamma_1,1}, \mathbf{s}_{\gamma_2,2}, \ldots, \mathbf{s}_{\gamma_k,k}$. Finally, it checks if $\forall\ i = 1, \ldots, k$, whether $\mathbf{s}_{\gamma_i,i}$ is a correct response to $\gamma_i$ with respect to $COM,\ \beta_i,\ \mathbf{t}_{1,i},\ \mathbf{g}_{1,i}$, i.e.,
  - $c_{0,i} \stackrel{?}{=} Comm(\mathbf{s}_{\gamma_i,i}, \beta_i \cdot \mathbf{s}_{\gamma_i,i} - \mathbf{t}_{1,i}, \beta_i \cdot \mathcal{R}(\mathbf{s}_{\gamma_i,i}) - \mathbf{g}_{1,i})$ (for $\gamma_i = 0$);
  - $c_{1,i} \stackrel{?}{=} Comm(\mathbf{s}_{\gamma_i,i}, \beta_i \cdot (\mathbf{w} - \mathcal{R}(\mathbf{s}_{\gamma_i,i})) - \mathcal{G}(\mathbf{t}_{1,i}, \mathbf{s}_{1,i}) - \mathbf{g}_{\gamma_i,i})$ (for $\gamma_i = 1$).

3.1. CORRECTNESS. In order to prove the correctness of the verification process, we have to show that for each $i = 1, \ldots, k$, the following equalities hold:

- $c_{0,i} = Comm(\mathbf{s}_{\gamma_i,i}, \beta_i \cdot \mathbf{s}_{\gamma_i,i} - \mathbf{t}_{1,i}, \beta_i \cdot \mathcal{R}(\mathbf{s}_{\gamma_i,i}) - \mathbf{g}_{1,i})$ (for $\gamma_i = 0$)
- $c_{1,i} = Comm(\mathbf{s}_{\gamma_i,i}, \beta_i \cdot (\mathbf{w} - \mathcal{R}(\mathbf{s}_{\gamma_i,i})) - \mathcal{G}(\mathbf{t}_{1,i}, \mathbf{s}_{1,i}) - \mathbf{g}_{\gamma_i,i})$ (for $\gamma_i = 1$).

This can be shown by considering the following two cases.
Case I: Consider $\gamma_i = 0$. Then,

$$\begin{aligned} &Comm(\mathbf{s}_{\gamma_i,i}, \beta_i \cdot \mathbf{s}_{\gamma_i,i} - t_{1,i}, \beta_i \cdot \mathcal{R}(\mathbf{s}_{\gamma_i,i}) - \mathbf{g}_{1,i}) \\ &= Comm(\mathbf{s}_{0,i}, \beta_i \cdot \mathbf{s}_{0,i} - \mathbf{t}_{1,i}, \beta_i \cdot \mathcal{R}(\mathbf{s}_{0,i}) - \mathbf{g}_{1,i}) \\ &= Comm(\mathbf{s}_{0,i}, \mathbf{t}_{0,i}, \mathbf{g}_{0,i}) = c_{0,i} = c_{\gamma_i,i}. \end{aligned}$$

Case II: Consider $\gamma_i = 1$. Then,

$$\begin{aligned} s_{\gamma_i,i} &= \mathbf{s}_{1,i} \\ &\beta_i \cdot (\mathbf{w} - \mathcal{R}(\mathbf{s}_{\gamma_i,i})) - \mathcal{G}(\mathbf{t}_{1,i}, \mathbf{s}_{1,i}) - \mathbf{g}_{\gamma_i,i} \\ &= \beta_i \cdot (\mathbf{w} - \mathcal{R}(\mathbf{s}_{1,i})) - \mathcal{G}(\beta_i \cdot \mathbf{s}_{0,i} - \mathbf{t}_{0,i}, \mathbf{s}_{1,i}) - \mathbf{g}_{1,i} \\ &= \beta_i \cdot (\mathbf{w} - \mathcal{R}(\mathbf{s}_{1,i})) - \beta_i \cdot \mathcal{G}(\mathbf{s}_{0,i}, \mathbf{s}_{1,i}) + \mathcal{G}(\mathbf{t}_{0,i}, \mathbf{s}_{1,i}) - \mathbf{g}_{1,i} \\ &= \beta_i \cdot (\mathbf{w} - \mathcal{R}(\mathbf{s}_{1,i})) - \beta_i \cdot (\mathcal{R}(\mathbf{s}_{0,i} + \mathbf{s}_{1,i}) - \mathcal{R}(\mathbf{s}_{0,i}) - \mathcal{R}(\mathbf{s}_{1,i})) \\ &\quad + \mathcal{G}(\mathbf{t}_{0,i}, \mathbf{s}_{1,i}) - \beta_i \cdot \mathcal{R}(\mathbf{s}_{0,i}) + \mathbf{g}_{0,i} \end{aligned}$$

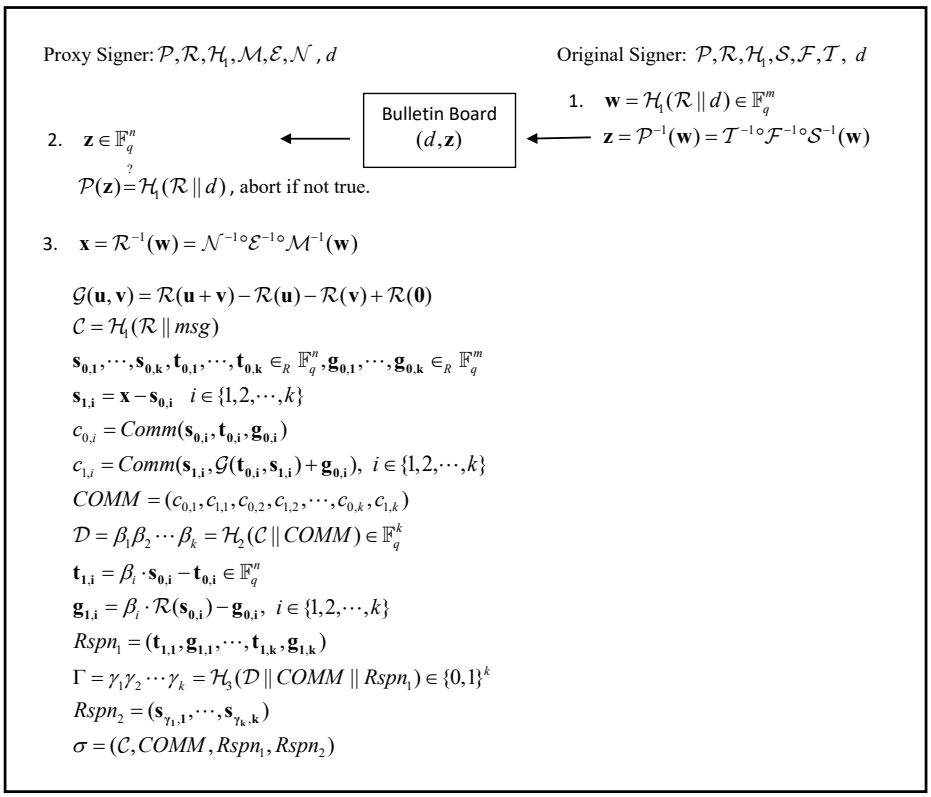


FIGURE 3. Our proxy signature protocol.

$$
\begin{aligned}
&= \beta_i(\mathbf{w}-\mathcal{R}(\mathbf{s}_{0,i}+\mathbf{s}_{1,i}))+\mathcal{G}(\mathbf{t}_{0,i},\mathbf{s}_{1,i})+\mathbf{g}_{0,i}\\
&= \beta_i(\mathbf{w}-\mathcal{R}(\mathbf{x}))+\mathcal{G}(\mathbf{t}_{0,i},\mathbf{s}_{1,i})+\mathbf{g}_{0,i}\\
&= \beta_i(\mathbf{w}-\mathbf{w})+\mathcal{G}(\mathbf{t}_{0,i},\mathbf{s}_{1,i})+\mathbf{g}_{0,i}\\
&= \mathcal{G}(\mathbf{t}_{0,i},\mathbf{s}_{1,i})+\mathbf{g}_{0,i},\\
&Comm(\mathbf{s}_{\gamma_i,i},\beta_i\cdot(\mathbf{z}-\mathcal{R}(\mathbf{s}_{\gamma_i,i}))-\mathcal{G}(\mathbf{t}_{1,i},\mathbf{s}_{1,i})-\mathbf{g}_{\gamma_i,i})\\
&= Comm(\mathbf{s}_{1,i},\mathcal{G}(\mathbf{t}_{0,i},\mathbf{s}_{1,i})+\mathbf{g}_{0,i})=c_{1,i}=c_{\gamma_i,i}.
\end{aligned}
$$

## 4. Security

Let $\mathcal{P}=\mathcal{S}\circ\mathcal{F}\circ\mathcal{T}:\mathbb{F}_q^n\to\mathbb{F}_q^m$ and $\mathcal{R}=\mathcal{M}\circ\mathcal{E}\circ\mathcal{N}:\mathbb{F}_q^n\to\mathbb{F}_q^m$ be public keys of an MQ based signature scheme (say, Rainbow). Suppose the hash functions $\mathcal{H}_1,\ \mathcal{H}_2,\ \mathcal{H}_3$ are designed as random oracles. Further, assume that the commitment scheme $Comm$ is computationally binding and statistically hiding. Then the proposed **Mult-proxy** attains unforgeability, identifiability, undeniability, verifiability, distinguishability, secrecy, prevention of misuse and revocability under the hardness of the $MQ$ problem.

- ***Unforgeability of Ordinary Signature:*** The proxy signer or any other party cannot derive the private key $(\mathcal{S},\ \mathcal{F},\ \mathcal{T})$ of the original signer from the warrant $(d,\ \mathbf{z})$ as it would directly imply breaking the Rainbow [9] signature scheme.

- ***Unforgeability of Proxy Signature:*** We now prove that the proposed proxy signature scheme possesses existential unforgeability under the chosen-message attack (uf-cma), assuming the hardness of $MQ$ problem. We show this by contradiction. If possible, let there be a forger $\mathsf{FRG}$ with non-negligible success probability in the uf-cma game for **Mult-proxy**. Then we will show that it is feasible to design an oracle machine $\mathcal{O}^{\mathsf{FRG}}$ to solve the MQ problem by using $\mathsf{FRG}$ and controlling the output of the random oracles $\mathcal{H}_1$,$\mathcal{H}_2$ and $\mathcal{H}_3$ in a series of games $\mathbf{Game}_0$, $\mathbf{Game}_1$, $\mathbf{Game}_2$, $\mathbf{Game}_3$, and $\mathbf{Game}_4$. Here $\mathbf{Game}_i$ is slightly different than $\mathbf{Game}_{i-1}$, for $1 \leq i \leq 4$. Suppose the success probability of $\mathsf{FRG}$ in the game $\mathbf{Game}_i$ is $\mathsf{Prob}[\mathbf{Game}_i]$, for $0 \leq i \leq 4$.
  - $\mathbf{Game}_0$ : $\mathbf{Game}_0$ is exactly the same as uf-cma game for **Mult-proxy**. Thus, $\mathsf{Adv}^{\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult\text{-}proxy}(1^\kappa)}}_{\mathsf{FRG}} = \mathsf{Prob}[\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult-proxy}(\mathbf{1}^\kappa)} = 1] = \mathsf{Prob}[\mathbf{Game}_0]$.
  - $\mathbf{Game}_1$ : It is similar to $\mathbf{Game}_0$ except that the oracle $\mathcal{O}^{\mathsf{FRG}}$ substitutes the $\mathcal{H}_1$ query of $\mathcal{R}||msg$ by a random $m$-tuple from $\mathbb{F}_q^m$ . Note that $|\mathsf{Prob}[\mathbf{Game}_1] - \mathsf{Prob}[\mathbf{Game}_0]|$ is non-negligible means that $\mathsf{FRG}$ distinguishes the output distributions of the random oracle $\mathcal{H}_1$, which is not possible. Thus, $|\mathsf{Prob}[\mathbf{Game}_1] - \mathsf{Prob}[\mathbf{Game}_0]| = \delta_1(\kappa)$, for some negligible function $\delta_1(\kappa)$.
  - $\mathbf{Game}_2$ : This game is the same as $\mathbf{Game}_1$ except that the oracle $\mathcal{O}^{\mathsf{FRG}}$ substitutes the output of $\mathcal{H}_2$ by a random $k$-tuple from $\mathbb{F}_q^k$. Note that $|\mathsf{Prob}[\mathbf{Game}_2] - \mathsf{Prob}[\mathbf{Game}_1]|$ is non-negligible implies that $\mathsf{FRG}$ distinguishes the output distributions of the random oracle $\mathcal{H}_2$, which is not possible. Therefore, $|\mathsf{Prob}[\mathbf{Game}_2] - \mathsf{Prob}[\mathbf{Game}_1]| = \delta_2(\kappa)$, where $\delta_2(\kappa)$ is a negligible function.
  - $\mathbf{Game}_3$ : This game is similar to $\mathbf{Game}_2$ except that the oracle $\mathcal{O}^{\mathsf{FRG}}$ replaces the output of $\mathcal{H}_3$ by a random binary bit string of length $k$. Note that $|\mathsf{Prob}[\mathbf{Game}_3] - \mathsf{Prob}[\mathbf{Game}_2]|$ is non-negligible means that $\mathsf{FRG}$ distinguishes the output distributions of the random oracle $\mathcal{H}_3$, which is not possible. Therefore, $|\mathsf{Prob}[\mathbf{Game}_3] - \mathsf{Prob}[\mathbf{Game}_2]| = \delta_3(\kappa)$ for some negligible function $\delta_3(\kappa)$.
  - $\mathbf{Game}_4$ : This game is similar to $\mathbf{Game}_3$ except during the challenge phase by $\mathsf{FRG}$ to forge a signature of a message $msg^*$, when the random oracle $\mathcal{O}^{\mathsf{FRG}}$ replaces $\mathcal{H}_1$ query of $\mathcal{R}||msg$ by a random $m$-tuple from $\mathbb{F}_q^m$, $\mathcal{H}_2$ query of $\mathcal{C}||COM$ by a random $k$-tuple from $\mathbb{F}_q^k$ and $\mathcal{H}_3$ query of $\mathcal{D}||COM||Rspn_1$ by a random $k$-bit binary string. Using the similar arguments of $\mathbf{Game}_1$, $\mathbf{Game}_2$ and $\mathbf{Game}_3$, we can assert that $|\mathsf{Prob}[\mathbf{Game}_4] - \mathsf{Prob}[\mathbf{Game}_3]| = \delta_4(\kappa)$ for some negligible function $\delta_4(\kappa)$.

  We have the following

$$
\begin{aligned}
&|\mathsf{Prob}[\mathbf{Game}_4] - \mathsf{Prob}[\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult\text{-}proxy}(\mathbf{1}^\kappa)} = 1]| \\
&= |\mathsf{Prob}[\mathbf{Game}_4] - \mathsf{Prob}[\mathbf{Game}_0]| \\
&\leq |\mathsf{Prob}[\mathbf{Game}_4] - \mathsf{Prob}[\mathbf{Game}_3]| + |\mathsf{Prob}[\mathbf{Game}_3] - \mathsf{Prob}[\mathbf{Game}_2]| \\
&+|\mathsf{Prob}[\mathbf{Game}_2] - \mathsf{Prob}[\mathbf{Game}_1]| + |\mathsf{Prob}[\mathbf{Game}_1] - \mathsf{Prob}[\mathbf{Game}_0]| \\
&= \delta_4(\kappa) + \delta_3(\kappa) + \delta_2(\kappa) + \delta_1(\kappa) = \rho(\kappa(\text{say})),
\end{aligned}
$$

  $\rho(\kappa)$ being a negligible function. Thus, the success probability $\mathsf{Prob}[\mathbf{Game}_4]$ of $\mathsf{FRG}$ in $\mathbf{Game}_4$ is same as the success probability $\mathsf{Adv}^{\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult-proxy}(1^\kappa)}}_{\mathsf{FRG}} =$

$\mathsf{Prob}[\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult-proxy}(1^\kappa)} = 1]$ of $\mathsf{FRG}$ in the uf-cma game. This ensures that $\mathsf{Adv}^{\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult-proxy}(1^\kappa)}}_{\mathsf{FRG}}$ is non-negligible if $\mathsf{Prob}[\mathbf{Game}_4]$ is so.

If possible, let $\mathsf{Prob}[\mathbf{Game}_4]$ be non-negligible. Then we show below that the oracle machine $\mathcal{O}^{\mathsf{FRG}}$ can easily solve the MQ problem by finding a solution $\mathbf{u}^*$ of the system $\mathbf{w}^* = \mathcal{R}(\mathbf{x})$ with $\mathsf{FRG}$'s help.

1. The oracle machine $\mathcal{O}^{\mathsf{FRG}}$ generates four valid transcripts $(Comm, \mathcal{D}^{(i)}, Rspn_1^{(i)}, \Gamma^{(j)}, Rspn_2^{(i,j)})_{\{i,j=0,1\}}$ with the help of $\mathsf{FRG}$ and controlling the output of random oracles $\mathcal{H}_1$, $\mathcal{H}_2$ and $\mathcal{H}_3$, where
$$\begin{aligned}
&Comm = (c_{0,1}, c_{1,1}, \ldots, c_{0,k}, c_{1,k})\\
&\Delta^{(i)} = \{\beta_1^{(i)}, \ldots, \beta_k^{(i)}\} \text{ with } \beta_l^{(0)} \neq \beta_k^{(1)} \text{ for } l = 1, \ldots, k\\
&Rspn_1^{(i)} = (\mathbf{t}_{1,1}^{(i)}, \mathbf{g}_{1,1}^{(i)}, \ldots, \mathbf{t}_{1,k}^{(i)}, \mathbf{g}_{1,k}^{(i)})\\
&\Gamma^{(j)} = (\gamma_1^{(j)}, \ldots, \gamma_k^{(j)}) \text{ with } \gamma_l^{(j)} = j \in \{0,1\} \text{ for } l = 1, \ldots, k\\
&Rspn_2^{(i,j)} = (\mathbf{s}_{j,1}^{(i)}, \ldots, \mathbf{s}_{j,k}^{(i)}).
\end{aligned}$$
2. Note that for $l \in \{1, \cdots, k\}$,
$$\begin{aligned}
c_{0,l} &= Comm\left(\mathbf{s}_{0,l}^{(0)},\ \beta_l^{(0)}\mathbf{s}_{0,l}^{(0)} - \mathbf{t}_{1,l}^{(0)},\ \beta_l^{(0)}\mathcal{R}(\mathbf{t}_{0,l}^{(0)}) - \mathbf{g}_{1,l}^{(0)}\right)\\
&= Comm\left(\mathbf{s}_{0,l}^{(1)},\ \beta_l^{(1)}\mathbf{s}_{0,l}^{(1)} - \mathbf{t}_{1,l}^{(1)},\ \beta_l^{(1)}\mathcal{R}(\mathbf{s}_{0,l}^{(1)}) - \mathbf{t}_{1,l}^{(1)}\right) \quad (2)\\
c_{1,l} &= Comm\left(\mathbf{s}_{1,l}^{(0)},\ \beta_l^{(0)}(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{0})) - \mathcal{G}(\mathbf{t}_{1,l}^{(0)}, \mathbf{s}_{1,l}^{(0)}) - \mathbf{g}_{1,l}^{(0)}\right)\\
&= Comm\left(\mathbf{s}_{1,l}^{(1)},\ \beta_l^{(1)}(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(1)}) + \mathcal{R}(\mathbf{0})) - \mathcal{G}(\mathbf{t}_{1,l}^{(1)}, \mathbf{s}_{1,l}^{(1)}) - \mathbf{g}_{1,l}^{(1)}\right). \quad (3)
\end{aligned}$$
3. Using the computationally binding property of the commitment scheme $Comm$, we argue that the arguments of $Comm$ for $c_{0,l}$ are equal in (2). Similarly, the arguments of $Comm$ for $c_{1,l}$ are equal in (3) due to the binding property of $Comm$. Thus, we have
$$\mathbf{s}_{0,l}^{(0)} = \mathbf{s}_{0,l}^{(1)} \quad (4)$$
$$\beta_l^{(0)}\mathbf{s}_{0,l}^{(0)} - \mathbf{t}_{1,l}^{(0)} = \beta_l^{(1)}\mathbf{s}_{0,l}^{(1)} - \mathbf{t}_{1,l}^{(1)} \quad (5)$$
$$\beta_l^{(0)}\mathcal{R}(\mathbf{s}_{0,l}^{(0)}) - \mathbf{g}_{1,l}^{(0)} = \beta_l^{(1)}\mathcal{R}(\mathbf{s}_{0,l}^{(1)}) - \mathbf{g}_{1,l}^{(1)} \quad (6)$$
$$\mathbf{s}_{1,l}^{(0)} = \mathbf{s}_{1,l}^{(1)} \quad (7)$$
$$\begin{aligned}
&\beta_l^{(0)}(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{0})) - \mathcal{G}(\mathbf{t}_{1,l}^{(0)}, \mathbf{s}_{1,l}^{(0)}) - \mathbf{g}_{1,l}^{(0)}\\
&= \beta_l^{(1)}(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(1)}) + \mathcal{R}(\mathbf{0})) - \mathcal{G}(\mathbf{t}_{1,l}^{(1)}, \mathbf{s}_{1,l}^{(1)}) - \mathbf{g}_{1,l}^{(1)}. \quad (8)
\end{aligned}$$
4. From Equations (7) and (8),
$$\begin{aligned}
&(\beta_l^{(0)} - \beta_l^{(1)})(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{0})) = \mathcal{G}(\mathbf{t}_{1,l}^{(0)}, \mathbf{s}_{1,l}^{(0)}) - \mathcal{G}(\mathbf{t}_{1,l}^{(1)}, \mathbf{s}_{1,l}^{(1)}) + \mathbf{g}_{1,l}^{(0)} - \mathbf{g}_{1,l}^{(1)}\\
&\Rightarrow (\beta_l^{(0)} - \beta_l^{(1)})(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{0})) = \mathcal{G}(\mathbf{t}_{1,l}^{(0)} - \mathbf{t}_{1,l}^{(1)}, \mathbf{s}_{1,l}^{(0)}) + \mathbf{t}_{1,l}^{(0)} - \mathbf{g}_{1,l}^{(1)}. \quad (9)
\end{aligned}$$
5. From Equations (4), (5), (6) and (9),
$$(\beta_l^{(0)} - \beta_l^{(1)})(\mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{0}))$$

$$= \mathcal{G}((\beta_l^{(0)} - \beta_l^{(1)})\mathbf{s}_{0,l}^{(0)}, \mathbf{s}_{1,l}^{(0)}) + (\beta_l^{(0)} - \beta_l^{(1)})\mathcal{R}(\mathbf{s}_{0,l}^{(0)})$$
$$\Rightarrow \mathbf{w}^* - \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{0}) = \mathcal{G}(\mathbf{s}_{0,l}^{(0)}, \mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{s}_{0,l}^{(0)}), \text{ since } \beta_l^{(0)} \neq \beta_l^{(1)}$$
$$\Rightarrow \mathbf{w}^* = \mathcal{R}(\mathbf{s}_{1,l}^{(0)}) + \mathcal{G}(\mathbf{s}_{0,l}^{(0)}, \mathbf{s}_{1,l}^{(0)}) + \mathcal{R}(\mathbf{s}_{0,l}^{(0)}) - \mathcal{R}(\mathbf{0})$$
$$= \mathcal{R}(\mathbf{s}_{0,l}^{(0)} + \mathbf{s}_{1,l}^{(0)}).$$

6. Hence, the oracle machine $\mathcal{O}^{\mathsf{FRG}}$ extracts a solution $\mathbf{s}_{0,l}^{(0)} + \mathbf{s}_{1,l}^{(0)}$ of $\mathbf{w}^* = \mathcal{R}(\mathbf{x})$

Thus, $\mathsf{Prob}[\mathbf{Game}_4]$ is non-negligible implies that $\mathcal{O}^{\mathsf{FRG}}$ can find a solution of the MQ problem $\mathbf{w}^* = \mathcal{R}(\mathbf{x})$, which contradicts the assumption that MQ problem is NP-hard. As a consequence, $\mathsf{Prob}[\mathbf{Game}_4]$ is negligible in the security parameter $\kappa$. This implies $\mathsf{Adv}^{\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult\text{-}proxy}(1^\kappa)}}_{\mathsf{FRG}} = \mathsf{Prob}[\mathsf{Ex}^{\text{uf-cma}}_{\mathbf{Mult\text{-}proxy}(1^\kappa)} = 1]$ is negligible. Therefore, we conclude that the proposed **Mult-proxy** is uf-cma secure.

- ***Identifiability:*** Using the warrant or proxy share $(d, \mathbf{z})$, the verifier can be convinced of the identity of the proxy signer as it has to satisfy $\mathcal{P}(\mathbf{z}) = \mathcal{H}_1(\mathcal{R}||d)$, where $d$ consists of information on identity, which is specific to the particular proxy signer and publicly accessible from the bulletin board.
- ***Undeniability:*** In a valid proxy signature, the presence of the public parameter $\mathcal{R}$ specific to a proxy signatures in the verification equation requires knowledge of the inverting $\mathcal{R}$, which is only available to that particular proxy signature. So once a proxy signature is generated, the proxy signer cannot deny its creation.
- ***Verifiability:*** The verifier aligns with the original signer's agreement on the signing process using the second component $\mathbf{z}$ of the warrant $(d,\ \mathbf{z})$. This is because it needs inversion under the public parameter $\mathcal{P}$ of the original signer, which is not possible without involvement of the original signer, as $\mathcal{P}(\mathbf{z}) = \mathcal{H}_1(\mathcal{R}||d)$ and $\mathcal{R},\ d$ are specific to the proxy signer.
- ***Distinguishability:*** The original signer's Rainbow signature would have been verified by using only the original signer's public parameter $\mathcal{P}$. On the other hand, the proxy verification revolves around the warrant and the proxy signature: the warrant revolves using the public parameter of the original signer, while the proxy signature uses the public parameter $\mathcal{R}$ for a zero knowledge proof and proxy signer's specific information $\mathbf{z}$. Thus, ordinary signatures and proxy signatures are distinguishable due to the use of different public keys in different ways.
- ***Secrecy:*** It is not possible to derive the original signer's private key from the warrant as it would directly require breaking the Rainbow [9] signature scheme. On the other hand, obtaining proxy signer's private key from proxy signatures would imply both breaking properties of a commitment scheme with very high probability and simultaneously breaking the Rainbow [9] signature scheme.
- ***Prevention of misuse:*** The proxy signer cannot use the proxy share $(d,\ \mathbf{z})$ here for other purposes. That is, he cannot proxy sign a message with different proxy shares that have not been defined in the warrant. Thereby, the generated proxy signature will be invalid then.
- ***Revocability:*** Once the time period $t$ in the warrant $(d,\ \mathbf{z})$ is over, automatically, the delegated signing rights are revoked. Apart from this, a signed

Table 1. General comparison of different key sizes of our scheme **Mult-proxy**, Tang and Xu's scheme [22] and Proxy Rainbow [6] with Rainbow [9] as central map

| Scheme | **Mult-proxy** | Tang and Xu's scheme [22] | Proxy Rainbow [6] |
|---|---|---|---|
| Delegation | Partial with warrant | Partial with warrant | Partial with warrant |
| O.S's pub-key | $\frac{mn^2+3mn+2m}{2} \cdot p$ | $(\frac{mn^2+3mn+2m}{2} + \xi) \cdot p$ | $\frac{mn^2+3mn+2m}{2} \cdot p$ |
| P.S's pub-key | $\frac{mn^2+3mn+2m}{2} \cdot p$ | $(\frac{mn^2+3mn+2m}{2} + \xi) \cdot p$ | $\frac{mn^2+3mn+2m}{2} \cdot p$ |
| O.S's sec-key | $(m^2+n^2+m+n+\xi) \cdot p$ | $(m^2+n^2+m+n) \cdot p$ | $(m^2+n^2+m+n+\xi) \cdot p$ |
| P.S's sec-key | $(m^2+n^2+m+n+\xi) \cdot p$ | $(m^2+n^2+m+n) \cdot p$ | $(m^2+n^2+m+n+\xi) \cdot p$ |
| Proxy share | $n \cdot p$ | $\frac{mn^2+2m^2+3mn+2n^2+4m+2n}{2} \cdot p$ | $\frac{mn^2+2m^2+3mn+2n^2+4m+2n}{2} \cdot p$ |
| Proxy sig | $2k \cdot \omega + (k(m+2n)+n) \cdot p$ | $k + k(m^2+n^2+m+n) \cdot p$ | $\frac{3mn^2+9mn+6m+6n}{2} \cdot p$ |

$p$ = length of one field element in $\mathbb{F}_q$ (in bits), $\xi = \sum_{l=1}^{u} o_l \left( v_l o_l + \frac{v_l(v_l+1)}{2} + v_{l+1} \right)$,
$\omega = |Commit|$ (in bits), $\sum_{l=1}^{u} o_l = m,\ v_1 + m = n,\ v_{l+1} = v_l + o_l$ for $l \in \{1, 2, \ldots, q\}$

message can be broadcasted by the original signer for announcing the invalidation of the warrant **w**. As a consequence, the proxy signature created by that particular delegate will no longer be valid.

Table 2. Numeric comparison of different key sizes of our scheme **Mult-proxy**, Tang and Xu's scheme [22] and Proxy Rainbow [6] with Rainbow [9] as central map.

| Scheme | Mult-proxy | Tang and Xu's scheme [22] | Proxy Rainbow [6] |
|---|---|---|---|
| Parameters | (256,18,12,12) | (256,18,12,12) | (256,18,12,12) |
| O.S's public key size (kB) | 177.4 | 297.9 | 177.4 |
| P.S's public key size (kB) | 177.4 | 297.9 | 177.4 |
| O.S's secret key size (kB) | 139.4 | 18.8 | 139.4 |
| P.S's secret key size(kB) | 139.4 | 18.8 | 139.4 |
| Proxy share size (kB) | 0.33 | 196.2 | 196.2 |
| Proxy signature size (kB) | 173.7 | 2424.9 | 533.1 |
| Parameters | (256,40,24,24) | (256,40,24,24) | (256,40,24,24) |
| O.S's public key size (kB) | 1501.9 | 2542.6 | 1501.9 |
| P.S's public key size (kB) | 1501.9 | 2542.6 | 1501.9 |
| O.S's secret key size (kB) | 1120.2 | 79.6 | 1120.2 |
| P.S's secret key size(kB) | 1120.2 | 79.6 | 1120.2 |
| Proxy share size (kB) | 0.7 | 1581.4 | 1581.4 |
| Proxy signature size (kB) | 290.9 | 10263.7 | 4507.7 |
| Parameters | (31,28,20,20,8) | (31,28,20,20,8) | (31,28,20,20,8) |
| O.S's public key size (kB) | 938.7 | 1935.5 | 938.7 |
| P.S's public key size (kB) | 938.7 | 1935.5 | 938.7 |
| O.S's secret key size (kB) | 1046.5 | 49.7 | 1046.5 |
| P.S's secret key size(kB) | 1046.5 | 49.7 | 1046.5 |
| Proxy share size (kB) | 0.43 | 988.4 | 988.4 |
| Proxy signature size (kB) | 206 | 6414.8 | 2817.3 |

## 5. Efficiency analysis

In this section, we compare the communication, storage requirement of our scheme with existing proxy signature schemes based on multivariate public key cryptography where the underlying signature scheme is considered as Rainbow. A comparative summary of our scheme with the existing multivariate proxy signature schemes is provided in Table 1 in terms of delegation type, original signer's

public key size (required for the warrant and proxy signature verification), proxy signer's public key size (required for proxy signature verification), original signer's secret key size, proxy signer's secret key size, proxy share size and proxy signature size in general setting. Table 2 compares the same attributes for specific parameters for each scheme. For a parameter set $(q, v_1, o_1, o_2, \ldots, o_u)$, we take the base field to be $\mathbb{F}_q$, and the central map $\mathcal{F}(\mathbf{x}) : \mathbb{F}_q^n \to \mathbb{F}_q^m$ to be an $r$-round Rainbow with parameters $(v_1, o_1, o_2, \ldots, o_u)$ of the corresponding scheme, where $\sum_{l=1}^{u} o_l = m,\ v_1 + m = n,\ v_{l+1} = v_l + o_l$ for $l \in \{1, 2, \ldots, q\}$. Here, $k$ denotes the number of rounds in the non-interactive zero knowledge proof used for the proxy signature generation in the corresponding scheme. By using the result stated in Subsection 2.4, we may choose 129 rounds for a 128-bit security level over $\mathbb{F}_{256}$. We instantiate the output lengths of the commitment scheme with SHA3-256 in our **Mult-proxy**. However, one may use a weaker commitment scheme to reduce the signature size.

## 6. Conclusion

This paper proposed a post-quantum secure multivariate proxy signature scheme utilizing the 5-pass identification protocol of [20]. Any MQ based multivariate signature scheme (say, Rainbow) can be used as the underlying signature scheme in our construction. Compared to the existing multivariate proxy signature schemes, our design performs betters in terms of public key size, proxy share size and generated proxy signature size. Furthermore, the computation for warrant publishing is less expensive in our scheme than in both of the other two schemes. In contrast to [6], our scheme achieves uf-cma security using random oracles.

## Acknowledgments

The authors express their deep appreciation to the editor for promptly handling our paper, as well as to the anonymous referee, whose thorough reading and constructive comments have greatly improved the paper. This work was supported by DRDO, India (ERIP/ER/202005001/M/01/1775).

## References

[1] A. K. Awasthi and S. Lal, Proxy blind signature scheme, *Trans. on Cryptology*, **2:1** (2005), 5–11.

[2] D. J. Bernstein, *Introduction to Post-Quantum Cryptography*, Post-Quantum Cryptography, Springer–Berlin, Heidelberg, 2009, 1–14.

[3] A. Bogdanov, T. Eisenbarth, A. Rupp and C. Wolf, Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves?, *Cryptographic Hardware and Embedded Systems*, **5154** (2008), 45–61.

[4] A. Boldyreva, A. Palacio and B. Warinschi, Secure proxy signature schemes for delegation of signing rights, *J. Cryptology*, **25** (2012), 57–115.

[5] A. I.-T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E. L.-H. Kuo, F. Y.-S. Lee and B.-Y. Yang, SSE implementation of multivariate PKCS on modern x86 CPUs, *International Workshop on Cryptographic Hardware and Embedded Systems*, (2009), 33–48.

[6] J. Chen, J. Ling, J. Ning, E. Panaousis, G. Loukas, K. Liang and J. Chen, Post quantum proxy signature scheme based on the multivariate public key cryptographic signature, *International J. Distributed Sensor Networks*, **16** (2020).

[7] M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska and P. Schwabe, From 5-pass MQ-based identification to MQ-based signatures, *Adv. Cryptology*, **10032** (2016), 135–165.

[8] J.-Zhu Dai, X.-H. Yang and J.-X. Dong, Designated-receiver proxy signature scheme for electronic commerce, *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483), IEEE*, **1** (2003), 384–389.
[9] J. Ding and D. Schmidt, Rainbow, a new multivariable polynomial signature scheme, *International Conference on Applied Cryptography and Network Security*, (2005), 164–175.
[10] G. Fuchsbauer and D. Pointcheval, Anonymous proxy signatures, *International Conference on Security and Cryptography for Networks*, (2008), 201–217.
[11] M. R. Garey and D. S. Johnson, Computers and Intractability: A guide to the theory of NP-completeness, *Freeman San Francisco*, **174** (1979).
[12] A. Kipnis, J. Patarin and L. Goubin, Unbalanced oil and vinegar signature schemes, *International Conference on the Theory and Applications of Cryptographic Techniques*, (1999), 206–222.
[13] Q. Lin, Jin Li, Z. Huang, W. Chen and J. Shen, A short linearly homomorphic proxy signature scheme, *IEEE Access* **6** (2018), 12966–12972.
[14] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures: Delegation of the power to sign messages, *IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences*, **79:9** (1996), 1338–1354.
[15] M. Mambo, K. Usuda and E. Okamoto, Proxy signatures for delegating signing operation, *Proceedings of the 3rd ACM conference on Computer and Communications Security*, (1996), 48–57.
[16] T. Matsumoto and H. Imai, Public quadratic polynomial-tuples for efficient signature-verification and message-encryption, *Workshop on the Theory and Application of Cryptographic Techniques*, (1988), 419–453.
[17] J. Patarin, Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms, *International Conference on the Theory and Applications of Cryptographic Techniques*, (1996), 33–48.
[18] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao and J. Ding, Design principles for HFEV-based multivariate signature schemes, *International Conference on the Theory and Application of Cryptology and Information Security*, (2015), 311–334.
[19] E. Sakalauskas, The multivariate quadratic power problem over ZN is NP-complete, *Information Technology and Control*, **41:1** (2012), 33–39.
[20] K. Sakumoto, T. Shirai and H. Hiwatari, Public-key identification schemes based on multivariate quadratic polynomials, *Advances in Cryptology*, **6841** (2011), 706–723.
[21] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review*, **41** (1999), 303–332.
[22] S. Tang and L. Xu, Proxy signature scheme based on isomorphisms of polynomials, in *International Conference on Network and System Security*, (2012), 113–125.
[23] G. Wang, F. Bao, J. Zhou and R. H Deng, Security analysis of some proxy signatures, *International Conference on Information Security and Cryptology*, (2003), 305–319.
[24] F. Wu, W. Yao, X. Zhang, W. Wang and Z. Zheng, Identity-based proxy signature over NTRU lattice, *International J. Communication Systems*, **32** (2019), e3867.
[25] K. Zhang, Threshold proxy signature schemes, *International Workshop on Information Security*, (1997), 282–290.
[26] H. Zhu, Y. Tan, X. Yu, Y. Xue, Q. Zhang, L. Zhu and Y. Li, An identity-based proxy signature on NTRU lattice, *Chinese J. Electronics*, **27:2** (2018), 297–303.

Received October 2020; revised February 2021.

*E-mail address*: sdebnath.math@nitjsr.ac.in
*E-mail address*: tc499180022@gmail.com
*E-mail address*: pstanica@nps.edu
*E-mail address*: kunaldey3@gmail.com
*E-mail address*: nknkundu@gmail.com