MQuBS: A Short, Round-Optimal Blind Signature with Post-Quantum Security

Dipayan Das
NTT Social Informatics Laboratories
Tokyo, Japan
dipayan.das@ntt.com

Anindya Ganguly, Angshuman Karmakar, Nitin Saxena
Department of CSE, IIT Kanpur
Kanpur, India
{anindyag,angshuman,nitin}@cse.iitk.ac.in

***Abstract*—The blind signature, proposed by Chaum [Crypto'82], allows user to obtain a signature on a message without revealing it to the signer. This ensures the anonymity of the user while maintaining its security. This cryptographic primitive is a key to privacy-preserving applications like e-cash, e-voting, and digital currencies. With the rise of digital currencies and the demand for online privacy, blind signatures have grown in importance. However, efficient blind signatures are only known from the classical number theoretic assumptions. The advent of quantum computing threatens such classical assumptions, making post-quantum (PQ) blind signatures crucial for long-term security.**

In this work, we propose MQuBS, a new short, round-optimal PQ blind signature scheme based on the multivariate assumptions. This is achieved by carefully adapting Fischlin's round-optimal blind signature framework [Crypto'07] in multivariate settings. We show that it achieves the standard one-more-unforgeability (in the random oracle model) and satisfies the blindness property. Additionally, MQuBS has the smallest signature size among all post-quantum blind signatures. For instance, at the 128-bit security level, the scheme by Agrawal *et al.* [ACM CCS-22] produces a 45KB signature, and the construction by Beullens *et al.* [ACM CCS-23] offers a 22KB signature. In contrast, MQuBS achieves a significantly smaller signature size of just 5KB.

***Index Terms*—Anonymity, Blind signature, Post-quantum cryptography, multivariate cryptography, UOV-signature**

1. Introduction

Blind signatures. Blind signature (BS) [1] enable privacy-preserving protocols where the *signer* and message owner or *user* are distinct entities. A user hides the message from the signer during the signing process. The BS is widely used in digital currency and e-voting [2]–[4]. It requires an interactive protocol between the user and signer (Figure 1).

In a BS protocol, a user interacts with a signer via interactive protocol in multiple rounds to obtain the final signature. The most efficient and cost-effective ones are two rounds, which are often called *round-optimal*. Furthermore, two-round protocols are not vulnerable to attack during concurrent execution of the signing algorithm [5]. Here, the user first *blinds* the message and sends it to a signer for *signing*, a signer then signs the blinded message and communicates the result to the user. The user then unblinds the message and publishes the message and signature pair to the verifier. Fischlin [6] first proposed a generic framework for constructing round-optimal blind signatures.

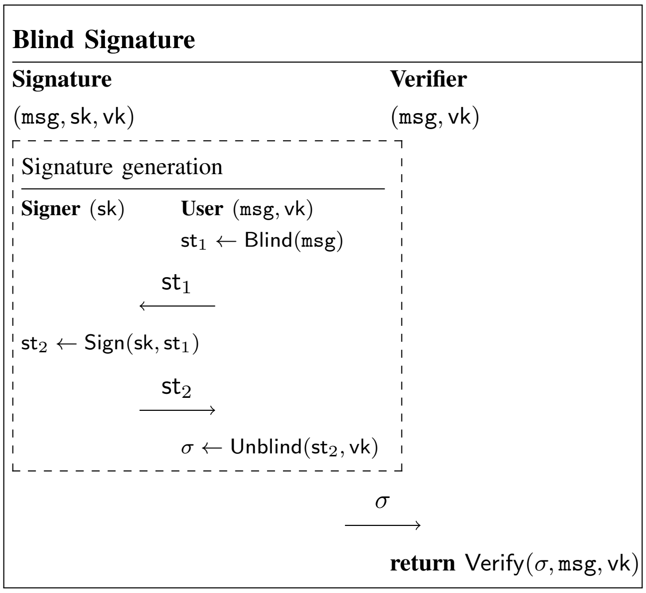
Blind Signature
Signature
Verifier
(msg, sk, vk)
(msg, vk)
Signature generation
Signer (sk)
User (msg, vk)
$\mathsf{st}_1 \leftarrow \mathsf{Blind}(\mathtt{msg})$
st_1
$\mathsf{st}_2 \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{st}_1)$
st_2
$\sigma \leftarrow \mathsf{Unblind}(\mathsf{st}_2, \mathsf{vk})$
σ
return $\mathsf{Verify}(\sigma, \mathtt{msg}, \mathsf{vk})$

Figure 1. A schematic representation of different parties involved and their interactions in a blind signature protocol.

A BS scheme needs to satisfy the two security properties, *blindness* and *one-more unforgeability* (explained in §3). In recent years, blind signatures have been studied extensively, leading to numerous instantiations based on various hardness assumptions. While the theoretical soundness of these schemes has been demonstrated, achieving practical efficiency, such as reducing signature and key size, reducing computation overheads, etc., remains a primary concern.

PQ blind signatures. Most of the currently existing BS schemes rely on the hardness of classical assumptions like integer factorization [7] and elliptic curve discrete logarithm problems [8], [9]. However, due to the Shor's [10] and Proos-Zalka's algorithm [11] a large quantum computer can subvert these assumptions. So, we need BS schemes based on PQ-assumptions for long-term security.

Recently, two new round-optimal BS schemes have been proposed based on (structured) lattice-based assumptions [5], [12], relying on the Fischlin's framework. Although, both of these schemes suffer from larger signature and public-key sizes.

The BS constructions based on the multivariate assumptions (for example, unbalanced oil and vinegar (UOV) problem [13]) can be a good solution to optimize the signature, public-key, and proof sizes (see Table 1). Also, the UOV signature scheme has been selected in the second round of National Institute of Standard and Technology's (NIST) ongoing additional digital signatures for the PQ cryptography standardization process [14]. However, as we will see, it is not so trivial to build a BS scheme from these assumptions.

Multivariate blind signature. In 2017, Petzoldt *et al.* [15] proposed a multivariate BS scheme. Their construction uses a commitment scheme, a multivariate trapdoor, and a zero-knowledge proof (ZKP) system. This ZKP allows users to prove they know a solution to a system of multivariate quadratic (MQ) equations without revealing the solution itself. The security of this approach [15] depends on the commitment scheme's perfect binding and hiding properties, as well as the inherent difficulty of solving multivariate quadratic (MQ) systems, which is believed to be quantum secure [16]. The commitment scheme in [15] utilizes a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$. Specifically, the message $\mathtt{msg}$ is committed by using a random vector $\mathbf{r} \in \mathbb{F}_q^m$ as follows:

$$\mathbf{b} = \mathcal{H}(\mathtt{msg}) - \mathcal{R}(\mathbf{r}), \tag{1}$$

where $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ is a random quadratic map. Subsequently, the user requests the signer to sign on the blind message $\mathbf{b}$. The signer applies a UOV-based signature algorithm [13] to generate a blind signature $\mathbf{s}$ and sends it to the user. The user constructs a non-interactive zero-knowledge (NIZK) proof π for the quadratic system $\mathcal{H}(\mathtt{msg}) = \mathcal{P}(\mathbf{s}) + \mathcal{R}(\mathbf{r})$, where $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ represents a public polynomial map based on the UOV scheme [13]. Here, n and m are the number of variables and the quadratic equations defined over the finite field $\mathbb{F}_q$ respectively. The user reveals this NIZK proof π as the final signature in the signature unblind phase.

On the one-more unforgeability of [15]: In [17], Beullens showed that the commitment scheme used in this BS scheme [15] is non-binding. This immediately breaks the one-more-unforgeability property of the scheme.[1]

In the same work, Beullens proposed a potential solution to mitigate the attack by abandoning the use of random polynomial system $\mathcal{R}$ in the commitment phase, and instead, employing a different commitment scheme based on AES with a NIZK proof [18]. However, this proposed solution leads to a large signature size, as the final signature will contain two proofs: one for the MQ problem and another for the AES-based commitment scheme [18].

As we see, constructing efficient BS from multivariate assumptions remains a challenge. In this work, we propose the first round optimal BS from multivariate assumptions, which achieves both the notion of *one-more-unforgeability* and *anonymity*. The resulting scheme also offers a smaller signature size than all the post-quantum solutions.

2. Our Contribution

We propose MQuBS, a new multivariate-based BS scheme, following Fischlin's round optimal framework [6]. For this, we introduce a new secure binding commitment scheme based on the multivariate assumptions (see Algorithm 1) as a building block for MQuBS.

Our MQuBS needs four cryptographic components. First, the new commitment scheme. Second, a hash-based commit-and-prove protocol [19]–[21]. Third, the UOV signature [13], [22] scheme as an underlying multivariate signature scheme [2].

Finally, to open the multivariate commitment scheme, we need a NIZK proof for the MQ problem. There exist many efficient and optimized NIZK proofs for MQ problem in the literature [18], [26]–[30]. In our construction, we use vector obvious linear evaluation in the head (VOLEitH)-based NIZK for the MQ problem to achieve smaller proof sizes and good execution time [18], [30].

In Section §5, we show that MQuBS offers anonymity (or blindness) and one-more-unforgeability in the random oracle model (ROM).

As claimed before, MQuBS offers a smaller signature (σ) compared to the previous post-quantum round-optimal BS schemes. The following Table 1 illustrates our claim[3].

TABLE 1. POST-QUANTUM ROUND OPTIMAL BLIND SIGNATURE

Scheme	Assumption	$\lvert\sigma\rvert$ (KB)
Agrawal *et al.* [12]	OM-ISIS	45
Beullens *et al.* [5]	MSIS and MLWE	22
Petzoldt *et al.* [15]	UOMUF-MQ	28.5
MQuBS (this work)	UOV and gWMQ	5

2.1. MQuBS: A brief overview

Below, we briefly introduce the key components and fundamental ideas of MQuBS (see Figure 3 for more details). The key generation algorithm of MQuBS is the same as the UOV-signature scheme [22]. The signature algorithm of MQuBS.Sign consists with three independent algorithms $\mathsf{Sign}_1, \mathsf{Sign}_2,$ and Sign_3.

Blinding phase $\mathcal{S} \leftarrow \mathcal{U}$: ($\beta \leftarrow \mathsf{Sign}_1(\mathsf{vk}, \mathtt{msg})$). In this phase, the user performs three steps. First, a random vector is generated, and its hash is concatenated with the message as part of the message digest computation. Next, the user runs two commitment schemes: Com_{MQ} and $\mathsf{Com}_{\mathsf{Hash}}$.

Fischlin's suggestions [6]. After computing the message digest $\mathbf{d} = \mathcal{H}(\mathtt{msg})$, it commits the $\mathbf{d}$ using a randomness r_1 to blind the message digest. (See Fig: 2). We add extra randomness during the computation of the message digest. Later it encrypts $\mathbf{d}$, r_1 along with a new randomness r_2. In addition, it also adds a proof of

[1] In the scheme, the authors achieved a weaker variant of the one-more-unforgeability, called universal one-more-unforgeable security (UOMUF). However, this property does not accurately reflect real-world attack scenarios for blind signatures. For more details, we refer the readers to [17]

[2] It is possible to incorporate any UOV-type signature schemes, like VDOO [23], Mayo [24], and QR-UOV [25] in MQuBS.

[3] The security model used in Petzoldt *et al.* construction is universal OMUF. Their construction is designed on the top of Rainbow [31] scheme, which is broken by [32]. So the parameters proposed in [15] will change. However, due to the vulnerability in the commitment scheme, the BS scheme is no longer secure.

Fischlin's Blind Signature [6]

User $\mathcal{U}$ $(\mathbf{d} \leftarrow \mathcal{H}(\texttt{msg}), \mathsf{vk})$		**Signer** $\mathcal{S}$ (sk)
$\beta \leftarrow \mathsf{Sign}_1(\mathsf{vk}, \mathbf{d})$		
choose $r_1, r_2, r_3 \xleftarrow{\$} \{0,1\}^{2\lambda}$		
$\mathbf{b} \leftarrow \mathsf{Com}(\mathbf{d}; r_1)$		
$\mathsf{ct} \leftarrow \mathsf{Enc}(\mathbf{d}, r_1; r_2)$		
$x_{\mathsf{Enc}} \leftarrow (\mathsf{ct}, \mathsf{pk}'_{\mathsf{Enc}})$ //statement		
$w_{\mathsf{Enc}} \leftarrow (\mathbf{d}, r_1, r_2)$ //witness		
compute $\pi_{\mathsf{Enc}} \leftarrow \mathsf{Proof}_{\mathsf{Enc}}(x_{\mathsf{Enc}}, w_{\mathsf{Enc}})$		
$\beta \leftarrow (\pi_{\mathsf{Enc}}, \mathbf{b}, \mathsf{ct})$		
	$\xrightarrow{\beta}$	
	$\psi \leftarrow \mathsf{Sign}_2(\beta, \mathsf{sk})$	
		$\texttt{if}$ $(\mathsf{Verify}_{\mathsf{Enc}}(\pi_{\mathsf{Enc}}, \mathsf{ct}))$
		$\mathbf{s} \leftarrow \mathsf{Sign}(\mathbf{b}, \mathsf{sk})$
	$\xleftarrow{\psi = \mathbf{s}}$	
$\sigma \leftarrow \mathsf{Sign}_3(\mathsf{vk}, \mathbf{s})$		
$\texttt{if}$ $(\mathsf{Verify}_{\mathsf{Sign}}(\mathbf{s}, \mathsf{vk}))$		
$Ct \leftarrow \mathsf{Enc}(\beta, \psi; r_3)$		
$x \leftarrow (Ct, \mathsf{pk}_{\mathsf{Enc}})$ // statement		
$w \leftarrow (\mathbf{b}, \mathbf{s}; r_3)$ // witness		
prepares a proof $\pi \leftarrow \mathsf{Proof}(x, w)$		
$\sigma \leftarrow (Ct, \pi)$		

Figure 2. Fischlin's round-optimal blind signature framework

encryption π_{Enc}. Instead of encryption, we use a hash-based commitment scheme. Finally, β contains a blind message, ciphertext, and proof of encryption π_{Enc}.

Computing the message digest. In MQuBS, the user initially uniformly generates a random vector $\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$. Then, it computes $\mathbf{t} = \mathcal{H}(\texttt{msg}, \mathcal{G}(\mathbf{r}))$, where $\mathcal{G} : \mathbb{F}_q^m \to \{0,1\}^{2\lambda}$, ($\lambda$ is a security parameter) and $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$ are collision-resistant hash functions[4]. These hash functions can be modelled as random oracles. Beullens *et al.* [5]'s lattice-based blind signature scheme used this trick to avoid one-more-ISIS assumptions. We employ this trick to achieve the same functionality. Basically it forces an adversary to fix $\mathbf{r}$ before querying the random oracle. It helps us to achieve the OMUF security of MQuBS.

Com_{MQ}: *New multivariate commitment scheme.* The user runs this scheme to blind the message digest. To build a secure BS , we propose a new commitment scheme designed to defend the Beullens's attack [17]. Our approach involves generating a random polynomial $\mathcal{R}$ during the signing phase by using the message as a seed for a pseudorandom generator (PRG) (see Algorithm 1). Our commitment scheme Com_{MQ} is given below.

$$\mathsf{Com}_{MQ}(\texttt{msg}; \mathbf{r}) = \mathbf{E}_1^{-1}\left(\mathcal{H}(\texttt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r})\right) \quad (2)$$

The random polynomial map $\mathcal{R}, \mathbf{E}_1,$ and $\mathbf{E}_2$ are generated using the message along with a random $\mathbf{r} \in \mathbb{F}_q^m$. We present this algorithm in Section §4.2.

Commit-and-prove a random vector and message. In Figure 2, the user commits the message digest and randomness using public-key encryption. Lattice-based BS constructions [5], [12] utilize lattice-based public-key encryption in an "*encryption to the sky*" fashion to achieve this goal. However, multivariate cryptography faces challenges. Most multivariate public key encryption schemes have either been broken in a short period [34]–[37] or result in inefficient constructions [38]. Therefore, we adopt standard commitment schemes in our scenario. To establish the OMUF security of MQuBS (see Theorem 4), we must decommit the committed value to retrieve the message and randomness. This is only feasible for a commitment scheme modeled as a random oracle. Consequently, hash-based commitment schemes are well-suited for our construction. The following hash-based commitment scheme commits $(\mathbf{r}, \texttt{msg})$ using a random number $u \in \{0,1\}^{2\lambda}$.

$$C = \mathsf{Com}_{\mathsf{Hash}}(\mathbf{r}, \texttt{msg}; u) = \mathcal{H}_{\mathsf{Com}}(\mathbf{r}, \texttt{msg}; u) \quad (3)$$

In addition, the user adds a NIZK proof π_{Com} with the blind message $\mathbf{b}$ and the committed value C; then it communicates to the signer. The proof π_{Com} includes two things. The first one is the proof of committed value C, that is the Equation 3. Here, the statement is the committed value C, and witness is $(\mathbf{r}, \texttt{msg}; u)$. And the second proof is the well-formedness of $\mathbf{b}$, that is, it should prove the Equation $\mathbf{b} = \mathbf{E}_1^{-1}\left(\mathcal{H}(\texttt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r})\right)$.

Observe that, the final signature does not include the proof, and the user can precompute the proof. Therefore it does not add to the performance of the blind signature scheme. At the end of this phase, the user sends $\beta = (C, \pi_{\mathsf{Com}}, \mathbf{b})$ to the signer and asks for a signature. Therefore, user sends $|\beta| = |C| + |\pi_{\mathsf{Com}}| + |\mathbf{b}| = 2\lambda + |\pi_{\mathsf{Com}}| + m\log q$-bit elements.

Signing phase $\mathcal{U} \leftarrow \mathcal{S}$**:** $(\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \beta))$. This procedure is the same as Fischlin's proposal [6]. After receiving β, it first verifies the proof π_{Com}, then it runs the UOV-signature algorithm [22] to compute the signature $\mathbf{s}$. It communicates $|\mathbf{s}| = n\log q$-bit elements to user.

Unblind the signature $\mathcal{V} \leftarrow \mathcal{U}$**:** $(\sigma \leftarrow \mathsf{Sign}_3(\mathsf{sk}, \psi))$. After verifying the validity of the signature, Fischlin [6] proposed committing to β and ψ using public key encryption and providing a NIZK proof of encryption. The final signature includes both the proof and the ciphertext. A key improvement in MQuBS is that we eliminate the need for public key encryptions in Sign_3, as the user instead provides a NIZK proof for opening the commitment Com_{MQ}.

As per Fischlin's proposal, the user first checks whether $\mathcal{H}(\texttt{msg}, \mathcal{G}(\mathbf{r})) = \mathbf{E}_1\mathcal{P}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$ holds. Once this verification is done, then it prepares a proof π_{MQ} for the solution $(\mathbf{s}, \mathbf{r})$ of the quadratic equations $\mathbf{t} = \tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{E}_1\mathcal{P}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2)$. The most efficient NIZK proof for a solution of multivariate quadratic system can be implemented using multi-party-computation-in-the-head (MPCitH)-based and vector-oblivious-linear-evaluation-in-the-head (VOLEitH)-based framework [18], [28]–[30]. The final signature σ contains three items, one is proof π_{MQ}, a seed to generate emulsifier maps, and a

[4] For notational purpose we use two hash function $\mathcal{G}$ and $\mathcal{H}$, in practice any one hash function is enough. For example SHA-3 [33].

random quadratic map, and the third component is $\mathcal{G}(\mathbf{r})$. The size of the final signature is $4\lambda + |\pi_{MQ}|$-bits, where 4λ equals to the size of a seed and $\mathcal{G}(\mathbf{r})$, and $|\pi_{MQ}|$ is the proof size of π_{MQ}.

Security. Here, we sketch the security arguments of our commitment scheme. The main underlying hardness assumption is the generalized whipped multivariate quadratic problem (gWMQ). It asks to solve a quadratic system of the form $\mathbf{E}_1\mathcal{R}_1(\mathbf{x}_1) - \mathbf{E}_2\mathcal{R}_2(\mathbf{x}_2) = \mathbf{t}$. We introduce the generalization of Beullens's whipped multivariate quadratic problem (WMQ) [24]. We also show that MQuBS offers blindness and OMUF.

Security of commitment scheme. Any secure commitment scheme has two properties, one is computationally hiding and another one is perfectly binding. The computationally hiding property of our commitment scheme can be derived from the collision resistance of the cryptographically secure hash function. However, the perfectly binding property is not so obvious. The perfectly binding property of our commitment scheme can be derived from the fact that finding a collision in the commitment scheme, that is, that is, $\mathsf{Com}(\mathtt{msg}_1, \mathcal{G}(\mathbf{r}_{\mathtt{msg}_1})) = \mathsf{Com}(\mathtt{msg}_2, \mathcal{G}(\mathbf{r}_{\mathtt{msg}_2}))$, which requires solving a special structure quadratic system. The following claim guarantees the perfectly binding property of our commitment scheme. We formally present this theorem in Theorem 2, and then we prove the following claim. The detailed proof of this claim can be found in Section §4.1.

Claim 1 (Perfect binding of the commitment scheme). The commitment scheme presented in Figure 1 is perfectly binding under the assumptions of gWMQ.

Blindness and One-more-unforgeability of MQuBS.The blindness of MQuBS depends on the zero-knowledge property of the NIZK proofs (see Theorem 3). We present this claim informally, and later in Theorem 4, we provide a formal statement and proof.

Claim 2 (*MQuBS *is OMUF). MQuBS achieves OMUF, depending on the EUF-CMA of the underlying signature and the soundness of the NIZK proof system.

We use three lemmas to prove the OMUF property of MQuBS. The first Lemma 1 tells that the underlying signature scheme is EUF-CMA. Since we have chosen UOV as an underlying signature, so we follow the EUF-CMA-security proof of UOV signature [22]. Note that our version of UOV differs slightly from standard UOV [13], [22] (see Figure 4). The EUF-CMA-security of underlying signature relies on the hardness of UOV, and gWMQ problem. The proof of the statement of Lemma 1 requires reduction from EUF-CMAto EUF-KO-security of the scheme (key-only-attack). Further, the hardness of UOV and gWMQ together implies the security against EUF-KO for the underlying signature. Since similar proofs for these three lemmas are available in [22], so we refer [22] for these proofs.

Practicality: efficiency and comparisons. The signature size of MQuBS is significantly smaller than the existing post-quantum round-optimal blind signatures [5], [39]. The initial communication from the user $\mathcal{U}$, to the signer $\mathcal{S}$, consists of $|\beta| = (2\lambda + |\pi_{\mathsf{Com}}| + m\log q)$ bits. The second interaction from $\mathcal{S}$ to $\mathcal{U}$ requires $n \log q$ bits. The size of the final signature is $4\lambda + |\pi_{MQ}|$, since it contains $\pi_{MQ}, \mathcal{G}(\mathbf{r})$ and a seed to generate emulsifier maps and random quadratic maps. We present the communication costs of MQuBS in Table 2.

TABLE 2. COMMUNICATION COSTS OF MQuBS

Communication	$\mathcal{U} \to \mathcal{S}$	$\mathcal{S} \to \mathcal{U}$	$\mathcal{U} \to \mathcal{V}$
Sizes (in bits)	$2\lambda + \lvert\pi_{\mathsf{Com}}\rvert + m\log q$	$n\log q$	$4\lambda + \lvert\pi_{\mathcal{MQ}}\rvert$

Communication cost for 128-bits security level. We use formulas from Table 2 to compute signature and public-key size for 128-bit security level. To estimate the signature size, we calculate the size of π_{Com}, the signature size of the underlying signature UOV, and the size of π_{Com}. The proof contains two things, one is the proof for the hash-based commitment, and the relation 2. According to [21], the proof size of the hash-based commitment is 33KB. The second component is a linear relation which requires 15KB. Thus total cost for π_{Com} is around 47KB. The UOV signature [13], [22] offers a 96-byte signature size for 128-bit security. According to the UOV SL-1 parameters (128-bit security) [22], the UOV public polynomial map has 64 quadratic equations with 160 variables (see Table 6.2). Therefore, a user prepares a NIZK proof for a quadratic system with $n+m$ variables and m- homogeneous quadratic equations. Bui [30] offers the most efficient proof size. Using the technique of [30], the proof size for our quadratic system is around 4.96KB. Therefore, the size of blind message β is 48KB, the signer communicates 96 bytes as a signature on the blind message, and then the user reveals 5KB as a final signature size. Since the secret and public keys are the same as the UOV signature, the secret and public keys for MQuBS are 48bytes, and 43.576KB respectively.

3. Background

In this section, we define the blind signature scheme Section §3.1 and its security properties.

Basic Notations. We denote $a \xleftarrow{\$} U$ to signify a is generated randomly from the set U. Any homogeneous quadratic map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ consisting m quadratic polynomials is denoted as $p_1, p_2, \ldots, p_m$. We define the polar form $\mathcal{DP} : \mathbb{F}_q^n \times \mathbb{F}_q^n \to \mathbb{F}_q^m$ of a quadratic map $\mathcal{P}$ as: $\mathcal{DP}(\mathbf{u}, \mathbf{v}) = \mathcal{P}(\mathbf{u} + \mathbf{v}) - \mathcal{P}(\mathbf{u}) - \mathcal{P}(\mathbf{v})$. Since $\mathcal{P}$ is a homogeneous quadratic map, we assume $\mathcal{P}(0) = 0$. The notation $\mathcal{DP}_{\mathbf{u}}(\mathbf{v})$ is employed when $\mathbf{u}$ is fixed, essentially representing the partial derivatives with respect to $\mathbf{u}$. We write $\mathbb{F}_q$ to denote the finite field with q elements, and $\mathbb{F}_q^n$ as a n-dimensional vector with elements from $\mathbb{F}_q$. We also denote $\mathsf{GL}(m, q)$ for the set of invertible $m \times m$ matrices over $\mathbb{F}_q$. Throughout this paper, λ serves as the security parameter.

3.1. Blind Signature

A *round-optimal blind signature* scheme, denoted as $\mathsf{BS} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$, requires only two rounds of interaction between the signer $\mathcal{S}$ and the user $\mathcal{U}$ to generate a blind signature. The Sign algorithm has three sub-algorithms: Sign_1, Sign_2, and Sign_3. Below, we describe each of these algorithms in detail.

$(\mathsf{vk}, \mathsf{sk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$: On input the security parameter λ, it outputs a verification key vk and a secret key sk.

$(\beta, S_{\mathcal{U}}) \leftarrow \mathsf{Sign}_1(\mathsf{vk}, \mathtt{msg})$. In the initial phase of the signing protocol, the user blinds the message and sends it to the signer. This probabilistic polynomial-time (PPT) algorithm computes a message state $S_{\mathcal{U}}$ and generates a first message β, which is then sent to the signer.

$\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \beta)$. The signer computes a signature ψ using its secret key sk on β. It runs the underlying signature algorithm to do this. Afterward, the signer sends the signature ψ back to the user as the signature for β.

$\sigma \leftarrow \mathsf{Sign}_3(\psi, S_{\mathcal{U}})$. The user removes the blindness of the received signature ψ using $S_{\mathcal{U}}$ and derives a signature σ of the original message msg.

$0/1 \leftarrow \mathsf{Verify}(\mathsf{vk}, \mathtt{msg}, \sigma)$. The verifier uses its verification key vk to verify whether σ is a correct signature on the message msg or not.

Properties. The blind signature algorithm must adhere to the *correctness* property. Essentially, this ensures that if a signature is generated honestly, it should be verified with a very high probability.

Definition 1 (Correctness). A BS is considered *correct* if for any message digest $\mathsf{d} = \mathcal{H}(\mathtt{msg})$.

$$\Pr\left[\mathsf{Verify}(\mathsf{vk}, \sigma, \mathtt{msg}) = 1 \middle| \begin{matrix} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ (\beta, S_{\mathsf{U}}) \leftarrow \mathsf{Sign}_1(\mathsf{pk}, \mathsf{d}) \\ \psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \beta) \\ \sigma \leftarrow \mathsf{Sign}_3(\psi, S_{\mathcal{U}}) \end{matrix}\right] \geq 1 - \mathsf{negl}(\lambda)$$

The two security properties of blind signatures are the *blindness* (or *anonymity*) and the *one-more-unforgeability*. The *blindness* ensures that the signer of a message receives no information about the content of the message to be signed.

Definition 2 (Anonymity/ Blindness). We call the BS offers *blindness* when, for each polynomial-time three-part stateful adversary $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$, there exists a negligible function negl such that, for any two messages $\mathsf{d}_0, \mathsf{d}_1$, the following condition holds.

$$\left|\Pr\left[\mathcal{A}_3(\sigma_0, \sigma_1) = b \middle| \begin{matrix} (\mathsf{vk}) \leftarrow \mathcal{A}_1(\lambda),\ \mathsf{b} \in_{\mathsf{U}} \{0,1\} \\ (\mathsf{d}_1^0, S_{\mathcal{U}}^0) \leftarrow \mathsf{Sign}_1(\mathsf{pk}, \mathsf{d}_0) \\ (\mathsf{d}_1, S_{\mathcal{U}}^1) \leftarrow \mathsf{Sign}_1(\mathsf{pk}, \mathsf{d}_1) \\ \mathsf{d}_2^b, \mathsf{d}_2^{1-b} \leftarrow \mathcal{A}_2(\mathsf{d}_1^b, \mathsf{d}_1^{1-b}) \\ \sigma_0 \leftarrow \mathsf{Sign}_3(\mathsf{d}_0^2, \mathsf{S}_{\mathcal{U}}^0) \\ \sigma_1 \leftarrow \mathsf{Sign}_3(\mathsf{d}_1^2, \mathsf{S}_{\mathcal{U}}^1) \end{matrix}\right] - \frac{1}{2}\right| < \mathsf{negl}(\lambda)$$

The *one-more unforgeability (OMUF)* ensures that if a malicious user interacts r times with an honest signer and receives r message-signature pairs, the probability that the malicious user can produce a $r+1$ message-signature pairs without further interaction with the signer is negligible.

Definition 3 (OMUF). The BS is said to have one-more unforgeable if for every PPT adversary $\mathcal{A}$ that makes at most r queries to the honest signer (where r upper bounded by $\mathsf{poly}(\lambda)$) can produce a $r+1$ blind message-blind signature pair with negligible probability negl. The following probability should be less than $\mathsf{negl}(\lambda)$.

$$\Pr\left[\begin{matrix} (\mathsf{d}_i \neq \mathsf{d}_j)_{\forall i,j=1,\ i\neq j}^{r+1} \\ (\mathsf{Verify}(\mathsf{vk}, \sigma_{\mathsf{i}}, \mathsf{d}_{\mathsf{i}}) = 1)_{\forall i=1}^{r+1} \end{matrix} \middle| \begin{matrix} (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \{(\mathsf{d}_i, \sigma_i)\}_{i\in[r]} \leftarrow \mathcal{A}^{\mathsf{Sign}_2(\mathsf{sk}, \cdot)}(\mathsf{vk}) \end{matrix}\right] < \mathsf{negl}(\lambda)$$

4. MQuBS: A compact, quantum-secure and round-optimal blind signature scheme

In this section, we describe the construction of our blind signature scheme MQuBS in detail. We also describe the major components of our scheme in this section.

We have presented our scheme in Figure 3. As can be seen in the Figure 3, for MQuBS, we require some cryptographic components as *ingredients*. First we need two hash functions $\mathcal{G} : \{0,1\}^* \to \{0,1\}^{2\lambda}$, $\mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$ and one hash based commitment scheme $\mathcal{H}_{\mathsf{Com}}$. Any cryptographically secure hash function can be deployed here such as the current NIST standard SHA-3 [33]. Later in the security proof (see Section §5), we model these functions as random oracles.

4.1. New Commitment Scheme

We need a secure commitment scheme to hide the message. We present our commitment scheme Com_{MQ} in Algorithm 1 below. Later, in Section §5, we will show that this scheme is perfectly binding and computationally hiding.

Algorithm 1 Com_{MQ}: Multivariate Commitment Scheme

$\mathbf{b} \leftarrow \mathsf{Com}_{MQ}(\mathtt{msg}; \mathbf{r}) = \mathbf{E}_1^{-1}\left(\mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r})\right)$

```
Input: msg
Output: $\mathbf{b}$, $(\mathbf{E}_1, \mathbf{E}_2, \mathcal{R})$
$\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$, $\rho \leftarrow \mathcal{G}(\mathbf{r})$
while $(\det(\mathbf{E}_1) \neq 0$ & $\det(\mathbf{E}_2) \neq 0)$ {
  rnd $\xleftarrow{\$} \{0,1\}^{2\lambda}$, seed1 ← Hash(msg|| $\mathbf{r}$||rnd)
  seed2 ← Hash(msg || seed1)
  $\mathbf{E}_1$ ← XOF(seed1);  $\mathbf{E}_2$ ← XOF(seed2) }
seed3 ← Hash(msg || seed2)
$\mathcal{R}$ ← XOF(seed3)
$\mathbf{b} \leftarrow \mathbf{E}_1^{-1}\left(\mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r})\right)$.
return $\mathbf{b}$, $(\mathbf{E}_1, \mathbf{E}_2, \mathcal{R})$
```

At first, the Algorithm 1 selects an $\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$. Then it computes, $\mathcal{G}(\mathbf{r})$. Using $\mathbf{r}$, the message msg, and the random string rnd it generates $\mathbf{E}_i$'s and $\mathcal{R}$, where $\mathbf{E}_i \in \mathbb{F}_q^{m\times m}$ and $\mathcal{R}$ is a random quadratic map from $\mathbb{F}_q^m$ to $\mathbb{F}_q^m$. Each generated $\mathbf{E}_i$ must be an invertible matrix; otherwise, the algorithm changes rnd and recomputes. Following Beullens [24], we refer to $\mathbf{E}_i$ as an *emulsifier map*. At the end, the commitment of a message is computed as $\mathbf{b} = \mathsf{Com}_{MQ}(\mathtt{msg}; \mathbf{r}) = \mathbf{E}_1^{-1}\left(\mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r})\right)$.

4.2. Algorithms of MQuBS

The complete MQuBS algorithm outlined in Figure: 3 includes three primary algorithms: MQuBS.KeyGen, MQuBS.Sign, and MQuBS.Verify. The MQuBS.Sign consists of three distinct algorithms: Sign_1, Sign_2, and Sign_3. The user runs Sign_1 and Sign_3; while the signer executes the Sign_2 algorithm.

Multivariate Blind Signature in the Fischlin's Framwork

Signature $(\mathtt{msg}, \mathsf{sk}, \mathsf{vk})$ **Verifier** $(\mathtt{msg}, \mathsf{vk} = \mathcal{P}_{UOV})$

Signature generation

User $(\mathtt{msg}, \mathsf{vk})$ **Signer** (sk)

$\underline{\beta \leftarrow \mathsf{Sign}_1(\mathcal{P}_{UOV}, \mathtt{msg})}$ (See Algorithm 3)

$\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m;\ \rho \leftarrow \mathcal{G}(\mathbf{r})$

Generate $\mathbf{E}_1$, $\mathbf{E}_2$, $\mathcal{R}$ from $\mathtt{msg}, \mathsf{vk},$ and $\mathbf{r}$

$\mathbf{b} \leftarrow \mathsf{Com}_{MQ}(\mathtt{msg}, \mathbf{r}) = \mathbf{E}_1^{-1}\Big(\mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}_{\mathtt{msg}}(\mathbf{r})\Big)$

$u \xleftarrow{\$} \{0,1\}^{2\lambda};\ C \leftarrow \mathcal{H}_{\mathsf{Com}}(\mathbf{r}, \mathtt{msg}; u)$

$x_{\mathsf{Com}} \leftarrow (\mathbf{E}_1, \mathbf{E}_2, \mathcal{R}, C, \mathbf{b})$

$w_{\mathsf{Com}} \leftarrow (\mathbf{r}, \mathtt{msg}, u)$

$\mathsf{Relation}_1 \leftarrow \mathbf{b} = \mathsf{Com}_{MQ}(\mathtt{msg}, \mathbf{r}) \bigwedge C = \mathcal{H}_{\mathsf{Com}}(\mathbf{r}, \mathtt{msg}; u)$

$\pi_{\mathsf{Com}} \leftarrow \mathsf{NIZK.Proof}_1(x_{\mathsf{Com}}, w_{\mathsf{Com}}, \mathsf{Relation}_1)$

$\beta \leftarrow (C, \pi_{\mathsf{Com}}, \mathbf{b})$

$$\xrightarrow{\beta}$$

$\underline{\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \mathbf{b})}$ (See Algorithm 4)

`if` $(\mathsf{NIZK.Verify}_1(x_{\mathsf{Com}}, \pi_{\mathsf{Com}}))$

compute $\mathbf{s} \leftarrow \mathsf{UOV.Sign}(\mathbf{b}, \mathsf{sk})$

set $\psi \leftarrow \mathbf{s}$

$$\xleftarrow{\psi}$$

$\underline{\sigma \leftarrow \mathsf{Sign}_3(\mathcal{P}_{UOV}, \psi)}$ (See Algorithm 5)

`if` $(\mathsf{UOV.Verify}(\psi, \mathsf{vk}))$

Define :: $\tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2)$

$x \leftarrow (\mathtt{msg}, \rho, \mathbf{E}_1, \mathbf{E}_2, \mathcal{P}_{UOV}, \mathcal{R})$

$w \leftarrow (\mathbf{s}, \mathbf{r})$

$\mathsf{Relation}_2 \leftarrow \{(\mathbf{s}, \mathbf{r}) : \mathcal{H}(\mathtt{msg}, \rho) = \tilde{\mathcal{P}}(\mathbf{s}, \mathbf{r})\}$

$\pi_{MQ} \leftarrow \mathsf{NIZK.Proof}_2(x, w, \mathsf{Relation}_2)$

$\sigma \leftarrow (\pi_{\mathcal{MQ}}, \rho, \tilde{\mathcal{P}})$

$$\xrightarrow{\sigma}$$

$\mathbf{t} \leftarrow \mathcal{H}(\mathtt{msg}, \rho)$

return $\mathsf{NIZK.Verify}_2(\sigma, \mathbf{t}, \mathsf{vk})$

Figure 3. MQuBS Multivariate Blind Signature

4.2.1. Key Generation of MQuBS .

The key generation algorithm MQuBS.KeyGen is shown in Algorithm 2, which generates a UOV secret and verification key pair. This algorithm is the same as the key generation algorithm of the UOV signature scheme [22]. The parameters for MQuBS are (n, m, q, r). Here, n is the number of variables present in m homogeneous quadratic equations defined over a finite field $\mathbb{F}_q$, and r is the number of repetitions required to execute the NIZK proof π_{MQ} for the solution of a MQ system.

Key Generation: $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{MQuBS.KeyGen}(1^\lambda)$. Let O is a *secret oil subspace* for the UOV signature, and $\mathcal{P}_{UOV} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is a public polynomial map (*public key*) so that $\mathcal{P}_{UOV}(O) = 0$. The secret linear subspace O is the row space of the matrix $\tilde{\mathbf{O}} = [\mathbf{O}\ \mathbf{I_m}]$, where the $\mathbf{O} \xleftarrow{\$} \mathbb{F}_q^{o \times n-o}$. Each polynomial in the public polynomial map is constructed using the following equation.

$$\begin{pmatrix}\mathbf{O}^\top & \mathbf{I_m}\end{pmatrix} \mathbf{P}_i \begin{pmatrix}\mathbf{O} \\ \mathbf{I_m}\end{pmatrix} = \mathbf{O}^\top\mathbf{P}_i^{(1)}\mathbf{O} + \mathbf{O}^\top\mathbf{P}_i^{(2)} + \mathbf{P}_i^{(3)} = 0$$

Here, $\mathbf{P}_i^{(1)} \in \mathbb{F}_q^{(n-m)\times(n-m)}$, $\mathbf{P}_i^{(3)} \in \mathbb{F}_q^{m \times m}$ are upper triangular matrices and $\mathbf{P}_i^{(2)} \in \mathbb{F}_q^{(n-m)\times(n-m)}$ so that

$$\mathbf{P}_i = \begin{pmatrix}\mathbf{P}_i^{(1)} & \mathbf{P}_i^{(2)} \\ 0 & \mathbf{P}_i^{(3)}\end{pmatrix}.$$

So, generates $\mathbf{P}_i^{(1)}$, $\mathbf{P}_i^{(2)}$ from seed using a CSPRNG, and set $\mathbf{P}_i^{(3)} \leftarrow \mathsf{Upper}(-\mathbf{O}^\top\mathbf{P}_i^{(1)}\mathbf{O} - \mathbf{O}^\top\mathbf{P}_i^{(2)})$.

Algorithm 2 Key Generation of MQuBS

```
$(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{MQuBS.KeyGen}(1^\lambda)$
**Input** :$\lambda$
**Output** :$\mathsf{sk}, \mathsf{vk}$
$\mathbf{O} \xleftarrow{\$} \mathbb{F}_q^{o\times n-o}$
$\mathsf{seed} \xleftarrow{\$} \{0,1\}^{2\lambda}$
for $1 \le i \le m$
    $\mathbf{P}_i^{(1)} \leftarrow \mathsf{CSPRNG}(\mathsf{seed}, i)$
    $\mathbf{P}_i^{(2)} \leftarrow \mathsf{CSPRNG}(\mathsf{seed}, i)$
    $\mathbf{P}_i^{(3)} \leftarrow \mathsf{Upper}(-\mathbf{O}^\top \mathbf{P}_i^{(1)} \mathbf{O} - \mathbf{O}^\top \mathbf{P}_i^{(2)})$
$\mathsf{sk} \leftarrow \{\mathbf{O}\}$
$\mathsf{vk} \leftarrow \mathcal{P}_{UOV} = \left\{\mathsf{seed}, \left(\mathbf{P}_i^{(3)}\right)_{i=1}^{m}\right\}$
**return** $(\mathsf{sk}, \mathsf{vk})$
```

4.2.2. Interactive Signing Algorithm of MQuBS : $\sigma \leftarrow \mathsf{MQuBS.Sign}(\mathtt{msg}, \mathsf{sk}, \mathsf{vk})$. The user blinds the message using Sign_1 (see Algorithm 3) and sends the masked message to the signer for signing. The signer, utilizing the Sign_2 algorithm (see Algorithm 4), generates a signature and returns it to the user. After receiving the blinded message's signature, the user finalizes the signature for the original message using the Sign_3 algorithm (see Algorithm 5). Then it publishes the signature. Now we describe each algorithm.

Blind the message: $\beta \leftarrow \mathsf{Sign}_1(\mathtt{msg}, \mathsf{vk})$. Here, the user has inputs a public key $\mathcal{P}_{UOV}$, and a message $\mathtt{msg}$. Then, it randomly generates $\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$. It further computes, $\mathcal{G}(\mathbf{r})$, and $\mathbf{t} \leftarrow \mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r}))$. Additionally, it generates two emulsifier maps $\mathbf{E}_1, \mathbf{E}_2$ ($m \times m$ invertible matrices over $\mathbb{F}_q$) and a random quadratic map $\mathcal{R} : \mathbb{F}_q^m \to \mathbb{F}_q^m$ from $\mathtt{msg}$, $\mathcal{G}(\mathbf{r})$, and $\mathcal{P}_{UOV}$. To do this, the user computes the following seeds

- $\mathsf{seed}_{\mathbf{E}_1} \leftarrow \mathsf{Hash}(\mathcal{H}(\mathtt{msg})||\ \mathcal{G}(\mathbf{r})||\ \mathcal{P}_{UOV})$,
- $\mathsf{seed}_{\mathbf{E}_2} \leftarrow \mathsf{Hash}(\mathcal{H}(\mathtt{msg})||\mathsf{seed}_{\mathbf{E}_1})$, and
- $\mathsf{seed}_{\mathcal{R}} \leftarrow \mathsf{Hash}(\mathcal{H}(\mathtt{msg})\ ||\ \mathsf{seed}_{\mathbf{E}_2}))$.

Now using these seeds, it generates $\mathbf{E}_1, \mathbf{E}_2$ and $\mathcal{R}$. It uses the eXtendable Output Function (XOF) to generate these outputs. The user changes $\mathbf{r}$, if $\mathbf{E}_1$, and $\mathbf{E}_2$ are not invertible. Note that each seed depends on the message $\mathtt{msg}$, so when the message changes, then $\mathbf{E}_1, \mathbf{E}_2$ and $\mathcal{R}$ also change. Then, it outputs the blind message $\mathbf{b} \leftarrow \mathbf{E}_1^{-1}(\mathbf{t} - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$. Further, it applies the hash-based commitment scheme to commit $\mathbf{r}$, and $\mathtt{msg}$ using a randomly generated 2λ-bit string u, that is, Equation 3. Later it provides a NIZK proof π_{Com} for C, and it also adds a proof for the well-formedness of $\mathbf{b} = \mathbf{E}_1^{-1}(\mathbf{t} - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$. In the proof π_{Com}, the statement x_{Com} are $(\mathbf{E}_1, \mathbf{E}_2, \mathcal{R}, C, \mathbf{b})$, and witness w_{Com} are $\mathbf{r}, \mathtt{msg}, u_1$. Finally, the user communicates $\beta = (\mathbf{b}, C, \pi_{\mathsf{Com}})$ to the signer.

NIZK proof π_{Com}. The algorithm $\mathsf{NIZK.Proof}_1$ prepares the proof and $\mathsf{NIZK.Verify}_1$ algorithm verifies the proof π_{Com}. We use a hash-based commitment scheme that employs a secure hash function, such as SHA-3 [33], [40]. The user adds a NIZK proof of the commitment. We use existing hash-based NIZK proof for this purpose. This is known before our work, so for more details about hash-based commitment and proof of the commitment we refer to [19]–[21]. We sketch brief details about these schemes.

Consider the hash function $\mathcal{H}_{\mathsf{Com}}$ in Equation 3, which can be computed by an N-gate circuit ϕ, such that $C = \mathcal{H}_{\mathsf{Com}}(\alpha) = \phi(\alpha)$, where $\alpha = (\mathtt{msg}, \mathbf{r}; u)$. The multi-party computation (MPC) protocol computes C as in Equation 3, with each player holding shares of the input α, and the output C being public.

Now, the user simulates an MPC protocol internally (i.e., *in their head*), committing to the state and transcripts of all participating parties. In the NIZK protocol (in this case, the Signer), the verifier then *corrupts* a random subset of the simulated players after observing their full internal state. The verifier verifies whether the computation was executed correctly from the viewpoint of the corrupted players. If this check passes, it convinces the verifier that the output is correct and that the prover (in our case, the User) knows the value of α. To ensure confidence in the verifier, this process is repeated over multiple rounds. For details understanding we refer the ZKBoo protocol [19]. The second part of π_{Com} is use same methodology like this. For the proof size and parameters, we refer once again to this source.

Communication cost. Therefore, the user needs to send a 2λ-bit string for C, $|\pi_{\mathsf{Com}}|$ bits for the NIZK proof, and $m \log q$ bits for the blind message. In total, this amounts to $2\lambda + |\pi_{\mathsf{Com}}| + m \log q$ bits to communicate with the signer as the blind message.

Algorithm 3 Sign_1: Message blinding

```
$\beta \leftarrow \mathsf{Sign}_1(\mathcal{P}_{UOV}, \mathtt{msg})$
**Input:** $\mathsf{vk} = \mathcal{P}_{UOV}$; $\mathtt{msg}$
**Output:** $\beta = (\mathbf{b}, C, \pi_{\mathsf{Com}})$
Hash: $\mathcal{G} : \{0,1\}^* \to \{0,1\}^{2\lambda}, \mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^n$
while $(\det(\mathbf{E}_1) \neq 0\ \&\ \det(\mathbf{E}_2) \neq 0)\{$
    $\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$, $\mathsf{seed}_{\mathbf{E}_1} \leftarrow \mathcal{G}(\mathtt{msg}||\mathcal{G}(\mathbf{r})||\mathcal{P}_{UOV})$
    $\mathsf{seed}_{\mathbf{E}_2} \leftarrow \mathcal{G}(\mathtt{msg}||\mathsf{seed}_{\mathbf{E}_1})$
    $\mathbf{E}_1 \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathbf{E}_1})$, $\mathbf{E}_2 \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathbf{E}_2})\}$
$\mathbf{t} \leftarrow \mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r}))$
$\mathsf{seed}_{\mathcal{R}} \leftarrow \mathcal{G}(\mathtt{msg}||\mathsf{seed}_{\mathbf{E}_2})$, $\mathcal{R} \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathcal{R}})$
$\mathbf{b} \leftarrow \mathbf{E}_1^{-1}(\mathbf{t} - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$
$u \xleftarrow{\$} \{0,1\}^\lambda$, $C \leftarrow \mathcal{H}(\mathtt{msg}, \mathbf{r}, u)$
$x_{\mathsf{Com}} \leftarrow C$, $w_{\mathsf{Com}} \leftarrow (\mathtt{msg}, \mathbf{t}, u)$
$\mathsf{Relation} :: \mathbf{b} = \mathbf{E}_1^{-1}(\mathbf{t} - \mathbf{E}_2\mathcal{R}(\mathbf{r})) \bigwedge C = \mathcal{H}(\mathtt{msg}, \mathbf{r}, u)$
$\pi_{\mathsf{Com}} \leftarrow \mathsf{NIZK.Proof}_1(x_{\mathsf{Com}}, w_{\mathsf{Com}}, \mathsf{Relation})$
$\beta \leftarrow (\mathbf{b}, C, \pi_{\mathsf{Com}})$
**return** $\beta$
```

Blind signature computation: $\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \beta)$. In this algorithm, the signer (or issuer) receives a blinded message $\mathbf{b}$, secret key sk, and a NIZK proof π_{Com} along with a statement x_{Com}. It first verifies π_{Com} using the verification algorithm $\mathsf{NIZK.Verify}_1$. If the verification is successful, it proceeds to compute a signature $\mathbf{s}$. This is done by executing the $\mathsf{UOV.Sign}$ algorithm on $\mathbf{b}$ and sk. Finally, the signer outputs $\psi = \mathbf{s}$ and delivers it to the user as a signature on the blinded message β.

UOV signature generation: $\mathbf{s} \leftarrow \mathsf{UOV.Sign}(\mathsf{sk}, \mathbf{b})$. The signer wants to compute $\mathbf{s} = \mathcal{P}_{UOV}^{-1}(\mathbf{b})$. Initially, the signer randomly selects vinegar vector $\mathbf{v} \xleftarrow{\$} \mathbb{F}_q^n$ and attempts to solve the subsequent linear system $\mathbf{b} = \mathsf{L}_\mathbf{v}(\mathbf{o})$:

$$\mathsf{L}_\mathbf{v} :: \mathbf{b} = \mathcal{P}_{UOV}(\mathbf{v}) + \mathcal{D}\mathcal{P}_{UOV\mathbf{v}}(\mathbf{o}).$$

Note that, $\mathcal{P}_{UOV}(\mathbf{o}) = 0$, since $\mathbf{o}$ belongs the secret linear subspace O. The linear system is invertible with an approximate probability of $(1 - \frac{1}{q})$. In cases, if it fails, the signer will re-sample $\mathbf{v}$ and reiterate the aforementioned procedure. This approach mirrors the methodology employed in UOV signature algorithm [22]. At the end, signer communicates $\psi = \mathbf{s}$ as a signature on the blind message.

Algorithm 4 Sign_2 : Signature computation by Signer

```
$\psi = (\mathbf{s}) \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \beta)$
Input: $\mathsf{sk}, \beta$
Output: $\psi = \mathbf{s}$
if (NIZK.Verify$_1(x_{\mathsf{Com}}, Com) \neq 1$)
    abort
while ($\det(\mathsf{L}_\mathbf{v}) \neq 0$){
    $\mathbf{v} \xleftarrow{\$} \mathbb{F}_q^n$
    $\mathsf{L}_\mathbf{v} :: \mathbf{b} = \mathcal{P}_{UOV}(\mathbf{v}) + \mathcal{D}\mathcal{P}_{UOV\mathbf{v}}(\mathbf{o})$ }
solve $\mathbf{b} = \mathsf{L}_\mathbf{v}(\mathbf{o})$, $\mathbf{s} \leftarrow \mathbf{v} + \mathbf{o}$
return $\psi = (\mathbf{s})$
```

Communication cost. The cost for this round is $|\psi|$, meaning the signer transmits an $n \log q$-bit string to the user.

Unblind the signature: $\sigma \leftarrow \mathsf{Sign}_3(\mathcal{P}_{UOV}, \psi)$. The user first runs UOV.Verify to ensure that the UOV signature s is correctly generated by the signer.

UOV verification algorithm: $0/1 \leftarrow \mathsf{UOV.Verify}(\mathsf{vk}, \mathbf{s}, \mathbf{b})$. It evaluates $\mathcal{P}_{UOV}(\mathbf{s})$ and returns 0 if the result doesn't match $\mathbf{b}$, otherwise returns 1.

If UOV verification fails, the user aborts the protocol. Otherwise, it prepares a NIZK proof π_{MQ} for a solution $(\mathbf{s}, \mathbf{r})$ of the following quadratic system.

$$\mathbf{t} = \tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2)$$

The witness w for π_{MQ} is the solution $(\mathbf{s}, \mathbf{r})$ of the quadratic system $\tilde{\mathcal{P}}$, and statement x is $(\mathbf{t}, \mathbf{E}_1, \mathbf{E}_2, \mathcal{R}, \mathcal{P}_{UOV})$.

Communication cost. Given that the combined size of $\mathtt{seed}_{\mathbf{E}_1}$ and $\mathcal{G}(\mathbf{r})$ is 4λ, the signature size is $4\lambda + |\pi_{\mathcal{MQ}}|$.

NIZK proof for π_{MQ}. An efficient NIZK proof plays an important role in our constructions. Because the signature size relies on the proof size. There are various NIZK proofs are available for the MQ problem [18], [27]–[30], [41]–[44]. In the below, we present the proof size for the 128-bit security level. Table 3 reflects that the VOLEitH-based NIZK proof provides smaller proof sizes, resulting in a short signature for MQuBS. There are two such constructions discussed in the literature, both based on VOLEitH NIZK proofs [18], [30]. In our case, the only difference is in the parameters: our quadratic system has $n + m$ variables and m equations. The corresponding proof size is presented in Table 5. For further details on the NIZK proof and verification algorithm, we refer to [18].

TABLE 3. PROOF SIZE FOR VARIOUS NIZK PROOF FOR MQ

ZKP	Five pass [26]	with helper [27]	MPCitH [42]	TCitH [44]	VOLEitH [18]	VOLEitH [30]
Proof size (KB)	29	14	6.9	4.2	2.6	3.6

Algorithm 5 Sign_3: Unblind the signature

```
$\sigma = (\mathtt{seed}_{\mathbf{E}_1}, \mathcal{G}(\mathbf{r}), \pi_{\mathcal{MQ}}) \leftarrow \mathsf{Sign}_3(\mathsf{vk}, \psi)$
Input: $\mathsf{vk}, \psi$
Output: $\mathbf{E}_1, \mathbf{E}_2, \mathcal{R}, \mathcal{G}(\mathbf{r}), \pi_{\mathcal{MQ}}$
if ($\mathcal{P}_{UOV}(\mathbf{s}) \neq \mathbf{b}$), abort.
Define $\tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) :: \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2)$
$x$ (statement) $\leftarrow (\mathbf{t}, \mathbf{E}_1, \mathbf{E}_2, \mathcal{P}_{UOV}, \mathcal{R})$;
$w$ (witness) $\leftarrow (\mathbf{s}, \mathbf{r})$
Relation :: $\mathbf{t} = \tilde{\mathcal{P}}(\mathbf{s}, \mathbf{r}) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$
$\pi_{\mathcal{MQ}} \leftarrow$ NIZK.Proof$_2$ ($x, w$, Relation)
return $\sigma = (\mathtt{seed}_{\mathbf{E}_1}, \mathcal{G}(\mathbf{r}), \pi_{\mathcal{MQ}})$
```

$0/1 \leftarrow \mathsf{MQuBS.Verify}(\mathsf{vk}, \sigma, \mathtt{msg})$: **Verification Phase.**
The verifier possesses the public key $\mathcal{P}_{UOV}$ and the message $\mathtt{msg}$. Upon receiving the signature σ, the verifier aims to determine whether it is the correct signature for the message $\mathtt{msg}$. Initially, the verifier $\mathbf{t}' \leftarrow \mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r}))$. Further, it expands the emulsifier matrices $\mathbf{E}_1$, $\mathbf{E}_2$, and the random quadratic map $\mathcal{R}$ from the seeds present in the signature. After completing these computations, it constructs the quadratic system $\tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2) = \mathbf{t}'$. Finally, it follows verifies the proof $\pi_{\mathcal{MQ}}$ by running $\mathsf{NIZK.Verify}_2$. We use the algorithm of [30] in this case.

Algorithm 6 MQuBS : Verification algorithm

```
$0/1 \leftarrow \mathsf{MQuBS.Verify}(\mathsf{vk}, \sigma, \mathtt{msg})$
$\mathbf{t}' \leftarrow \mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r}))$
Construct $\mathbf{E}_1, \mathbf{E}_2, \mathcal{R}$ from seed
Construct $\tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2)$
return NIZK.Verify$_2(\pi_{MQ}, \tilde{\mathcal{P}}, \mathbf{t}')$
```

4.2.3. Correctness of MQuBS.

We establish the correctness of our blind signature scheme in the following theorem.

Theorem 1 (Correctness). For properly executed $\mathsf{MQuBS}(n, m, q, r)$ protocol, if the signature on message $\mathtt{msg}$ is generated as σ. Then $\mathsf{MQuBS.Verify}(\mathsf{vk}, \sigma, \mathtt{msg}) = 1$ holds with probability $1 - \mathsf{negl}(\lambda)$.

Sketch. The correctness of MQuBS relies on the correctness of the UOV signature algorithm along with two NIZK proofs: π_{Com} and π_{MQ}. Together, these results establish the correctness of MQuBS. The proof of this theorem is presented in Appendix A.

5. Security Analysis

First, we define the UOV problem, which has been extensively studied and believe to be hard [13], [22]

Definition 4. UOV **Problem.** [13] Let $\mathcal{MQ}_{n,m,q}$ be the family of the random quadratic map; and $\mathcal{MQ}_{n,m,q}^{UOV}$ is the family of UOV-public polynomial map. The problem asks to distinguish between $\mathcal{P} \in \mathcal{MQ}_{n,m,q}$ or $\mathcal{P} \in \mathcal{MQ}_{n,m,q}^{UOV}$. Suppose $\mathcal{D}$ denotes the distinguisher algorithm for UOV, then the distinguishing advantage for $\mathcal{D}$ is defined as below.

$$\mathsf{Adv}_{UOV}^{(n,m,q)}(\mathcal{D}) = \Big| \Pr\left[\mathcal{D}(\mathcal{P}) = 1 \mid \mathcal{P} \leftarrow \mathcal{MQ}_{n,m,q}\right] - \Pr\left[\mathcal{D}(\mathcal{P}) = 1 \mid \mathcal{P} \leftarrow \mathcal{MQ}_{n,m,q}^{UOV}\right] \Big|$$

We define $\mathcal{P}$ as belonging to $\mathcal{MQ}_{n,m,q}^{UOV}$, such that for a secret linear subspace O, $\mathcal{P}(O) = 0$. It is widely believed that, for all probabilistic polynomial time distinguisher $\mathcal{D}$, the advantage $\mathsf{Adv}_{UOV}^{(n,m,q)}(\mathcal{D}) \leq \mathsf{negl}(\lambda)$. Now we propose a new hard problem called gWMQ problem. Beullens first introduced the WMQ problem for the Mayo digital signature scheme [24], and the gWMQ problem extends this to a more generalized form.

Definition 5. **Generalized Whipped Multivariate Quadratic** (gWMQ) **Problem**. Suppose $\mathcal{R}_1, \cdots \mathcal{R}_k \in \mathcal{MQ}_{n,m,q}$ are random polynomial maps, and $\mathbf{E}_{ij}^{\mathcal{R}_i}$ are $m \times m$ invertible matrices. Let $\mathbf{t} \in \mathbb{F}_q^m$ be the target vector. Now the problem asks to find $\mathbf{s}_1, \cdots, \mathbf{s}_k$, such that

$$\sum_{i=1}^{k} \mathbf{E}_{ii}^{\mathcal{R}_i} \mathcal{R}_i(\mathbf{s}_i) + \sum_{1 \leq i < j \leq k} \mathbf{E}_{ij}^{\mathcal{R}_i} \mathcal{D}\mathcal{R}_i(\mathbf{s}_i, \mathbf{s}_j) = \mathbf{t}.$$

In WMQ, Beullens used *cross terms* for higher values of k, since the algorithm for solving k-Sum algorithm is efficient for higher values of k [45]. However, in our case since $k = 2$, we do not need those cross terms. The gWMQ problem asks for a solution $(\mathbf{x}_1, \mathbf{x}_2)$ from the quadratic system $\mathbf{t} = \mathbf{E}_1\mathcal{R}_1(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}_2(\mathbf{x}_2)$ for a given $\mathbf{t}$, $\mathbf{E}_i$, and $\mathcal{P}_i$. Let's say $\mathcal{A}$ represents the adversary attempting to solve this problem. Then, the adversary's advantage against the problem is defined as follows.

$$\mathsf{Adv}_{gWMQ}^{(n,m,q)}(\mathcal{A}) = \left| \Pr \left[\begin{array}{c|} \mathbf{E}_1\mathcal{R}_1(\mathbf{s}_1) + \mathbf{E}_2\mathcal{R}_2(\mathbf{s}_2) = \mathbf{t} \\ \mathcal{R}_1, \mathcal{R}_2 \leftarrow \mathcal{MQ}_{n,m,q} \\ \left(\mathbf{E}_1, \mathbf{E}_2\right) \leftarrow \mathsf{GL}(m,q) \\ \mathbf{t} \leftarrow \mathbb{F}_q^m \\ (\mathbf{s}_1, \mathbf{s}_2) \leftarrow \mathcal{A}(\mathbf{t}, \mathcal{R}_1, \mathcal{R}_2, \\ \mathbf{E}_1, \mathbf{E}_2) \end{array} \right] \right|$$

To the best of our knowledge, there is no known cryptanalysis for Beullens's WMQ problem [24]. Consequently, we assume that for any probabilistic polynomial-time adversary $\mathcal{A}$, the advantage of $\mathcal{A}$ against the gWMQ problem satisfies $\mathsf{Adv}_{gWMQ}^{(n,m,q)} \leq \mathsf{negl}(\lambda)$.

5.1. Security of the commitment scheme

The following proof demonstrates that the security of the commitment scheme relies on the hardness of the gWMQ problem. We use the following instance of the gWMQ problem throughout our work: find a solution $(\mathbf{x}_1, \mathbf{x}_2)$ of a quadratic system $\mathbf{t} = \mathbf{E}_1\mathcal{R}_1(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}_2(\mathbf{x}_2)$ where $\mathbf{E}_1, \mathbf{E}_2, \mathcal{R}_1,$ and $\mathcal{R}_2$ are known.

Theorem 2 (Perfectly binding). The commitment scheme presented in Algorithm 1 is *perfectly binding* under the gWMQ assumptions. Specifically, if an adversary $\mathcal{A}$ has an advantage $\mathsf{Adv}_{\mathsf{COM}}(\mathcal{A})$ in the perfectly binding game, then there exist adversaries $\mathcal{B}$ that solve the gWMQ problem with advantages $\mathsf{Adv}_{\mathsf{gWMQ}}(\mathcal{B})$, such that

$$\mathsf{Adv}_{\mathsf{COM}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{gWMQ}}(\mathcal{B})$$

Proof: At first, we simplify Equation 2, and rewrite it as $\mathsf{Com}_{MQ} = \mathbf{E}_1'\mathcal{H}(\mathtt{msg}_1, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2'\mathcal{R}(\mathbf{r})$, where $\mathbf{E}_1' = \mathbf{E}_1^{-1}$, and $\mathbf{E}_2' = \mathbf{E}_1^{-1}\mathbf{E}_2$. Let $\mathtt{msg}_1$ and $\mathtt{msg}_2$ are two different messages for which the adversary $\mathcal{A}$ attempts to find a collision in the commitment. In addition, the adversary got $(\mathcal{R}_1, \mathbf{E}_{11}', \mathbf{E}_{12}')$ for the commitment on $\mathtt{msg}_1$; and $(\mathcal{R}_2, \mathbf{E}_{21}', \mathbf{E}_{22}')$ for the commitment on $\mathtt{msg}_2$, along with $\mathcal{G}(\mathbf{r}_1)$, and $\mathcal{G}(\mathbf{r}_2)$.

The goal of $\mathcal{A}$ is to find $(\mathbf{r}_1, \mathbf{r}_2)$ such that

$$\mathbf{E}_{12}'\mathcal{R}_1(\mathbf{r}_1) - \mathbf{E}_{22}'\mathcal{R}_2(\mathbf{r}_2) = \mathbf{E}_{11}'\mathcal{H}(\mathtt{msg}_1, \mathcal{G}(\mathbf{r}_1)) - \mathbf{E}_{21}'\mathcal{H}(\mathtt{msg}_2, \mathcal{G}(\mathbf{r}_2)). \tag{4}$$

Thus, the right-hand side of the above expression is known and can be computed by the adversary. Therefore, the adversary $\mathcal{A}$ fixes $\mathbf{t} \leftarrow \mathbf{E}_{11}'\mathcal{H}(\mathtt{msg}_1, \mathcal{G}(\mathbf{r}_1)) - \mathbf{E}_{21}'\mathcal{H}(\mathtt{msg}_2, \mathcal{G}(\mathbf{r}_2))$. Now rewrite Equation 4 in the following manner.

$$\mathbf{t} = \mathbf{E}_{12}'\mathcal{R}_1(\mathbf{r}_1) - \mathbf{E}_{22}'\mathcal{R}_2(\mathbf{r}_2) \tag{5}$$

Now, $\mathcal{A}$ invokes the adversary $\mathcal{B}$ which can break the gWMQ problem with the advantage $\mathsf{Adv}_{\mathsf{gWMQ}}(\mathcal{B})$. Then, $\mathcal{A}$ supplies $(\mathbf{t}, \mathbf{E}_{12}', \mathcal{R}_1, \mathbf{E}_{22}', \mathcal{R}_2)$ to $\mathcal{B}$. The adversary $\mathcal{B}$ computes $(\mathbf{r}_1, \mathbf{r}_2)$ and returns it to $\mathcal{A}$. This completes the proof.

□

5.2. Security proof for MQuBS

5.2.1. Blindness.

Theorem 3. For an adversary $\mathcal{A}$ which can subvert the blindness of MQuBS with advantage $\mathsf{Adv}_{\mathsf{BLND}}(\mathcal{A})$, there exists an adversary $\mathcal{B}$ that can distinguish simulated NIZK proofs from real ones with advantage $\mathsf{Adv}_{NIZK}(\mathcal{B})$, an adversary $\mathcal{C}$ that can break multivariate commitment scheme Com_{MQ} defined in Algorithm 1 with advantage $\mathsf{Adv}_{\mathsf{Com}_{MQ}}(\mathcal{C})$, and an adversary $\mathcal{D}$ that can break hash-based commitment scheme $\mathsf{Com}_{\mathsf{Hash}}$ with advantage $\mathsf{Adv}_{\mathsf{Com}_{\mathsf{Hash}}}(\mathcal{D})$, so that the following condition holds.

$$\mathsf{Adv}_{\mathsf{BLND}}(\mathcal{A}) \leq \mathsf{Adv}_{NIZK}(\mathcal{B}) + \mathsf{Adv}_{\mathsf{Com}_{MQ}}(\mathcal{C}) + \mathsf{Adv}_{\mathsf{Com}_{\mathsf{Hash}}}(\mathcal{D})$$

Proof: Before we proceed, let's replace two NIZKs π_{Com}, π_{MQ} and the translated-UOV signature schemes with their respective simulations. This alteration allows the adversary to differentiate this scenario with an advantage denoted as $\mathsf{Adv}_{NIZK}(\mathcal{B})$. At this moment, the two proofs π_{Com}, π_{MQ} are become independent from $\mathtt{msg}$ and $(\mathbf{s}, \mathbf{r})$. We are assuming that the adversary has access to values $\mathcal{G}(\mathbf{r})$, and $\mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r}))$. Therefore, it is enough to show that the committed value C and the RHS of $\mathbf{t} = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$ does not leak any information about the message $\mathtt{msg}$, and $(\mathbf{s}, \mathbf{r})$. Now the adversary can break the $\mathsf{Com}_{\mathsf{Hash}}$ commitment scheme with the advantage of $\mathsf{Adv}_{\mathsf{Com}_{\mathsf{Hash}}}$, which implies the adversary $\mathcal{A}$ can learn about the message $\mathtt{msg}$, and $(\mathbf{s}, \mathbf{r})$ with the same advantage.

Now we move to the second part. The Theorem 2 already established that the commitment scheme Com_{MQ} ensures perfecty hiding. Thus the adversary can break the commitment Com_{MQ} with the advantage $\mathsf{Adv}_{\mathsf{Com}_{MQ}}$. We present a more detailed idea. Now, the crucial aspect remaining to demonstrate is that $\mathbf{E}_1^{-1}(\mathcal{H}(\mathtt{msg}, \rho) - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$ does not disclose any information about the message. Essentially, the adversary selects a pair $(\rho^*, \mathtt{msg}^*)$ from one interaction and aims to associate it with the blind message $\mathbf{b}$ from a separate interaction. As previously noted, the signature σ offers an advantage $\mathsf{Adv}_{NIZK}(\mathcal{B})$ for linking an equivalent $\mathbf{b}^*$ to $\mathtt{msg}^*$. Subsequently, the adversary randomly chooses $\mathbf{r}^*$ and computes $\mathcal{R}(\mathbf{r}^*)$. Given the randomness of $\mathbf{r}^*$ and $\mathcal{R}$, $\mathcal{R}(\mathbf{r}^*)$ remains indistinguishable from a random distribution. Therefore, the adversary's ability to distinguish the blind signature $\mathbf{b}^* = \mathbf{E}_1^{-1}(\mathcal{H}(\mathtt{msg}^*, \rho^*) - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$ is determined by the advantage $\mathsf{Adv}_{\mathsf{Com}_{MQ}}(\mathcal{C})$. This implies that the adversary can compute any predicate of $\mathtt{msg}^*$ from $\mathbf{b}^*$ with an advantage of $\mathsf{Adv}_{\mathsf{Com}_{MQ}}$. □

5.2.2. One More Unforgeability (OMUF). At first, we define of EUF-CMA, and EUF-KO-security of the underlying signature scheme $\mathsf{SIG} = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Verify})$.

Definition 6 *(*EUF-CMA*-security).* The underlying signature scheme SIG is considered EUF-CMA-secure if any polynomial-time adversary $\mathcal{A}$ has only a negligible advantage in the EUF-CMA game, defined as follows.

$$\mathsf{Adv}_{\mathsf{SIG}}(\mathcal{A}) = \Pr\left[\begin{array}{l|l} \mathsf{Verify}(\mathsf{vk}, \mathtt{msg}^*, \sigma^*) = 1 & (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ \mathtt{msg}^* \text{ not queried} & (\sigma, \mathtt{msg}^*) \leftarrow \mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk}) \end{array}\right]$$

The notation $\mathcal{A}^{\mathsf{Sign}(\mathsf{sk}, \cdot)}(\mathsf{vk})$ signifies that the adversary $\mathcal{A}$ has access to the signing oracle $\mathcal{O}_{\mathsf{Sign}}(\mathsf{sk}, \cdot)$. Also, the adversary does not query on the message $\mathtt{msg}^*$.

Definition 7 *(*EUF-KO*-security).* A signature scheme SIG is EUF-KO-secure if any polynomial-time adversary $\mathcal{A}$ has a negligible advantage in the EUF-KO game. It is defined as follows.

$$\mathsf{Adv}_{\mathsf{KO}}(\mathcal{A}) = \Pr\left[\begin{array}{l|l} \mathsf{Verify}(\mathsf{vk}, \mathtt{msg}^*, \sigma^*) = 1 & (\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KeyGen}(1^\lambda) \\ & (\sigma, \mathtt{msg}^*) \leftarrow \mathcal{A}(\mathsf{vk}) \end{array}\right]$$

A key difference between the EUF-CMA game and EUF-KO is that the adversary in EUF-KO does not have access to the signing oracle.

```
Signing Phase: Translated-UOV
Ingredients. Three hash functions, and a XOF
$\mathcal{G} : \{0,1\}^* \to \{0,1\}^{2\lambda}, \mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$
$\mathcal{K} : \{0,1\}^* \to \{0,1\}^{2\lambda}$, and a XOF
Inputs. $\mathtt{msg} \in \{0,1\}^*, \mathsf{sk} = (O)$,
$\mathsf{vk} = \mathcal{P}_{UOV},\ \mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$.
Computes $\rho = \mathcal{G}(\mathbf{r}),\ \mathbf{t} \leftarrow \mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r}))$.
$\mathtt{seed}_{\mathbf{E}_1} \leftarrow \mathcal{K}(\mathtt{msg}||\mathcal{G}(\mathbf{r})),\ \mathtt{seed}_{\mathbf{E}_2} \leftarrow \mathcal{K}(\mathtt{msg}||\mathtt{seed}_{\mathbf{E}_1})$,
$\mathtt{seed}_{\mathcal{R}} \leftarrow \mathcal{K}(\mathtt{msg}||\mathtt{seed}_{\mathbf{E}_2}),\ \mathbf{E}_1 \leftarrow \mathsf{XOF}(\mathtt{seed}_{\mathbf{E}_1})$
$\mathbf{E}_2 \leftarrow \mathsf{XOF}(\mathtt{seed}_{\mathbf{E}_2}),\ \mathcal{R} \leftarrow \mathsf{XOF}(\mathtt{seed}_{\mathcal{R}})$
$\mathbf{b} \leftarrow \mathbf{E}_1^{-1}(\mathbf{t} - \mathbf{E}_2\mathcal{R}(\mathbf{r})),\ \mathbf{s} \leftarrow \mathcal{P}_{UOV}^{-1}(\mathbf{b})$
$\sigma = (\mathbf{s}, \rho, \mathtt{seed}_{\mathbf{E}_1}, \mathtt{seed}_{\mathbf{E}_2}, \mathtt{seed}_{\mathcal{R}})$
```

Figure 4. Translated-UOV Signature

We used UOV as the underlying signature scheme [13], [22]. The existential unforgeability (EUF-CMA) in the random oracle model of the UOV signature scheme has been studied extensively in the literature [39], [41], [46], [47]. Note that, we have modified the UOV signature to employ it in the blind signature settings. We call this modified signature as *translated-UOV*, and we denote the signature scheme as $\mathsf{tran\text{-}UOV}_{(q,n,m)}$, where (q, n, m) is the algorithm's parameter. We detail the signature algorithm in Figure: 4, and the verification algorithm in Figure: 5. The key generation algorithm of tran-UOV is the same as the key generation algorithm of the UOV-signature scheme [22].

Lemma 1 demonstrates that $\mathsf{tran\text{-}UOV}(q, n, m)$ is EUF-CMA-secure. The security of tran-UOV relies on the hardness of the UOV problem and the gWMQ problem. We use a similar proof style like the UOV-signature scheme [22]. Several UOV-based signature schemes [23]–[25] follow this approach.

Lemma 1 *(*EUF-CMA*-security of* **tran-UOV***).* The translated UOV-signature scheme is one-more-unforgeable under the UOV and gWMQ assumptions when $\mathcal{G}$ and $\mathcal{H}$ are modeled as random oracles. Basically for an adversary $\mathcal{A}$ in the signature forgery game that makes upto q_h random oracle queries and q_s signing oracle queries, and has advantages $\mathsf{Adv}_{SIG}(\mathcal{A})$, then there exists an adversary $\mathcal{B}$ that distinguishes UOV public key with advantage $\mathsf{Adv}_{UOV}(\mathcal{B})$ and an adversary $\mathcal{C}$ that solve the gWMQ with the advantage $\mathsf{Adv}_{gWMQ}(\mathcal{C})$ in time $t + (1 + q_s + q_h) \cdot \mathrm{poly}(q, n, m)$ so that,

$$\mathsf{Adv}_{SIG}(\mathcal{A}) \leq \mathsf{Adv}_{UOV}(\mathcal{B}) + q_h \cdot \mathsf{Adv}_{gWMQ}(\mathcal{C}) + \frac{q_s(q_h + q_s)}{2^{2\lambda}} + \frac{1}{q^m}$$

To prove Lemma 1, we need two more lemmas. Lemma 2 gives a reduction from the EUF-CMA-security of tran-UOV to its EUF-KO-security. Further, we require Lemma 3. It presents a reduction from UOV and gWMQ problem to the EUF-KO-security of tran-UOV signature scheme. Combining these two lemmas, we establish the claim of Lemma 1.

```
Verification Phase: Translated-UOV
Ingredients: Three hash functions, and a XOF
$\mathcal{G} : \{0,1\}^* \to \{0,1\}^{2\lambda}, \mathcal{H} : \{0,1\}^* \to \mathbb{F}_q^m$,
$\mathcal{K} : \{0,1\}^* \to \{0,1\}^{2\lambda}$, and a XOF
Inputs. $\mathbf{r}, \texttt{msg}, \sigma, \mathsf{vk} = \mathcal{P}_{UOV}$
Computes $\rho = \mathcal{G}(\mathbf{r}), \mathbf{t} \leftarrow \mathcal{H}(\texttt{msg}, \mathcal{G}(\mathbf{r}))$
If $\rho$ does not match, then abort.
If any seed is not matching, then abort.
Compute $\mathbf{t}_{temp} \leftarrow \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$
Return 1, if $\mathbf{t}_{temp} = \mathbf{t}$, else 0.
```

Figure 5. Verification of the translated-UOV Signature

Lemma 2 (*EUF-CMA *to* EUF-KO *security). For a PPT adversary $\mathcal{A}$ which runs against the EUF-CMA security game of the translated-UOV signature scheme with parameter (n, m, q) in the random oracle model and it makes q_s signing oracle and q_h random oracle queries. Then there exists an adversary $\mathcal{B}$ against the EUF-KO security of the translated-UOV signature, which runs in time $t + O(q_s + q_h)\text{poly}(n, m, q)$ so that the following conditions will hold.

$$\mathsf{Adv}_{\mathsf{SIG}}(\mathcal{A}) \leq \mathsf{Adv}_{\mathsf{KO}}(\mathcal{B}) + \frac{q_s(q_s + q_h)}{2^{2\lambda}}$$

Lemma 3 (UOV and gWMQ to* EUF-KO-*security). Let $\mathcal{A}$ be a EUF-KO adversary that runs in time t against the $\mathsf{tran\text{-}UOV}_{(q,n,m)}$ signature in the ROM and it makes q queries to the random oracle. Then there exists an adversaries $\mathcal{B}$ against the $\mathsf{UOV}_{n,m,q}$ problem and $\mathcal{C}$ against the $\mathsf{gWMQ}_{n,m,q}$ problem, that runs in time $t + O(1+q_h)$ poly(q, n, m) so that, the following condition hold.

$$\mathsf{Adv}_{\mathsf{KO}}(\mathcal{A}) \leq \mathsf{Adv}_{UOV}(\mathcal{B}) + (1+q_h)\mathsf{Adv}_{gWMQ}(\mathcal{C}) + \frac{1}{q^m}$$

The proof of three lemmas can be found in [22]. The EUF-CMA-security proof of many UOV-based signature schemes [23], [25] [24]follows the same approach. We understood that the three lemmas collectively prove the underlying signature is EUF-CMA-secure in the ROM when $\mathcal{G}$, $\mathcal{K}$, and $\mathcal{H}$ are implemented using cryptographically secure hash functions like SHA-2 or SHA-3 [33]. Therefore, we can say that MQuBS is OMUF, since the basic signature scheme is EUF-CMA secure and the soundness and zero-knowledge property of NIZK proofs.

Theorem 4. The EUF-CMA security of the underlying signature scheme and the soundness of NIZK proofs jointly implies the one-more unforgeability of MQuBS. In particular, for any adversary $\mathcal{A}$ that makes at most q_{RO} oracle queries in the OMUF game, there exist adversaries $\mathcal{B}$ and $\mathcal{C}$ such that $\mathcal{B}$ challenges the soundness of the NIZK and $\mathcal{C}$ targets the unforgeability of the basic signature scheme. The advantage of $\mathcal{A}$ in breaking the one-more unforgeability of the blind signature scheme is bounded by:

$$\mathsf{Adv}_{\mathsf{MQuBS}}(\mathcal{A}) \leq (q_{RO} + 1)\mathsf{Adv}_{\mathsf{SND}}(\mathcal{B}) + \mathsf{Adv}_{\mathsf{SIG}}(\mathcal{C}).$$

where,
-$\mathsf{Adv}_{\mathsf{MQuBS}}(\mathcal{A})$ is the advantage of an adversary $\mathcal{A}$ in breaking the one-more unforgeability of MQuBS.
- $\mathsf{Adv}_{\mathsf{SND}}(\mathcal{B})$ is the advantage of a adversary $\mathcal{B}$ in breaking the soundness of the NIZK proof system.
- $\mathsf{Adv}_{\mathsf{SIG}}(\mathcal{C})$ is the advantage of a adversary $\mathcal{C}$ in breaking the unforgeability of the basic signature scheme.

Proof: We use a hash list $\mathsf{List}_{\mathsf{Com}}$ in the reduction phase. When $\mathcal{A}$ queries for a signature, we verify the NIZK proof π_{Com} first. Then we look into the list $\mathsf{List}_{\mathsf{Com}}$ and find the values $(\texttt{msg}, \mathbf{r})$. The verification of the proof π_{Com} ensures two things, one is the well-formedness of $\mathbf{b}$ and another is the proof of the commitment scheme. The adversary can bluff with the advantage $\mathsf{Adv}_{\mathsf{SND}}(\mathcal{B})$. So, we can assume that the relation in the proof π_{Com} is correct. Now, we query the signing oracle from the unforgeability game with the message $(\texttt{msg}, \mathbf{r})$ and receive the corresponding signature $(\rho, \mathbf{s})$.

The adversary manages to generate $(q_{RO} + 1)$ signed messages $(\texttt{msg}_i, \rho_i, \pi_{MQ_i})$ for $i = 1, \ldots, q_{RO} + 1$. We can verify whether each message was included in any of the earlier at most q_{RO} signing queries. There must be at least one signed message, $(\texttt{msg}^*, \rho^*, \pi^*_{MQ})$, for which $\texttt{msg}^*$ is not previously queried. Using a NIZK extractor, we can extract a witness $(\mathbf{s}^*, \mathbf{r}^*)$ for the relation $\mathbf{E}_1\mathcal{P}(\mathbf{s}^*) + \mathbf{E}_2\mathcal{R}^*(\mathbf{r}^*) = \mathcal{H}(\rho^*, \texttt{msg}^*)$. This gives us a forged signature $(\rho, \mathbf{s})$ for the message $(\texttt{msg}, \mathbf{r})$, which was not part of any previous queries in the unforgeability game for the underlying signature. □

5.3. Algebraic Cryptanalysis

In this section, we describe possible algebraic attacks against MQuBS.

5.3.1. Attack on the Com_{MQ}.

We proved the security of our commitment scheme relies on the gWMQ problem. To find a collision in our commitment scheme, an attacker tries to solve Equation 5 which is an instance of gWMQ problem. To our best knowledge, there is no better algorithm known for the WMQ problem [24].

Since the quadratic mappings are generated randomly, and the inversion of any random quadratic map is challenging, therefore computing $(\mathbf{r}_1, \mathbf{r}_2)$ is presumed to be as hard as solving the MQ problem. To find $(\mathbf{r}_1, \mathbf{r}_2)$ using the MQ-solving algorithm, the attacker will perform the following steps.

The attacker starts by selecting a random m-tuple $\mathbf{r}_{\mathsf{adv}}$, followed by the computation of $\mathcal{R}_1(\mathbf{r}_{\mathsf{adv}})$. The only remaining unknown in the Equation 5 is $\mathcal{R}_2(\mathbf{r}_2)$. Consequently, the adversary must tackle the MQ problem to ascertain $\mathbf{r}_2$.

The alternative method to find a solution $(\mathbf{r}_1, \mathbf{r}_2)$ is to use an algorithm that solves the k-SUM problem. It's worth noting that $\mathbf{t}$ represents the sum of two functions with independent inputs. The adversary simplifies the task of finding a pre-image of the quadratic map to an instance of the k-sum problem.

Initially, the attacker constructs two lists, List_1 and List_2. Here, these lists has the evaluations of $\mathbf{E}'_{12}\mathcal{R}_1(\mathbf{x})$, and $\mathbf{E}'_{22}\mathcal{R}_2(\mathbf{x})$ respectively. Subsequently, the adversary searches for one value in each list to ensure that their sum equals $\mathbf{t}$. This task can be done in $O(q^m)$ time using the Wagner k-tree algorithm [45].

5.3.2. Beullens's [17] attack is not applicable. Since in our case, we have random polynomials, so the Equation 5 remains quadratic. Therefore, Beullens's polar form attack [17] can not convert the quadratic system to a linear system.

5.3.3. Direct Attack. The most fundamental attack on UOV and many other multivariate cryptosystems is the direct attack. Here, the attacker picks a message $\mathtt{msg}^*$ and a salt $\mathcal{G}(\mathbf{r})$, computes their hash value $\mathbf{t}$, and then focuses on uncovering a preimage $\mathbf{s}, \mathbf{r}$ for $\mathbf{t}$ using quadratic system-solving techniques under the quadratic system $\tilde{\mathcal{P}}$. At first, the attacker converts the underdetermined system to a system with $m' = m-1$ equations in $n' = m-1$ variables using the approach developed by Thomae and Wolf [48]. Then it runs the hybrid WiedemannXL algorithm [49] to find a solution for the quadratic system. The time complexity of this algorithm is as follows.

$$\min_k q^k \cdot 3\binom{n'-k+d_{n'-k,m}}{d_{n'-k,m}}^2 \cdot \binom{n'-k+2}{2}(2r^2+r)$$

and represents the expenditure associated with the direct assault on UOV. Here, $d_{N,M}$ denotes the operational degree of XL, which is defined as the smallest $d > 0$ such that the coefficient of t^d in the power series expansion of

$$\frac{(1-t^2)^M}{(1-t)^{N+1}}$$

is non-positive.

5.3.4. Min-Rank Attack. The attacker can use a min-rank algorithm to find the secret of the quadratic map $\tilde{\mathcal{P}}$. In our case, the secret oil space of the quadratic map $\tilde{\mathcal{P}}$ is $\tilde{O} = \{(\mathbf{s}, \mathbf{r}) : \mathbf{s}, \mathbf{r} \in \mathbb{F}_q^m\}$. The dimension of the secret oil space $\tilde{O}$ is $2m$.

In the MinRank attack, the adversary aims to find a linear combination Q of the public polynomials represented by matrices $P_1, \cdots, P_m$ in a quadratic system $\mathcal{P}$, such that the rank of Q does not exceed a specified threshold r. Mathematically, this can be expressed as:

$$Q = \sum_{i=1}^{m} c_i \cdot P_i$$

where c_i are the coefficients chosen by the adversary, and the objective is to minimize rank(Q) subject to rank$(Q) \leq r$. Various methods have been developed to address the MinRank problem, ranging from linear algebraic techniques to specialized algorithms such as the Kipnis-Shamir method and Minors Modeling [50], [51].

5.3.5. Intersection Attack. The intersection attack builds upon the principles underlying the Kipnis-Shamir method and integrates a system-solving strategy akin to the reconciliation attack [52]. This attack is used to find k vectors within the secret oil space $\tilde{O}$, defined as the collection of vectors $\mathbf{u}$ in $\mathbb{F}_q^n$ satisfying $\tilde{\mathcal{P}}(\mathbf{u}) = \mathbf{0}_m$. By solving a system of quadratic equations, the attack endeavours to locate a vector common to the intersections of $\mathbf{M}_i O$ for k different matrices $\mathbf{M}_i$. Successful execution of the attack relies on the existence of a non-empty intersection, which occurs when $n < \frac{2k-1}{k-1}m$. The primary computational effort involves solving a random system of equations with $M = \binom{k+1}{2}m - \binom{k}{2}$ equations in $N = kn - (2k-1)m$ variables. In the context of UOV with $k = 3$, the certainty of finding a non-trivial intersection is not guaranteed, thus the effectiveness of the attack may vary. However, analysis suggests that for these parameters, the intersection is non-trivial with a probability of $1/(q-1)$. Consequently, the attack may need to be repeated approximately $q - 1 = 15$ times on average, rendering it more cost-effective than a single attack employing $k = 2$.

6. Parameter Selection

The security of MQuBS fundamentally relies on several key aspects. First, solving the quadratic system should be hard. Our scheme employs two quadratic systems: a random quadratic system $\mathcal{R}$ and the UOV quadratic system. This leads to two critical observations.

1. Finding a solution in the m variables and m constraints random quadratic system should be difficult.

2. Inverting the UOV map should be computationally hard, or equivalently, retrieving an oil vector in the secret oil space should be hard.

6.1. Communication Cost

We now recall the communication cost for each round of interaction. Table 2 summarizes the communication costs incurred during each round of interaction between the user, signer, and verifier. The size of β, which is the output of the Sign_1 algorithm, is $2\lambda + m\log q + |\pi_{\mathsf{Com}}|$. The signer sends an $m \log q$-bit string to the user as the blind signature, and the final signature size is $|\sigma| = 4\lambda + |\pi_{MQ}|$.

NIZK proof size for π_{Com}. As per the security level λ, we pick parameters from [19] for a hash-based NIZK proof to build π_{Com}. Note that, this proof size will not be added with the final signature.

Size of π_{MQ}, *NIZK Proof for the MQ problem:* The proof π_{MQ} is included in the final signature. To minimize the signature size, we need a small size NIZK proof. For this purpose, we employ VOLEitH-based constructions. We compute the proof size using the formula presented in the Subsection 5.3 of [30]. Unlike the standard case where the number of variables equals the number of equations, in our scenario, there are $n+m$ variables and m equations present in the quadratic system. This increases the proof size by $nr \log q + |\pi|$ bits over the proof size $|\pi|$ given in [30]. [5] The term $nr \log q$ arises because each iteration of the proof π_{MQ} involves an additional n variables, with r being the total number of rounds repeated to boost the soundness error.

6.2. Parameters Selection

We follow the security level (SL) definitions provided by the National Institute of Standards and Technology (NIST) [53]. First, we set the parameters for the underlying signature scheme based on the security parameter λ. After configuring the UOV parameters and constructing the quadratic system $\tilde{\mathcal{P}}$, we then design the parameters for the NIZK proof π_{MQ}.

[5] The $|\pi|$ denote the proof size of [30].

TABLE 4. UOV- PARAMETERS ACCORDING TO [22]

UOV	NIST SL	n	m	q	$\lvert\sigma_{UOV}\rvert$ (B)	|sk| (B)	|vk| (KB)
`uov-Ip`	1	112	44	256	128	48	43.576
`uov-Is`	1	160	64	16	96	48	66.576
`uov-III`	3	184	72	256	200	48	189.232
`uov-V`	5	244	99	256	260	48	446.992

TABLE 5. PROOF SIZE (KB) FOR OUR CASE

NIST SL	Parameters (n, m, q) Table 6.2	MQDSS [41] (KB)	with Helper (KB) [27]	MPCitH (KB) [28]	TCitH (KB) [44]	VOLEitH (KB) [30]
1	(112,44,256)	85.184	22.262	9.061	5.9	5.455
1	(160,64,16)	81.664	21.212	8.261	5.369	4.968
3	(184,72,256)	198.816	74.288	19.897	13.529	12.535
5	(244,96,256)	367.488	170.944	35.053	24.5371	22.784

Parameters for UOV-signature. Based on the security parameter λ, we first configure the parameters for the underlying UOV signature scheme as outlined in the UOV specifications document [22]. Table 6.2 presents the parameters for UOV across different security levels: 128-bit (SL-1), 192-bit (SL-3), and 256-bit (SL-5). Specifically, λ determines the values of q, n, and m, which represent the field size, the number of variables, and the number of homogeneous quadratic equations needed to construct the UOV public key $\mathcal{P}_{UOV}$, respectively.

Suppose $\lambda = 128$ bit, then as per `uov-Is` of the Table 6.2, $n = 160$, $m = 64$, and $q = 16$. For 128-bit security level, the size of π_{Com} is 47KB. Hence the size of blind message is $2 * 128 + 64 * \log 16 + |\pi_{\mathsf{Com}}|$-bits. This leads to the size of a blind message β is 47.288KB. Based on the parameters `uov-Is` in Table 6.2, the size of ψ is 96 bytes. To determine the total size of the final signature σ, we must also calculate the size of π_{MQ}.

Parameters for NIZK proof π_{MQ}. Now we turn our attention to the NIZK. The homogeneous multivariate quadratic system $\tilde{\mathcal{P}}$ has $(n+m)$ variables and m equations and defined over $\mathbb{F}_q$. Since $\mathcal{P}_{UOV}$ has n variables and $\mathcal{R}$ has m variables. The user prepared a NIZK proof π_{MQ} which involves the quadratic system $\tilde{\mathcal{P}}$. The parameters for π_{MQ} are underlying field size, number of variables, number of constraints, and the number of repetition to achieve the soundness property of the NIZK. Earlier, we fixed field size, number of variables, number of constraints. According to the security level number of repetition r.

The NIZK proof π_{MQ} for our multivariate blind signature can be implemented using several techniques, including Sakumoto et al.'s five-round NIZK [26], Beullens's *helper* approach [27], the MPCitH framework [28], the TCitH paradigm [44], and the VOLEitH technique [30]. To compute the proof size of π_{MQ}, we follow the formulas presented in each of these references. The table 5 illustrates the proof size for various security levels.

6.3. Size of the Keys and Signature

Keys sizes for MQuBS. In the MQuBS.KeyGen algorithm (see Algorithm 2), we noted that the UOV key generation algorithm is used to produce the public and secret keys. As a result, the key sizes are determined entirely by the UOV signature algorithm. Thus, Table 6.2 also reflects the key sizes for MQuBS. Table 6 shows the key and signature sizes of the MQuBS blind signature algorithm for different security levels.

TABLE 6. KEY AND SIGNATURE SIZES FOR MQuBS AT VARIOUS SECURITY LEVELS.

NIST SL	|sk| (B)	|vk| (B)	$\mathcal{U} \to \mathcal{S}$ (B)	$\mathcal{S} \to \mathcal{U}$ (B)	$\lvert\sigma\rvert$ (KB)
`MQuBS.SL-1p`	48	43.576	352	896	5.5
`MQuBS.SL-1s`	48	66.576	256	640	5
`MQuBS.SL-3`	48	189.232	576	1472	12.65
`MQuBS.SL-5`	48	446.992	768	1952	23

Size of MQuBS *final signature* σ. The final signature has a seed, $\mathcal{G}(\mathbf{r})$, and a NIZK proof π_{MQ}. Therefore, the signature size is $4\lambda + |\pi_{MQ}|$. According to Table 5, the most efficient NIZK proof has a proof size of 4.968KB for our parameters. Hence, the size of the final signature according to the formula $4\lambda + |\pi_{MQ}|$ is 5KB (approximately) for SL-1.

7. Conclusion

In this work, we investigated multivariate PQ BS schemes. Currently, the most efficient PQ blind signatures are based on the lattice assumption. There is very little exploration for other quantum hard problems in the context of designing BS. So, we decided to use multivariate assumptions. We are the first to adapt Fischlin's framework in multivariate settings. Our construction used the well-studied UOV signature as the underlying signature. The UOV signature is also submitted in the NIST additional round PQ-signature standardization process [14]. We established that it offers *blindness*, and *one-more unforgeable*. We also introduced the gWMQ problem. The security of our construction relies on the hardness of UOV and the gWMQ problem.

MQuBS used an efficient and shorter NIZK proof for a solution to the MQ problem. This eliminated one of the major shortcomings of the lattice-based blind signatures. We gave a shorter signature size of 5KB for a 128-bit PQ security level. We compared our results with the state-of-the-art round-optimal post-quantum blind signatures. The lattice-based blind signature proposed by Agrawal *et al.* [12] offered a 45KB signature scheme, while an upgraded version proposed by Beullens *et al.* [5] offered a 22KB signature size. This concludes that our design MQuBS offers the shortest signature among PQ round-optimal blind signatures.

Acknowledgments

N.S. thanks the funding support from DST-SERB (CRG/2020/45 + JCB/2022/57) and N. Rama Rao Chair. A.G. is supported by a TCS research fellowship. A part of this work was done while D.D. visited IIT Kanpur.

References

[1] D. Chaum, "Blind signatures for untraceable payments," in *Advances in Cryptology: Proceedings of Crypto 82*. Springer, 1983, pp. 199–203.

[2] D. Chaum and T. P. Pedersen, "Wallet databases with observers," in *Annual international cryptology conference*. Springer, 1992, pp. 89–105.

[3] S. Brands, "Untraceable off-line cash in wallet with observers," in *Advances in Cryptology—CRYPTO'93: 13th Annual International Cryptology Conference Santa Barbara, California, USA August 22–26, 1993 Proceedings 13*. Springer, 1994, pp. 302–318.

[4] A. Szepieniec and B. Preneel, "New techniques for electronic voting," in *USENIX Journal of Election Technology and Systems (JETS)*, 2015.

[5] W. Beullens, V. Lyubashevsky, N. K. Nguyen, and G. Seiler, "Lattice-based blind signatures: Short, efficient, and round-optimal," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 16–29.

[6] M. Fischlin, "Round-optimal composable blind signatures in the common reference string model," in *Annual International Cryptology Conference*. Springer, 2006, pp. 60–77.

[7] F. Denis, F. Jacobs, and C. Wood, "RFC 9474 RSA Blind signatures," 2023.

[8] G. Fuchsbauer and M. Wolf, "Concurrently secure blind schnorr signatures," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2024, pp. 124–160.

[9] G. Fuchsbauer, A. Plouviez, and Y. Seurin, "Blind schnorr signatures and signed elgamal encryption in the algebraic group model," in *Advances in Cryptology–EUROCRYPT 2020: 39th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Zagreb, Croatia, May 10–14, 2020, Proceedings, Part II 30*. Springer, 2020, pp. 63–95.

[10] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[11] J. Proos and C. Zalka, "Shor's discrete logarithm quantum algorithm for elliptic curves," *arXiv preprint quant-ph/0301141*, 2003.

[12] S. Agrawal, E. Kirshanova, D. Stehlé, and A. Yadav, "Practical, round-optimal lattice-based blind signatures," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 39–53.

[13] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced oil and vinegar signature schemes," in *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 1999, pp. 206–222.

[14] National Institute of Standards and Technology (NIST), "NIST announces second round of post-quantum cryptography digital signature standardization," 2024, accessed: October 24, 2024. [Online]. Available: https://csrc.nist.gov/news/2024/pqc-digital-signature-second-round-announcement

[15] A. Petzoldt, A. Szepieniec, and M. S. E. Mohamed, "A practical multivariate blind signature scheme," in *Financial Cryptography and Data Security: 21st International Conference, FC 2017, Sliema, Malta, April 3-7, 2017, Revised Selected Papers 21*. Springer, 2017, pp. 437–454.

[16] D. S. Johnson and M. R. Garey, *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman, 1979.

[17] W. Beullens, "Multivariate Blind Signatures Revisited," Cryptology ePrint Archive, Paper 2024/720, 2024.

[18] C. Baum, W. Beullens, S. Mukherjee, E. Orsini, S. Ramacher, C. Rechberger, L. Roy, and P. Scholl, "One Tree to Rule Them All: Optimizing GGM Trees and OWFs for Post-Quantum Signatures," Cryptology ePrint Archive, Paper 2024/490, 2024.

[19] I. Giacomelli, J. Madsen, and C. Orlandi, "ZKBoo: Faster Zero-Knowledge for boolean circuits," in *25th usenix security symposium (usenix security 16)*, 2016, pp. 1069–1083.

[20] M. Chase, D. Derler, S. Goldfeder, C. Orlandi, S. Ramacher, C. Rechberger, D. Slamanig, and G. Zaverucha, "Post-quantum zero-knowledge and signatures from symmetric-key primitives," in *Proceedings of the 2017 acm sigsac conference on computer and communications security*, 2017, pp. 1825–1842.

[21] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The sphincs+ signature framework," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 2129–2146.

[22] W. Beullens, M.-S. Chen, J. Ding, B. Gong, M. J. Kannwischer, J. Patarin, B.-Y. Peng, D. Schmidt, C.-J. Shih, C. Tao, and B.-Y. Yang, "UOV: Unbalanced Oil and Vinegar Algorithm Specifications and Supporting Documentation Version 1.0," 2018. [Online]. Available: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/UOV-spec-web.pdf

[23] A. Ganguly, A. Karmakar, and N. Saxena, "VDOO: A short, fast, post-quantum multivariate digital signature scheme," in *International Conference on Cryptology in India*. Springer, 2023, pp. 197–222.

[24] W. Beullens, "MAYO: practical post-quantum signatures from Oil-and-Vinegar maps," in *Selected Areas in Cryptography: 28th International Conference, Revised Selected Papers*, 2022, pp. 355–376.

[25] H. Furue, Y. Ikematsu, F. Hoshino, Y. Kiyomura, T. Saito, and T. Takagi, "QR-UOV," 2023.

[26] K. Sakumoto, T. Shirai, and H. Hiwatari, "Public-key identification schemes based on multivariate quadratic polynomials," in *Advances in Cryptology–CRYPTO 2011: 31st Annual Cryptology Conference, Santa Barbara, CA, USA, August 14-18, 2011*. Springer, 2011, pp. 706–723.

[27] W. Beullens, "Sigma protocols for MQ, PKP and SIS, and fishy signature schemes," Cryptology ePrint Archive, Paper 2019/490, 2019.

[28] T. Feneuil and M. Rivain, "MQOM: MQ on my mind-algorithm specifications and supporting documentation. Version 1.0-31 May 2023."

[29] R. Benadjila, T. Feneuil, and M. Rivain, "MQ on my Mind: Post-Quantum Signatures from the Non-Structured Multivariate Quadratic Problem," *Cryptology ePrint Archive*, 2023.

[30] D. Bui, "Shorter VOLEitH Signature from Multivariate Quadratic," Cryptology ePrint Archive, Paper 2024/465, 2024.

[31] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *International conference on applied cryptography and network security*. Springer, 2005, pp. 164–175.

[32] W. Beullens, "Breaking rainbow takes a weekend on a laptop," in *Annual International Cryptology Conference*. Springer, 2022, pp. 464–479.

[33] M. J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," 2015.

[34] D. Cabarcas, D. Smith-Tone, and J. A. Verbel, "Key recovery attack for ZHFE," in *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8*. Springer, 2017, pp. 289–308.

[35] R. Perlner, D. Moody, and D. Smith-Tone, "Key recovery attack on the cubic abc simple matrix multivariate encryption scheme," in *23rd International Workshop, Selected Areas in Cryptography (SAC 2016); 08/10/2016-08/12/2016; St. John's, Newfoundland, Canada*. Springer, 2017, pp. 542–558.

[36] M. Øygarden, P. Felke, H. Raddum, and C. Cid, "Cryptanalysis of the multivariate encryption scheme eflash," in *Topics in Cryptology–CT-RSA 2020: The Cryptographers' Track at the RSA Conference 2020, San Francisco, CA, USA, February 24–28, 2020, Proceedings*. Springer, 2020, pp. 85–105.

[37] P. Briaud, J.-P. Tillich, and J. Verbel, "A polynomial time key-recovery attack on the sidon cryptosystem," in *International Conference on Selected Areas in Cryptography*. Springer, 2021, pp. 419–438.

[38] D. Smith-Tone, "2F-A new method for constructing efficient multivariate encryption schemes," in *International Conference on Post-Quantum Cryptography*. Springer, 2022, pp. 185–201.

[39] H. Kosuge and K. Xagawa, "Probabilistic Hash-and-Sign with Retry in the Quantum Random Oracle Model," Cryptology ePrint Archive, Paper 2022/1359, 2022.

[40] S.-j. Chang, R. Perlner, W. E. Burr, M. S. Turan, J. M. Kelsey, S. Paul, and L. E. Bassham, "Third-round report of the SHA-3 cryptographic hash algorithm competition," *NIST Interagency Report*, vol. 7896, p. 121, 2012.

[41] K. Sakumoto, T. Shirai, and H. Hiwatari, "On provable security of UOV and HFE signature schemes against chosen-message attack," in *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011*. Springer, 2011, pp. 68–82.

[42] T. Feneuil, "Post-quantum signatures from secure multiparty computation," Ph.D. dissertation, Sorbonne Université, 2023.

[43] C. Baum, L. Braun, C. D. de Saint Guilhem, M. Klooß, E. Orsini, L. Roy, and P. Scholl, "Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head," in *Annual International Cryptology Conference*, 2023, pp. 581–615.

[44] T. Feneuil and M. Rivain, "Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments," Cryptology ePrint Archive, Paper 2023/1573, 2023.

[45] D. Wagner, "A generalized birthday problem," in *Annual International Cryptology Conference*. Springer, 2002, pp. 288–304.

[46] S. Chatterjee, M. P. L. Das, and T. Pandit, "Revisiting the security of salted uov signature," in *International Conference on Cryptology in India*. Springer, 2022, pp. 697–719.

[47] B. Cogliati, P.-A. Fouque, L. Goubin, and B. Minaud, "New Security Proofs and Techniques for Hash-and-Sign with Retry Signature Schemes," Cryptology ePrint Archive, Paper 2024/609, 2024.

[48] E. Thomae and C. Wolf, "Solving underdetermined systems of multivariate quadratic equations revisited," in *International workshop on public key cryptography*. Springer, 2012, pp. 156–171.

[49] L. Bettale, J.-C. Faugere, and L. Perret, "Hybrid approach for solving multivariate systems over finite fields," *Journal of Mathematical Cryptology*, vol. 3, no. 3, pp. 177–197, 2009.

[50] A. Kipnis and A. Shamir, "Cryptanalysis of the Oil and Vinegar signature scheme," in *Annual international cryptology conference*. Springer, 1998, pp. 257–266.

[51] M. Bardet, M. Bros, D. Cabarcas, P. Gaborit, R. Perlner, D. Smith-Tone, J.-P. Tillich, and J. Verbel, "Improvements of algebraic attacks for solving the rank decoding and minrank problems," in *Advances in Cryptology–ASIACRYPT 2020*, 2020, pp. 507–536.

[52] W. Beullens, "Improved cryptanalysis of UOV and Rainbow," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2021, pp. 348–373.

[53] L. Chen, D. Moody, and Y. Liu, "NIST post-quantum cryptography standardization," *Transition*, vol. 800, p. 131A, 2017.

Appendix A. Security Proofs

Proof of Theorem 1: The Correctness of MQuBS

***Theorem*.** For properly executed $\mathsf{MQuBS}(n, m, q, r)$ protocol, if the signature on message $\mathtt{msg}$ is generated as σ. Then $\mathsf{MQuBS.Verify}(\mathsf{vk}, \sigma, \mathtt{msg}) = 1$ holds with probability $1 - \mathsf{negl}(\lambda)$.

Proof: Suppose the probability of correctness for MQuBS is denoted by $\mathsf{Pr}_{\mathsf{MQuBS}}$, which is defined as the following probability:

$$\Pr\left[\mathsf{MQuBS.Verify}(\mathsf{vk}, \sigma, \mathtt{msg}) = 1 \middle| \begin{array}{l} (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{MQuBS.KeyGen}(1^{\lambda}) \\ \sigma \leftarrow \mathsf{MQuBS.Sign}(\mathtt{msg}, \mathsf{sk}, \mathsf{vk}) \end{array}\right].$$

Now the correctness of UOV signature algorithm is denoted by Pr_{UOV}, and defined as following.

$$\Pr\left[\begin{array}{l} \mathsf{UOV.Verify}(\mathcal{P}_{UOV}, \mathbf{s} \\ , \mathtt{msg}) = 1 \end{array} \middle| \begin{array}{l} (O, \mathcal{P}_{UOV}) \leftarrow \mathsf{UOV.KeyGen}(1^{\lambda}) \\ \mathsf{s} \leftarrow \mathsf{UOV.Sign}(\mathtt{msg}, \mathsf{O}, \mathcal{P}_{\mathsf{UOV}}) \end{array}\right].$$

Similarly, the correctness of NIZK proofs is defined as follows.

$$\Pr\left[\begin{array}{l} \mathsf{NIZK.Verify}(x, \pi) = 1 \\ (x, w) \in \mathsf{Relation} \end{array} \middle| \begin{array}{l} \mathsf{st} \leftarrow \mathsf{SetUp}(1^{\lambda}) \\ \pi \leftarrow \mathsf{NIZK.Proof}(x, w, \mathsf{Relation}, \mathsf{st}) \end{array}\right]$$

We denote $\mathsf{Pr}_{\pi_{\mathsf{Com}}} = \Pr[\mathsf{NIZK.Verify}_1(x_{\mathsf{Com}},\ \pi_{\mathsf{Com}}) = 1|$ given that the proofπ_{Com} is generated correctly], and $\mathsf{Pr}_{\pi_{MQ}} = \Pr[\mathsf{NIZK.Verify}_2(x,\ \pi_{MQ}) = 1|$ given that the proof π_{MQ} is generated correctly] respectively. Since the correctness of the UOV signature, π_{Com}, and π_{MQ} are independent of each other, the correctness of MQuBS can be expressed as follows.

$$\mathsf{Pr}_{\mathsf{MQuBS}} = \mathsf{Pr}_{\mathsf{UOV}} \times \mathsf{Pr}_{\pi_{\mathsf{Com}}} \times \mathsf{Pr}_{\pi_{MQ}}$$

First, the signer verifies the NIZK proof π_{Com} for the hash-based commitment scheme, as the correctness of the NIZK proof guarantees its validity. If the proof is correct, the signer proceeds to compute the signature. The correctness of the NIZK proof π_{Com} is $\mathsf{Pr}_{\pi_{\mathsf{Com}}} = 1 - \mathsf{negl}_1(\lambda)$.

In the second step, we show that, at the end of the interactive process, the user obtains $\mathbf{s}$ as a pre-image of $\mathbf{b}$ under the map $\mathcal{P}_{UOV}$. The correctness of UOV signature ensures that $\mathcal{P}_{UOV}(\mathbf{s}) = \mathbf{b}$ holds. Therefore, we can say that, user has a solution $(\mathbf{s}, \mathbf{r})$ of the system $\tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{t}$, that is $\mathbf{t} = \tilde{\mathcal{P}}(\mathbf{s}, \mathbf{r}) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$. Hence, correctness of the UOV signature algorithm is $\mathsf{Pr}_{UOV} = 1 - \mathsf{negl}_2(\lambda)$.

In the third part, we use the correctness of the π_{MQ} protocol (see [30]). An honest prover (in our case, the user) provides a NIZK proof π_{MQ} for a quadratic system. This correctness of the NIZK proof π_{MQ} ensures that a proof generated by an honest user who knows a solution to the public system $\tilde{\mathcal{P}}$ will be verified by an honest verifier with probability $\mathsf{Pr}_{\pi_{MQ}} = 1 - \mathsf{negl}_3(\lambda)$.

Now, combine all values to compute the correctness of MQuBS. Finally, $\mathsf{Pr}_{\mathsf{MQuBS}} = (1 - \mathsf{negl}_1(\lambda)) \times (1 - \mathsf{negl}_2(\lambda)) \times (1 - \mathsf{negl}_3(\lambda)) = (1 - \mathsf{negl}_4(\lambda))$. Here, $\mathsf{negl}_4(\lambda) = \mathsf{poly}(\mathsf{negl}_1(\lambda), \mathsf{negl}_2(\lambda), \mathsf{negl}_3(\lambda)$, basically it consumes all negligible values. Therefore, combining all the probabilities, we can claim that the verifier of MQuBS blind signature will correctly verify the signature with overwhelming probability. □