MQuBS: A Short, Round-Optimal Blind Signature from Multivariate Quadratic Assumption

Anonymous Author(s)

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-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.

CCS Concepts

• **Security and privacy → Digital signatures**.

Keywords

Multivariate cryptography, Post-quantum cryptography, Blind signatures, Round optimal

ACM Reference Format:

Anonymous Author(s). 2025. MQuBS: A Short, Round-Optimal Blind Signature from Multivariate Quadratic Assumption . In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX).* ACM, New York, NY, USA, 18 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Blind signatures. Blind signature (BS) [23] 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 [19, 24, 58]. It requires an interactive protocol between the user and the 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 *round-optimal* interaction is of two rounds. Furthermore, two-round protocols are immune to attack during concurrent execution of the signing algorithm [16]. In this protocol, the user first *blinds* the message and sends it to the signer for *signing*, the 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 [36] first proposed a generic framework for constructing round-optimal blind signatures.

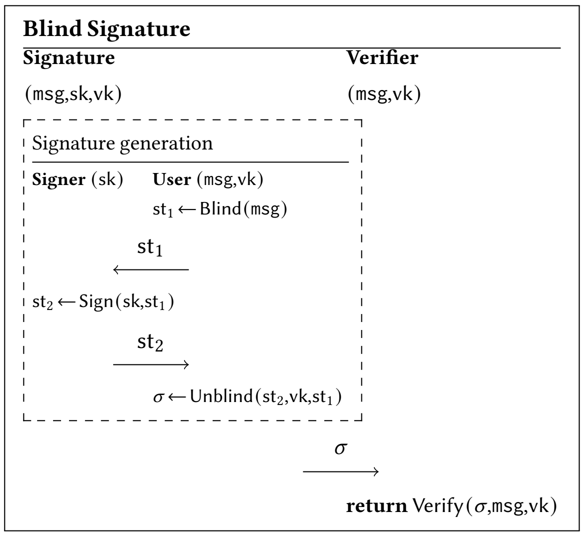

Figure 1: A schematic representation of different parties involved and their interactions in a blind signature protocol.

A BS scheme needs to satisfy the two security properties, *blindness* and *one-more unforgeability*. 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 [28] and elliptic curve discrete logarithm problems [38, 39]. However, due to the Shor's [57] and Proos-Zalka's algorithm [54] 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 [1, 16], relying on the Fischlin's framework. Although, both of these schemes suffer from larger signature and public-key sizes.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, Woodstock, NY
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

Problems in PQ cryptography (PQC). PQC has some major challenges. Many schemes are secure but have large keys or large signature sizes, which makes them hard to use in practice. It's also not fully clear how hard the math problems behind PQC are against quantum attacks. A clever new algorithm might one day break one of these problems, which could collapse an entire direction of research. There's also the practical challenge of adapting PQC schemes to work well with existing systems and protocols. For example, the Table 1 shows the signature sizes of different pre- and post-quantum digital signature algorithms (DSAs). It assumes 128-bit security.

Table 1: Signature sizes of pre- and post-quantum DSAs.

Signatures	EC-DSA	Dilithium [30]	Falcon [37]	SPHINCS+ [8]	UOV [15, 48]
Assumption	ECC	Lattices	Lattices	Hash-based	MQ-based
Quantum safe	×	✓	✓	✓	✓
Signature size (B)	64	2420	666	7856	**96**

Multivariate blind signature. In 2017, Petzoldt *et al.* [53] 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 [53] depends on the commitment scheme's binding and hiding properties, as well as the inherent hardness of solving multivariate quadratic (MQ) systems, which is believed to be NP-complete [45]. The commitment scheme in [53] 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 $\mathbf{r} \in \mathbb{F}_q^m$ and a random quadratic map $\mathcal{R}:\mathbb{F}_q^m \to \mathbb{F}_q^m$ as follows:

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

Subsequently, the user requests the signer to sign on the blind message $\mathbf{b}$. The signer applies a UOV-based signature algorithm [48] 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 [48]. 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 [53]: In [14], Beullens showed that the commitment scheme used in this BS scheme [53] is non-binding. This immediately breaks the one-more-unforgeability property of the scheme.[1]

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.

[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 [14]

2 Our Contribution

We propose MQuBS, a new multivariate-based BS scheme, following Fischlin's round optimal framework [36]. 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 IND-CPA secure MQ-based public-key encryption scheme (HFERP) [43]. Third, the UOV signature [15, 48] 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 [3, 7, 10, 20, 34, 56]. 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 [3, 20].

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 2 illustrates our claim[3].

Table 2: Post-quantum round optimal blind signature

Scheme	Assumption	$\|\sigma\|$ (KB)
Agrawal *et al.* [1]	OM-ISIS	45
Beullens *et al.* [16]	MSIS and MLWE	22
Petzoldt *et al.* [53]	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 [15]. The signature algorithm of MQuBS.Sign consists with three independent algorithms Sign_1,Sign_2, and Sign_3 (see Algorithm 3).

Blinding phase $\mathcal{S} \leftarrow \mathcal{U}$: $(\beta \leftarrow \mathsf{Sign}_1(\mathsf{vk}, \mathsf{msg}))$. 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 a commitment scheme, and a public-key encryption scheme: Com_{MQ} and $\mathsf{PKE}_{\mathsf{MQ}}$.

Fischlin's suggestions [36]. 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 encryption π_{Enc}. 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

[2]It is possible to incorporate any UOV-type signature schemes, like VDOO [41], Mayo [13], and QR-UOV [40] in MQuBS.
[3]The security model used in Petzoldt *et al.* construction is universal OMUF. Their construction is designed on the top of Rainbow [29] scheme, which is broken by [12]. So the parameters proposed in [53] will change. However, due to the vulnerability in the commitment scheme, the BS scheme is no longer secure.

```
Fischlin's Blind Signature
User $\mathcal{U}$ ($\mathbf{d} \leftarrow \mathcal{H}(\texttt{msg}), \mathsf{vk}$)            Signer $\mathcal{S}$ ($\mathsf{sk}$)
$\beta \leftarrow \mathsf{Sign}_1(\mathsf{vk}, \mathbf{d})$
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 $\pi_{\mathsf{Enc}} \leftarrow \mathsf{Proof}_{\mathsf{Enc}}(x_{\mathsf{Enc}}, w_{\mathsf{Enc}})$
$\beta \leftarrow (\pi_{\mathsf{Enc}}, \mathbf{b}, \mathsf{ct})$
                              $\xrightarrow{\beta}$
                              $\psi \leftarrow \mathsf{Sign}_2(\beta, \mathsf{sk})$
                                        if ($\mathsf{Verify}_{\mathsf{Enc}}(\pi_{\mathsf{Enc}}, \mathsf{ct})$)
                                          $\mathbf{s} \leftarrow \mathsf{Sign}(\mathbf{b}, \mathsf{sk})$
                              $\xleftarrow{\psi = \mathbf{s}}$
$\sigma \leftarrow \mathsf{Sign}_3(\mathsf{vk}, \mathbf{s})$
if ($\mathsf{Verify}_{\mathsf{Sign}}(\mathbf{b}, \mathbf{s}, \mathsf{vk})$)
$Ct \leftarrow \mathsf{Enc}(\beta, \psi; r_3)$
$x \leftarrow (Ct, \mathsf{pk}_{\mathsf{Enc}})$ // 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 [36] framework

$\mathbf{t} = \mathcal{H}(\texttt{msg}, \mathcal{G}(\mathbf{r}))$, where $\mathcal{G} : \mathbb{F}_q^m \rightarrow \{0,1\}^{2\lambda}$, ($\lambda$ is a security parameter) and $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^m$ are collision-resistant hash functions[4]. These hash functions can be modelled as random oracles. Beullens *et al.* [16]'s lattice-based BS 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 make the BS scheme secure, we design a new commitment scheme that resists Beullens's attack [14]. Our commitment scheme Com_{MQ} uses two random matrices $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{F}_q^{m \times m}$ and a random quadratic map $\mathcal{R} : \mathbb{F}_q^m \rightarrow \mathbb{F}_q^m$, all derived from the message and a random $\mathbf{r} \in \mathbb{F}_q^m$ using a pseudorandom generator (PRG) (see Algorithm 1). The commitment equation is defined as follows (for more details, see Section §4).

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

Encryption of message digest and randomness. In Figure 2, the user encrypts the message digest and randomness using a public-key encryption (PKE). Lattice-based BS constructions [1, 16] utilize lattice-based public-key encryption in an "*encryption to the sky*" fashion to achieve this goal. In similar fashion, we adopt standard MQ-public-key encryption. To establish the OMUF security of MQuBS (see Theorem 5), we must decrypt the ciphertext to retrieve the message and randomness. We use the HFERP encryption scheme [43] to encrypt $\mathbf{r}$ and $\texttt{msg}$.

$$CT = \mathsf{PKE}_{\mathsf{MQ}}(\mathbf{r}, \texttt{msg}; u) \qquad (3)$$

In addition, the user generates a NIZK proof π_{PKE}. It consists of two parts: the first ensures the well-formedness of $\mathbf{b}$, as defined by Equation 2; the second verifies the well-formedness of CT, as defined by Equation 3. Observe that, the final signature does not include the proof π_{PKE}, and so the user can precompute the proof (offline phase). Therefore it does not add to the performance of the blind signature scheme. At the end of this phase, the user sends $\beta = (CT, \pi_{\mathsf{PKE}}, \mathbf{b})$ to the signer and asks for a signature. Therefore, user sends $|\beta| = |CT| + |\pi_{\mathsf{PKE}}| + |\mathbf{b}|$-bit elements, where $|\mathbf{b}| = m \log_2 q$.

Signing phase $\mathcal{U} \leftarrow \mathcal{S}$: $(\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \beta))$. This procedure is the same as Fischlin's proposal [36]. After receiving β, it first verifies the proof π_{PKE}, then it runs the UOV-signature algorithm [15] 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 [36] 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 [3, 7, 20, 34]. The final signature σ contains three items, one is proof π_{MQ}, a seed to generate emulsifier maps, and a random quadratic map, and the third component is $\mathcal{G}(\mathbf{r})$. The size of $|\sigma|$ 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)[13]. We also show that MQuBS offers blindness and OMUF.

Security of commitment scheme. The primary security properties of a cryptographically secure commitment scheme are *computationally hiding* and *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 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 4). We present this claim informally, and later in Theorem 5, 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.*

[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 [32].

We use three lemmas to prove the OMUF property of MQuBS. The 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 [15]. Note that our version of UOV differs slightly from standard UOV [15, 48] (see Figure 5). 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. The proofs of these lemmas are given in the Appendix, and similar arguments can be found in [15].

Practicality: efficiency and comparisons. The signature size of MQuBS is significantly smaller than the existing post-quantum round-optimal blind signatures [16, 50]. The initial communication from the user $\mathcal{U}$, to the signer $\mathcal{S}$, consists of $|\beta| = (|CT| + |\pi_{\mathsf{PKE}}| + m\log q)$ 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 3.

Table 3: Communication costs of MQuBS

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

Key and Signature sizes for 128-bits security level. We use formulas from Table 3 to compute signature size for 128-bit security level. The private and public key sizes for our scheme are same as UOV signature [15]. Since β and ψ are not part of the final signature, we focus solely on computing the size of σ, i.e., $|\sigma|$. The UOV signature[15, 48] offers a 96-byte signature size for 128-bit security. According to the UOV SL-1 parameters (128-bit security) [15], the UOV public polynomial map has 64 quadratic equations with 160 variables (see Table 5). Therefore, a user prepares a NIZK proof for a quadratic system with $n+m$ variables and m- homogeneous quadratic equations. Bui [20] offers the most efficient proof size for MQ relation. Using this technique, the proof size for our quadratic system is approximately 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.

2.2 Post-quantum BS from other hard problems

Due to the growing interest in post-quantum cryptography, numerous BS schemes have been proposed in the literature [17, 18, 31, 47, 51]. Previously, we discussed some round-optimal lattice-based BS constructions. However, there exist BS constructions from other problems also.

Designs from code-based cryptography. In a parallel development, two independent works [31, 51] appeared on ePrint. However, both designs follow a three-pass blind signature (BS) framework. As a result, we do not compare their signature sizes with those from round-optimal constructions, as three-pass schemes inherently require higher bandwidth. Additionally, these works did not report any performance benchmarks for their constructions. Blazy *et al.* [18] introduced a round-optimal BS, which was later revised [17] to address certain security issues. According to [17], the reported signature size for this scheme is 86.7MB at the 80-bit security level.

Designs from isogeny-based cryptography. A notable blind signature scheme based on isogeny-based assumptions was introduced by Katsumata et al. [47] This construction also follows a three-pass framework.

2.3 Estimated computation time

Our constructions is heavily rely on the NIZK proofs, HFERP encryption and UOV signature scheme. It uses two core NIZK proofs, one is the NIZK proof of a hash-based commitments, and the other is the NIZK proof for MQ systems. As per specification of [34], the estimated running time to generate a NIZK proof for a quadratic systems with 48 variables and 48 quadratic equations over $\mathbb{F}_{256}$ is 18M clock cycles. The key difference is our quadratic system has $n+m$ variables, and m-equations, and for 128-bit security level, the quadratic system has 112 variables and 44 equations over $\mathbb{F}_{256}$. The estimated proof generation time for quadratic system is approximately 50M clock cycles. Now we come to the NIZK proof for the hash-based commitments. In this case, we use efficient NIZK proof constructions called ZKBoo [42]. This proof generation requires around 72M clock cycles, and verification will take approximately 124M clock cycle. Generating the proof takes approximately 72 million clock cycles, while verification requires around 124 million clock cycles. The signature generation of UOV requires 90M clock cycle for skc+pkc variant and 2.5M clock cycle for classic and pkc versions. The verification UOV signature requires 11.5M clock cycle for pkc and skc+pkc variant and 1M for classic variant. For more details, see Table 10 of UOV specifications [15]. All clock cycle counts are taken from their respective references.

3 Background

In this section, we define the blind signature scheme Section §3.1 and its security properties. Due to space constraints, we refer the reader to the Appendix C for brief details about the commitment scheme and the NIZK proof for the MQ problem.

Basic Notations. We denote $a \overset{\$}{\leftarrow} U$ 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 $\mathrm{GL}(m, q)$ 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* (ROBS) scheme, denoted as BS = (KeyGen,Sign,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* $\mathsf{d} = \mathcal{H}(\mathtt{msg})$.

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

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.*

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

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* $\mathsf{negl}(\lambda)$*.*

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

4 MQuBS: A Round-Optimal Blind Signature

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.

```
Multivariate Blind Signature in the Fischlin's Framework
Signature                                                   Verifier
$(\mathtt{msg}, \mathsf{sk}, \mathsf{vk})$                  $(\mathtt{msg}, \mathsf{vk} = \mathcal{P}_{UOV})$
  Signature generation
  User $(\mathtt{msg}, \mathsf{vk})$                        Signer $(\mathsf{sk})$
  $\beta \leftarrow \mathsf{Sign}_1(\mathcal{P}_{UOV}, \mathtt{msg})$
  $\mathbf{r} \leftarrow \mathbb{F}_q^m;\ \rho \leftarrow \mathcal{G}(\mathbf{r})$
  Generate $\mathbf{E}_1,\ \mathbf{E}_2,\ \mathcal{R}$ from $\mathtt{msg}, \mathsf{vk}, \mathbf{r}$
  $\mathbf{b} \leftarrow \mathsf{Com}_{MQ}(\mathtt{msg}, \mathbf{r}) = \mathbf{E}_1^{-1}\left(\mathcal{H}(\mathtt{msg}, \mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r})\right)$
  $u \leftarrow \{0,1\}^{2\lambda};\ CT \leftarrow \mathsf{PKE_{MQ}}(\mathbf{r}, \mathtt{msg}; u)$
  $x_{\mathsf{PKE}} \leftarrow (CT, \mathbf{b})$
  $w_{\mathsf{PKE}} \leftarrow (\mathbf{r}, \mathtt{msg}, u)$
  $\mathsf{Relation}_1 \leftarrow \mathbf{b} = \mathsf{Com}_{MQ}(\mathtt{msg}, \mathbf{r}) \wedge C = \mathcal{H}_{\mathsf{Com}}(\mathbf{r}, \mathtt{msg}; u)$
  $\pi_{\mathsf{PKE}} \leftarrow \mathsf{NIZK.Proof}_1(x_{\mathsf{PKE}}, w_{\mathsf{PKE}}, \mathsf{Relation}_1)$
  $\beta \leftarrow (CT, \pi_{\mathsf{PKE}}, \mathbf{b})$
                                        $\xrightarrow{\beta}$
                                          $\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk}, \mathbf{b})$
                                          if $\mathsf{NIZK.Verify}_1(x_{\mathsf{PKE}}, \pi_{\mathsf{PKE}})$
                                            $\mathbf{s} \leftarrow \mathsf{UOV.Sign}(\mathbf{b}, \mathsf{sk})$
                                            $\psi \leftarrow \mathbf{s}$
                                        $\xleftarrow{\psi}$
$\sigma \leftarrow \mathsf{Sign}_3(\mathcal{P}_{UOV}, \psi)$
if $\mathsf{UOV.Verify}(\psi, \mathsf{vk})$
  Define $\tilde{\mathcal{P}}(\mathbf{x}_1, \mathbf{x}_2) = \mathbf{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1) + \mathbf{E}_2\mathcal{R}(\mathbf{x}_2)$
$x \leftarrow (\mathtt{msg}, \rho, \mathbf{E}_1, \mathbf{E}_2, \mathcal{P}_{UOV}, \mathcal{R})$
$w \leftarrow (\mathbf{s}, \mathbf{r})$
$\mathsf{Relation}_2 \leftarrow \{(\mathbf{s}, \mathbf{r}) : \mathcal{H}(\mathtt{msg}, \rho) = \tilde{\mathcal{P}}(\mathbf{s}, \mathbf{r})\}$
$\pi_{MQ} \leftarrow \mathsf{NIZK.Proof}_2(x, w, \mathsf{Relation}_2)$
$\sigma \leftarrow (\pi_{MQ}, \rho, \tilde{\mathcal{P}})$
                                                            $\xrightarrow{\sigma}$
                                                            $\mathbf{t} \leftarrow \mathcal{H}(\mathtt{msg}, \rho)$
                                                            return $\mathsf{NIZK.Verify}_2(\sigma, \mathbf{t}, \mathsf{vk})$
```

Figure 3: MQuBS **Multivariate Blind Signature**

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 $\mathcal{G} : \{0,1\}^* \rightarrow \{0,1\}^{2\lambda}$, $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^m$ and one MQ-based public key encryption scheme $\mathsf{PKE_{MQ}}$. Any cryptographically secure hash function can be deployed here such as the current NIST standard SHA-3 [32]. Later in the security proof (see Section §5), we model these hash 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

```
b ← Com_MQ(msg;r) = $\mathbf{E}_1^{-1}(\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$
1:  Input: msg
2:  Output: $\mathbf{b},(\mathbf{E}_1,\mathbf{E}_2,\mathcal{R})$
3:  $\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m,\ \rho \leftarrow \mathcal{G}(\mathbf{r})$
4:  while $(\det(\mathbf{E}_1) \neq 0\ \&\ \det(\mathbf{E}_2) \neq 0)$ {
5:    $\mathtt{rnd} \xleftarrow{\$} \{0,1\}^{2\lambda}$, seed1 ← Hash(msg || r || rnd)
6:    seed2 ← Hash(msg || seed1)
7:    $\mathbf{E}_1$ ← XOF(seed1); $\mathbf{E}_2$ ← XOF(seed2) }
8:  seed3 ← Hash(msg || seed2)
9:  $\mathcal{R}$ ← XOF(seed3)
10: $\mathbf{b} \leftarrow \mathbf{E}_1^{-1}(\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$.
11: return 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 [13], we refer to $\mathbf{E}_i$ as an *emulsifier map*. At the end, the commitment of a message is computed as $\mathbf{b} = \mathrm{Com}_{MQ}(\mathtt{msg};\mathbf{r}) = \mathbf{E}_1^{-1}(\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r})) - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$.

4.2 Algorithms of MQuBS

The complete MQuBS algorithm includes three main algorithms: MQuBS.KeyGen, MQuBS.Sign, and MQuBS.Verify. The interactive signing algorithm, 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.

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 [15]. 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: $(\mathsf{sk},\mathsf{vk}) \leftarrow \mathrm{MQuBS.KeyGen}(1^\lambda)$. 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 $\mathbf{P}_i^{(3)} \leftarrow \mathrm{Upper}(-\mathbf{O}^\top\mathbf{P}_i^{(1)}\mathbf{O} - \mathbf{O}^\top\mathbf{P}_i^{(2)})$.

Algorithm 2 Key Generation of MQuBS

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

4.2.2 *Interactive Signing Algorithm of* MQuBS.

The signing operation is given by: $\sigma \leftarrow \mathrm{MQuBS.Sign}(\mathtt{msg},\mathsf{sk},\mathsf{vk})$ 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: $\beta \leftarrow \mathrm{Sign}_1(\mathtt{msg},\mathsf{vk})$. Here, the user has inputs a public key $\mathcal{P}_{UOV}$, and a message 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 msg, $\mathcal{G}(\mathbf{r})$, and $\mathcal{P}_{UOV}$. To do this, the user computes the following seeds

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

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 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 PKE scheme to commit $\mathbf{r}$, and msg using a randomly generated 2λ-bit string u, that is, Equation 3. Later it provides a NIZK proof π_{PKE} for CT, and it also adds a proof for the well-formedness of $\mathbf{b} = \mathbf{E}_1^{-1}(\mathbf{t} - \mathbf{E}_2\mathcal{R}(\mathbf{r}))$.

The proof of well-formedness is a bit tricky here. Since, if $\mathbf{E}_1$, $\mathbf{E}_2$, and $\mathcal{R}$ are published as a statement in the NIZK proof π_{PKE}, and these values also present in the proof π_{MQ}. So it breaks the blindness property of the scheme.

Well-formedness of **b**

```
1: Sample $u_1,u_2,u_3,u_4,u_5 \in \{0,1\}^{2\lambda}$
2: $\mathsf{seed}_{\mathbf{E}_1^*} \leftarrow \mathcal{H}(\mathsf{seed}_{\mathbf{E}_1^*};u_1)$
3: $\mathbf{E}_1^* \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathbf{E}_1^*})$; if $\det(\mathbf{E}_1^*)=0$ then, change $u_1$
4: $\mathsf{seed}_{\mathbf{E}_2^*} \leftarrow \mathcal{H}(\mathsf{seed}_{\mathbf{E}_2^*};u_2)$
5: $\mathbf{E}_2^* \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathbf{E}_2^*})$; if $\det(\mathbf{E}_2^*)=0$ then, change $u_2$
6: $\mathsf{seed}_{\mathcal{R}^*} \leftarrow \mathcal{H}(\mathsf{seed}_{\mathcal{R}};u_3)$
7: $\mathcal{R}^* \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathcal{R}^*})$
8: $\mathbf{b}^* \leftarrow (\mathbf{E}_1^*)^{-1}(\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r}))-\mathbf{E}_2^*\mathcal{R},^*(\mathbf{r}))$
9: $\mathcal{L}^*=\{(\mathbf{E}_1,\mathbf{E}_2,\mathcal{R},\mathbf{r},\mathtt{msg}):\mathsf{seed}_{\mathbf{E}_1^*} \leftarrow \mathcal{H}(\mathsf{seed}_{\mathbf{E}_1^*})\wedge$
10: $\mathsf{seed}_{\mathbf{E}_2^*} \leftarrow \mathcal{H}(\mathsf{seed}_{\mathbf{E}_2^*})\wedge \mathsf{seed}_{\mathcal{R}^*} \leftarrow \mathcal{H}(\mathsf{seed}_{\mathcal{R}};u_3)\wedge$
11: $\mathbf{b}_1 \leftarrow (\mathbf{E}_1^*)^{-1}(\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r}))-\mathbf{E}_2^*\mathcal{R},^*(\mathbf{r}))$,
12: $\wedge c_1=\mathsf{Hash}\Big((\mathbf{E}_1)^{-1}\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r});u_4)\Big)\wedge$
13: $c_2=\mathsf{Hash}\Big((\mathbf{E}_1)^{-1}\mathbf{E}_2\mathcal{R}(\mathbf{r});u_5\Big)\}$
14: $x_{\mathcal{L}^*}::(\mathsf{seed}_{\mathbf{E}_1^*},\mathsf{seed}_{\mathbf{E}_2^*},\mathsf{seed}_{\mathcal{R}^*},c_1,c_2)$
15: $w_{\mathcal{L}^*}::(\mathbf{E}_1,\mathbf{E}_2,\mathcal{R},\mathbf{r},\mathtt{msg})$
```

Figure 4: Well-formedness proof of b.

To avoid this situation, each seeds of $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathcal{R}$ commits, and computes a new seed to generate $\mathbf{E}_1^*$, $\mathbf{E}_2^*$ and $\mathcal{R}^*$. Once these values are generated, then it computes the following

$$\mathbf{b}^* \leftarrow (\mathbf{E}_1^*)^{-1}(\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r}))-\mathbf{E}_2^*\mathcal{R},^*(\mathbf{r})),$$

and add a well-formedness proof for $\mathbf{b}^*$. It also adds proof for the above commitments of each seeds, and the commitment value $c_1 = \mathsf{Hash}\left((\mathbf{E}_1)^{-1}\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r});u_4)\right)$, and $c_2 = \mathsf{Hash}\left((\mathbf{E}_1)^{-1}\mathbf{E}_2\mathcal{R}(\mathbf{r});u_5\right)$. Through the commitment of each seeds, user convinces that it has knowledge about $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathcal{R}$. Now, the commitment $c_1 = \mathsf{Hash}\left((\mathbf{E}_1)^{-1}\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r});u_4)\right)$, and $c_2 = \mathsf{Hash}\left((\mathbf{E}_1)^{-1}\mathbf{E}_2\mathcal{R}(\mathbf{r});u_5\right)$ ensures that, the message msg, and $\mathbf{r}$ is used during the computation of $\mathbf{b}$. It also ensures that user picks $\mathbf{b}$ before processing the query. In the proof π_{PKE}, the statement is x_{PKE} are $x_{\mathcal{L}^*},\mathbf{b},CT$, and witness w_{PKE} are $\mathbf{r},\mathtt{msg},u_1$. Finally, the user communicates $\beta=(\mathbf{b},CT,\pi_{\mathsf{PKE}})$ to the signer.

NIZK proof π_{PKE}. The algorithm $\mathsf{NIZK.Proof}_1$ prepares the proof and $\mathsf{NIZK.Verify}_1$ algorithm verifies the proof π_{PKE}. As we mentioned, the proof π_{PKE} consists of two things, one is the wellformedness of $\mathbf{b}$, and another is the wellformedness of CT. The first part, we already described in Fig 4. The second is simple, we just provide a proof of encryption for MQ-based PKE. This can be achieved from the construction of Bui [20]. Several algorithms have also been proposed to construct efficient NIZK proofs for multivariate quadratic (MQ) relations (see Table 6, where $\lambda=128$-bits).

Communication cost. Therefore, the user needs to send $|CT|$-bits for the ciphertext, $|\pi_{\mathsf{PKE}}|$ bits for the NIZK proof, and $m\log q$ bits for the blind message. In total, this amounts to $|CT|+|\pi_{\mathsf{PKE}}|+m\log q$ bits to communicate with the signer as the blind message.

Blind signature computation: $\psi \leftarrow \mathsf{Sign}_2(\mathsf{sk},\beta)$. In this algorithm, the signer (or issuer) receives a blinded message $\mathbf{b}$, ciphertext CT, secret key sk, and a NIZK proof π_{PKE} along with a statement x_{PKE}.

Algorithm 3 Sign_1: Message blinding

```
$\beta \leftarrow \mathsf{Sign}_1(\mathcal{P}_{UOV},\mathtt{msg})$
1: **Input:** $\mathsf{vk}=\mathcal{P}_{UOV}$; msg
2: **Output:** $\beta=(\mathbf{b},CT,\pi_{\mathsf{PKE}})$
3: Hash: $\mathcal{G}:\{0,1\}^* \rightarrow \{0,1\}^{2\lambda}$, $\mathcal{H}:\{0,1\}^* \rightarrow \mathbb{F}_q^n$
4: while $(\det(\mathbf{E}_1)\neq 0 \ \&\ \det(\mathbf{E}_2)\neq 0)\{$
5:     $\mathbf{r} \xleftarrow{\$} \mathbb{F}_q^m$, $\mathsf{seed}_{\mathbf{E}_1} \leftarrow \mathcal{G}(\mathtt{msg}||\mathcal{G}(\mathbf{r})||\mathcal{P}_{UOV})$
6:     $\mathsf{seed}_{\mathbf{E}_2} \leftarrow \mathcal{G}(\mathtt{msg}||\mathsf{seed}_{\mathbf{E}_1})$
7:     $\mathbf{E}_1 \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathbf{E}_1})$, $\mathbf{E}_2 \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathbf{E}_2})\}$
8: $\mathbf{t} \leftarrow \mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r}))$
9: $\mathsf{seed}_{\mathcal{R}} \leftarrow \mathcal{G}(\mathtt{msg}||\mathsf{seed}_{\mathbf{E}_2})$, $\mathcal{R} \leftarrow \mathsf{XOF}(\mathsf{seed}_{\mathcal{R}})$
10: $\mathbf{b} \leftarrow \mathbf{E}_1^{-1}(\mathbf{t}-\mathbf{E}_2\mathcal{R}(\mathbf{r}))$
11: $u \xleftarrow{\$} \{0,1\}^{\lambda}$, $CT \leftarrow \mathsf{PKE}_{\mathsf{MQ}}(\mathtt{msg},\mathbf{r},u)$
12: $x_{\mathsf{PKE}} \leftarrow CT$, $w_{\mathsf{PKE}} \leftarrow (\mathtt{msg},\mathbf{t},u)$
13: Relation :: $\mathbf{b}=\mathbf{E}_1^{-1}(\mathbf{t}-\mathbf{E}_2\mathcal{R}(\mathbf{r}))\bigwedge C=\mathcal{H}(\mathtt{msg},\mathbf{r},u)$
14: $\pi_{\mathsf{PKE}} \leftarrow \mathsf{NIZK.Proof}_1(x_{\mathsf{PKE}},w_{\mathsf{PKE}},\mathsf{Relation})$
15: $\beta \leftarrow (\mathbf{b},CT,\pi_{\mathsf{PKE}})$
16: **return** $\beta$
```

It first verifies π_{PKE} 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 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 $\mathbf{b}=\mathsf{L}_{\mathbf{v}}(\mathbf{o})$:

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

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 [15]. At the end, signer communicates $\psi=\mathbf{s}$ as a signature on the blind message.

Algorithm 4 Sign_2 : Signature computation by Signer

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

Table 4: Proof size for various NIZK proof for MQ

ZKP	Five pass [56]	with helper [10]	MPCitH [33]	TCitH [35]	VOLEitH [3]	VOLEitH [20]
Proof size (KB)	29	14	6.9	4.2	2.6	3.6

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.

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

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

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

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 [3, 4, 7, 10, 20, 33–35, 55]. In the below, we present the proof size for the 128-bit security level. Table 4 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 [3, 20]. Both constructions extend the idea of [4]. 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 6. For further details on the NIZK proof and verification algorithm, we refer to [3].

Algorithm 5 Sign_3: Unblind the signature

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

$0/1 \leftarrow \mathsf{MQuBS.Verify}(\mathsf{vk},\sigma,\mathsf{msg})$ **: Verification Phase.** The verifier possesses the public key $\mathcal{P}_{UOV}$ and the message msg. Upon receiving the signature σ, the verifier aims to determine whether it is the correct signature for the message msg. Initially, the verifier $\mathbf{t}' \leftarrow \mathcal{H}(\mathsf{msg},\mathcal{G}(\mathbf{r}))$. Further, it expands the emulsifier matrices $\mathrm{E}_1,\mathrm{E}_2$, and the random quadratic map $\mathcal{R}$ from the seeds present in the signature. After completing these computations, it constructs the quadratic system $\tilde{\mathcal{P}}(\mathbf{x}_1,\mathbf{x}_2) = \mathrm{E}_1\mathcal{P}_{UOV}(\mathbf{x}_1)+\mathrm{E}_2\mathcal{R}(\mathbf{x}_2) = \mathbf{t}'$. Finally, it follows verifies the proof π_{MQ} by running $\mathsf{NIZK.Verify}_2$. We use the algorithm of [20] in this case.

Algorithm 6 MQuBS : Verification algorithm

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

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* msg *is generated as* σ. *Then* $\mathsf{MQuBS.Verify}(\mathsf{vk},\sigma,\mathsf{msg}) = 1$ *holds with probability* 1.

Sketch. The correctness of MQuBS relies on the correctness of the UOV signature algorithm along with two NIZK proofs: π_{PKE} 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 [15, 48]. The algebraic cryptanalysis of MQuBS can be found in Appendix C.

Definition 4. *UOV **Problem.** [48] Let $\mathcal{MQ}_{n,m,q}$ be the family of the random quadratic map; and $\mathcal{MQ}^{UOV}_{n,m,q}$ is the family of UOV-public polynomial map. The problem asks to distinguish between $\mathcal{P} \in \mathcal{MQ}_{n,m,q}$ or $\mathcal{P} \in \mathcal{MQ}^{UOV}_{n,m,q}$. Suppose $\mathcal{D}$ denotes the distinguisher algorithm for UOV, then the distinguishing advantage for $\mathcal{D}$ is defined as below.*

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

We define $\mathcal{P}$ as belonging to $\mathcal{MQ}^{UOV}_{n,m,q}$, such that for a secret linear subspace O, $\mathcal{P}(O) = 0$. It is widely believed that, for all probabilistic polynomial time (PPT) distinguisher $\mathcal{D}$, the advantage $\mathsf{Adv}^{(n,m,q)}_{UOV}(\mathcal{D}) \leq \mathsf{negl}(\lambda)$. Now we propose a new hard problem called gWMQ problem. Beullens first introduced the WMQ problem for the Mayo digital signature scheme [13], 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 $\mathrm{E}^{\mathcal{R}_i}_{ij}$ 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*

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

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 [60]. However, in our case since $k=2$, we do not need those cross terms. The* gWMQ *problem asks for a solution* $(\mathbf{s}_1,\mathbf{s}_2)$ *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* $\mathcal{R}_i$. *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.*

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

To the best of our knowledge, there is no known cryptanalysis for Beullens's WMQ problem [13]. 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

5.1.1 Binding property. The *binding property* of a commitment scheme ensures that once a value is committed, it cannot be changed. This prevents a sender from opening the commitment to different values later. The following proof demonstrates that the binding property of our commitment scheme (see Algorithm 1) 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 (computationally binding). *The commitment scheme presented in Algorithm 1 is computationally binding under the* gWMQ *assumptions. In other words, if an adversary* $\mathcal{A}$ *has an advantage* $\mathsf{Adv}_{\mathsf{COM}}(\mathcal{A})$ *in the computationally 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, the adversary $\mathcal{A}$ 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. □

5.1.2 Hiding property. The *hiding property* of a commitment scheme ensures that the commitment does not reveal any information about the committed value until it is later revealed. A commitment scheme satisfies the *hiding property* if, for any two messages μ_0 and μ_1, the distributions of $\mathsf{Com}(\mu_0,r)$ and $\mathsf{Com}(\mu_1,r')$ are computationally (or statistically) indistinguishable. This ensures that an adversary cannot learn anything about μ from *Com* (where *Com* $\leftarrow \mathsf{Com}(\mu,r)$), preserving secrecy until the commitment is opened. Let's recall our commitment scheme.

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

The hiding property of our commitment scheme is coming from the pre-image resistance of the hash function $\mathcal{H}$.

Theorem 3 (Computationally hiding). *The commitment scheme presented in Algorithm 1 is computationally hiding under the pre-image resistance of the hash function. Specifically, if an adversary* $\mathcal{A}$ *has an advantage* $\mathsf{Adv}_{\mathsf{COM}}^{\mathsf{Hiding}}(\mathcal{A})$ *in the hiding game, then there exist adversaries* $\mathcal{B}$ *that can find the pre-image of hash function with advantages* $\mathsf{Adv}_{\mathsf{Hash}}(\mathcal{B})$, *such that*

$$\mathsf{Adv}_{\mathsf{COM}}^{\mathsf{Hiding}}(\mathcal{A})\leq \mathsf{Adv}_{\mathsf{Hash}}(\mathcal{B})$$

Proof. The adversary has input $\mathbf{b}$, and tries to learn about the message. To do this adversary tries to find the pre-image of the hash function with the advantages $\mathsf{Adv}_{\mathsf{Hash}}(\mathcal{B})$. Once it finds the pre-image, it learns about the message. □

5.2 Security proof for MQuBS

5.2.1 Blindness or Anonymity.

Theorem 4. *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* C *that can break multivariate commitment scheme* Com_{MQ} *defined in Algorithm 1 with advantage* $\mathsf{Adv}_{\mathsf{Com}_{MQ}}(C)$, *and an adversary* $\mathcal{D}$ *that can break the IND-CPA secure PKE scheme* $\mathsf{MQ}_{\mathsf{PKE}}$ *with advantage* $\mathsf{Adv}_{\mathsf{PKE}_{\mathsf{MQ}}}(\mathcal{D})$, *so that the following condition holds.*

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

Proof. To prove our claim we use the following hybrids.

Hybrid$_0$. It denotes the original *blindness* or *anonymity* game of the honest signer.

Hybrid$_1$. It replaces the two non-interactive zero-knowledge proofs π_{PKE} and π_{MQ} with their respective simulated versions. That is, instead of generating the NIZKs honestly using the corresponding witnesses, the challenger simulates them without access to the witnesses.

Hybrid$_2$. In this hybrid, the main change from the previous one is that, both ciphertexts ct_0 and ct_1 are generated by encrypting $\mathbf{0}$, rather than the original message and randomness pairs $(\mathtt{msg}_0,\mathbf{r}_0)$ and $(\mathtt{msg}_1,\mathbf{r}_1)$.

Hybrid$_3$. This hybrid differs from the previous hybrid in the way the challenger computes $\mathbf{b}_0$ and $\mathbf{b}_1$. Instead of computing $\mathbf{b}_i \leftarrow {\mathbf{E}_1}_i^{-1}(\mathcal{H}(\mathsf{msg},\mathcal{G}(\mathbf{r}))-{\mathbf{E}_2}_i\mathcal{R}_i(\mathbf{r}_i))$, it computes $\mathbf{b}_i \leftarrow {\mathbf{E}_1}_i^{-1}(\mathcal{H}(\mathsf{msg},\mathcal{G}(\mathbf{r}))-{\mathbf{E}_2}_i\mathbf{u}_i)$ (where $i=0,1$).

From the definition of Hybrid$_0$, we can write $\Pr[\text{Hybrid}_0 = 1] = \mathsf{Adv}_{\mathsf{BLND}}(\mathcal{A})$.

In Hybrid$_1$, the only difference is the way of computing the NIZK proof π_{PKE} and π_{MQ}. This scenario can be distinguished due to the zero-knowledge property of the NIZK with an advantage denoted as $\mathsf{Adv}_{NIZK}(\mathcal{B})$. Thus adversary's distinguishing advantage in Hybrid$_1$ differs from its distinguishing advantage in Hybrid$_0$ by $\mathsf{negl}(\lambda)$.

The Hybrid$_2$ differs from Hybrid$_1$ from the way of generating the ciphertext ct_i in β_i. Since PKE is IND-CPA secure, so the two hyrbids are computationally indistinguishable. If $\mathsf{Adv}_{\mathsf{PKE}_{MQ}}(\mathcal{D})$ denotes the advantage for the adversary C in the honest signer blindness game in Hybrid$_2$, then the adversary's distinguishing advantage in Hybrid$_2$ differs from its advantage in Hybrid$_1$ by at most $\mathsf{negl}(\lambda)$.

The only difference between Hybrid$_2$ and Hybrid$_3$ is in way of masking the message. To mask the message, we have introduced a commitment scheme, which is computationally hiding. Therefore, due to the computationally hiding property of $\mathsf{Com}_{\mathsf{MQ}}$, two hybrids Hybrid$_2$ and Hybrid$_3$ are computationally indistinguishable. Therefore, we can write, the difference in the adversary's ability to distinguish between Hybrid$_2$ and Hybrid$_3$ is negligible.

Note that, information theoretically the adversary $\mathcal{A}$ in the hybrid Hybrid$_3$ has zero advantage to guess the bit b. As a result, the adversary's distinguishing advantage in Hybrid$_0$ must be negligible. Hence the result follows.

□

5.2.2 One More Unforgeability (OMUF). In this part, we first define 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.*

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

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

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

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. We used UOV as the underlying signature scheme [15, 48]. The existential unforgeability (EUF-CMA) in the random oracle model of the UOV signature scheme has been studied extensively in the literature [22, 26, 50, 55]. Note that, we have modified the UOV signature to employ it in the blind signature settings. We call this modified

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

Figure 5: Translated-UOV Signature

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

Figure 6: Verification of the translated-UOV Signature

signature as *translated-UOV*, and we denote the signature scheme as tran-UOV$_{(q,n,m)}$, where (q,n,m) is the algorithm's parameter. We detail the signature algorithm in Figure: 5, and the verification algorithm in Figure: 6. The key generation algorithm of tran-UOV is the same as the key generation algorithm of the UOV-signature scheme [15].

Lemma 1 demonstrates that tran-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 [15]. Several UOV-based signature schemes [13, 40, 41] 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 C that solve the gWMQ with the advantage $\mathsf{Adv}_{gWMQ}(C)$ in time $t + (1+q_s+q_h)\cdot poly(q,n,m)$ so that,*

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

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.

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 $t+O(q_s+q_h)poly(n,m,q)$ 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 tran-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 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.*

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

We outline the main proof ideas for these lemmas in Appendix A. 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 [32]. 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 5. *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 with advantage $\mathsf{Adv}_{\mathsf{MQuBS}}(\mathcal{A})$, there exist adversaries $\mathcal{B}$ and C such that $\mathcal{B}$ challenges the soundness of the NIZK with advantage $\mathsf{Adv}_{\mathsf{SND}}(\mathcal{B})$ and C targets the unforgeability of the basic signature scheme with advantage $\mathsf{Adv}_{\mathsf{SIG}}(C)$. The advantage of $\mathcal{A}$ in breaking the one-more unforgeability of the blind signature scheme is bounded by:*

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

Proof. If the adversary $\mathcal{A}$ successfully breaks the OMUF security of MQuBS, it must generate $N+1$ valid message-signature pairs after making N queries to the blind oracle. Our goal is to use this adversary to break the EUF-CMA security of the transUOV signature scheme. To achieve this, we design a simulator as follows. First, we obtain the verification key and the hash functions—modeled as random oracles—from the challenge oracle associated with the transUOV signature. We then generate all necessary parameters for MQuBS. Finally, we carefully manage the hash and signing queries to complete the simulation.

In this proof, we assume that we can decrypt the ciphertext CT to retrieve the witness r and the message $\mathtt{msg}$. When a signing query is received from the adversary $\mathcal{A}$, we first verify the proof π_{PKE}. If the verification succeeds, we proceed to decrypt CT to retrieve the witness r and the message $\mathtt{msg}$. Due to the soundness property of π_{PKE}, the adversary can only forge a valid proof with an advantage of at most $\mathsf{Adv}_{\mathsf{SND}}(\mathcal{B})$. The successful verification ensures the well-formedness of $\mathbf{b}$ and CT, allowing us to assume the correctness of the relation encoded in the proof. We then query the signing oracle from the unforgeability game with the message $(\mathtt{msg},\mathbf{r})$ and receive the corresponding signature $(\rho,\mathbf{s},\mathsf{seed}_{\mathbf{E}_1})$.

The adversary manages to generate $(q_{RO}+1)$ signed messages $(\mathtt{msg}_i,\rho_i,\mathsf{seed}_{\mathbf{E}_1 i},\pi_{MQ_i})$ 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, that is, $(\mathtt{msg}^*,\rho^*,\mathsf{seed}^*_{\mathbf{E}_1},\pi^*_{MQ})$, for which $\mathtt{msg}^*$ is not previously queried. Using a NIZK extractor, we can extract a witness $(\mathbf{s}^*,\mathbf{r}^*)$ for the relation $\mathbf{E}_1^*\mathcal{P}(\mathbf{s}^*)+\mathbf{E}_2^*\mathcal{R}^*(\mathbf{r}^*)=\mathcal{H}(\rho^*,\mathtt{msg}^*)$. This gives us a forged signature $(\rho,\mathbf{s},\mathsf{seed}_{\mathbf{E}_1})$ for the message $(\mathtt{msg},\mathbf{r})$, which was not part of any previous queries in the unforgeability game for the underlying signature. □

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 3 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 $|CT|+m\log q+|\pi_{\mathsf{PKE}}|$. 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_{PKE}. As per the security level λ, we pick parameters from [42] for a hash-based NIZK proof to build the well-formedness of $\mathbf{b}$ in π_{PKE}. For the MQ-based PKE, we select parameters for HFERP from [43]. However, due to improved cryptanalysis of HFERP [21] that has reduced its security, we carefully choose the parameters for HFERP. Another NIZK proof is required for the well-formedness of CT. Since this is an MQ relation, a NIZK proof for MQ will help us design this proof. Note that the proof size, $|\pi_{\mathsf{PKE}}|$, is not a part of the final signature σ.

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 [20]. 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 [20]. [5] The term $nr\log q$ arises because each iteration of the proof π_{MQ} involves an additional n

[5]The $|\pi|$ denote the proof size of [20].

Table 5: UOV- parameters according to [15]

UOV	NIST SL	n	m	q	$\|\sigma_{UOV}\|$ (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 6: Proof size (KB) for our case

NIST SL	Parameters (n,m,q) Table 5	MQDSS[55] (KB)	with Helper (KB)[10]	MPCitH (KB)[34]	TCitH (KB)[35]	VOLEitH (KB)[20]
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

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) [25]. 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}.

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 [15]. Table 5 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 5, $n = 160$, $m = 64$, and $q = 16$. For 128-bit security level, the size of π_{Com} is 47KB. Hence the size of blind message is $2 * 128 + 64 * \log 16 + |\pi_{\mathrm{Com}}|$-bits. This leads to the size of a blind message β is 47.288KB. Based on the parameters `uov-Is` in Table 5, 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 [56], Beullens's *helper* approach [10], the MPCitH framework [34], the TCitH paradigm [35], and the VOLEitH technique [20]. To compute the proof size of π_{MQ}, we follow the formulas presented in each of these references. The table 6 illustrates the proof size for various security levels.

Table 7: Key and signature sizes for MQuBS at various security levels.

NIST SL	|sk| (B)	|vk| (B)	$\mathcal{U} \rightarrow \mathcal{S}$ (B)	$\mathcal{S} \rightarrow \mathcal{U}$ (B)	$\|\sigma\|$ (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

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 5 also reflects the key sizes for MQuBS. Table 7 shows the key and signature sizes of the MQuBS blind signature algorithm for different security levels.

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 6, 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 [52]. 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 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.* [1] offered a 45KB signature scheme, while an upgraded version proposed by Beullens *et al.* [16] offered a 22KB signature size. This concludes that our design MQuBS offers the shortest signature among PQ round-optimal blind signatures.

However, the main challenge in multivariate cryptography is the efficient PKE. Our construction uses PKE as *encryption to the sky*. It would be more beneficial if we could remove the dependency on MQ-based PKE to build a BS. Although the hardness of the problem relies on the MQ problem, another open issue is proving blindness and anonymity in the QROM model. Thus, we leave two open problems: removing the dependency on MQ-based PKE to build a BS and proving blindness and anonymity in the QROM model.

References

[1] Shweta Agrawal, Elena Kirshanova, Damien Stehlé, and Anshu Yadav. 2022. Practical, round-optimal lattice-based blind signatures. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security.* 39–53.

[2] Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier Verbel. 2020. Improvements of algebraic attacks for solving the rank decoding and MinRank problems. In *Advances in Cryptology–ASIACRYPT 2020*. 507–536.
[3] Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. 2024. One Tree to Rule Them All: Optimizing GGM Trees and OWFs for Post-Quantum Signatures. Cryptology ePrint Archive, Paper 2024/490.
[4] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. 2023. Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. In *Annual International Cryptology Conference*. 581–615.
[5] Carsten Baum, Cyprien Delpech de Saint Guilhem, Daniel Kales, Emmanuela Orsini, Peter Scholl, and Greg Zaverucha. 2021. Banquet: Short and Fast Signatures from AES. Cryptology ePrint Archive, Paper 2021/068.
[6] Carsten Baum and Ariel Nof. 2019. Concretely-Efficient Zero-Knowledge Arguments for Arithmetic Circuits and Their Application to Lattice-Based Cryptography. Cryptology ePrint Archive, Paper 2019/532.
[7] Ryad Benadjila, Thibauld Feneuil, and Matthieu Rivain. 2023. MQ on my Mind: Post-Quantum Signatures from the Non-Structured Multivariate Quadratic Problem. *Cryptology ePrint Archive* (2023).
[8] Daniel J. Bernstein, Andreas Hülsing, Stefan Kölbl, Ruben Niederhagen, Joost Rijneveld, and Peter Schwabe. 2019. The SPHINCS+ Signature Framework. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 2129–2146. doi:10.1145/3319535.3363229
[9] Luk Bettale, Jean-Charles Faugere, and Ludovic Perret. 2009. Hybrid approach for solving multivariate systems over finite fields. *Journal of Mathematical Cryptology* 3, 3 (2009), 177–197.
[10] Ward Beullens. 2019. Sigma protocols for MQ, PKP and SIS, and fishy signature schemes. Cryptology ePrint Archive, Paper 2019/490.
[11] Ward Beullens. 2021. Improved Cryptanalysis of UOV and Rainbow. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 348–373.
[12] Ward Beullens. 2022. Breaking rainbow takes a weekend on a laptop. In *Annual International Cryptology Conference*. Springer, 464–479.
[13] Ward Beullens. 2022. MAYO: practical post-quantum signatures from Oil-and-Vinegar maps. In *Selected Areas in Cryptography: 28th International Conference, Revised Selected Papers*. 355–376.
[14] Ward Beullens. 2024. Multivariate Blind Signatures Revisited. Cryptology ePrint Archive, Paper 2024/720.
[15] Ward Beullens, Ming-Shing Chen, Jintai Ding, Boru Gong, Matthias J. Kannwischer, Jacques Patarin, Bo-Yuan Peng, Dieter Schmidt, Cheng-Jhih Shih, Chengdong Tao, and Bo-Yin Yang. 2018. UOV: Unbalanced Oil and Vinegar Algorithm Specifications and Supporting Documentation Version 1.0. https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/UOV-spec-web.pdf
[16] Ward Beullens, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. 2023. Lattice-based blind signatures: Short, efficient, and round-optimal. In *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*. 16–29.
[17] Olivier Blazy, Philippe Gaborit, and Dang Truong Mac. 2022. A Correction to a Code-Based Blind Signature Scheme. In *Code-Based Cryptography*, Antonia Wachter-Zeh, Hannes Bartz, and Gianluigi Liva (Eds.). Springer International Publishing, Cham, 84–94.
[18] Olivier Blazy, Philippe Gaborit, Julien Schrek, and Nicolas Sendrier. 2017. A code-based blind signature. In *2017 IEEE International Symposium on Information Theory (ISIT)*. 2718–2722. doi:10.1109/ISIT.2017.8007023
[19] Stefan Brands. 1994. 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, 302–318.
[20] Dung Bui. 2024. Shorter VOLEitH Signature from Multivariate Quadratic. Cryptology ePrint Archive, Paper 2024/465.
[21] Max Cartor, Ryann Cartor, Hiroki Furue, and Daniel Smith-Tone. 2024. Improved Cryptanalysis of HFERP. In *Public-Key Cryptography – PKC 2024*, Qiang Tang and Vanessa Teague (Eds.). Springer Nature Switzerland, Cham, 413–440.
[22] Sanjit Chatterjee, M Prem Laxman Das, and Tapas Pandit. 2022. Revisiting the security of salted UOV signature. In *International Conference on Cryptology in India*. Springer, 697–719.
[23] David Chaum. 1983. Blind signatures for untraceable payments. In *Advances in Cryptology: Proceedings of Crypto 82*. Springer, 199–203.
[24] David Chaum and Torben Pryds Pedersen. 1992. Wallet databases with observers. In *Annual international cryptology conference*. Springer, 89–105.
[25] Lily Chen, D Moody, and YK Liu. 2017. NIST Post-Quantum Cryptography Standardization. *Transition* 800 (2017), 131A.
[26] Benoît Cogliati, Pierre-Alain Fouque, Louis Goubin, and Brice Minaud. 2024. New Security Proofs and Techniques for Hash-and-Sign with Retry Signature Schemes. Cryptology ePrint Archive, Paper 2024/609.
[27] Cyprien Delpech de Saint Guilhem, Emmanuela Orsini, and Titouan Tanguy. 2021. Limbo: Efficient Zero-knowledge MPCitH-based Arguments. Cryptology ePrint Archive, Paper 2021/215.
[28] F Denis, F Jacobs, and CA Wood. 2023. RFC 9474 RSA Blind Signatures. (2023).
[29] Jintai Ding and Dieter Schmidt. 2005. Rainbow, A New Multivariable Polynomial Signature Scheme. In *International conference on applied cryptography and network security*. Springer, 164–175.
[30] Leo Ducas, Tancrede Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehle. 2017. CRYSTALS – Dilithium: Digital Signatures from Module Lattices. Cryptology ePrint Archive, Paper 2017/633. https://eprint.iacr.org/2017/633 https://eprint.iacr.org/2017/633.
[31] Dung Hoang Duong, Xuan Thanh Khuc, Youming Qiao, Willy Susilo, and Chuanqi Zhang. 2025. Blind Signatures from Cryptographic Group Actions. Cryptology ePrint Archive, Paper 2025/397. https://eprint.iacr.org/2025/397
[32] Morris J Dworkin. 2015. SHA-3 standard: Permutation-based hash and extendable-output functions. (2015).
[33] Thibauld Feneuil. 2023. *Post-Quantum Signatures from Secure Multiparty Computation*. Ph. D. Dissertation. Sorbonne Université.
[34] Thibauld Feneuil and Matthieu Rivain. [n. d.]. MQOM: MQ on my mind-algorithm specifications and supporting documentation. Version 1.0-31 May 2023.
[35] Thibauld Feneuil and Matthieu Rivain. 2023. Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments. Cryptology ePrint Archive, Paper 2023/1573.
[36] Marc Fischlin. 2006. Round-optimal composable blind signatures in the common reference string model. In *Annual International Cryptology Conference*. Springer, 60–77.
[37] Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Prest, Thomas Ricosset, Gregor Seiler, William Whyte, and Zhenfei Zhang. 2020. Falcon: Fast-Fourier Lattice-based Compact Signatures over NTRU. https://falcon-sign.info/falcon.pdf.
[38] Georg Fuchsbauer, Antoine Plouviez, and Yannick Seurin. 2020. 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, 63–95.
[39] Georg Fuchsbauer and Mathias Wolf. 2024. Concurrently Secure Blind Schnorr Signatures. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 124–160.
[40] Hiroki Furue, Yasuhiko Ikematsu, Fumitaka Hoshino, Yutaro Kiyomura, Tsunekazu Saito, and Tsuyoshi Takagi. 2023. QR-UOV. (2023).
[41] Anindya Ganguly, Angshuman Karmakar, and Nitin Saxena. 2023. VDOO: A Short, Fast, Post-Quantum Multivariate Digital Signature Scheme. In *International Conference on Cryptology in India*. Springer, 197–222.
[42] Irene Giacomelli, Jesper Madsen, and Claudio Orlandi. 2016. ZKBoo: Faster Zero-Knowledge for Boolean Circuits. In *25th usenix security symposium (usenix security 16)*. 1069–1083.
[43] Yasuhiko Ikematsu, Ray Perlner, Daniel Smith-Tone, Tsuyoshi Takagi, and Jeremy Vates. 2018. HFERP - A New Multivariate Encryption Scheme. In *Post-Quantum Cryptography*, Tanja Lange and Rainer Steinwandt (Eds.). Springer International Publishing, Cham, 396–416.
[44] Yuval Ishai, Eyal Kushilevitz, Rafail Ostrovsky, and Amit Sahai. 2007. Zero-knowledge from secure multiparty computation. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. 21–30.
[45] David S Johnson and Michael R Garey. 1979. *Computers and Intractability: A Guide to the Theory of NP-completeness*. WH Freeman.
[46] Daniel Kales and Greg Zaverucha. 2022. Efficient Lifting for Shorter Zero-Knowledge Proofs and Post-Quantum Signatures. Cryptology ePrint Archive, Paper 2022/588.
[47] Shuichi Katsumata, Yi-Fu Lai, Jason T LeGrow, and Ling Qin. 2024. CSI-Otter: Isogeny-based (partially) blind signatures from the class group action with a twist. *Designs, Codes and Cryptography* 92, 11 (2024), 3587–3643.
[48] Aviad Kipnis, Jacques Patarin, and Louis Goubin. 1999. Unbalanced Oil and Vinegar Signature Schemes. In *International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 206–222.
[49] Aviad Kipnis and Adi Shamir. 1998. Cryptanalysis of the Oil and Vinegar Signature Scheme. In *Annual international cryptology conference*. Springer, 257–266.
[50] Haruhisa Kosuge and Keita Xagawa. 2022. Probabilistic Hash-and-Sign with Retry in the Quantum Random Oracle Model. Cryptology ePrint Archive, Paper 2022/1359.
[51] Veronika Kuchta, Jason T. LeGrow, and Edoardo Persichetti. 2025. Post-Quantum Blind Signatures from Matrix Code Equivalence. Cryptology ePrint Archive, Paper 2025/274. https://eprint.iacr.org/2025/274
[52] National Institute of Standards and Technology (NIST). 2024. NIST Announces Second Round of Post-Quantum Cryptography Digital Signature Standardization. https://csrc.nist.gov/news/2024/pqc-digital-signature-second-round-announcement Accessed: October 24, 2024.
[53] Albrecht Petzoldt, Alan Szepieniec, and Mohamed Saied Emam Mohamed. 2017. 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, 437–454.
[54] John Proos and Christof Zalka. 2003. Shor's discrete logarithm quantum algorithm for elliptic curves. *arXiv preprint quant-ph/0301141* (2003).
[55] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. 2011. On provable security of UOV and HFE signature schemes against chosen-message attack. In *Post-Quantum Cryptography: 4th International Workshop, PQCrypto 2011.* Springer, 68–82.
[56] Koichi Sakumoto, Taizo Shirai, and Harunaga Hiwatari. 2011. 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, 706–723.
[57] Peter W Shor. 1994. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science.* Ieee, 124–134.
[58] Alan Szepieniec and Bart Preneel. 2015. New techniques for electronic voting. In *USENIX Journal of Election Technology and Systems (JETS).*
[59] Enrico Thomae and Christopher Wolf. 2012. Solving underdetermined systems of multivariate quadratic equations revisited. In *International workshop on public key cryptography.* Springer, 156–171.
[60] David Wagner. 2002. A generalized birthday problem. In *Annual International Cryptology Conference.* Springer, 288–304.

A Security Proofs

Proof of Theorem 1: The Correctness of MQuBS

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

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

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

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

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

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

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

Since the correctness of the UOV signature, and the correctness of NIZK proofs (both π_{PKE}, and π_{MQ}) are independent of each other, the correctness of MQuBS can be expressed as follows.

$$\Pr_{\mathsf{MQuBS}}=\Pr_{\mathsf{UOV}}\times\Pr_{\pi}$$

First, the signer verifies the NIZK proof π_{PKE} for the well-formedness of $\mathbf{b}$ and CT, as the correctness of the NIZK proof guarantees its validity. If the proof is correct, the signer proceeds to compute the signature.

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 $\Pr_{UOV}=1$.

In the third part, we use the correctness of the π_{MQ} protocol (see [20]). 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 1.

First random oracle programming: $\mathcal{G}(\mathbf{r})$
```
1:  if $\exists\,\rho\in\{0,1\}^{2\lambda}:(\mathbf{r},\rho)\in\mathsf{List}_{\mathcal{G}}$
2:      $\rho\leftarrow\{\rho_1\in\{0,1\}^{2\lambda}\mid(\mathbf{r},\rho_1)\in\mathsf{List}_{\mathcal{G}}\}$
3:      return $\rho$
4:  $\rho\leftarrow\{0,1\}^{2\lambda}$
5:  $\mathsf{List}_{\mathcal{G}}=\mathsf{List}_{\mathcal{G}}\cup\{(\mathbf{r},\rho)\}$
6:  return $\rho$
```

Second random oracle programming: $\mathcal{K}(\mathsf{msg},\rho)$
```
1:  if $\exists\,\mathsf{seed}\in\{0,1\}^{2\lambda}:(\mathsf{seed},\mathsf{msg},\rho)\in\mathsf{List}_{\mathcal{K}}$
2:      $\mathsf{seed}\leftarrow\{\mathsf{seed}_1\in\{0,1\}^{2\lambda}\mid(\mathsf{seed}_1,\mathsf{msg},\rho)\in\mathsf{List}_{\mathcal{K}}\}$
3:      return seed
4:  $\mathsf{seed}\leftarrow\{0,1\}^{2\lambda}$
5:  if $(\mathbf{r},\rho)\in\mathsf{List}_{\mathcal{G}}$
6:      $\mathsf{List}_{\mathcal{K}}=\mathsf{List}_{\mathcal{K}}\cup\{(\mathsf{seed},\mathsf{msg},\rho)\}$
7:  else
8:      $\rho\leftarrow\{0,1\}^{2\lambda}$
9:      $\mathsf{List}_{\mathcal{G}}=\mathsf{List}_{\mathcal{G}}\cup\{(\mathbf{r},\rho)\}$
10:     $\mathsf{List}_{\mathcal{K}}=\mathsf{List}_{\mathcal{K}}\cup\{(\mathsf{seed},\mathsf{msg},\rho)\}$
11: return seed
```

Figure 7: Two Random Oracle Programming

Now, combine all values to compute the correctness of MQuBS. Finally, $\Pr_{\mathsf{MQuBS}}=1$. Therefore, combining all the probabilities, we can claim that the verifier of MQuBS blind signature will correctly verify the signature with overwhelming probability. □

Proof of Lemma 2: EUF-CMA to EUF-KO-security

Statement. Let $\mathcal{A}$ be a PPT adversary, 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 $t+O(q_s+q_h)\mathrm{poly}(n,m,q)$ so that the following conditions will hold.

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

Proof. The proof start with simulating the Hash and the Signing oracles. The EUF-KO adversary $\mathcal{B}$ follows the following steps. It has the public key $\mathcal{P}_{UOV}$ and it starts simulating $\mathcal{A}$ on the input $\mathcal{P}_{UOV}$. The adversary $\mathcal{B}$ starts simulating the random oracle queries by maintaining lists. To simulate the signing oracle for the adversary, we assume hash function $\mathcal{G},\mathcal{K}$ and $\mathcal{H}$ as random oracles (RO), producing outputs that follow a random distribution and remain consistent for repeated queries.

Hash oracle simulation. To manage hash queries, we maintain three lists: $\mathsf{List}_{\mathcal{G}}$, $\mathsf{List}_{\mathcal{K}}$ (see Algorithm 7) and $\mathsf{List}_{\mathcal{H}}$ (see Algorithm 8). For each new query, $\mathcal{G}$ randomly selects $\rho\in_U\{0,1\}^{2\lambda}$. Similarly, for each new query $\mathcal{K}$ randomly selects $\mathsf{seed}\in_U\{0,1\}^{2\lambda}$. Simultaneously, the oracle $\mathcal{H}$ computes and lists three seeds $\mathsf{seed}\leftarrow(\mathsf{seed}_1,\mathsf{seed}_2,\mathsf{seed}_3)$ to construct $\mathbf{E}_1$, $\mathbf{E}_2$, and $\mathcal{R}$. It also selects a

```
Third random oracle programming: $\mathcal{H}(\mathtt{msg},\rho)$
if $\exists\, \mathbf{t} \in \mathbb{F}_q^m$: $(\mathbf{t},\mathtt{msg},\rho,\mathtt{seed}) \in \mathtt{List}_{\mathcal{H}}$
    $\mathbf{t} \leftarrow \{\mathbf{t}_1 \in \mathbb{F}_q^m \mid (\mathbf{t}_1,\mathtt{msg},\rho) \in \mathtt{List}_{\mathcal{H}}\}$
    return $\mathbf{t}$
if $(\cdot,\mathtt{msg},\rho) \in \mathtt{List}_{\mathcal{K}}$
    $\mathbf{E}_1 \leftarrow \mathtt{XOF}(\mathtt{seed}_1)$
else
    $\mathtt{seed}_1 \in \{0,1\}^{2\lambda}$
    $\mathtt{List}_{\mathcal{K}} = \mathtt{List}_{\mathcal{K}} \cup \{(\mathtt{seed}_1,\mathtt{msg},\rho)\}$
$\mathtt{seed}_2 \in \{0,1\}^{2\lambda}$, $\mathtt{seed}_3 \in \{0,1\}^{2\lambda}$
$\mathtt{List}_{\mathcal{K}} = \mathtt{List}_{\mathcal{K}} \cup \{(\mathtt{seed}_2,\mathtt{msg},\mathtt{seed}_1)\}$
$\mathtt{List}_{\mathcal{K}} = \mathtt{List}_{\mathcal{K}} \cup \{(\mathtt{seed}_3,\mathtt{msg},\mathtt{seed}_2)\}$
$\mathbf{E}_2 \leftarrow \mathtt{XOF}(\mathtt{seed}_2)$, $\mathcal{R} \leftarrow \mathtt{XOF}(\mathtt{seed}_3)$
sample $\mathbf{s} \in \mathbb{F}_q^m$
set $\mathbf{t} \leftarrow \mathbf{E}_1\mathcal{P}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$
sample $\mathbf{s} \xleftarrow{\$} \mathbb{F}_q^n$, and compute $\mathcal{P}_{UOV}(\mathbf{s})$
$\mathtt{seed} \leftarrow (\mathtt{seed}_1,\mathtt{seed}_2,\mathtt{seed}_3)$
$\mathtt{List}_{\mathcal{H}} = \mathtt{List}_{\mathcal{H}} \cup \{(\mathbf{t},\mathtt{msg},\rho,\mathtt{seed})\}$
$\mathtt{List}'_{\mathcal{H}} = \mathtt{List}'_{\mathcal{H}} \cup \{(\mathbf{r},\mathbf{t},\mathbf{s})\}$
return $\mathbf{t}$

Simulating signing oracle: $O_{\mathsf{Sign}}(\mathtt{msg},\mathbf{r})$
$\rho \leftarrow \mathcal{G}(\mathbf{r})$
$\mathbf{t} \leftarrow \mathcal{H}(\rho,\mathtt{msg})$
$(\mathbf{r}',\mathbf{s}') \leftarrow \{(\mathbf{r}',\mathbf{s}') \in \mathbb{F}_q^m \times \mathbb{F}_q^n \mid (\mathbf{r}',\mathbf{s}',\mathbf{t}) \in \mathtt{List}'_{\mathcal{H}}\}$
if $\mathbf{r}' \neq \mathbf{r}$, abort
return $\mathbf{s}'$
```

Figure 8: Oracle Programming for $\mathcal{H}$ and O_{Sign}.

random $\mathbf{s} \in_U \mathbb{F}_q^n$ and computes $\mathbf{t} \leftarrow \mathbf{E}_1\mathcal{P}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$. Because $\mathcal{P}$ is uniformly distributed and $\mathcal{R}$ is a random map, $\mathbf{t}$ is statistically indistinguishable from a uniform random distribution. The pre-image $(\mathbf{r},\mathbf{s})$ is stored in a new list, $\mathtt{List}'_{\mathcal{H}}$, which tracks the pre-images for all $\mathbf{t}$ values output by $\mathcal{H}$.

Signing oracle O_{Sign}. We start with simulating the signing oracle for the adversary. Since we are assuming $\mathcal{G}$, $\mathcal{K}$ and $\mathcal{H}$ are random oracles (RO), so the output of the oracle follows random distribution. Additionally, outputs are consistent for repeated queries. To ensure consistency, the algorithm uses three lists, $\mathtt{List}_{\mathcal{G}}$, $\mathtt{List}_{\mathcal{K}}$ and $\mathtt{List}_{\mathcal{H}}$ to handle random oracle queries. For random outputs, $\mathcal{G}$ samples $\rho \in_U \{0,1\}^{2\lambda}$ for each new query. The oracle $\mathcal{K}$ helps to compute seed for random outputs. Now $\mathcal{H}$ compute three seeds using $\mathcal{K}$ and generates $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathcal{R}$ using the XOF. It also generates a random $\mathbf{s} \in_U \mathbb{F}_q^n$ and computes $\mathbf{t} \leftarrow \mathbf{E}_1\mathcal{P}(\mathbf{s}) + \mathbf{E}_2\mathcal{R}(\mathbf{r})$. Clearly, $\mathbf{t}$ is statistically indistinguishable from the uniform random distribution. Since $\mathcal{P}$ has uniform and $\mathcal{R}$ is a random map. Now the pre-image $(\mathbf{r},\mathbf{s})$ is listed in a new list $\mathtt{List}'_{\mathcal{H}}$. The list $\mathtt{List}'_{\mathcal{H}}$ contains pre-image $(\mathbf{r},\mathbf{s})$ for $\mathbf{t}$ that ever been output by $\mathcal{H}$.

When the adversary $\mathcal{A}$ asks for a signature on $(\mathtt{msg},\mathbf{r})$, $\mathcal{B}$ follows the earlier step. It outputs $(\rho,\mathbf{s},\mathtt{seed}_{\mathbf{E}_1})$ as signature to the adversary $\mathcal{A}$. However, it aborts when $\mathbf{r} \neq \mathbf{r}'$. Now, when $\mathcal{A}$ outputs the message-signature pair, the adversary $\mathcal{B}$ outputs the same pair.

The time complexity to perform these steps is $t + O(1 + q_h + q_s)\text{poly}(q,n,m)$. So the only thing that remains in the proof is to compute the probability of $\mathcal{B}$ to succeed the EUF-KO game.

Hybrid_0 : This game is played by the adversary $\mathcal{A}$ against the EUF-CMA game of the translated − UOV signature. Hence, from the definition, $\Pr[\mathsf{Hybrid}_0 = 1] = \mathsf{Adv}_{\mathsf{SIG}}(\mathcal{A})$.

Hybrid_1 : The hybrid Hybrid_1 is identical with Hybrid_0, except that the game aborts and outputs 0, if to answer a signing query $(\mathtt{msg},\mathbf{r})$, the random oracle already queried on input $(\mathtt{msg},\mathbf{r})$. Hence the probability of an abort is at most $\frac{(q_s+q_h)}{2^{2\lambda}}$ for each signing query. Hence the total probability of an abort is $\frac{q_s(q_s+q_h)}{2^{2\lambda}}$. So we have, $\Pr[\mathsf{Hybrid}_1() = 1] \geq \Pr[\mathsf{Hybrid}_0() = 1] - \frac{q_s(q_s+q_h)}{2^{2\lambda}}$.

Hybrid_2 : This game is the EUF-KO game played by the adversary $\mathcal{B}$. If the earlier game does not abort, then the views of $\mathcal{A}$ are same in both hybrids. Since there is no abort, so $\mathcal{B}$ simulates the random oracles perfectly. As per UOV signature, $\mathbf{v} \in \mathbb{F}_q^n$ is random, so $\mathbf{o}$ is random; and hence $\mathbf{s}$ is random. The other values in the signature ρ, $\mathtt{seed}_{\mathbf{E}_1}$ is also random due their properties. Finally, the probability that $\mathcal{A}$ outputs a forgery in Hybrid_1 is at least as big as the probability that it outputs a forgery in Hybrid_2. Therefore, we have $\mathsf{Adv}_{\mathsf{KO}}(\mathcal{B}) > \Pr[\mathsf{Hybrid}_1() = 1]$. Hence, by combining all the inequalities, we arrive at the final result. □

Proof of Lemma 3: The EUF-KO-security

Lemma 4 (UOV and gWMQ to EUF-KO-security). *Let $\mathcal{A}$ be a EUF-KO adversary that runs in time t against the tran-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 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.*

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

Proof. The following hybrids help to construct the proof.

Hybrid_0: This hybrid is played by the adversary $\mathcal{A}$ against the EUF-KO game.

Hybrid_1: This hybrid experiment is structurally identical to the previous one, with a primary difference in the key generation phase. Specifically, the challenger samples a uniformly random polynomial map $\mathcal{P} : \mathbb{F}_q^n \to \mathbb{F}_q^m$ in place of the structured map $\mathcal{P}_{\mathsf{UOV}}$. Consequently, if the adversary $\mathcal{B}$ is provided with $\mathcal{P}_{\mathsf{UOV}}$, the experiment corresponds to Hybrid_0; if it is provided with the random map $\mathcal{P}$, it corresponds to Hybrid_1. In the case where $\mathcal{B}$ receives $\mathcal{P}$, it computes the matrix representations of the component polynomials $\{\mathbf{P}_i^{(1)}, \mathbf{P}_i^{(2)}, \mathbf{P}_i^{(3)}\}_{i\in[m]}$ constituting $\mathcal{P}$. Subsequently, it samples a seed seed and queries the oracle $\mathcal{A}$ on the input $\{\mathtt{seed}, \mathbf{P}_i^{(3)}\}_{i\in[m]}$. The adversary's distinguishing advantage between the two hybrids is thus defined as

$$\mathsf{Adv}_{\mathsf{UOV}}(\mathcal{B}) = \left|\Pr[\mathsf{Hybrid}_0() = 1] - \Pr[\mathsf{Hybrid}_1() = 1]\right|.$$

Hybrid$_2$: We now proceed to the next hybrid. In this experiment, the adversary C operates against an instance of the generalized Witness Multivariate Quadratic (gWMQ) problem. Given an instance $(\mathcal{P},\mathcal{R},\mathbf{E}_1,\mathbf{E}_2,\mathbf{t})$, the adversary C performs the same steps as in the previous hybrid. The primary difference lies in the decision procedure. Specifically, C selects an index $l \in [q_h]$ uniformly at random, corresponding to one of the q_h distinct random oracle queries made by $\mathcal{A}$. It then programs the oracle to return $\mathcal{H}(\mathtt{msg},\mathcal{G}(\mathbf{r}))$ as the response to the l-th distinct query. If $\mathcal{A}$ outputs a valid signature $(\mathtt{msg},\mathbf{r};\rho,\mathbf{s},\mathtt{seed}_{\mathbf{E}_1})$, then C checks whether $(\mathtt{msg},\mathbf{r})$ was the l-th random oracle query. If so, C outputs $(\rho,\mathbf{s})$; otherwise, it aborts. Since the view of $\mathcal{A}$ in this hybrid is identical to that in Hybrid$_1$, the probability that it outputs a valid signature is $\Pr[\text{Hybrid}_1()=1]$. The probability that $\mathcal{A}$ produces a valid signature without having queried the random oracle on $(\mathtt{msg},\mathbf{r})$ is at most $1/q^m$.

Information-theoretically, $\mathcal{A}$ does not know the index l. Hence, the probability that it outputs a valid forgery corresponding to the l-th random oracle query is at most $1/(1+q_h)$. Therefore, we obtain the following lower bound on the advantage of C in solving the gWMQ problem:

$$\mathsf{Adv}_{\mathsf{gWMQ}}(C) \geq \frac{\left(\Pr[\text{Hybrid}_1()=1]-\frac{1}{q^m}\right)}{1+q_h}.$$

Combining all the hybrids, we derive the final bound:

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

□

B Algebraic Cryptanalysis

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

B.0.1 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 [13].

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 $\mathbf{r}_{\text{adv}}$, followed by the computation of $\mathcal{R}_1(\mathbf{r}_{\text{adv}})$. 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 [60].

B.0.2 Beullens's [14] attack is not applicable. Since in our case, we have random polynomials, so the Equation 5 remains quadratic. Therefore, Beullens's polar form attack [14] can not convert the quadratic system to a linear system.

B.0.3 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 [59]. Then it runs the hybrid WiedemannXL algorithm [9] 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.

B.0.4 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 $\operatorname{rank}(Q)$ subject to $\operatorname{rank}(Q)\leq r$. 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 [2, 49].

B.0.5 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 [11]. 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 $\mathbf{M}_iO$ 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 $k=2$.

C Basic Cryptographic Primitives

C.1 Commitment Schemes

A *commitment scheme* Com = {Com.Setup, Commit, Open} enables a user to commit to a message while keeping it hidden from the verifier at the time of commitment. Subsequently, the verifier can open the commitment to verify the value.

$\underline{\mathsf{Com.Setup}(1^{\lambda})\rightarrow \mathsf{crs}}$. It takes a security parameter λ as input and outputs a common reference string (CRS) crs.

$\underline{\mathsf{Commit}(\mathsf{crs},\mu;r)\rightarrow c}$. The user first commits a chosen message μ using a crs and randomness r. The resultant value c is known as *committed* value. The user sends the c to the receiver.

$\underline{\mathsf{Open}(\mathsf{crs},\mu,r,c)\rightarrow 0/1}$. In the opening phase, the user sends the crs, μ, and r to the verifier to check whether the committed value c is valid or not. After opening the commitment, the verifier outputs 1 if *accept*, else 0.

C.1.1 Security properties of a commitment scheme. In the following, we define key security properties of a commitment scheme.

Correctness. A Com is considered *correct* if, for every $\lambda\in\mathbb{N}$, any message μ, and a correctly generated crs $\leftarrow$ Com.Setup(1^{λ}), the following expression holds:

$$\Pr\left[\mathsf{Open}(\mathsf{crs},\mu,r,c)=1\,\middle|\,\begin{array}{l}\mathsf{crs}\leftarrow\mathsf{Com.Setup}(1^{\lambda})\\ r\leftarrow\{0,1\}^{\lambda},\ c\leftarrow\mathsf{Commit}(\mathsf{crs},\mu;r)\end{array}\right]=1$$

Perfectly binding. A commitment scheme is known to be binding if, for any two different chosen messages, the committed value should be different.

Definition 8 (Perfectly binding). *A commitment scheme* Com = {Com.Setup, Commit, Open} *is* perfectly binding, *if for any* $\lambda\in\mathbb{N}$, crs $\leftarrow$ Com.Setup(1^{λ}), *and for two different message* $\mu_0\neq\mu_1$, *the following conditions hold for at least one* $i\in\{0,1\}$.

$$\Pr[\mathsf{Open}(\mathsf{crs},\mu_i,r_i,c)=1]=0$$

Computationally hiding. This property ensures that no information about the message can be inferred from its commitment as long as the openings remain hidden.

Definition 9 (Computationally hiding). *A* Com *is computationally hiding if, for any probabilistic polynomial-time (PPT) algorithm* $\mathcal{A}$ *and for all* $\lambda\in\mathbb{N}$, *the following conditions are satisfied.*

$$\Pr\left[\mathcal{A}(c)=b\,\middle|\,\begin{array}{l}\mathsf{crs}\leftarrow\mathsf{Com.Setup}(1^{\lambda})\\ m_0,m_1\leftarrow\mathcal{A}(\mathsf{crs}),b\leftarrow\{0,1\}\\ r\leftarrow\{0,1\}^{\lambda},\ c\leftarrow\mathsf{Commit}(\mathsf{crs},\mu_b;r)\end{array}\right]=1$$

C.1.2 Security properties of a proof system. The key security properties of a NIZK proof system are outlined below.

Definition 10 (Completeness). *The* completeness *property ensures that, for any security parameter* $\lambda\in\mathbb{N}$, crs $\leftarrow$ NIZK.SetUp(1^{λ}), *and a random instance* $x\in\mathcal{L}$ *with its corresponding witness* w, *the following expression will hold.*

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

Definition 11 (Soundness). *A* NIZK *proof is considered* computationally sound *if, for every stateful PPT adversary* $\mathcal{A}$ *and for each* $\lambda\in\mathbb{N}$, *there exists a negligible function* negl(λ) *such that the following condition holds.*

$$\Pr\left[\mathsf{NIZK.Verify}(\mathsf{crs},x,\pi)=1\wedge x\notin\mathcal{L}\,\middle|\,\begin{array}{l}\mathsf{crs}\leftarrow\mathsf{NIZK.SetUp}(1^{\lambda})\\ (x,\pi)\leftarrow\mathcal{A}(1^{\lambda},\mathsf{crs})\end{array}\right]\leq\mathsf{negl}(\lambda)$$

Definition 12 (Knowledge extractor). *A PPT extractor* $\mathcal{W}$ *of a* NIZK *proof system is aid to be a* knowledge extractor, *if for every PPT adversary* $\mathcal{A}$ *and all* $\lambda\in\mathbb{N}$ *the following conditions hold.*

$$\Pr\left[\begin{array}{l}\mathsf{NIZK.Verify}(\mathsf{crs},x,\pi)=1\bigwedge\\ w=\mathcal{W}(x,\pi)\ \textit{is not a valid}\\ \textit{witness for } x\in\mathcal{L}\end{array}\,\middle|\,\begin{array}{l}\mathsf{crs}\leftarrow\mathsf{NIZK.SetUp}(1^{\lambda})\\ (x,\pi)\leftarrow\mathcal{A}(1^{\lambda},\mathsf{crs})\end{array}\right]\leq\mathsf{negl}(\lambda)$$

C.2 Zero Knowledge Proof for MQ problem

Sakumoto et al [55] first built a three-round and a five-round zero-knowledge proof from the MQ problem. The soundness error for the five round protocols is $\frac{1}{2}+\frac{1}{2q}$ where the three round protocol has the soundness error $\frac{2}{3}$. This clearly shows that the five-round protocol is much more practical. Later, Beullens used *helper* to further reduced the soundness error to $\frac{1}{q'}$ (q' is any number bounded by the field size q) and made it more practical [10]. However, this needs a lot of computation. To overcome these issues, Feneuil used the MPC-in-the-Head paradigm and built a ZKP for the MQ problem [33].

The MPC-in-the-Head paradigm offers a flexible framework for crafting zero-knowledge proofs of knowledge, leveraging the secure multi-party computation (MPC) techniques [44]. In this paradigm, the prover divides the secret witness and, mentally, engages in an MPC protocol with N parties, independently committing to each party's view. The verifier then challenges the prover to disclose the views of a random subset of parties. Due to the privacy of the MPC protocol, no information about the witness is revealed, ensuring the zero-knowledge property. Conversely, a malicious prover attempting to deceive at least one party will likely be exposed by the verifier, ensuring the soundness property. When combined with the Fiat-Shamir transform, the MPCitH paradigm becomes a valuable tool for constructing practical signatures. The security of the resulting scheme depends solely on the security of the commitment and hash functions, as well as the security of a one-way function—wherein we employ the MQ problem as one such function.

C.2.1 MPC protocols for the MQ problem. Consider a scenario where the prover seeks to demonstrate knowledge of the solution vector, denoted as $\mathbf{x}$, for the quadratic system $\mathcal{P}$. In this context, even without direct access to $\mathbf{x}$, the verifier can be convinced of the prover's knowledge through a proof accompanied by the tuple $\langle\mathbf{w},\mathcal{P}\rangle$. Here, $\mathbf{w}=\mathcal{P}(\mathbf{x})$ represents the evaluation of $\mathcal{P}$ at $\mathbf{x}$. This quadratic system is expanded below.

$$\begin{aligned}w_1&=\mathbf{x}^{\top}P_1\mathbf{x}\\ w_2&=\mathbf{x}^{\top}P_2\mathbf{x}\\ &\vdots\\ w_m&=\mathbf{x}^{\top}P_m\mathbf{x}\end{aligned}$$

Here, P_i denotes a matrix representation of a homogeneous quadratic polynomial p_i in the quadratic system $\mathcal{P}$ and $\mathbf{w}:=(w_1,w_2,\cdots,w_m)\in$

$\mathbb{F}_q^m$. Now the above system can be easily batched (according to [7, 33]) as follows.

$$\sum_{i=1}^{m}\gamma_i(w_i - \mathbf{x}^\top P_i \mathbf{x}) = 0$$

$$\begin{aligned}\sum_{i=1}^{m}\gamma_i(w_i) &= \sum_{i=1}^{m}\gamma_i \mathbf{x}^\top P_i \mathbf{x}\\ &= \sum_{i=1}^{m}\gamma_i \mathbf{x}^\top P_i \mathbf{x}\\ &= \langle \mathbf{x}, \mathrm{p}\rangle \text{ where } \mathrm{p} = (\sum_{i=1}^{m}\gamma_i P_i)\mathbf{x}\end{aligned}$$

Here, γ_i sampled randomly from an extension field $\mathbb{F}_{q^\eta}$. To prove, the solution, it is enough to prove the following identity.

$$\mathrm{w} = \langle \mathbf{x}, \mathrm{p}\rangle; \text{ where } \mathrm{w} = \sum_{i=1}^{m}\gamma_i(w_i).$$

The only remaining task is to deploy an MPC protocol that verifies three matrices X, Y, and Z, ensuring that Z is the product of X and Y. The literature contains established MPC protocols for checking inner products or matrix multiplication, which are extensively used [5, 6, 27, 46].