Advances in Mathematics of Communications
Vol. 19, No. 5, October 2025, pp. 1336-1359
doi:10.3934/amc.2024057

QUANTUM SECURE DIGITAL SIGNATURE SCHEME BASED ON MULTIVARIATE QUADRATIC QUASIGROUPS (MQQ)

Satish Kumar[1], Indivar Gupta[*2] and Ashok Ji Gupta[1]

[1]Department of Mathematical Sciences,
Indian Institute of Technology (BHU), Varanasi-221005, India

[2]Scientific Analysis Group,
Defence Research & Development Organisation, New Delhi-110054, India

(Communicated by Massimiliano Sala)

Abstract. In this paper, we propose a digital signature scheme, the MQQ-Sigv scheme, that relies on the difficulty of solving the multivariate quadratic (MQ) problem. The central map of the proposed scheme will be designed using the multivariate quadratic quasigroup (MQQ). We will prove that the MQQ-Sigv scheme is secure against various attacks including existential unforgeability under chosen message attack (EUF-CMA), Min-Rank attack, High-Rank attack, Direct attack, and Differential attack. Furthermore, we will prove that finding an equivalent good key for the MQQ-Sigv scheme is infeasible in polynomial time, and analyze the operating characteristics of the scheme.

1. **Introduction.** In this modern era, where communication and data sharing via the internet is very common, ensuring the confidentiality and integrity of the data transferred between two systems is one of the biggest concern. Public-key cryptographic primitives have played a crucial role in safeguarding data security. Classical public-key cryptographic algorithms like RSA and ElGamal cryptosystems rely on the difficulty of finding the prime factorization of large positive integers and solving the discrete logarithmic problem (DLP), respectively. In [46], Peter Shor proposed an algorithm that proves factoring a large positive integer using quantum computers is possible in polynomial time. Subsequently, solving the DLP over natural numbers and algebraic groups can also be feasible in polynomial time by using quantum computers. As quantum computers advance, various classical public-key cryptographic algorithms are anticipated to become vulnerable to security breaches. Post-quantum cryptography plays a critical role following the emergence of quantum computers. Multivariate public-key cryptography (MPKC) falls within this category and has been an active research area for over two decades. MPKC schemes offer several advantages compared to the classical public-key cryptosystems such as RSA, ElGamal, or elliptic curve cryptography (ECC). These advantages stem from MPKC's highly parallelizable nature, and, as of now, no known quantum algorithm exist which can break multivariate quadratic public-key cryptographic schemes. Finding the solution of a system of multivariate polynomial equations over a finite field is

2020 *Mathematics Subject Classification.* 20N05, 14G50, 12E20, 94A62.

Key words and phrases. Non-associative group, quasigroups, public-key cryptography, Multivariate public-key cryptosystem, multivariate quadratic quasigroup (MQQ)..

*Corresponding author: Indivar Gupta.

widely recognized as an NP-complete problem, and MPKC relies on the inherent complexity of this problem.

In 1849, Euler [8] published a paper introducing a new theory on Latin squares with the property of being pairwise orthogonal, termed as mutually orthogonal Latin squares (MOLS). In the following years, Cayley [21] introduced the concept of group multiplication tables, and demonstrated that such tables could be viewed as bordered Latin squares. In 1935, Moufang [30] first introduced the term 'quasigroup', and later defined a loop as a quasigroup with an identity element. The versatility of quasigroups, with their unique properties and the existence of quasigroups of specific orders, enables their application in diverse fields, including coding theory, cryptography, telecommunications, and beyond. For further details, readers may refer to [30, 21, 45, 26, 25], which contain extensive research on the applications of quasigroups in various fields such as cryptology and coding theory.

A quasigroup, by definition, does not have the associative property, which is a defining characteristic of a group. The properties of algebraic structures, such as closure and element inversion, are widely acknowledged to play a crucial role in the design of various cryptographic primitives. Groups, rings, and finite fields, which are all associative algebraic structures, have been widely employed in creating diverse cryptographic primitives, as well as in developing algebraic codes for simultaneous error detection and correction. As mentioned earlier, Dénes and Keedwell [21] designed various cryptographic primitives using these non-associative structures (quasigroups).

1.1. **Related research works in multivariate public-key cryptosystems (MPKC).** In literature [50, 6], various multivariate polynomials based public key cryptographic schemes (also referred as multivariate public key cryptosystems (MPKC)) have been proposed. Generally, they can be divided into four categories depending upon their non-linear quadratic part. These include the Matsumoto-Imai (MI) scheme [27], Hidden Field Equation (HFE) scheme [35], Oil-Vinegar (OV) scheme [24], and Stepwise Triangular System (STS) scheme [19, 48]. The MI cryptosystem [27] was proposed by Matsumoto and Imai in 1988. It was the first multivariate quadratic (MQ) problem- based public-key cryptosystem. In 1995, Patarin [33, 32] proved that the MI cryptosystem could be easily cryptanalyzed using the linearization equation attack. In 1996, Patarin proposed a cryptosystem named Hidden Field Equation (HFE) cryptosystem [35] and its different variants, including the HFEv-cryptosystem which can be viewed as an extension of the MI cryptosystem that resists the linearization equation attack. In HFE cryptosystem, Patarin transformed the central map of a MI cryptosystem using a Frobenius automorphism map $x \mapsto x^{q^i}$, $i \in \mathbb{N}$ of a finite field $\mathbb{F}_q$. In 1997, Patarin proposed a signature scheme named it *Oil-Vinegar (OV) signature scheme* [34]. The basic idea behind the OV signature scheme [34] was motivated by the Linearization equation attack on the MI cryptosystem. In this scheme, Patarin utilized the quadratic maps containing oil variables in linear form and the vinegar variables in quadratic form. In 1998, Kipnis and Shamir [23] proved that in OV signature scheme when number of oil and vinegar variables are same then it is not secure against an attack based on the oil-vinegar separation technique. In 1999, Kipnis et al. [22] proposed a variant of OV signature scheme called the *Unbalanced Oil-Vinegar (UOV) signature scheme*. In the UOV signature scheme, the number of vinegar variables is greater than the number of oil variables. Additionally, they proved that it was very efficient

and immune against the oil-vinegar separation attack. However, the public key size in this scheme was still extremely large. In [36, 37], Patarin et al. proposed two digital signature schemes based on the MQ problem, namely, the *SFlash signature scheme* and *Quartz signature scheme*. The idea of the SFlash signature scheme was based on the MI cryptosystem, and the Quartz signature scheme was based upon the HFEv- cryptosystem. Dubois et al. [7] have practically shown that the SFlash signature scheme was not secure against direct attacks. In 2003, Chen and Yang [3] proposed a multivariate signature scheme called the *Stepwise Triangular Scheme (STS)*, based on Tame Transformations or the Tame map. In 2006, Ding et al. [5] gave the cryptanalysis of the STS scheme.

In 2005, Ding et al. [4] came up with the idea of a multilayered Oil-Vinegar signature scheme, i.e. *Rainbow*. They have shown that it has shorter signature size and smaller public-key size than the classical OV signature scheme. The Rainbow signature scheme was submitted as a robust candidate for the NIST competition [11]. Thereafter, Petzoldt et al.,[39] proposed a variant of the Rainbow signature scheme, i.e. *Cyclic Rainbow*. The public key size of this scheme is reduced by 62%, and the number of required field multiplications operations in verifying the legitimate signature is reduced by 30%. In 2015, Petzoldt et al. [41] proposed a signature scheme, referred to as *GUI*, which was based upon the HFEv- framework. This scheme utilized the small finite fields to increase its efficiency; however, it could not resolve the problem of large key size. In 2017, Petzoldt et al. [40] tried to resolve that issue by applying vinegar variations to the MultiHFE [1] scheme, resulting in the development of a HMFEv signature scheme, defined over any random finite field. Later on, Hashimoto [20] showed that the HMFEv signature scheme is not secure. Parallely, in 2016 Chen et al. [2] proposed a multivariate signature scheme, namely *MQDSS*, which was proved to be secure with respect to the random oracle model. This scheme was based upon the 5-pass identification protocol explained in [12, 42].

In 2008, Gligoroski et al. [14, 15] proposed a public key cryptosystem based on multivariate quadratic quasigroup (MQQ). The central map of the proposed cryptosystem was based on the string transformations of quasigroups. They had shown that the proposed cryptosystem was more efficient in both hardware and software compared to existing multivariate polynomial based public key cryptosystems. However, in the same year, Mohamed et al. [28] analyzed the proposed cryptosystem and proved that it is susceptible to the MutantXL attack [50]. In 2010, Faugére et al. [10] and Ødegård et al. [31] proved that the proposed cryptosystem was also susceptible to the Gröbner basis attack. Additionally, Faugére et al. [10] had shown that the central map in existing multivariate quadratic quasigroup based public key cryptosystem has a weakness in central map which can be exploited very easily. In the following years, Gligoroski et al. improved their scheme, and in 2012 they proposed a digital signature scheme, namely *MQQ-SIG* [16]. In the scheme, authors used the minus modifier variation in the public key to safeguard it against direct attacks. Additionally, they demonstrated that their scheme is ultra-fast compared to the existing MQ-based cryptosystems, and proved that it is secure against chosen message attacks (CMA). Following that, in the same year Gligoroski and Samardjiska [17] also proposed a probabilistic encryption scheme referred to as *MQQ-ENC*. In this scheme, they utilized the left multivariate quadratic quasigroups (LMQQ) for the construction of a central map. Despite the differences in central map of both

the MQQ-SIG and MQQ-ENC schemes, in 2015 Faugére et al. [9] proved that for both schemes, finding equivalent good keys is feasible in polynomial time.

This motivation drives us to do further research into refining the MQQ-SIG scheme, aiming to address all existing drawbacks while ensuring the scheme's resilience against Faugére's attack [9].

Main results of the paper.

(*i*) We propose a digital signature scheme, MQQ-Sigv, whose security relies on the difficulty of solving a system of multivariate polynomial equations over a finite field. To construct the central map, we utilize a specific variant of the bilinear MQQ scheme, known as the vinegar variant. We then present an efficient algorithm to find the inverse of the central map.

(*ii*) The large size of private and public keys is a primary constraint in any multivariate polynomial based cryptosystem. To overcome this problem, we designed the secret key by utilizing a Toeplitz matrix instead of a random matrix. We will utilize the minus modifier technique to reduce the public key size.

(*iii*) In security analysis of the MQQ-Sigv scheme, we will prove that it is secure against EUF-CMA attack, Min-rank attack, High-rank attack, Direct attack, and differential attack. Additionally, we will prove that it is infeasible for any legitimate adversary to find an equivalent good key in the polynomial time for the proposed signature scheme. This is achieved by utilizing a randomly selected invertible map along with two hash maps as described by Wang and Yang in [49]. Finally, we give the operating characteristics of the proposed signature scheme.

This paper is divided into several sections. Section 2 covers basic definitions from quasigroup theory, the multivariate quadratic (MQ), problem and the MQQ. We discuss how a general quasigroup can be represented as the multivariate quadratic quasigroup, and examine the conditions necessary for the T-function to define a MQQ over the finite field $\mathbb{F}_{p^k}$. In Section 3, we discuss a general digital signature scheme based on the MQ problem. In Section 4, we propose a signature scheme and its verification based on MQQ over $\mathbb{F}_{p^k}$. In Section 5, we discuss the security analysis of the MQQ-Sigv scheme and prove some main results and theorems. In Section 6, we analyze the operating characteristics and efficiency of the proposed signature scheme. Finally, in Section 7 we draw the conclusion of the paper.

2. Mathematical preliminaries.

This section contains basic definitions of the quasigroups, T-functions, and multivariate quadratic (MQ) problem. Additionally, we discuss the construction of MQQ over the finite field, and how to find the parastrophe for the given MQQ. For an in-depth understanding of the system of multivariate polynomials and the multivariate public key cryptosystems, readers are encouraged to refer to the survey article [50]. Likewise, for the theory of quasigroups and its application, please refer to [45].

A non-empty set Q with a binary operation $\mathfrak{q}$ is called a quasigroup if for all $a, b \in Q$ there exist unique $x, y \in Q$ satisfying the equations $\mathfrak{q}(a, x) = b$ and $\mathfrak{q}(y, a) = b$. A quasigroup of order n means it has cardinality n.

For every quasigroup we can derive left ($\mathfrak{q}_\backslash$) and right ($\mathfrak{q}_/$) parastrophes by utilizing the following identities:

$$\begin{cases} \mathfrak{q}_\backslash(x, y) = z \iff \mathfrak{q}(x, z) = y \\ \mathfrak{q}_/(x, y) = z \iff \mathfrak{q}(z, y) = x \end{cases} \tag{1}$$

respectively. The algebra $(Q, \mathfrak{q}, \mathfrak{q}_\backslash, \mathfrak{q}_/)$ with three operations satisfies the following identities:

$$\begin{cases} \mathfrak{q}_\backslash(x, \mathfrak{q}(x,y)) = y \\ \mathfrak{q}_/(\mathfrak{q}(x,y), y) = x \\ \mathfrak{q}(x, \mathfrak{q}_\backslash(x,y)) = y \\ \mathfrak{q}(\mathfrak{q}_/(x,y), y) = x \end{cases} \tag{2}$$

Additionally, $(Q, \mathfrak{q}_\backslash)$ and $(Q, \mathfrak{q}_/)$ also form quasigroups.

Definition 2.1 (T-function)**.** Consider a map $f : (\mathbb{F}_2^d)^m \longrightarrow (\mathbb{F}_2^d)^l$ such that, for every $x \in (\mathbb{F}_2^d)^m$, the k^{th} bit of the j^{th} component, denoted as $f(x)_k^j$, depends only on the rightmost k bits of each component of x for all $j \in \{1, \ldots, l\}$. Then, the function f is defined as a *T-function*.

Definition 2.2. A quasigroup $(Q, \mathfrak{q})$ is an isotopic image of another quasigroup $(Q, \mathfrak{q}')$ if there exist permutations α, β, and γ such that, for all $l, m \in Q$, $\mathfrak{q}(l,m) = \gamma^{-1}\mathfrak{q}'(\alpha(l), \beta(m))$. The triplet (α, β, γ) is referred as an isotopism from $(Q, \mathfrak{q})$ to $(Q, \mathfrak{q}')$.

Now we formally define the multivariate quadratic problem (MQ) problem.

Definition 2.3. (MQ Problem [50]) Consider a system of m multivariate quadratic polynomials as given below:

$$\begin{cases} p_1(x_1, \cdots, x_n) & = \sum_{1 \le i \le j \le n} \alpha_{1ij} x_i x_j + \sum_{i=1}^n \beta_{1i} x_i + \gamma_1 \\ p_2(x_1, \cdots, x_n) & = \sum_{1 \le i \le j \le n} \alpha_{2ij} x_i x_j + \sum_{i=1}^n \beta_{2i} x_i + \gamma_2 \\ \vdots & \\ p_m(x_1, \cdots, x_n) & = \sum_{1 \le i \le j \le n} \alpha_{mij} x_i x_j + \sum_{i=1}^n \beta_{mi} x_i + \gamma_m \end{cases} \tag{3}$$

in n variables $x_1, \ldots, x_n$ over the finite field, $\mathbb{F}_{p^n}$ where p is prime and $n \in \mathbb{N}$. Finding a solution $x = (x_1, \ldots, x_n)$ of the system of equations $p_1(x) = \ldots, p_m(x) = 0$ is a challenging and NP-complete problem.

Definition 2.4. A map $f : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is said to be an *affine map* if it is defined as $f(x) = A \cdot x + b$, where A is an arbitrary $m \times n$ matrix and b is an arbitrary m-vector. An affine map is said to be linear if it is defined as $f(x) = A \cdot x$.

2.1. **Multivariate quadratic quasigroups (MQQ).** Gligoroski et al. [15] presented that every quasigroup $(Q, \mathfrak{q})$ with order 2^d and $d \ge 2$ can be represented as a vector valued Boolean function. This means that a quasigroup operation $\mathfrak{q}$ can be represented as a map $\mathfrak{q}_{vv} : \mathbb{F}_2^{2d} \to \mathbb{F}_2^d$ and for each $x, y, z \in Q$, with their binary representations $(x_1, x_2, \ldots, x_d)$, $(y_1, y_2, \ldots, y_d)$, and $(z_1, z_2, \ldots, z_d)$, respectively. The operation $x * y = z$ can be represented as

$$\mathfrak{q}(x,y) = z \Longleftrightarrow \mathfrak{q}(x_1, x_2, \ldots, x_d, y_1, y_2, \ldots, y_d) = (z_1, z_2, \ldots, z_d) \tag{4}$$

where $i = 1, 2, \ldots, d$, and each z_i can be uniquely expressed as $z_i = f_i(x_1, \ldots, x_d, y_1, \ldots, y_d)$, and $f_i : \mathbb{F}_2^{2d} \to \mathbb{F}_2$ is dependent on the operation $\mathfrak{q}$.

Every k-ary Boolean function $f(x_1, \ldots, x_k)$ can be uniquely represented using its algebraic normal form (ANF). So, the ANF of f can be represented as

$$f(x_1, \ldots, x_k) = \sum_{I \subseteq \{1, \ldots, k\}} a_I \left(\prod_{i \in I} x_i \right) \tag{5}$$

Here, the coefficients $a_I \in \mathbb{F}_2$. The addition and multiplication are done with respect to the finite field $\mathbb{F}_2$. In Equation (4), all z_i can be represented in their ANF forms up to degree 2 for the $2d$-ary boolean function (i.e., all z_i is a multivariate polynomial over $\mathbb{F}_2$).

The ANF of the function f_i offers insights into the complexity of the quasigroup (Q, q) by analyzing the degree of the boolean function f_i. In the case of randomly generated quasigroups with an order of 2^d, with $d \geq 4$, the degrees are greater than 2. Gligoroski et al. [15] gave the following definitions of multivariate quadratic quasigroups (MQQ) that have been used in designing public key cryptosystems [14, 15].

Definition 2.5. [15] Consider a quasigroup $(Q, *)$ of order 2^d referred to as *the multivariate quadratic quasigroup (MQQ)* of type $Quad_{d-k}Lin_k$ if its vector valued Boolean representation contains exactly $d-k$ quadratic polynomials and k linear polynomials, where $0 \leq k < d$.

The sufficient condition for a quasigroup $(Q, *)$ to be MQQ is given by the following theorem.

Theorem 2.6. [15] *Consider two $d \times d$ sized matrices P and Q of linear Boolean expressions, and two vectors b_1 and b_2 of linear or quadratic Boolean expression. Let elements of P, b_1 depends on the variables $x = (x_1, \ldots, x_d)$, and the elements of Q, b_2 depend upon the variables $y = (y_1, y_2, \ldots, y_d)$. If $\det(P) = \det(Q) = 1$ and $P \cdot y^T + b_1 \equiv Q \cdot x^T + b_2$ in $\mathbb{F}_2$, then the vector-valued operation $*_{vv}$ of the given quasigroup $(Q, *)$ with order 2^d can be expressed as*

$$\mathfrak{q}_{vv}(x, y) = P \cdot y^T + b_1$$

The quasigroup Q with vector valued operation $*_{vv}$ is known as "multivariate quadratic quasigroup (MQQ)"

2.1.1. *Multivariate quadratic quasigroups (MQQ) over finite field.* In this section, we discuss a construction of MQQ over the finite field $\mathbb{F}_{p^k}$, where p is prime and $k \geq 1$ [43]. This construction utilizes T-functions, and, consequently, the MQQ is referred to as a "T-multivariate quadratic quasigroups (T-MQQ)". Additionally, we explore an efficient method for determining the parastrophes of the T-MQQ.

The conditions under which the T-function defines a quasigroup are outlined in Theorem 2.7. For better comprehension, we specifically discuss the case for $p = 2$.

Theorem 2.7. [44] *A Boolean T-function $\mathfrak{q} : \mathbb{F}_2^{2d} \longrightarrow \mathbb{F}_2^d$ defines a quasigroup operation if and only if it is of the form $\mathfrak{q} = (\mathfrak{q}^{(d)}, \mathfrak{q}^{(d-1)}, \ldots, \mathfrak{q}^{(1)})$, where for every $s = 1, \ldots, d$ and $(x, y) = (x_d, \ldots, x_1, y_d, \ldots, y_1)$ we define $\mathfrak{q}^{(s)}(x, y)$ as*

$$\mathfrak{q}^{(s)}(x, y) = x_s \oplus y_s \oplus \left(\bigoplus_{j=(j_{s-1},\ldots,j_0)\in\mathbb{F}_2^s, k=(k_{s-1},\ldots,k_0)\in\mathbb{F}_2^s} b_{jk} x_{s-1}^{j_{s-1}} \ldots x_1^{j_1} y_{s-1}^{k_{s-1}} \ldots y_1^{k_1} \right) \tag{6}$$

Theorem 2.8. [43] *Consider $(Q, \mathfrak{q})$ as a quasigroup defined using a T-function over $\mathbb{F}_{p^k}$ of order p^{kd}. For a map $\mathfrak{q} : \mathbb{F}_{p^k}^{2d} \longrightarrow \mathbb{F}_{p^k}^d$ such that $\mathfrak{q} = (\mathfrak{q}^{(1)}, \ldots, \mathfrak{q}^{(d)})$, for all $s = 1, \ldots, d$ and $(x, y) = (x_d, \ldots, x_1, y_d, \ldots, y_1)$, we define each coordinate $\mathfrak{q}^{(s)}$ of $\mathfrak{q}$ as*

$$\mathfrak{q}^{(s)}(x, y) = p_1^{(s)}(x_s) + p_2^{(s)}(y_s) + \sum_{l,m>s} \alpha_{l,m}^{(s)} x_l x_m$$

$$+\sum_{l,m>s}\beta_{l,m}^{(s)}y_l y_m+\sum_{l,m>s}\gamma_{l,m}^{(s)}x_l y_m$$
$$+\sum_{l>s}\delta_l^{(s)}x_l+\sum_{l>s}\epsilon_l^{(s)}y_l+\eta^{(s)}. \tag{7}$$

where $p_1^{(s)}(x_s)$ *and* $p_2^{(s)}(y_s)$ *are a quadratic permutation polynomials over* $\mathbb{F}_{p^k}$*, and defines a multivariate quadratic quasigroup* $MQQ(\mathbb{F}_{p^k}^d, q)$ *of order* p^{kd}*.*

Proposition 2.9. [43] *Consider* $(Q,\mathfrak{q})$ *as an MQQ over* $\mathbb{F}_{p^k}$ *of order* p^{kd} *as defined in Theorem 2.8. Let* $[D]_{d\times d}, [D_1]_{d\times d}$*, and* $[D_2]_{d\times d}$ *be three non-singular matrices, whereas* c, c_1*, and* c_2 *are non-zero vectors of dimension* d*. The quasigroup* $(Q,\mathfrak{q}_0)$ *is isotopic to the quasigroup* $(Q,\mathfrak{q})$ *if the following condition holds:*

$$\mathfrak{q}_0(x,y)=D\cdot\mathfrak{q}(D_1\cdot x+c_1, D_2\cdot y+c_2)+c \tag{8}$$

Consider a quasigroup $(Q,\mathfrak{q})$ of order p^{kd}. In the cryptographic algorithms, finding the left $\mathfrak{q}_\backslash$ or right $\mathfrak{q}_/$ parastrophes of the given quasigroup operation $\mathfrak{q}$ is necessary for the decryption or signature process. However, determining the explicit form of the left parastrophe $\mathfrak{q}_\backslash$ can be computationally intensive in terms of space and time, especially when the degree of parastrophe can be arbitrary ($2 \le deg \le d$). In this context, we explore two distinct approaches for obtaining the parastrophe of $\mathfrak{q}$, with the choice depending on the system architecture:

1. Finding and storing the multiplication table (Latin square) of the quasigroup $\mathfrak{q}$, and compute the parastrophe $\mathfrak{q}_\backslash$ by referring to the multiplication table (Latin square). But, in this process architecture, the machine utilizes the most memory in the system. Since, this approach is independent of the type of quasigroup, we can compute the parastrophe $\mathfrak{q}_\backslash$ for any type of MQQ.
2. Another way of computing the parastrophe $\mathfrak{q}_\backslash$ of the quasigroup $\mathfrak{q}$ is by solving the system of d equations given by Equation (9), and to find out the unknowns $\{y_1,\cdots,y_d\}$ for memory constrained architectures. By this approach, we can avoid finding the explicit form of the parastrophe $\mathfrak{q}_\backslash$.

Our primary focus here is on the second approach for determining the parastrophe $\mathfrak{q}_\backslash$ of quasigroup operation $\mathfrak{q}$. Instead of directly obtaining the explicit form of $\mathfrak{q}_\backslash$ and evaluating $y=\mathfrak{q}_\backslash(u,v)$ for given $u,v\in\mathbb{F}_{p^k}$, $k\ge 1$, we choose to find y for the bilinear MQQ operation $(\mathfrak{q})$ by leveraging the identity $\mathfrak{q}_\backslash(u,v)=y \iff \mathfrak{q}(u,y)=v$. In simpler terms, we transform the task of evaluating $\mathfrak{q}_\backslash$ into solving a system of d equations in d variables $y_1,\ldots,y_d$ over $\mathbb{F}_{p^k}$

$$\mathfrak{q}(u,y)=v. \tag{9}$$

Solving this system is generally a non-trivial problem. However, owing to the specific structure of MQQ, this system can be efficiently solved in polynomial time. The reason lies in the conversion of Equation (9) into a system of linear equations involving 'd' variables.

2.2. Existential unforgeability under chosen-message attack (EUF-CMA).

In this section we discuss a notion of security designed for digital signature schemes. This standard notion of security is referred to as *Existential Unforgeability Under Chosen-Message Attack (EUF-CMA)* [18], and this concept is established through an experiment or game conducted between an adversary $\mathcal{A}$ and a challenger Ch.

Consider a digital signature scheme $Sig=\{\mathcal{K}, Gen, Ver\}$, where $\mathcal{K}$ is the *Key-Generation algorithm*, *Gen* is the *Signature-Generation algorithm*, and *Ver* is the

Signature Verification algorithm. Suppose $\mathcal{A}$ represents an adversary that can access the signature protocol as a black box (meaning it takes a message as an input and gives the signature as an output) and Ch represents as challenger of game. Now, we define the following experiment $Exp_{Sig(k)}^{EUF-CMA}$, which runs between $\mathcal{A}$ and Ch:

Algorithm 1 Experiment $Exp_{Sig(k)}^{EUF-CMA}$

1: The challenger (Ch) generates keys $(sk, pk) \leftarrow \mathcal{K}(k)$ by running Key-Generation algorithm which takes security parameter k as an input. Additionally, gives pk to $\mathcal{A}$.
2: The adversary $\mathcal{A}$ requests signatures for chosen n messages $\{m_i\}_{i=1}^n$ and obtains the valid signatures $\{\sigma_i\}_{i=1}^n$ in response, where $\sigma_i \leftarrow Gen(sk, m_i)$ for $i = 1, \ldots, n$.
3: The adversary $\mathcal{A}$ adaptively produces the message-signature pair (σ^*, m^*).
4: The output of the experiment $Exp_{Sig(k)}^{EUF-CMA}$ is

$$\begin{cases} 1, \text{ if } Ver(pk, m^*, \sigma^*) = \text{accept and } \ m^* \notin \{m_i\}_{i=1}^n \\ 0, \ otherwise \end{cases}$$

The success probability (or, Advantage (Adv)) of $\mathcal{A}$ can be defined as

$$Adv(\mathcal{A}_{Sig(k)}^{EUF-CMA}) = Prob[Exp_{Sig(k)}^{EUF-CMA} = 1].$$

The signature scheme $Sig(k)$ is EUF-CMA-secure if $Adv(\mathcal{A}_{Sig(k)}^{EUF-CMA})$ of any probabilistic polynomial time (PPT) adversary $\mathcal{A}$ is negligible concerning the security parameter k.

3. **Multivariate quadratic public-key signature scheme.** The multivariate public key signature scheme [50] relies on difficulty of solving the the multivariate quadratic (MQ) problem (3). For the multivariate public key signature scheme, we need to construct a central map $F : \mathbb{F}_{p^n}^n \longrightarrow \mathbb{F}_{p^n}^m$, where $\mathbb{F}_{p^n}$, p is prime, $n \in \mathbb{N}$, and it must satisfy the following two criteria:

1. $F(x_1, \ldots, x_n) = (f_1(x_1, \ldots, x_n), \ldots, f_m(x_1, \ldots, x_n))$ where $f_i \in \mathbb{F}_{p^n}[x_1, \ldots, x_n]$;
2. Equation $F(x_1, \ldots, x_n) = (y_1, \ldots, y_m)$ can be solved easily, and we can find the pre-image of $(y_1, \ldots, y_m)$ efficiently.

After the construction of the central map F, we define the public key function $\bar{F}$ by masking the central map F using two invertible affine maps S and T. The public key function $\bar{F}$ is

$$\bar{F} = S \circ F \circ T \tag{10}$$

Signatures scheme: To sign the document M, the system (user) utilizes the private key (S, F, T) and hash function $H : \{0,1\}^* \longrightarrow \mathbb{F}_{p^n}^m$. First, the system (user) computes $d = H(M) \in \mathbb{F}_{p^n}^m$ and then computes $x = T^{-1}(d),\ y = F^{-1}(x), and\ z = S^{-1}(y)$. Finally, the signature of the document M is $z \in \mathbb{F}_{p^n}^n$

Verification: For the verification process, if the user has the signed document vector $z \in \mathbb{F}_{p^n}^n$ and the public key function $\bar{F}$, The system (user) authenticates the signature scheme by verifying the equation

$$\bar{F}(z) \stackrel{?}{=} H(M) \tag{11}$$

and return either true or false.

4. **Construction of central map based on MQQ.** We introduce a new method to construct a central map using the MQQ with vinegar variables. The idea is motivated by the structure of the central map of the HFEv- signature scheme [41]. We will add some vinegar variables in Equation (7) to design the central map for the signature scheme.

Consider a quasigroup $(Q, \mathfrak{q})$ of order p^{kd}. The map $\mathfrak{q} = (\mathfrak{q}^{(1)}, \ldots, \mathfrak{q}^{(d)}) : \mathbb{F}_{p^k}^{2d+v} \longrightarrow \mathbb{F}_{p^k}^{d}$, with components $\mathfrak{q}^{(s)}$, $\forall\ s = 1, \ldots, d$ of $\mathfrak{q}$, is of the form

$$\mathfrak{q}^{(s)}(x_1, \cdots, x_d, y_1, \cdots, y_d, z_1, \ldots, z_v) = p^{(s)}(x_s) + p^{(s)}(y_s) + \sum_{l,m>s} \alpha_{l,m}^{(s)} x_l x_m + \sum_{l,m>s} \beta_{l,m}^{(s)} y_l y_m + \sum_{l,m>s} \gamma_{l,m}^{(s)} x_l y_m + \sum_{l>s} \delta_l^{(s)}(z_1, \ldots, z_v) x_l + \sum_{l>s} \epsilon_l^{(s)}(z_1, \ldots, z_v) y_l + \eta^{(s)}(z_1, \ldots, z_v) + c \tag{12}$$

where $p^{(s)}(y_s)$ and $p^{(s)}(x_s)$ are quadratic permutation polynomials [29] over $\mathbb{F}_{p^k}$ and the coefficients $\alpha_{l,m}^{(s)}, \beta_{l,m}^{(s)}$, *and* $\gamma_{l,m}^{(s)} \in \mathbb{F}_{p^k}$, the coefficients $\delta_l^{(s)}(z_1, \ldots, z_v)$, $\epsilon_l^{(s)}(z_1, \ldots, z_v)$ must be linear, while the coefficient $\eta^{(s)}(z_1, \ldots, z_v)$ must be quadratic in the vinegar variable. We use Algorithm 2 for the generation of the central map.

Algorithm 2 Construction of central map by utilizing the vinegar variation of MQQ

Input: $d, k \in \mathbb{N}$ and p-prime

1: Randomly generate the coefficients $\alpha_{l,m}^{(s)}, \beta_{l,m}^{(s)}, \gamma_{l,m}^{(s)}, c \in \mathbb{F}_{p^k}\ \forall\ l, m > s,\ \forall\ s \in \{1, \ldots, d\}$

2: For all $s \in \{1, \ldots, d\}$,
- Construct randomly the coefficients $\delta_i^{(s)} = \sum_{i=1}^{v} a_i z_i$ and $\epsilon_i^{(s)} = \sum_{i=1}^{v} b_i z_i$, where $a_i, b_i \in \mathbb{F}_{p^k}$.
- Construct randomly the coefficient $\eta^{(s)} = \sum_{i=0}^{v} c_i z_i^2$, where $c_i \in \mathbb{F}_{p^k}$.

3: For all $s \in \{1, \ldots, d\}$,
- Generate two random bits $r_1, r_2 \in \mathbb{F}_2$ if $p = 2$, otherwise set $r_1 = 0, r_2 = 0$.
- If $r_l = 0$ construct $p_l^{(s)} = a_l^{(s)} x_s + b_l^{(s)}$, otherwise construct $p_l^{(s)} = a_l^{(s)} x_s^2 + b_l^{(s)}$, for $l \in \{1, 2\}$.

4: Construct the polynomial map $\mathfrak{q}^{(s)}(x_1, \ldots, x_d, y_1, \ldots, y_d, z_1, \ldots, z_v)$ for all $s \in \{1, \ldots, d\}$ by utilizing the Equation (12), and the map

$$\mathfrak{q} = (\mathfrak{q}^{(1)}, \mathfrak{q}^{(2)}, \ldots, \mathfrak{q}^{(d)}).$$

5: Randomly generate three non-singular $d \times d$ matrices D, D_1, D_2 and vectors c, c_1, c_2 with dimension d over $\mathbb{F}_{p^k}$.

Output: The tuple $(\mathfrak{q}, D^{-1}, D_1^{-1}, D_2^{-1})$ and the central map: $\mathfrak{q}_0(x, y, z) = D \cdot \mathfrak{q}(D_1 \cdot x + c_1, D_2 \cdot y + c_2, z) + c$.

In Algorithm 2, if we chose random values of the vinegar variables $\{z_1, \ldots, z_v\}$ where each $z_i \in \mathbb{F}_{p^k}$ for $i = 1, \ldots, v$, the coefficients $\delta_l^{(s)}, \epsilon_l^{(s)}$, and $\eta^{(s)}$ become the constant terms and belong to the field $\mathbb{F}_{p^k}$. As a consequence, we give the following

theorem. This theorem plays a crucial role in determining the inverse of central map.

Theorem 4.1. *Consider a map* $\mathfrak{q} = (\mathfrak{q}^{(1)}, \ldots, \mathfrak{q}^{(d)}) : \mathbb{F}_{p^k}^{2d+v} \longrightarrow \mathbb{F}_{p^k}^{d}$, *and the components* $\mathfrak{q}^{(s)}$, $\forall\ s = 1, \ldots, d$ *of the quasigroup operation* $\mathfrak{q}$, *are of the form*

$$\mathfrak{q}^{(s)}(x_1, \cdots, x_d, y_1, \cdots, y_d, z_1, \ldots, z_v) = p^{(s)}(x_s) + p^{(s)}(y_s) + \sum_{l,m>s} \alpha_{l,m}^{(s)} x_l x_m + \sum_{l,m>s} \beta_{l,m}^{(s)} y_l y_m + \sum_{l,m>s} \gamma_{l,m}^{(s)} x_l y_m + \sum_{l>s} \delta_l^{(s)}(z_1, \ldots, z_v) x_l + \sum_{l>s} \epsilon_l^{(s)}(z_1, \ldots, z_v) y_l + \eta^{(s)}(z_1, \ldots, z_v) + c \tag{13}$$

where $p^{(s)}(y_s)$ *and* $p^{(s)}(x_s)$ *are quadratic permutations over* $\mathbb{F}_{p^k}$ *and the coefficients* $\alpha_{l,m}^{(s)}, \beta_{l,m}^{(s)}$, *and* $\gamma_{l,m}^{(s)} \in \mathbb{F}_{p^k}$, *and the coefficients* $\delta_l^{(s)}(z_1, \ldots, z_v)$ *and* $\epsilon_l^{(s)}(z_1, \ldots, z_v)$ *must be linear, where the coefficient* $\eta^{(s)}(z_1, \ldots, z_v)$ *must be quadratic in the vinegar variable. After choosing random values of the vinegar variables* $\{z_1, \ldots, z_v\}$ *where each* $z_i \in \mathbb{F}_{p^k}$ *for* $i = 1, \ldots, v$, *the above Equation (13) defines a T-MQQ* $(\mathbb{F}_{p^k}^d, q)$ *of order* p^{kd}.

4.1. **Generation of private-key pair.** This section unveils an algorithm for generating two different affine maps S and T. We will utilize the symmetric Toeplitz matrix instead of some random permutation matrix to save the memory consumption, and speed up the signing process significantly. The idea of the generation of the private key pair (S, T) is motivated by the MQQ-SIG signature scheme [16].

According to Algorithm 3, to construct the pair of affine maps (S, T) we just need to store the $(\sigma_1, \sigma_2, M_0, (\alpha_i)_n, P_{\rho_i}, P_{\rho_j})$ parameters. This requires only $(2n + \frac{rn}{8} + \frac{kn}{8} + n(u_1 + u_2))$ byte spaces where n, k, r, u_1, and u_2 are input parameters, which is proportional to $\mathcal{O}(n)$. The randomness of the private key generated using Algorithm 3 is equivalent to the random matrix. For better understanding of the generation of the private key pair S, T, reader may refer to the toy example mentioned in Appendix.

4.2. **Generation of public-key.** We construct a public key by utilizing the central map, which is defined by Equation (12), and the private key pair, which is defined by Algorithm 3. For the public key, we also utilized the minus modifier technique to decrease the public key size. The construction of the public key is inspired by the *Rainbow signature scheme*.

4.3. **Signature scheme.** In this section, we propos a signing scheme for a given document $M \in \mathbb{F}_{p^k}^n$ in which we need to find the inverse of the tuple (S, F, T). To find the inverse of central map F, we have used the previous crucial Theorem 4.1. For this, the first step is to convert the central map into the T-MQQ by choosing random values of the vinegar variables from field $\mathbb{F}_{p^k}$, then calculate the parastrophe of the given quasigroup operation $\mathfrak{q}$ using Proposition 2.9.

The procedure consists of a step in which we first concatenate the hash of the message M with random values from the field $\mathbb{F}_{p^k}$. This concatenation helps in improving the security of the scheme against direct attacks. Then, we generate the signature for the resultant string.

Algorithm 3 Affine maps S and T

Input: $n, u_1, u_2, r, k \in \mathbb{N}$ and p-prime

1: Generate two permutation matrices p_{σ_1} and p_{σ_2} randomly over field $\mathbb{F}_2$ corresponding to two different permutations σ_1 and σ_2 respectively on the set $\{1, \ldots, n\}$.

2: Randomly generate the matrices which satisfy the condition such that $p_{\rho_i^{(1)}} = [p_{\rho_i}[row, col]]$ and $p_{\rho_j^{(2)}} = [p_{\rho_j}[row, col]]$ over $\mathbb{F}_{p^k}, \forall\ i \in \{0, \ldots, u_1\}, j \in \{0, \ldots, u_2\}$ such that $[p_{\rho_i}[row, col]] = [p_{\rho_i}[row+1, col+1]]$ and $[p_{\rho_j}[row, col]] = [p_{\rho_j}[row+1, col+1]]$, respectively, with the symmetric condition of matrix.

3: Generate a random matrix $M_0 = [m^0_{i,j}]_{r\times n}$ over the set $\{0, 1\} \subset \mathbb{F}_{p^k}$, such that for each column j there exits at least one row i where $m_{i,j} \neq 0$.

4: For the fixed ordering of $\mathbb{F}_{p^k}\backslash\{0\}$, construct a repeating sequence $(\alpha_i)_n$ of length n. Form a new matrix $M = [\alpha_{\sigma_1(j)} \cdot m^{(0)}_{i,j}]_{r\times n}$, and then obtain the matrix I_M by replacing the last r rows of the identity matrix I_n by M.

5: Compute the matrices

$$S^{'}_{inv} = \sum_{i=0}^{u_1} p_{\rho_i^{(1)}} \cdot p_{\sigma_2} + \sum_{j=0}^{v_1} p_{\rho_j^{(2)}} \cdot P_{\sigma_1}, \tag{14}$$

$$T^{'}_{inv} = \sum_{i=0}^{u_1} p_{\rho_i^{(1)}} \cdot p_{\sigma_1} \cdot I_M + \sum_{j=0}^{v_1} p_{\rho_j^{(1)}} \cdot p_{\sigma_2} \cdot I_M \tag{15}$$

6: Let $\mathbf{SubT}^{'}_{inv}$ be the $r \times n$ matrix of the first r rows of $T^{'}_{inv}$. If $\mathbf{SubT}^{'}_{inv}$ has zero column or $det(T^{'}_{inv}) = 0$ or $det(S^{'}_{inv}) = 0$, then go to step 1, else create the matrices $S = ((S^{'}_{inv})^{transpose})^{-1}$ and $T = ((T^{'}_{inv})^{transpose})^{-1}$ and the column vector $v_s = (\alpha_{\sigma_1(1)} \cdot \alpha_{\sigma_2(1)}, \ldots, \alpha_{\sigma_1(n)}\alpha_{\sigma_2(n)})^T$.

7: The affine maps are $S(x) = S \cdot x + v_s$ and $T(x) = T \cdot x$.

Output: The tuple $(\sigma_1, \sigma_2, M_0, (\alpha_i)_n)$ and the affine maps S and T.

5. **Security analysis of MQQ-SIGv scheme.** We will prove that the MQQ-Sigv scheme is EUF-CMA secure, assuming the hardness of the MQ problem. Additionally, we present the comprehensive analysis demonstrating its resistance to Min-Rank, High-Rank, Direct, and Differential attack for different parameters, and it is discussed in the respective theorems. Finally, we will show that after applying the transformation proposed by Wang et al. [49] on the proposed signature scheme, it will be infeasible to find an equivalent good key in polynomial time.

Theorem 5.1. *Consider a cryptographically secure collision-resistant hash function H as a random oracle. Then, the proposed MQQ-Sigv scheme is EUF-CMA secure under the hardness of the MQ problem.*

Proof. We will prove the result by contradiction. Assume there exists an adversary $\mathcal{A}$ with a non-negligible winning probability in the EUF-CMA game for MQQ-Sigv. We then demonstrate the construction of an oracle machine $\mathcal{O}^{\mathcal{A}}$ capable of solving the MQ problem.

The proof proceeds through a series of games, denoted as G_0, G_1, and G_2. Each game G_i modifies G_{i-1} slightly for $i = 1,\ 2$. The probability of $\mathcal{A}$ winning game

Algorithm 4 Generation of public key

Input: leader element $b = (11\ldots1)$

1: First we obtain the private key pair S, T using Algorithm 3.
2: Construct a central map $\mathfrak{q}_0$ using Algorithm 2.
3: Suppose $x = (x_1, \ldots, x_n)$ is a vector with $x_i \in \mathbb{F}_{p^k}$, we transform vector x into $X = (X_1, \ldots, X_l)$ over field $\mathbb{F}_{p^k}^8$, where $n = 8l$ and every $X_i = (x_{8i-7}, \ldots, x_{8i})$.
4: We construct a mapping $F : \mathbb{F}_{p^k}^n \longrightarrow \mathbb{F}_{p^k}^n$ by utilizing the string transformations which we mentioned in Equation (12) as

$$F(x_1, \ldots x_n) = (y_1, \ldots, y_n) \Longleftrightarrow (Y_1, \ldots, Y_l)$$
$$= ((\mathfrak{q}_0(b, X_1, Z_1), \mathfrak{q}_0(X_1, X_2, Z_2), \ldots, \mathfrak{q}_0(X_{l-1}, X_l, Z_l))) \quad (16)$$

where $b = (11\ldots1)$ is the leader element and $Z = (Z_1, \ldots, Z_l)$, $Z_i = (z_{8i-7}, \ldots, z_{8i})$ the vector containing vinegar variables.
5: Now, we construct a map $P' : \mathbb{F}_{p^k} \longrightarrow \mathbb{F}_{p^k}$ as $P' = T \circ F \circ S = (P_1, \ldots, P_n)$, where $P_i(x_1, \ldots, x_n), 1 \leq i \leq n$.
6: We remove r quadratic equations from the map $P^{'}$ to construct the public key map $P : \mathbb{F}_{p^k}^n \longrightarrow \mathbb{F}_{p^k}^{n-r}$ as $P = (P_{r+1}, \ldots, P_n)$.
7: Choose a cryptographically secure hash map $H : \{0,1\}^* \longrightarrow \{0,1\}^n$

Output: Public key pair is (P, H)

Algorithm 5 Inverse of the function

Input: $d, k \in \mathbb{N}$, p-prime,

1: Randomly choose the vinegar variables $\{z_1, \ldots, z_v\} \in \mathbb{F}_{p^k}$. Substitute these variables into the map $\mathfrak{q}(x, y, z)$ defined by Equation (12). Then, this map converts into Equation (7) as described by Theorem (4.1).
2: Compute $u_1 = D_1 \cdot u + c_1$ and $v_1 = D^{-1} \cdot (v - c)$.
3: Solve the system of equations $\mathfrak{q}(u_1, y_1) = v_1$ for the unknown y_1.
4: If the aforementioned system of equations has no solution, a new set of vinegar variables is chosen, resulting in a new system of equations which needs to be solved. If the solution is unique, then proceed to the next step. In case of more than one solution, use other form of technique to find the correct set of y_1 like hash functions.
5: Compute $y = D_2^{-1} \cdot (y_1 - c_2)$

Output: $y = F^{-1}$, inverse of function 'F'.

G_i is denoted as $Prob[G_i]$. We assume that $\mathcal{O}^{\mathcal{A}}$ runs $\mathcal{A}$ and controls the output of the random oracle H in these games.

- $\mathbf{G_0}$: This game corresponds to the EUF-CMA game for MQQ-Sigv. Therefore, $Adv(\mathcal{A}_{Sig(1^k)}^{\text{EUF-CMA}}) = Prob[Exp_{Sig(1^k)}^{\text{EUF-CMA}} = 1] = Prob[G_0]$.
- $\mathbf{G_1}$: $\mathbf{G_1}$ is identical to $\mathbf{G_0}$, except that for $j = 1, \ldots, n$, $\mathcal{O}^{\mathcal{A}}$ substitutes the output of the hash H query of msg_j by $m_j = P(\sigma_0) \parallel P(\sigma_1) = S \circ F \circ T(\sigma_0) \parallel S \circ F \circ T(\sigma_1) \in \mathbb{F}_q^n$ and the signature query with σ_j, where σ_j is randomly selected from $\mathbb{F}_q^n$. It is important to note that if $\mid Prob(G_1) - Prob(G_0) \mid$ is non-negligible, it implies that $\mathcal{A}$ can distinguish the output of H. However, this is not possible since H is chosen to be a cryptographically secure collision-resistant hash function. Therefore, $\mid Prob(G_1) - Prob(G_0) \mid = \epsilon_1(k)$ is negligible.

Algorithm 6 Signature algorithm

Input: Message $M = (m_1, \ldots, m_l, m_{l+1}, m_{l+2}, \ldots, m_n) \in \mathbb{F}_{p^k}^n$, secret affine map pair (S, T), and the central function F.

1: Compute $h = H(m_1, m_2, \ldots, m_l, m_{l+1}, \ldots, m_n) = h_0 \mid\mid h_1$; here, h_0 is first $n-l$-bits of h and h_1 is remaining l-bits long.
2: Choose $r_0 = (r_{01}, r_{02}, \ldots, r_{0l}) \in \mathbb{F}_{p^k}^l$ and $r_1 = (r_{11}, r_{12}, \ldots, r_{1(n-l)}) \in \mathbb{F}_{p^k}^{n-l}$ randomly and uniformly. Compute $x_0 = r_0 \mid\mid h_0$ and $x_1 = r_1 \mid\mid h_1$.
3: First, we compute $y_0 = T^{-1}(x_0)$ and $y_1 = T^{-1}(x_1)$
4: Then, compute $z_0 = F^{-1}(y_0)$ and $z_1 = F^{-1}(y_1)$.
5: Finally, the signature of message M is $\sigma = (\sigma_0, \sigma_1)$, where $\sigma_0 = S^{-1}(z_0)$ and $\sigma_1 = S^{-1}(z_1)$.

Output: Signature of the message M as $\sigma(M) = (\sigma_0, \sigma_1)$.

Algorithm 7 Verification algorithm

Input: Message $M \in \mathbb{F}_{p^k}^n$, signature of message M is $\sigma = (\sigma_0, \sigma_1)$, and public key $pk = (p, H)$.

1: First, compute $h = H(m_1, m_2, \ldots, m_l, m_{l+1}, \ldots, m_n) = h_0 \mid\mid h_1$; here, h_0 is first $n-l$ bits of h and h_1 is remaining l-bits long.
2: Compute h_0' as the last $n-l$ bits of $P(\sigma_0)$ and h_1' as the last l bits of $P(\sigma_1)$
3: Check if $h_0' = h_0$ and $h_1' = h_1$ and return true; else, return false.

Output: True or False

- $\mathbf{G_2}$: This game is identical to $\mathbf{G_1}$ except that $\mathcal{O}^{\mathcal{A}}$ substitutes the output of the hash (H) query of msg^* with a random element m^* from $\mathbb{F}_q^n$. Similar to that argument for $\mathbf{G_1}$, it can be asserted that $\mid Prob(G_2) - Prob(G_1) \mid = \epsilon_2(k)$ is negligible.

Now we have

$$\mid Prob[G_2] - Prob[Exp_{Sig(1^k)}^{EUF-CMA} = 1] \mid = \mid Prob[G_2] - Prob[G_0] \mid$$
$$\leq \mid Prob[G_2] - Prob[G_1] \mid + \mid Prob[G_1] - Prob[G_0] \mid = \epsilon_1(k) + \epsilon_2(k) = \epsilon(k).$$

Here, $\epsilon(k)$ is a negligible function. Consequently, the probability $Prob[G_2]$ that $\mathcal{A}$ wins the game G_2 is same as the probability $Adv(\mathcal{A}_{Sig(1^k)}^{\text{EUF-CMA}}) = Prob[Exp_{Sig(1^k)}^{\text{EUF-CMA}} = 1]$ that $\mathcal{A}$ wins EUF-CMA game. In other words, if $Prob[G_2]$ is non-negligible, then $\mathcal{A}_{Sig(1^k)}^{\text{EUF-CMA}}$ is also non-negligible. It is important to note that $Prob[G_2]$ being non-negligible implies that $\mathcal{A}$ can generate the signature σ^* for msg^* such that $P(\sigma^*) = S \circ F \circ T(\sigma^*) = m^*$ with non-negligible probability. Thus, with the assistance of $\mathcal{A}$, the machine $\mathcal{O}^{\mathcal{A}}$ can solve the MQ problem $P(x) = S \circ F \circ T(x) = \sigma^*$, contradicting our assumption that the MQ problem is hard. Therefore, the $Adv(\mathcal{A}_{Sig(1^k)}^{EUF-CMA}) = Prob[Exp_{Sig(1^k)}^{EUF-CMA} = 1]$ is negligible. Consequently, we conclude that MQQ-Sigv is EUF-CMA secure. □

Theorem 5.2. *The proposed MQQ-Sigv signature scheme achieves $l(y)$-bits security against Min-Rank attack if the number of variables $v_{min-rank} \geq \lceil r_1 + 2^{val} \rceil$, where $val = \frac{l(y) - r_1 \log_2 q}{3}$.*

Proof. For the Min-rank attack [50], we need to find the tuple $(\omega_1, \ldots, \omega_k) \in \mathbb{F}_{p^k}$ such that

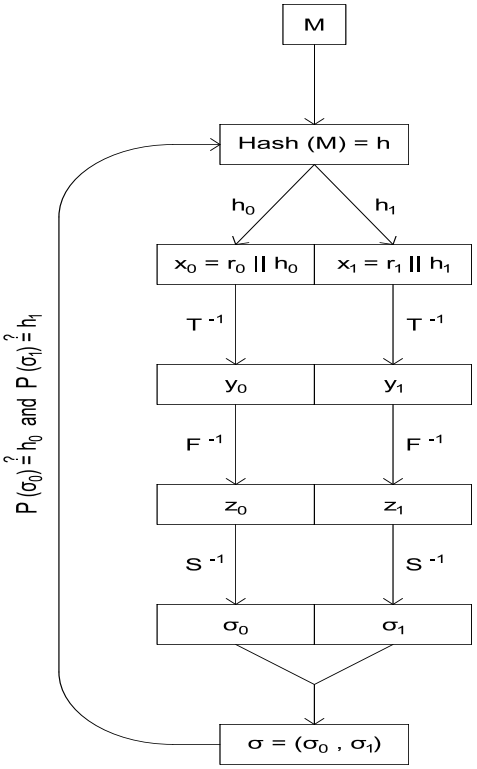

FIGURE 1. Pictorial representation of the signature scheme and its verification

$Rank(\sum_{i=1}^{k} \omega_i M_i - M_0) \leq r$ for given $r \in \mathbb{N}$. In [51], the Min-Rank complexity of the multivariate digital signature is $q^{s\lceil \frac{m}{n} \rceil} m^3$, where m is number of equations, n the numbers of variables, and s is the minimum rank. For the MQQ-Sigv scheme, we consider $n = v_{min-rank}, m = v_{min-rank} - r_1$, and $s \approx n - (n - r_1) = r_1$. As a result, to obtain an $l(y)$ bit security against Min-Rank attack, $q^{s\lceil \frac{m}{n} \rceil} m^3 \geq 2^{l(y)}$, which implies $q^{r_1}(v_{min-rank} - r_1)^3 \geq 2^{l(y)}$ means $v_{min-rank} \geq \lceil r_1 + 2^{val} \rceil$, where $val = \frac{l(y) - r_1 \log_2 q}{3}$. Therefore, the MQQ-Sigv signature scheme would be safe to Min-Rank attack if $v_{min-rank} \geq \lceil r_1 + 2^{val} \rceil$ where, $val = \frac{l(y) - r_1 \log_2 q}{3}$. □

Remark. For finite field $\mathbb{F}_{2^k}$, the MQQ-Sigv signature scheme would be immune to Min-Rank attack with $l(y)$-bit security if $v_{min-rank} \geq \lceil r + 2^{val_1} \rceil$, where $val_1 = \frac{l(y) - r_1 k}{3}$.

Theorem 5.3. *The proposed MQQ-Sigv signature scheme achieves $l(y)$-bits security against High-Rank attack over $\mathbb{F}_q$, where $q = 2^k$, if the number of variables $v_{high-rank} \approx \frac{l(y)}{2 \log_2 q}$.*

Proof. In [51], Yang and Chen mentioned that the complexity of High-Rank attack is $q^{\bar{\omega}}(\bar{\omega} n^2 + \frac{n^3}{6})$ where n is the number of plaintext variables, and for any plaintext

variables, the minimal of appearances in central map is $\bar{\omega}$. For MQQ-Sigv scheme, we fix $n = 2d$ and $\bar{\omega} = v$. To obtain $l(y)$ bit security against High-Rank attack, $q^v(4vd^2 + \frac{8d^3}{6}) \geq 2^{l(y)}$. In particular, if $q = 2^k$ and $d \approx \frac{l(y)}{2k} = \frac{l(y)}{2\log_2 q}$, then the complexity is greater than $2^{l(y)}$. This implies that MQQ-Sigv signature scheme would be immune against High Rank attack if $v_{high-rank} \approx \frac{l(y)}{2\log_2 q}$. □

Security level $l(y)$ (in bits)	Number of removed equations (r_1)	Number of variables to resist min-rank attack	Number of varibles to resist high-rank attack
80	8	264	5
	9	59	5
	10	20	5
96	9	265	6
	10	50	6
	11	17	6
112	10	1635	7
	11	267	7
	12	52	7
128	11	10332	8
	12	1637	8
	15	21	8

TABLE 1. Least number of variables required to resist min-rank and high-rank attack over the finite field $\mathbb{F}_{2^8}$.

Theorem 5.4. *The MQQ-Sigv signature scheme achieves $l(y)$-bit security level against Direct attack under the hardness of the MQ problem if the number of equations d is chosen accordingly to Table 2.*

Proof. In this method, the adversary tries to solve the system of equations $p(x) = y$ for fixed y directly by utilizing the different algebraic method like Gröbner bases methods, such as the F_5 algorithm [47]. In [38], Petzoldt determined the minimal number of equations d needed to attain $l(y)$-bit security level experimentally, and given in Table the 2.

Security level $l(y)$ (in bits)	Number of equations		
	$\mathbb{F}_{2^4}$	$\mathbb{F}_{31}$	$\mathbb{F}_{2^8}$
80	30	28	26
100	39	36	33
128	51	48	43
192	80	75	68
256	110	103	93

TABLE 2. Minimal number of equations needed to achieve a given security level

Assuming the hardness of MQ problem, MQQ-Sigv remains secure against Direct attack provided the minimum number of equations are chosen according to the Table 2. □

Theorem 5.5. *The proposed MQQ-Sigv is secure against Differential attack if the MQQ operations* $\mathfrak{q}^{(i)},\ i = 1, \ldots, n$ *are chosen uniformly at random.*

Proof. The notion of differential cryptanalysis introduced by Fouque, Granboulan, and Stern [13] has been effectively applied to various symmetric cryptographic algorithms and multivariate schemes. The fundamental concept involves considering a finite field $\mathbb{F}_q$ with characteristics p and any multivariate quadratic function $G : \mathbb{F}_q^n \to \mathbb{F}_q^m$. The differential operator between two points $x, y \in \mathbb{F}_q^n$ can be expressed as $L_{G,k} = G(x + k) - G(x) - G(k) + G(0)$, where this operator is indeed a bilinear function.

The MQQ-Sigv scheme employs randomly selected multivariate quadratic quasigroups $\mathfrak{q}_1, \ldots, \mathfrak{q}_n$. The non-linear component of the MQQ-Sigv scheme does not follow a function of the form $F : x \to x^{q^l+1}$, and its details are concealed from potential attackers as it is a part of the private key. Consequently, the MQQ-Sigv scheme is immune to differential attack. □

5.1. **Resistance against good-key attack.** In [10], Faugére et al. showed that the MQQ construction is a layered single field scheme and has Rainbow-like structure. In [9], Faugére et al. proved that, due to the structure of MQQ with only 2-variables, the MQQ-SIG [16] is vulnerable against the equivalent good-key attack.

Definition 5.6. [9] Both keys (F, S, T) and $(F^{'}, S^{'}, T^{'})$ are equivalent keys, if and only if $(T \circ F \circ S) = (T^{'} \circ F^{'} \circ S^{'}) \wedge (F\ |_I = F^{'}\ |_I)$, $\wedge$ is the 'and' operator, whereas $F, F^{'} \in \mathbb{F}_q[x_1, \ldots, x_n]^m$ $S, S^{'} \in GL_n(\mathbb{F}_q)$, and $T, T^{'} \in GL_m(\mathbb{F}_q)$, i.e., the central function F and $F^{'}$ have a similar structure when restricted to a limited set $I = \{I^{(1)}, \ldots, I^{(m)}\}$.

Definition 5.7. [9] We can say the key $(F^{'}, S^{'}, T^{'})$ is a good key for (F, S, T) iff $(T \circ F \circ S = T^{'} \circ F^{'} \circ S^{'}) \wedge (F\ |_J = F^{'}\ |_J)$ for the fixed set J such that J satisfies the following criteria:

- Consider two fixed sets $I = \{I^{(1)}, \ldots, I^{(m)}\}$ and $J = \{J^{(1)}, \ldots, J^{(m)}\}$ such that $J^{(k)} \subsetneq I^{(k)}\ \forall\ k, 1 \leq k \leq m$ with at least one $J^{(k)} \neq \emptyset$.

Essentially, the equivalent key notion reduces the number of variables by introducing two more linear maps (K, L), where $K \in GL_m(\mathbb{F}_q)$ and $L \in GL_n(\mathbb{F}_q)$, such that $P = T \circ K^{-1} \circ K \circ F \circ L \circ L^{-1} \circ S$. If F and $F^{'} = K \circ F \circ L$ have similar structure according to Definition (5.6), then $T^{'} = T \circ K^{-1}$ and $S^{'} = L^{-1} \circ S$ will be the equivalent keys. Furthermore, the good key concept [47] lowers the number of unknowns or unfixed coefficients in $(S^{'}, T^{'})$.

In the MQQ-Sigv scheme, we introduced vinegar variables into the MQQ structure and the construction of the signature scheme, as outlined in Algorithm 6, follows a Rainbow-like signature scheme. However, the proposed signature scheme is also based on MQQ, and according to Faugére et al. [9], MQQ-based signature schemes are susceptible to the equivalent good key attack.

To enhance the resistance of the MQQ-Sigv signature scheme against good key attacks, the transformation proposed by Wang and Yang [49] introduces several key elements to the scheme. Here is a detailed breakdown of the process and its goals:

1. **Quadratic Map Selection**: The transformation involves randomly selecting an invertible quadratic map over $\mathbb{F}_{p^k}$, where p is a prime and k is an integer. A quadratic map is a polynomial function of degree 2, and its use introduces additional complexity into the cryptographic scheme. The randomness in the

selection of this map helps ensure that the scheme is less predictable and harder to attack.

2. **Incorporation of Hash Maps**: Two hash maps are used in conjunction with the quadratic map. Hash maps are typically used to transform data in cryptographic schemes. They provide a way to securely process and incorporate information into the signature scheme, adding another layer of security.
3. **Additional Public Key Pairs**: The transformation introduces four additional pairs of public keys. This augmentation increases the number of public keys that must be handled and validated. By doing so, it strengthens the authentication condition, making the scheme more robust against attacks.
4. **Enhanced Verification Process**: The verification process is updated to include both the original external information and new, precise internal kernel information. This combined approach means that the verification not only checks against the initial external data, but also considers additional internal data provided by the new public keys and quadratic map.

The primary objective of these enhancements is to make it computationally infeasible for an adversary to find an equivalent key in polynomial time. By adding complexity through the quadratic map, hash maps, and additional public keys, the transformation increases the difficulty of solving the underlying mathematical problems that an attacker would face.

In the proposed scheme, we used the above ideas to protect the scheme against good key attacks.

Transformation of MQQ-Sigv scheme. Consider an MQQ-Sigv=$\{\mathcal{K}, Sig, Ver\}$ as a digital signature described in Section 4.3. To secure the MQQ-Sigv against good key attack, the user needs to randomly select an invertible quadratic polynomial $G : \mathbb{F}_{p^k}^{2n} \to \mathbb{F}_{p^k}^{2n}$ as a part of the private key and two random irreversible hash maps $H : \mathbb{F}_{p^k}^{n} \to \mathbb{F}_{p^k}^{n}$ and $\widetilde{H} : \mathbb{F}_{p^k}^{n} \to \mathbb{F}_{p^k}^{n}$.

Suppose the transformed version of the MQQ-Sigv signature scheme is $\widetilde{MQQ-Sigv} = \{\widetilde{\mathcal{K}}, \widetilde{Sig}, \widetilde{Ver}\}$, where $\widetilde{\mathcal{K}}$ is the key generation algorithm, $\widetilde{Sig}$ is the signature generation algorithm, and $\widetilde{Ver}$ is the verification algorithm.

1. **Key Generation algorithm**: The affine maps S and T can be generated by following Algorithm 3. Subsequently, $(P, Hash)$ can be generated by utilizing Algorithm 4. The secret key consists of a map $G : \mathbb{F}_{p^k}^{2n} \to \mathbb{F}_{p^k}^{2n}$, which is an invertible map. Consequently, the public key tuple is $\{P, H \circ S, H \circ G^{-1} \circ S, \widetilde{H} \circ F \circ G^{-1} \circ S, \widetilde{H} \circ T^{-1}\}$ and the private keys tuple is: $\{S, T, G\}$ for $\widetilde{MQQ-Sigv}$.

   To construct the affine map pair (S, T), it is sufficient to store the parameters $(\sigma_1, \sigma_2, M_0, (\alpha_i)_n, P_{\rho_i}, P_{\rho_j})$ as mentioned in Section 4.1. Additionally, to store the map $G : \mathbb{F}_{p^k}^{2n} \to \mathbb{F}_{p^k}^{2n}$, it requires $4n^2$ space. Therefore, the computation of the private key tuple (S, T, G) requires $(2n + \frac{rn}{8} + \frac{kn}{8} + n(u_1 + u_2) + 4n^2)$ space, which is proportional to $O(n^2)$, where $n, k, r, u_1, and u_2$ are input parameters.
2. **Signature Generation algorithm**: Consider $M = (m_1, m_2, \ldots, m_l, m_{l+1}, \ldots \ldots, m_n) \in \mathbb{F}_{p^k}^{n}$ needs to be signed using the private key pair (S, T, G). Algorithm 8 can be used to generate the signature of M.
3. **Signature Verification algorithm**: For the verification protocol, the input is message $M \in \mathbb{F}_{p^k}^{n}$, signature $(\sigma = (\sigma_0, \sigma_1), \sigma_g = (\sigma_{g_0}, \sigma_{g_1}))$, and the public

Algorithm 8 Transformed signature scheme $\widetilde{MQQ-Sigv}$

Input: Message $M = (m_1, m_2, \ldots, m_l, m_{l+1}, \ldots, m_n) \in \mathbb{F}_{p^k}^n$, secret key tuple (S, T, G), and the central function F.

1: Compute $h = Hash(m_1, \ldots, m_n) = h_0 \,||\, h_1$; here, h_0 is the first $n-l$ bits of h, and h_1 is the remaining l-bits long;

2: Choose $r_0 = (r_{01}, r_{02}, \ldots, r_{0l}) \in \mathbb{F}_{p^k}^l$ and $r_1 = (r_{11}, r_{12}, \ldots, r_{1(n-l)}) \in \mathbb{F}_{p^k}^{n-l}$ randomly and uniformly. Compute $x_0 = r_0 \,||\, h_0$ and $x_1 = r_1 \,||\, h_1$;

3: Calculate the following:

(i) $y_0 = T^{-1}(x_0)$ and $y_1 = T^{-1}(x_1)$;

(ii) $z_0 = F^{-1}(y_0)$ and $z_1 = F^{-1}(y_1)$,

(a) $\sigma_0 = S^{-1}(z_0)$ and $\sigma_1 = S^{-1}(z_1)$;

(b) $g(z_0, z_1) = g_0 \,||\, g_1$, $\sigma_{g_0} = S^{-1}(g_0)$ and $\sigma_{g_1} = S^{-1}(g_1)$.

Output: The signature of message M is (σ, σ_g), where $\sigma = (\sigma_0, \sigma_1)$ and $\sigma_g = (\sigma_{g_0}, \sigma_{g_1})$ is referred to as *the forward signature* and *backward signature*, respectively.

key tuple $(P, Hash, H \circ S, H \circ G^{-1} \circ S, \widetilde{H} \circ F \circ G^{-1} \circ S, \widetilde{H} \circ T^{-1})$. By using Algorithm 9, the user can verify whether the signature is legitimate or not.

Algorithm 9 Verification algorithm

Input: The message $M = (m_1, m_2, \ldots, m_l, m_{l+1}, \ldots, m_n) \in \mathbb{F}_{p^k}^n$, the signature pair $(\sigma = (\sigma_0, \sigma_1), \sigma_g = (\sigma_{g_0}, \sigma_{g_1}))$ and the public key tuple $(P, Hash, H \circ S, H \circ G^{-1} \circ S, \widetilde{H} \circ F \circ G^{-1} \circ S, \widetilde{H} \circ T^{-1}$.

1: Compute $h = Hash(m_1, m_2, \ldots, m_n) = h_0 \,||\, h_1$; here, h_0 is the first $n-l$ bits of h and h_1 is the remaining l-bits long.

2: Compute h_0' as the last $n-l$ bits of $P(\sigma_0)$ and h_1' the last l bits of $P(\sigma_1)$.

3: Check whether $h_0' = h_0$ and $h_1' = h_1$, if yes then

(a) Check $\widetilde{H} \circ T^{-1}(h_0) || \widetilde{H} \circ T^{-1}(h_1) = \widetilde{H} \circ F \circ G^{-1} \circ S(\sigma_g)$; if yes, then

(i) Check $\widetilde{H} \circ G^{-1} \circ S(\sigma_g) = H \circ S(\sigma)$.

If any one condition in step (3) is not true, then the verification fails.

Output: Either true or false.

The pictorial representation of the transformed MQQ-Sigv signature scheme is given in Figure 2. In the transformed MQQ-Sigv signature scheme, there are two signature pairs (σ, σ_g), where σ is the forward signature and σ_g is the backward signature. Suppose an adversary $\mathcal{A}$ tries to find an equivalent key (S', F', T') of (S, F, T) for the given MQQ-Sigv signature scheme. There may be a possibility that $\mathcal{A}$ can find these pairs of the equivalent key corresponding to the forward signature σ, but to find the random map $G : \mathbb{F}_{p^k}^{2n} \to \mathbb{F}_{p^k}^{2n}$ is not feasible in polynomial time according to [49]. Therefore, for an adversary to find the equivalent good key corresponding to the tuple (S, F, T, G) is not feasible in polynomial time. As a consequence, we have the following result.

Theorem 5.8. *The transformed digital signature scheme* $\widetilde{MQQ-Sigv}$ *based on MQQ is secure against the equivalent good key attack.*

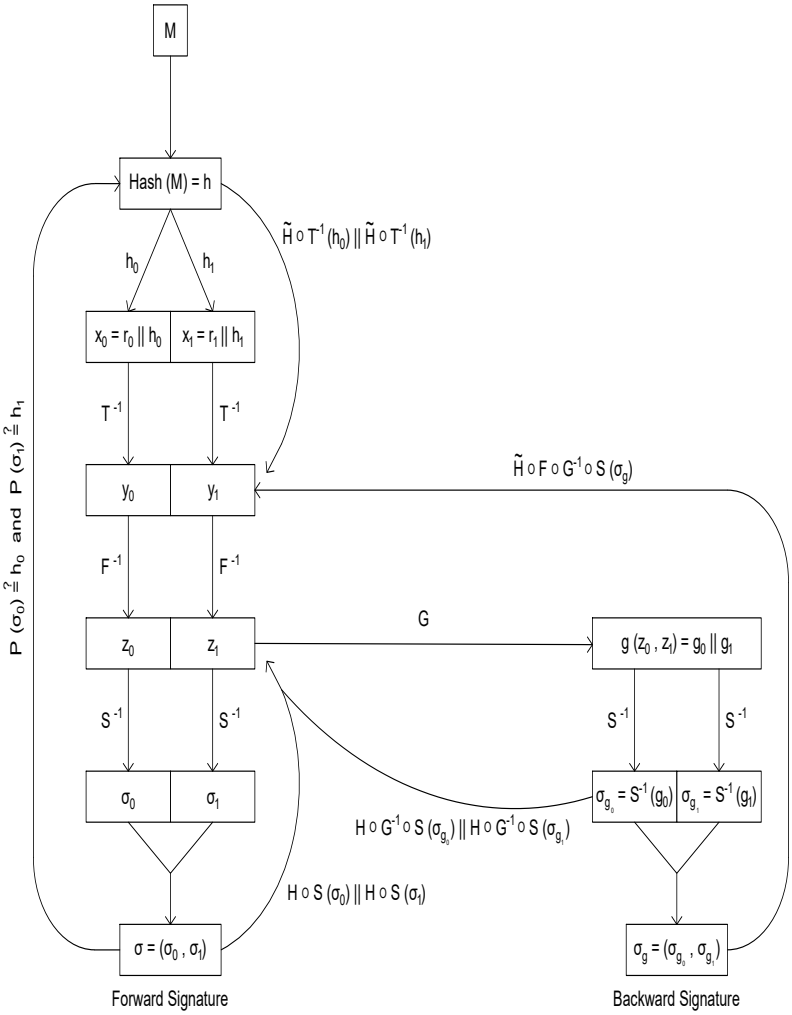

FIGURE 2. Pictorial representation of the transformed MQQ-Sigv scheme

6. **Operating characteristics of MQQ-Sigv signature scheme.** In this section, we discuss the operating characteristics, like public key, private key size, and signature size of the MQQ-Sigv scheme. Additionally, we compare these characteristics with the existing schemes, like MQQ-SIG and Rainbow, experimentally.

The proposed scheme includes the public key tuple (P, H), where P consists of $(n - r)$ polynomials in n variables over $\mathbb{F}_{2^k}$ and a standardized collision free hash function H. We analyzed the public key size of the proposed signature scheme over

$\mathbb{F}_{2^k}$:

$$\text{Size of public key } (pk) := \begin{cases} \frac{k}{8}(n-r)\left(1+\frac{(n+v)(n+v+1)}{2}\right) & \text{if } k=1; \\ \frac{k}{8}(n-r)\left(1+\frac{(n+v)(n+v+3)}{2}\right) & \text{if } k>1 \end{cases} \tag{17}$$

It was already mentioned in Section 4.1 that to construct the pair of affine maps (S,T), we need the $(\sigma_1,\sigma_2,M_0,(\alpha_i)_n,P_{\rho_i},P_{\rho_j})$ parameters only. This requires $\left(2n+\frac{rn}{8}+\frac{kn}{8}+n(u_1+u_2)\right)$ byte space for $k>1$, where n,k,r,u_1 and u_2 are input parameters.

Finite field	Security level $l(y)$ (in bits)	Number of variables (n)	Number of removed equations (r)	Number of vinegar variables (v)	Public key size (in bytes)	Private key size (in bytes) for fixed $u_1=u_2=10$
128	128	25	15	8	5206	618
		35	15	10	18917	866
256	128	45	20	15	47275	1147
		45	20	20	102410	1147

TABLE 3. For 128-bit security, the size of the public and private key

Security level $l(y)$ (in bits)	Algorithm	Number of variables (n)	Number of vinegar variables (v)	Public key size (in bytes)	Private key size (in bytes)	Signature size (in bytes)
80	MQQ-SIG	50	20	137408	401	40
	Rainbow	50	20	30240	23408	42
	MQQ-Sigv	50	20	12780	1156	100
96	MQQ-SIG	70	20	222360	465	48
	MQQ-Sigv	70	20	32441	1618	140
112	MQQ-SIG	90	30	352828	529	56
	MQQ-Sigv	90	30	73810	2081	180
128	MQQ-SIG	100	40	526368	593	64
	MQQ-Sigv	100	40	100110	2312	200

TABLE 4. Comparative analysis of the MQQ-SIG, Rainbow, and MQQ-Sigv schemes in terms of key size and signature size

7. **Conclusion.** We have proposed a quantum secure digital signature scheme based on the multivariate quadratic quasigroup structure. For the proposed scheme, we use two different variations of the bilinear MQQ: one is the vinegar variation, and the other is the minus modifier variation to the public key. This results in reducing the public key size. We have performed the security analysis of the proposed scheme and proved that it is secure against existential unforgeability under chosen message attack, Min-rank attack, High-rank attack, direct attack, and differential under given conditions. Most importantly, using the transformation proposed by Wang et al. [49], we secure the proposed scheme against equivalent good key attacks, i.e., it is infeasible for a computationally bounded adversary to find an equivalent good key of the proposed digital signature scheme in polynomial time. Finally, we discussed the operating characteristics and efficiency of the proposed scheme.

As a part of future work, to design an encryption technique based on the proposed algebraic structure or by leveraging the central map represented by Equation (13) and its optimized implementation.

8. Appendix.

8.1. Generation of affine map S and T.

For the construction of an affine map described by Algorithm 3, we assume particular values for $n = 5, u_1 = 2, u_2 = 3, p = 5$, and $k = 2$. We describe the construction of the private key pair step-by-step as follows:

1. Let us consider the permutations $\sigma_1 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 4 & 5 & 1 \end{pmatrix}$ and $\sigma_2 = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 5 & 4 \end{pmatrix}$. Generate the permutation matrix as per the permutations σ_1 and σ_2 as
$$p_{\sigma_1} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \; p_{\sigma_2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$
2. For $i \in \{0, 1, 2\}$ and $j \in \{0, 1, 2, 3\}$, consider the Toeplitz matrix with the symmetric conditions as
$$p^{(1)}_{\rho_0} = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 2 & 3 & 4 & 5 \\ 3 & 3 & 2 & 3 & 4 \\ 4 & 2 & 3 & 2 & 3 \\ 5 & 5 & 4 & 3 & 2 \end{bmatrix}, \; p^{(1)}_{\rho_1} = \begin{bmatrix} 2 & 3 & 4 & 5 & 6 \\ 3 & 2 & 3 & 4 & 5 \\ 4 & 3 & 2 & 3 & 4 \\ 5 & 4 & 3 & 2 & 3 \\ 6 & 5 & 4 & 3 & 2 \end{bmatrix}, \; p^{(1)}_{\rho_2} = \begin{bmatrix} 2 & 0 & 5 & 2 & 2 \\ 0 & 2 & 0 & 5 & 2 \\ 5 & 0 & 2 & 0 & 5 \\ 2 & 5 & 0 & 2 & 0 \\ 2 & 2 & 5 & 0 & 2 \end{bmatrix}$$
$$p^{(2)}_{\rho_0} = \begin{bmatrix} 6 & 7 & 8 & 0 & 6 \\ 7 & 6 & 7 & 8 & 0 \\ 8 & 7 & 6 & 7 & 8 \\ 0 & 8 & 7 & 6 & 7 \\ 6 & 0 & 8 & 7 & 6 \end{bmatrix}, \; p^{(2)}_{\rho_1} = \begin{bmatrix} 1 & 5 & 4 & 1 & 7 \\ 5 & 1 & 5 & 4 & 1 \\ 4 & 5 & 1 & 5 & 4 \\ 1 & 4 & 5 & 1 & 5 \\ 7 & 1 & 4 & 5 & 1 \end{bmatrix}, \; p^{(2)}_{\rho_2} = \begin{bmatrix} 0 & 1 & 3 & 5 & 9 \\ 1 & 0 & 1 & 3 & 5 \\ 3 & 1 & 0 & 1 & 3 \\ 5 & 3 & 1 & 0 & 1 \\ 9 & 5 & 3 & 1 & 0 \end{bmatrix}$$
$$p^{(2)}_{\rho_3} = \begin{bmatrix} 6 & 1 & 4 & 5 & 3 \\ 1 & 6 & 1 & 4 & 5 \\ 4 & 1 & 6 & 1 & 4 \\ 5 & 4 & 1 & 6 & 1 \\ 3 & 5 & 4 & 1 & 6 \end{bmatrix}.$$
3. Consider a matrix $M_0 = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{bmatrix}$ of order 2×5.
4. We create a matrix $M = [\alpha_{\sigma_{1(j)}} \cdot m^{(0)}_{i,j}]_{2\times 5} = \begin{bmatrix} 0 & \alpha_3 & \alpha_4 & 0 & \alpha_1 \\ \alpha_2 & 0 & \alpha_4 & \alpha_5 & 0 \end{bmatrix}$ and the matrix $I_M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & \alpha_3 & \alpha_4 & 0 & \alpha_1 \\ \alpha_2 & 0 & \alpha_4 & \alpha_5 & 0 \end{bmatrix}$. So, after choosing the particular values of $\alpha_1 = 2, \alpha_2 = 4, \alpha_3 = 5, \alpha_4 = 7$, and $\alpha_5 = 9$, we get the matrix $I_M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 5 & 7 & 0 & 2 \\ 4 & 0 & 7 & 9 & 0 \end{bmatrix}$.

5. Now, we compute the matrices

$$S'_{inv} = \sum_{k=0}^{2} p^{(1)}_{\rho_k} \cdot p_{\sigma_2} + \sum_{l=0}^{3} p^{(2)}_{\rho_l} \cdot p_{\sigma_1},\ T'_{inv} = \sum_{k=0}^{2} p^{(1)}_{\rho_k} \cdot p_{\sigma_1} \cdot I_M + \sum_{l=0}^{3} p^{(2)}_{\rho_l} \cdot p_{\sigma_2} \cdot I_M$$

$$S'_{inv} = \begin{bmatrix} 19 & 31 & 16 & 38 & 24 \\ 19 & 20 & 24 & 23 & 27 \\ 20 & 19 & 26 & 32 & 25 \\ 30 & 20 & 24 & 20 & 17 \\ 23 & 32 & 27 & 19 & 31 \end{bmatrix},\ T'_{inv} = \begin{bmatrix} 144 & 153 & 417 & 270 & 52 \\ 84 & 155 & 320 & 144 & 52 \\ 143 & 180 & 461 & 279 & 64 \\ 125 & 104 & 319 & 225 & 34 \\ 135 & 182 & 416 & 234 & 62 \end{bmatrix}$$

6. Consider the column vector $v_s = (\alpha_2 \cdot \alpha_2, \alpha_3 \cdot \alpha_3, \alpha_4 \cdot \alpha_1, \alpha_5 \cdot \alpha_5, \alpha_1 \cdot \alpha_4)^T = (16, 25, 14, 81, 14)^T$.
7. Finally, the affine maps $s(x) = (S^{transpose}_{inv})^{-1}(x) + v_s$ and $T(x) = (T^{transpose}_{inv})^{-1}(x)$.

REFERENCES

[1] C.-O. Chen, M.-S. Chen, J. Ding, F. Werner and B.-Y. Yang, Odd-char multivariate hidden field equations, *Cryptology ePrint Archive*, 2008.

[2] M.-S. Chen, A. Hülsing, J. Rijneveld, S. Samardjiska and P. Schwabe, From 5-pass-based identification to-based signatures, *Application of Cryptology and Information Security, Springer*, **10032** (2016) 135-165.

[3] W. Chen and Y. Shaoquan, Algorithm for modultion classification of mpsk signals based on cyclic cumulant invariants, *Journal of Electronics and Information*, **25** (2003), 320-325.

[4] J. Ding and D. Schmidt, Rainbow, a new multivariable polynomial signature scheme, *Applied Cryptography and Network Security*, (2005), 164-175.

[5] J. Ding, D. Schmidt and Z. Yin, Cryptanalysis of the new TTS scheme in CHES 2004, *International Journal of Information Security, Springer*, **5** (2006), 231–240.

[6] J. Ding and B.-Y. Yang, Multivariate public key cryptography, *Post-Quantum Cryptography, Springer Berlin Heidelberg*, (2009) 193-241 .

[7] V. Dubois, P. A. Fouque, A. Shamir and J. Stern, Practical cryptanalysis of SFLASH, *Advances in Cryptology - CRYPTO 2007*, **4622** (2007), 1-12.

[8] L. Euler, Commentationes arithmeticae collectae, *Typis ac impensis Academiae Imperialis Scientiarum*, **2** (1849).

[9] J.-C. Faugére, D. Gligoroski, L. Perret, S. Samardjiska and E. Thomae, A polynomial-time key-recovery attack on MQQ cryptosystems, *IACR International Workshop on Public Key Cryptography, Springer Berlin Heidelberg*, **9020** (2015), 150-174.

[10] J.-C. Faugére, R. S. Ødegård, L. Perret and D. Gligoroski, Analysis of the MQQ public key cryptosystem, *Cryptology and Network Security: 9th International Conference, CANS 2010*, Kuala Lumpur, Malaysia, Springer, (2010), 169-183.

[11] A. Ferozpuri and K. Gaj, High-speed FPGA implementation of the NIST round 1 rainbow signature scheme, *2018 International Conference on ReConFigurable Computing and FPGAs (ReConFig)*, IEEE, (2018), 1-8.

[12] A. Fiat and A. Shamir, How to prove yourself: Practical solutions to identification and signature problems, *Lecture Notes in Comput. Sci.*, **263** (1986), 186-194.

[13] P.-A. Fouque, L. Granboulan and S. Jacques. Differential cryptanalysis for multivariate schemes, *Advances in Cryptology—EUROCRYPT*, **3494** (2005), 341-353.

[14] D. Gligoroski, S. Markovski and S. J. Knapskog, Multivariate quadratic trapdoor functions based on multivariate quadratic quasigroups, *Proceedings of the American Conference on Applied Mathematics*, (2008), 44-49.

[15] D. Gligoroski, S. Markovski and S. J. Knapskog, A public key block cipher based on multivariate quadratic quasigroups, preprint, arXiv:0808.0247, 2008.

[16] D. Gligoroski, R. S. Ødegård, R. E. Jensen, L. Perret, J. C. Faugére, S. J. Knapskog and S. Markovski, MQQ-SIG: An ultra-fast and provably CMA resistant digital signature scheme, *International Conference on Trusted Systems, Springer*, (2011), 184-203.

[17] D. Gligoroski and S. Samardjiska, The multivariate probabilistic encryption scheme MQQ-ENC, *IACR Cryptology ePrint Archive*, 2012.

[18] S. Goldwasser, S. Micali and R. L. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, *SIAM Journal on Computing*, **17** (1988), 281-308.
[19] L. Goubin and N. T. Courtois, Cryptanalysis of the TTM cryptosystem, *Advances in Cryptology—ASIACRYPT 2000*, **1976** (2000), 44-57.
[20] Y. Hashimoto, On the security of HMFEv, *Cryptology ePrint Archive*, 2017.
[21] A. D. Keedwell and J. Dénes, *Latin Squares and Their Applications*, 2^{nd} edition, Elsevier, 2015.
[22] A. Kipnis, J. Patarin and L. Goubin, Unbalanced oil and vinegar signature schemes, *Advances in Cryptology—EUROCRYPT '99*, Springer, **1592** (1999), 206-222.
[23] A. Kipnis and A. Shamir, Cryptanalysis of the oil and vinegar signature scheme, *Advances in Cryptology—CRYPTO '98*, Springer, **1462** (1998), 257-266.
[24] A. Kipnis and A. Shamir, Cryptanalysis of the HFE public key cryptosystem by relinearization, *Annual International Cryptology Conference*, Springer, **1666** (1999), 19-30.
[25] S. Kumar, I. Gupta and A. J. Gupta, A study of public key cryptosystems based on quasigroups, *Cryptologia*, **47** (2023), 511-540.
[26] S. Kumar, H. Singh, I. Gupta and A. J. Gupta, MDS codes based on orthogonality of quasigroups, *Applicable Algebra in Engineering, Communication and Computing*, (2023), 1-22.
[27] T. Matsumoto and H. Imai, Public quadratic polynomial-tuples for efficient signature-verification and message-encryption, *Advances in Cryptology—EUROCRYPT'88*, **330** (1988), 419-453.
[28] S. E. Mohamed, J. Ding and J. Buchmann, *Algebraic Cryptanalysis of MQQ Public Key Cryptosystem by MutantXL*, IACR Cryptology ePrint Archive, 2008.
[29] R. A. Mollin and S. Charles, On permutation polynomials over finite fields, *International Journal of Mathematics and Mathematical Sciences*, Hindawi **10** (1987), 535-543.
[30] R. Moufang, Zur struktur von alternativkörpern, *Mathematische Annalen*, Springer, **110** (1935), 416-430.
[31] R. Ødegård, L. Perret, J. C. Faugére and D. Gligoroski, Analysis of the MQQ public key cryptosystem, *Conference on Symbolic Computation and Cryptography*, 2010.
[32] J. Patarin, Cryptanalysis of the matsumoto and imai public key scheme of eurocrypt'98, *Designs, Codes and Cryptography*, **20** (2000), 175-209.
[33] J. Patarin, Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt'88, *Advances in Cryptology—CRYPT0'95*, **963** (1995), 248-261.
[34] J. Patarin, *The Oil and Vinegar Signature Scheme*, Presented at the Dagstuhl Workshop on Cryptography September, 1997.
[35] J. Patarin, Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms, *International Conference on the Theory and Applications of Cryptographic Techniques*, Springer, (1996) 33-48.
[36] J. Patarin, N. Courtois and L. Goubin, Flash, a fast multivariate signature algorithm, *Topics in Cryptology — CT-RSA 2001: The Cryptographers' Track at RSA Conference 2001 San Francisco, C A*, **2020** (2001), 298-307.
[37] J. Patarin, N. Courtois and L. Goubin, QUARTZ, 128-bit long digital signatures, , *Topics in Cryptology-CT-RSA 2001*, Springer, (2001), 282-297.
[38] A. Petzoldt, Selecting and reducing key sizes for multivariate cryptography, *PhD Thesis, Darmstadt Tuprints*, 2013.
[39] A. Petzoldt, S. Bulygin and J. Buchmann, CyclicRainbow – a multivariate signature scheme with a partially cyclic public key, *Progress in Cryptology-INDOCRYPT 2010*, **6498** (2010), 33-48.
[40] A. Petzoldt, M.-S. Chen, J. Ding and B.-Y. Yang, HMFEv - an efficient multivariate signature scheme, *Post-Quantum Cryptography*, **10346** (2017), 205-223.
[41] A. Petzoldt, M. S. Chen, B. Y. Yang, C. Tao and J. Ding, Design principles for HFEv-based multivariate signature schemes, *Advances in Cryptology-ASIACRYPT 2015* Springer, **9452** (2015), 311-334.
[42] K. Sakumoto, T. Shirai and H. Hiwatari, Public-key identification schemes based on multivariate quadratic polynomials, *Advances in Cryptology-CRYPTO 2011*, Springer, **6841** (2011), 706-723.
[43] S. Samardjiska, Y. Chen and D. Gligoroski, Algorithms for construction of Multivariate Quadratic Quasigroups (MQQs) and their parastrophe operations in arbitrary Galois fields, *Journal of Information Assurance and Security*, **7** (2012).

[44] S. Samardjiska, S. Markovski and D. Gligoroski, Multivariate quasigroups defined by t-functions, *Conference on Symbolic Computation and Cryptography*, (2010), 117.
[45] V. Shcherbacov, *Elements of Quasigroup Theory and Applications*, CRC Press, Boca Raton, FL, 2017.
[46] P. W. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, *SIAM Review*, **41** (1999), 303-332.
[47] E. Thomae, About the security of multivariate quadratic public key schemes, PhD Thesis, Ruhr-Universität Bochum, Universitätsbibliothek, 2013.
[48] E. Thomae and C. Wolf, Cryptanalysis of enhanced TTS, STS and all its variants, or: Why cross-terms are important, *Progress in Cryptology-AFRICACRYPT 2012*, Springer, **7374** (2012), 188-202.
[49] X. Wang and B. Yang, An improved signature model of multivariate polynomial public key cryptosystem against key recovery attack, *Mathematical Biosciences and Engineering*, **16** (2019), 7734-7750.
[50] C. Wolf and B. Preneel, *Taxonomy of Public Key Schemes Based on the Problem of Multivariate Quadratic Equations*, IACR Cryptology ePrint Archive, 2005.
[51] B.-Y. Yang and J.-M. Chen, Building secure tame-like multivariate public-key cryptosystems: The new TTS, *Australasian Conference on Information Security and Privacy*, Springer, (2005), 518-531.

Received April 2024; revised August 2024; early access December 2024.