New Practical Multivariate Signatures from a Nonlinear Modifier

Daniel Smith-Tone[1,2]

[1]National Institute of Standards and Technology, USA
[2]University of Louisville, USA

daniel.smith@nist.gov

Abstract. Multivariate cryptography is dominated by schemes supporting various tweaks, or "modifiers," designed to patch certain algebraic weaknesses they would otherwise exhibit. Typically these modifiers are linear in nature— either requiring an extra composition with an affine map, or being evaluated by a legitimate user via an affine projection. This description applies to the minus, plus, vinegar and internal perturbation modifiers, to name a few. Though it is well-known that combinations of various modifiers can offer security against certain classes of attacks, cryptanalysts have produced ever more sophisticated attacks against various combinations of these linear modifiers.
In this article, we introduce a more fundamentally nonlinear modifier, called Q, that is inspired from relinearization. The effect of the Q modifier on multivariate digital signature schemes is to maintain inversion efficiency at the cost of slightly slower verification and larger public keys, while altering the algebraic properties of the public key. Thus the Q modifier is ideal for applications of digital signature schemes requiring very fast signing and verification without key transport. As an application of this modifier, we propose new multivariate digital signature schemes with fast signing and verification that are resistant to all known attacks.

Keywords: post-quantum, digital signature, multivariate

1 Introduction

The National Institute of Standards and Technology (NIST) is currently engaged in a process to establish new cryptographic standards [19] that offer security against adversaries with access to large scale quantum computing technology. This process aims to "Shor"-up NIST's public key suite of algorithms as a response to the exponential speed-ups offered by Shor's quantum algorithms [32] for solving the problems on which the current public key infrastructure is based. NIST's process is currently in the third round [26] and consists of 9 public key encryption or key-establishment algorithms and 6 digital signature schemes, see [20].

While the majority of the diverse array of key-establishment candidates target general use applications and offer good performance in many metrics, the

situation for digital signatures is very different. First, applications of digital signatures are extremely diverse and often different applications require dramatically different performance characteristics; moreover, many "niche" applications are actually quite pervasive. Secondly, there are very few candidates that are general purpose or that offer acceptable performance for some applications. The situation is of sufficient concern that NIST has asked for public feedback on the issue of signature scheme diversity on the NIST Post-Quantum Cryptography (PQC) Forum [11].

Part of this concern arises from the recent cryptanalyses [30, 4, 37] of two of the non-lattice-based digital signature schemes that made it to the third round of NIST's post-quantum standardization process. These candidate algorithms, Rainbow [21] and GeMSS [1], are both multivariate signature schemes with long histories. If neither scheme can be repaired in such a way that public confidence in the approach is restored, then there can be no Federal Information Processing Standard-compliant (FIPS-compliant) alternative to the lattice signatures CRYSTALS-Dilithium [42] and Falcon [39] for applications requiring signatures significantly shorter than a kilobyte in length.

Not only are the above cryptanalyses concerning, also the recent advances in generic techniques have contributed to apprehension about the security of multivariate signature schemes in general. In particular, the most effective attack [4] on the NIST round 3 finalist Rainbow is made efficient by the support minors method of solving the MinRank problem, see [2]. This advance alone changes the complexity of rank attacks on schemes like Rainbow and GeMSS by orders of magnitude *in the exponent*.

In addition, the cryptanalysis of GeMSS in [37] bypasses the combination of the vinegar and minus modifiers, one of the last remaining combinations of modifiers for multivariate systems that was believed to offer security for the so-called "big field" schemes— schemes requiring the multiplicative structure of an extension field. This advance invites the question of whether big field schemes are at all viable or whether secure multivariate digital signatures require a structure like that of Unbalanced Oil-Vinegar (UOV), see [23].

In this article we suggest a very strange answer to the above question. We propose that a big field scheme may be secure by turning it into an odd form of a UOV scheme by way of a new nonlinear modifier. This modifier, called Q, transforms any quadratic map into a UOV map in a way that preserves the structure of the original map in the sense that with secret information, the legitimate user can use the inversion procedure for the original central map to find a preimage.

As an application of this modifier, we construct multivariate digital signatures by applying the Q modifier to C^* and show that the resulting scheme, QC^*, is secure against all known attacks. We also select a "small field" cryptosystem, the Step-wise Triangular System (STS) multivariate encryption scheme, and use the Q modifier to create QSTS. Thus, we use the Q modifier to convert two insecure encryption schemes into secure digital signature schemes, which is quite humorous.

This article is organized as follows. In the next section, we introduce some of the multivariate cryptosystems we have discussed above and which we will be modifying. In Section 3, we present and discuss the common modifiers of multivariate schemes and their security properties. We then introduce the new Q modifier in the subsequent section. Next, we present a few new schemes based on the Q modifier, illustrating the breadth of possible schemes it can produce. In Section 6, we present a thorough analysis of the security of these schemes. We next propose parameters for the focus of future study and application of these schemes in Section 7. Finally, we conclude, discussing the possible directions to which this work leads.

2 Multivariate Signature Schemes

Multivariate cryptosystems can broadly be categorized as "big field" or "small field" schemes. Big field schemes rely on the multiplicative structure of an extension field to provide a nonlinear efficiently invertible function. In contrast, small field schemes accomplish this task directly by selecting nonlinear functions with some special structure embedded. In both cases, the structure that allows for efficient inversion is hidden with the application of some morphism of polynomials.

2.1 Unbalanced Oil-Vinegar (UOV)

The unbalanced oil-vinegar (UOV) signature scheme [23] is the oldest small field scheme still considered secure. Like most small field schemes, UOV relies on the sequential derivation of preimage variables for the inversion of the private key.

Given the finite field $\mathbb{F}_q$, one selects integers $v \approx 3o$ and constructs the vector space $O \oplus V \approx \mathbb{F}_q^{o+v}$, where $O \approx \mathbb{F}_q^o$ is called the oil subspace and $V \approx \mathbb{F}_q^v$ is known as the vinegar subspace. The private key then consists of a random linear map $L : \mathbb{F}_q^{o+v} \to \mathbb{F}_q^{o+v}$, and a random quadratic function F that is affine on cosets of O. Specifically, the map F is defined by

$$F(x_1, x_2, \ldots, x_{o+v}) = \sum_{i=o+1}^{o+v} \sum_{j=1}^{o+v} a_{ij} x_i x_j.$$

Each coordinate of F can be written as a quadratic form of the shape presented in Figure 1. Given any constant vector $\begin{bmatrix} c_{o+1} & \ldots & c_{o+v} \end{bmatrix} \in V$, we have that $F(\cdot, \ldots, \cdot, c_{o+1}, \ldots, c_{o+v})$ is an affine function on O. The public key is then the composition $P = F \circ L$.

A preimage for any element in the codomain of P can be efficiently found by a legitimate user by randomly selecting an element $\mathbf{c}$ of V, inverting the affine map $F(\cdot, \mathbf{c})$ and finally inverting L. Verification is accomplished by merely evaluating the public key at a given signature.

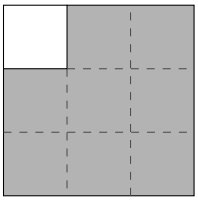
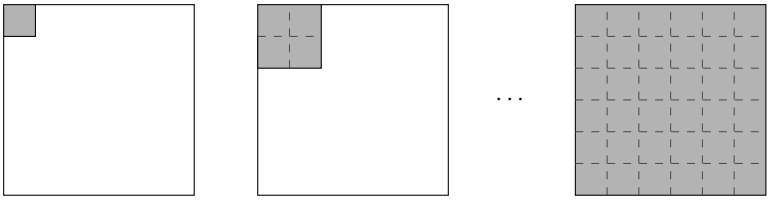


Fig. 1. The shape of the matrix representations of each central quadratic form of unbalanced oil-vinegar (UOV). The shaded regions represent possibly nonzero values while unshaded areas have coefficients of zero.



(a) Layer 1 (b) Layer 2 (c) Layer k

Fig. 2. The shape of the matrix representations of quadratic forms from each layer of the central map of a generic STS system. The shaded regions represent possibly nonzero values while unshaded areas have coefficients of zero.

2.2 Step-wise Triangular System (STS)

The main line of what we would today call step-wise triangular schemes originated in Shamir's birational permutation scheme over large rings in [31]. A very similar idea emerged which was called the sequential solution method (SSM) in [41]. These ideas were extended to construct the RSE system of [22] and were further adapted in [18] where the authors made it clear that these schemes were broken. This more general scheme was named triangle-plus-minus (TPM), which was further generalized into what we now call step-wise triangular schemes (STS) in [43]. There have since been numerous variations on the theme including [40, 36, 17]. They are all very similar and the simplest exposition to provide a good understanding of all of them is to present the generic STS constructions of [43].

Unlike UOV, the STS-style schemes are designed for encryption. Also unlike UOV, STS cryptosystems have a special differentiation in the structure of equations as well as the structure of the space of variables. As such, STS schemes require affine maps mixing both the inputs and outputs of the secret central map F. Thus a public key looks like $P = T \circ F \circ U$. The critical structure in the STS family is the structure of the central map.

The central map of a generic STS instance is defined by selecting integers $0 = u_0 < u_1 < \ldots < u_k = n$, and random quadratic maps $\mathbf{y}_i = \psi_i(\mathbf{x}_i)$, where $\mathbf{x}_i = (x_1, \ldots, x_{u_i})$ and $dim(\mathbf{y}_i) = u_i - u_{i-1}$ for $i = \{1, \ldots, k\}$. The central map is then the direct sum $\bigoplus_{i=1}^{k} \psi_i$, see Figure 2 for a visualization.

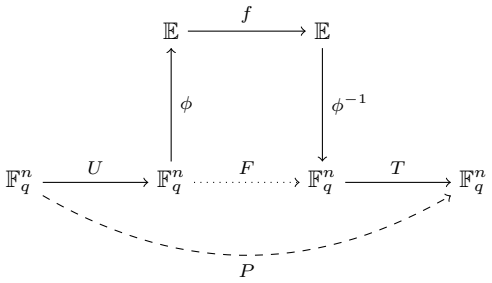


Fig. 3. The structure of a C^* scheme. The map ϕ is a $\mathbb{F}_q$-vector space isomorphism, F is a vector-valued function on $\mathbb{F}_q^n$, and f is an univariate function over $\mathbb{E}$.

Again, the technique for inversion of the secret map F is sequential. One first parses the output vector $\mathbf{y}$ into the component vectors $\mathbf{y}_i$ for each of the k layers. Then sequentially, the quadratic equations $\psi_i(\mathbf{x}_i) = \mathbf{y_i}$ are solved using the coordinates previously solved for $\mathbf{x}_{i-1}$ as a prefix of $\mathbf{x}_i$.

All of these constructions are vulnerable to generic combinatorial rank attacks as shown in [43]. In fact, all such schemes are vulnerable to both the MinRank attack— finding a low rank non-zero linear combination of the public quadratic forms— and the dual rank attack— finding a small subspace that is in the kernel of a large subspace of the quadratic forms.

2.3 *C**

The progenitor of all "big field" schemes is commonly known as C^*, or the Matsumoto-Imai scheme, see [25]. This scheme exploits the fact that an extension field $\mathbb{E}$ of $\mathbb{F}_q$ is an $\mathbb{F}_q$-algebra to produce two versions of a function— a vector-valued version which is quadratic over the base field, and a monomial function whose input and output lie in the extension field. Specifically, the C^* central map is the univariate function $f : \mathbb{E} \to \mathbb{E}$ defined by

$$f(X) = X^{q^\theta+1},$$

where $|\mathbb{E} : \mathbb{F}_q| = n$ and $(q^\theta + 1, q^n - 1) = 1$. The final condition ensures that the power map is invertible in $\mathbb{E}^*$. To complete the construction, one composes invertible affine maps to produce the public key $P(\mathbf{x}) = T \circ F \circ U$, see Figure 3. The C^* scheme can be considered a sort of multivariate version of RSA; in fact, the design of C^* intends for the inversion of F to be accomplished in exactly the same way as RSA, that is, by exponentiation by the multiplicative inverse of the encryption exponent modulo the size of the unit group.

C^* was broken by Patarin in [27] by way of linearization equations. Patarin discovered that there is a bilinear relationship between the plaintext $\mathbf{x}$ and ciphertext $\mathbf{y}$. In all but a few pathological cases, an adversary can interpolate this bilinear function by generating many plaintext-ciphertext pairs. Once recovered,

these linearization equations provide an even faster method of decryption than using the private key. Indeed later derivatives of C^* derive linearization equations from the private key as a fast method of inversion, see [9, 7, 8].

3 Modifiers

The cryptanalysis of the C^* scheme by Patarin in [27] inspired the creation of modifiers to make certain attacks infeasible. There are two categories of such alterations: one can modify the central map in some specific way preserving efficient invertibility; or one can make one or both affine transformations non-invertible. Of course, various modifications can be taken together as well. We present here some prominent modifiers.

Shortly after the cryptanalysis of C^*, Patarin introduced in [29] three modifiers aiming to enhance the security of C^*. These three modifiers include the minus (-) modifier (the removal of public equations), the plus (+) modifier (the addition of random equations in the central map that can be ignored on inversion) and the projection (p) modifier (the assignment of one or more input variables to constant values before the publication of the key).

The purpose of the minus modifier is clear. The idea is to remove some public equations and thereby change the algebraic structure of the central map. This method is equivalent to making the output transformation T singular. An immediate consequence in the case of C^* is that the minus modified scheme, C^*-, no longer has linearization equations. Still, C^*- was proven weak by an attack [14] exploiting a symmetric relation satisfied by the public key.

The projection modifier is the analogous modification on the input space. Instead of making the output transformation T singular, the input transformation U is made singular. Interestingly, this modification does not prevent the linearization equations attack if applied to C^*. The only cryptosystem proposed that is essentially of the pC^* form is SQUARE, see [10], which was broken by an attack analogous to that on C^*-, see [5].

The plus modifier is in some sense the opposite of the minus modifier. Additional random equations are added to the central map and then mixed via the output transformation. In the case of C^*, the plus modifier does not enhance security. The MinRank attack of [3] with a target rank of 2 recovers an equivalent C^* key. Still, this modifier has found use in numerous schemes, most recently including the, so named, PCBM scheme, see [35].

In [28], the vinegar (v) modifier (the addition of variables in the central map, the values of which can be randomly assigned upon inversion) is introduced in the QUARTZ scheme. QUARTZ is a parametrization of Hidden Field Equations with the vinegar and minus modifiers (HFEv-), the same construction as used in GeMSS, see [1]. Thus, the attack of [37] breaks the vinegar modification, even in conjunction with the minus modifier, if the central map is of low rank.

In [13], the internal perturbation (ip) modifier (the addition of a random summand with a small support) is used to produce the Perturbed Matsumoto-Imai (PMI) cryptosystem. The random summand introduced by the internal

perturbation modifier has such small support that its value can be guessed and subtracted from the output of the central map before inversion. This modifier applied to C^* was also broken, see [16].

All of these modifiers share the property that they either constitute a linear action on the public key or can be removed by a linear action on the public key. More specifically, the projection and minus modifiers are obviously linear projections and are dual to each other, while the vinegar and plus modifiers can both be removed via the application of the appropriate linear projection on the input or output space. Even the internal perturbation modifier can be removed via a projection, though the resulting scheme is the same as the original with an application of the projection modifier.

4 The Q Modifier

In this section we introduce a new generic modifier for multivariate schemes, named Q, that is inspired by relinearization, see [24]. As we will see, the Q modifier is not linear in the sense that each of the modifiers in the previous section are. Q is not a linear function on a public key nor can it be removed by a linear function on the public key.

First let us recall the relinearization technique first introduced in [24]. The idea of the technique is to symbolically solve a system of nonlinear equations by iteratively linearizing the system and recalling relations between the variables. Specifically, given a multivariate system in the variables $x_1, \ldots, x_n$, the relinearization technique assigns a new variable y_{ij} to each monomial of the form $x_i x_j$, attempts to solve the resulting linear system, and recalls the relations of the form $y_{ij}y_{k\ell} = y_{ik}y_{j\ell}$, among others. While relinearization did not provide the originally promised performance in solving overdefined systems, it did inspire the development of XL, see [12], and offers a new technique for modifying quadratic systems.

We begin the description in as general a context as possible and then discuss the specifications required to apply Q in special contexts. First, let $F : \mathbb{F}_q^m \to \mathbb{F}_q^m$ be an arbitrary homogenous quadratic function in the variable $\mathbf{x} = \begin{bmatrix} x_1 \ldots x_m \end{bmatrix}$. We select a short vector of auxiliary variables $\mathbf{w} = \begin{bmatrix} w_1, \ldots, w_\ell \end{bmatrix}$ and form products between these variables and terms of F (at this point, in an arbitrary way) to create a cubic map $\widetilde{F} : \mathbb{F}_q^{m+\ell} \to \mathbb{F}_q^m$. We then consider the general monomial of the form $x_i x_j w_k$. Such a monomial must always contain exactly one variable from $\mathbf{w}$. We define a vector $\mathbf{z}$ of $m\ell$ new variables $z_{ik} = x_i w_k$. Thus we have the relations

$$x_i x_j w_k = x_i z_{jk} = x_j z_{ik}. \tag{1}$$

We replace $\widetilde{F}$ with a new function $\widehat{F} : \mathbb{F}_q^{(\ell+1)m} \to \mathbb{F}_q^m$ in a two step process. First, we use relations of the form of Equation (1) to replace every cubic monomial in $\mathbf{x}$ with a monomial bilinear in $\mathbf{x}$ and $\mathbf{z}$ randomly. Second, we introduce new quadratic summands of the form $\alpha x_i z_{jk} - \alpha x_j z_{ik}$ and $\alpha z_{ij} z_{rs} - \alpha z_{is} z_{rj}$ for

randomly selected $\alpha \in \mathbb{F}_q$. These summands must equal zero by the definition of the variables in $\mathbf{z}$. The function $\widehat{F}$ is now a new quadratic function.

We illustrate with a small example. Suppose that $\left[y_1\ y_2\right] = F(x_1, x_2, x_3)$ over $\mathbb{F}_7$ is given by

$$\begin{aligned} y_1 &= 2x_1x_2 + 3x_1x_3 + x_2x_3 \\ y_2 &= x_1^2 + 5x_1x_3 + 2x_2x_3. \end{aligned}$$

We multiply by the variables w_1 and w_2 in an arbitrary way producing $\widetilde{F}$ defined by

$$\begin{aligned} y_1 &= 2x_1x_2w_2 + 3x_1x_3w_1 + 3x_1x_3w_2 + x_2x_3w_1 \\ y_2 &= x_1^2w_1 + x_1^2w_2 + 5x_1x_3w_2 + 2x_2x_3w_1. \end{aligned}$$

Next we substitute for x_iw_j and add cancelling terms (in parentheses below) in the new variables $z_{11}, z_{12}, \ldots, z_{32}$ to produce $\widehat{F}$ of the form

$$\begin{aligned} y_1 &= 2x_2z_{12} + 3x_1z_{31} + 3x_1z_{32} + x_3z_{21} + (4z_{12}z_{31} + 3z_{11}z_{32} + x_1z_{22} + 6x_2z_{12}) \\ y_2 &= x_1z_{11} + x_1z_{12} + 5x_3z_{12} + 2x_2z_{31} + (x_3z_{12} + 6x_1z_{32} + 4z_{22}z_{11} + 3z_{12}z_{21}). \end{aligned}$$

There are three things to notice. First, the resulting function $\widehat{F}$ is a UOV map. The map is clearly linear in $\mathbf{x}$ and quadratic in $\mathbf{z}$. Therefore, we can find a preimage under $\widehat{F}$ by using the inversion procedure for UOV. Consequently, we can see that the Q modifier embeds some distribution of quadratic maps into a subspace of the space of UOV keys necessarily having less entropy.

Second— and this is a key point— if there is an assignment of the ℓ variables $\mathbf{w}$ that makes $\widetilde{F}(\cdot, \mathbf{w})$ an efficient to invert quadratic system, then we have a second way to invert $\widehat{F}$. Specifically, the user assigns values to $\mathbf{w}$, solves for $\mathbf{x}$ such that $\widetilde{F}(\mathbf{x}, \mathbf{w}) = \mathbf{y}$, and computes $\mathbf{z} = \mathbf{x} \otimes \mathbf{w}$. We note here that quadratic terms in $\mathbf{z}$ never need to be computed unlike in the case of inversion as a UOV map. Thus, for functions $\widetilde{F}(\cdot, \mathbf{w})$ with sufficiently efficient inversion, the inversion of the maps transformed by Q is more efficient than UOV inversion.

Finally, since the original monomials are gone, there exists no linear projection on the input nor the output that transforms $\widehat{F}$ into a linear function of F. In fact, the Q transformation is a quadratic substitution, hence the name. Therefore attacks exploiting projections away from a modifier are ineffective against Q.

Thus, the Q modifier is particularly useful in cases in which we have families of efficiently invertible quadratic maps that can be parametrized by an additional auxiliary set of variables. In such a case for any fixed $\mathbf{w}$, the function $\widetilde{F}(\cdot, \mathbf{w})$ is efficiently invertible. Then we may use the inversion procedure for $\widetilde{F}(\cdot, \mathbf{w})$ to find preimages of $\widehat{F}$ with greater efficiency than the UOV inversion procedure. We present some explicit examples of constructing such parametrized families $\widetilde{F}$ in Section 5.

5 New Schemes

We can now explain the most complicated part of the Q modifier, the task of creating the parametrized family of efficiently invertible functions $\widetilde{F}(\mathbf{x}, \mathbf{w})$ from an efficiently invertible function F. The key is to use the structure that makes F efficiently invertible.

5.1 QC^*

Let $F(\mathbf{x}) = \phi^{-1} \circ f \circ \phi(\mathbf{x})$ where $f(X) = X^{q^\theta+1}$ is a C^* central map. We may select a linear transformation $B : \mathbb{F}_q^\ell \to \mathbb{F}_q^n$ and construct the function

$$\widetilde{F}(\mathbf{x}, \mathbf{w}) = \phi^{-1}\left(\phi(B(\mathbf{w}))f(\phi(\mathbf{x}))\right).$$

For any fixed nonzero $\mathbf{w}$, the quantity $a_{\mathbf{w}} = \phi(B(\mathbf{w}))$ is just some constant in $\mathbb{E}$, therefore the family of functions is simply the small field representations of the functions $a_{\mathbf{w}}X^{q^\theta+1}$, a collection of C^* maps with coefficients other than 1. Every such function has linearization equations which are trivial for the user to derive and use for extremely efficient inversion.

In fact, when ℓ is very small, linearization equations can be derived for all nonzero values of $\mathbf{w}$ and inversion is accomplished with a very small number of multiplications. Specifically, let $\mathbf{L}_i^{\mathbf{w}}$ be the ith linearization equation corresponding to $a_{\mathbf{w}}X^{q^\theta+1}$. Then we may invert $P(\tilde{\mathbf{x}}) = T \circ \widehat{F}(U\tilde{\mathbf{x}}) = \mathbf{y}$ by first computing a left kernel element $\mathbf{u}$ of the block matrix

$$\left[\mathbf{L}_1^{\mathbf{w}}\mathbf{T}^{-\top}\mathbf{y}^\top \cdots \mathbf{L}_m^{\mathbf{w}}\mathbf{T}^{-\top}\mathbf{y}^\top\right],$$

appending $\mathbf{u} \otimes \mathbf{w}$, and multiplying on the right by $\mathbf{U}^{-1}$. Since $\mathbf{L}_i^{\mathbf{w}}\mathbf{T}^{-\top}$ are all precomputed as part of the private key, inversion only involves computing $m+1$ matrix vector products, an $m\ell$ dimensional Kronecker product and solving a linear system.

Thus, the complexity of inversion is $m^3+m^\omega+m^2(\ell+1)^2+m\ell$, multiplications in $\mathbb{F}_q$ where $2 \leq \omega \leq 3$ is the linear algebra constant. For comparison, the complexity of inverting UOV(m,$m\ell$) using the structure of equivalent keys, see [44], is $\frac{1}{2}m^3\ell^2 + m^3\ell + m^\omega + \frac{3}{2}m^2\ell$ multiplications in $\mathbb{F}_q$.

5.2 QSTS

Let $F(\mathbf{x})$ be a step-wise triangular function with m steps of size 1. For any vector $\mathbf{w}$ we can construct the function $\widetilde{F}(\mathbf{x}, \mathbf{w})$ from F by randomly multiplying each term by a linear form in $\mathbf{w}$. For all constant nonzero assignments $\mathbf{w} = \mathbf{c}$ the resulting function of $\mathbf{x}$, $\widetilde{F}(\mathbf{x}, \mathbf{c})$ is still a triangular map, so inversion can proceed as normal.

Inversion of the public key is straightforward. Given $\mathbf{y} = P(\tilde{\mathbf{x}}) = T \circ \widehat{F}(U'\tilde{\mathbf{x}}, U''\tilde{\mathbf{x}})$, the user simply inverts T, finds the preimage $\mathbf{u}$ under $\widetilde{F}(\cdot, \mathbf{w})$, appends $\mathbf{u} \otimes \mathbf{w}$ and inverts the input transformation U.

Since the inversion process for $\widetilde{F}(\cdot, \mathbf{w})$ is inversion of a triangular map, it is very efficient. In total, inversion requires $m^3 + 2\binom{m+2}{3} + m^2(\ell+1)^2 + m\ell$ multiplications in $\mathbb{F}_q$.

6 Security Analysis

In this section we consider the security of the schemes introduced in the previous section as well as some general considerations for the security of Q modified schemes. Attacks on the UOV structure are well known and easy to avoid. Thus, we consider four main attack avenues.

6.1 Q Kernel Attacks

In the case of using the Q modifier generically, there exists an injection $M : \mathbb{F}_q^m \to \mathbb{F}_q^{m(\ell+1)}$ such that $\mathbf{M}\mathbf{P}_i\mathbf{M}^\top = \mathbf{0}_{m\times m}$ for all $1 \leq i \leq m$. Notice also, though, since monomials of the form $z_{ik}z_{jk}$ do not occur in $\widehat{F}$ that there also exist injections $M' : \mathbb{F}_q^\ell \to \mathbb{F}_q^{m(\ell+1)}$ such that $\mathbf{M}'\mathbf{P}_i\mathbf{M}'^\top = \mathbf{0}_{\ell\times\ell}$ for all $1 \leq i \leq m$. Thus, we either have a system of m^3 homogeneous quadratic equations in the $m^2(\ell+1)$ unknown coefficients of $\mathbf{M}$ or a system of $m\ell^2$ homogeneous quadratic equations in the $m\ell(\ell+1)$ unknown coefficients of $\mathbf{M}'$.

Such systems can be solved via Gröbner basis methods. Given a hybrid approach of guessing k variables and resolving the system, we either obtain a system of m^3 equations in $m^2(\ell+1)-k$ variables or a system of $m\ell^2$ equations in $m\ell(\ell+1)-k$ variables. Let d_{sr} and d'_{sr} represent the semi-regular degrees of such systems. These values are given by the degree of the first nonpositive coefficient in the series expansions of

$$S(t) = \frac{(1-t^2)^{m^3}}{(1-t)^{m^2(\ell+1)-k}},\ S'(t) = \frac{(1-t^2)^{m\ell^2}}{(1-t)^{m\ell(\ell+1)-k}}.$$

Assuming that such systems are semi-regular, we find a complexity

$$\mathcal{O}\left(q^k\binom{m^2(\ell+1)-k+d_{sr}}{d_{sr}}^\omega\right), \text{ or } \mathcal{O}\left(q^k\binom{m\ell(\ell+1)-k+d'_{sr}}{d'_{sr}}^\omega\right).$$

6.2 Direct Attacks

Direct attacks try to invert the public key directly as a quadratic function. Typically this process involves using some polynomial system solver based on either XL, see [12], or F4, see [15].

Since the public key of a Q modified scheme is underdetermined, we can employ the reduction procedure from [38] to convert the public key into a system of $m-\ell-1$ equations in $m-\ell-1$ variables. We can then take a hybrid approach and guess the values of k variables. The semi-regular degree for systems

of $m - \ell - 1$ equations in $m - \ell - 1 - k$ variables is the degree d_{sr} of the first nonpositive coefficient in the series

$$S(t) = \frac{(1-t^2)^{m-\ell-1}}{(1-t)^{m-\ell-1-k}}.$$

Under the assumption that the system derived from the public key is semi-regular, the complexity of the direct attack is

$$\mathcal{O}\left(q^k \binom{m-\ell-1-k+d_{sr}}{d_{sr}}^{\omega}\right).$$

6.3 Rank Attacks

The STS cryptosystem is vulnerable to every type of rank attack, as shown in [43]. The Q modification, because it introduces terms involving all variables, in general makes all of the maps full rank when the field is large enough. Thus QSTS has no rank defect.

The C^* scheme does have a rank defect with respect to the extension field $\mathbb{E}$. We note, however, that due to the addition of the cancelling terms of the form $x_i z_{jk} - x_j z_{ik}$ and $z_{ij} z_{rs} - z_{is} z_{rj}$ that there is no longer an $\mathbb{E}$ combination of the public quadratic forms with low rank. In particular, there exists no linear injection $M : \mathbb{F}_q^m \to \mathbb{F}_q^{m(\ell+1)}$ such that $P \circ M$ is a C^* public key; thus, QC^* is safe from rank attack.

6.4 Differential Attacks

The C^* scheme and higher degree analogues are also vulnerable to differential attacks directly as shown, for example, in [34]. Therefore, we need to verify that the Q transformation prevents such an attack.

As outlined in [33], the only maps that satisfy a differential symmetry on an $\mathbb{E}$-algebra are componentwise multiples of C^* monomial maps. Thus the attack is only possible if there exists a linear injection M such that $P \circ M$ is componentwise C^*. Due to the quadratic substitution, there exists no such injection.

7 Parameters and Performance

Selecting parameters to achieve security against the attacks from Section 6, we find that the limiting attack is the direct attack. With the complexity estimate then given in Section 6, we find that the optimal attack classically uses a hybrid approach with $k = 3$ in the case of $q = 2^8$ for all realistic parameters.

Using a linear algebra exponent of $\omega = 2.8$, we find that $m = 44$ and $\ell = 3$ are sufficient to achieve 151-bit security, which is comfortably NIST Level I. For a fair comparison, we implemented simple proof of concept implementations of

QC^*, QSTS and UOV with the same parameters in the MAGMA Computer Algebra System[1] see [6]. We observed that at the precision of measurement we were able to make that the performance of the Q modified schemes was extremely consistent between the variants and was better than that of our implementation of UOV. The results are presented in Table 1. Please note that these implementations are not at all optimized.

Table 1. The parameters and performance of QC^* and QSTS in comparision to UOV. The Q schemes performance data were essentially identical and are presented under the row labelled Q-schemes.

	q	m	ℓ	# Eqs.	# Vars.	sig. (B)	PK (B)	sign (ms)	ver. (ms)
Q-schemes	2^8	44	3	44	176	176	677600	0.6	2.9
UOV	2^8	N/A	N/A	44	176	176	677600	3.7	2.9

8 Conclusion

Digital signature schemes based on systems of nonlinear multivariate equations have been around for a long time. The break-and-patch evolution of the discipline as well as the multitude of attack paths available has always made multivariate cryptography a somewhat risky venture. The appeal of some of the performance characteristics of these schemes (e.g., very short signatures, very fast verification) has helped to keep alive the hope that multivariate schemes will find a permanent home in our future standards.

Recent advances in cryptanalytic techniques, however, have further shaken public confidence in certain multivariate approaches. Most multivariate schemes rely on a low rank property at some point in the inversion process. The new support minors method introduced in [2] is a dramatic improvement in generic technique and led to a significant attack against Rainbow, see [4]. Another recent advance, see [37], shows that the combination of vinegar and minus modifiers are not sufficient alone to secure big field schemes. As a result, there are no remaining multivariate candidates in NIST's post-quantum standardization process that have not suffered some significant attack.

In this work we present the Q modifier and show that it is qualitatively different from the modifiers that have been studied for a couple of decades. Q is inherently nonlinear and creates a new map divorced from the algebraic properties of the original map. Still, the new map, which is of UOV form, is related via a hidden quadratic relationship to the original map, so that inversion can still be accomplished with the original structure.

The fact that the Q modifier is generic suggests that it may be a promising direction requiring further study. In particular, it is possible to eliminate the

[1] Any mention of commercial products does not indicate endorsement by NIST.

UOV structure of the resulting scheme by appending a 1 at the end of the vector $\mathbf{w}$ defined in Section 4. The consequence of this change is that one may include terms quadratic in $\mathbf{x}$ in the central map. Thus, depending on the structure of the map, there may exist a linear projection onto the prototype function for the scheme. This alteration seems risky for systems with a rank defect, but is a topic worthy of further research in the general case.

References

1. A. Casanova, J.-C. Faugère, G. Macario-Rat, J. Patarin, L. Perret, J. Ryckeghem. GeMSS: A Great Multivariate Short Signature. available at `https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/GeMSS-Round3.zip`, 2020. Technical report, National Institute of Standards and Technology.
2. Magali Bardet, Maxime Bros, Daniel Cabarcas, Philippe Gaborit, Ray A. Perlner, Daniel Smith-Tone, Jean-Pierre Tillich, and Javier A. Verbel. Improvements of algebraic attacks for solving the rank decoding and minrank problems. In Shiho Moriai and Huaxiong Wang, editors, *Advances in Cryptology - ASIACRYPT 2020 - 26th International Conference on the Theory and Application of Cryptology and Information Security, Daejeon, South Korea, December 7-11, 2020, Proceedings, Part I*, volume 12491 of *Lecture Notes in Computer Science*, pages 507–536. Springer, 2020.
3. Luk Bettale, Jean-Charles Faugère, and Ludovic Perret. Cryptanalysis of HFE, multi-HFE and variants for odd and even characteristic. *Des. Codes Cryptography*, 69(1):1–52, 2013.
4. Ward Beullens. Improved cryptanalysis of UOV and rainbow. *IACR Cryptol. ePrint Arch.*, 2020:1343, 2020.
5. O. Billet and G. Macario-Rat. Cryptanalysis of the square cryptosystems. *ASIACRYPT 2009, LNCS*, 5912:451–486, 2009.
6. Wieb Bosma, John Cannon, and Catherine Playoust. The magma algebra system i: The user language. *J. Symb. Comput.*, 24(3–4):235–265, October 1997.
7. Ryann Cartor and Daniel Smith-Tone. EFLASH: A new multivariate encryption scheme. In Carlos Cid and Michael J. Jacobson Jr., editors, *Selected Areas in Cryptography - SAC 2018 - 25th International Conference, Calgary, AB, Canada, August 15-17, 2018, Revised Selected Papers*, volume 11349 of *Lecture Notes in Computer Science*, pages 281–299. Springer, 2018.
8. Ryann Cartor and Daniel Smith-Tone. All in the c^* family. *Des. Codes Cryptogr.*, 88(6):1023–1036, 2020.
9. M.-S. Chen, B.-Y. Yang, and D. Smith-Tone. Pflash - secure asymmetric signatures on smart cards. Lightweight Cryptography Workshop 2015, 2015. http://csrc.nist.gov/groups/ST/lwc-workshop2015/papers/session3-smith-tone-paper.pdf.
10. Crystal Clough, John Baena, Jintai Ding, Bo-Yin Yang, and Ming-Shing Chen. Square, a New Multivariate Encryption Scheme. In Marc Fischlin, editor, *CT-RSA*, volume 5473 of *Lecture Notes in Computer Science*, pages 252–264. Springer, 2009.
11. International Community. Nist pqc-forum (email forum). Google Groups, 2021. https://groups.google.com/a/list.nist.gov/g/pqc-forum.

12. N. Courtois, A. Klimov, J. Patarin, and A.Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. *EUROCRYPT 2000, LNCS*, 1807:392–407, 2000.
13. Jintai Ding. A new variant of the matsumoto-imai cryptosystem through perturbation. In Feng Bao, Robert H. Deng, and Jianying Zhou, editors, *Public Key Cryptography - PKC 2004, 7th International Workshop on Theory and Practice in Public Key Cryptography, Singapore, March 1-4, 2004*, volume 2947 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 2004.
14. Vivien Dubois, Pierre-Alain Fouque, Adi Shamir, and Jacques Stern. Practical Cryptanalysis of SFLASH. In Alfred Menezes, editor, *CRYPTO*, volume 4622 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2007.
15. J. C. Faugere. A new efficient algorithm for computing grobner bases (f4). *Journal of Pure and Applied Algebra*, 139:61–88, 1999.
16. Pierre-Alain Fouque, Louis Granboulan, and Jacques Stern. Differential cryptanalysis for multivariate schemes. In Ronald Cramer, editor, *Advances in Cryptology - EUROCRYPT 2005, 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Aarhus, Denmark, May 22-26, 2005, Proceedings*, volume 3494 of *Lecture Notes in Computer Science*, pages 341–353. Springer, 2005.
17. M. Gotaishi and S. Tsujii. Hidden pair of bijection signature scheme. Cryptology ePrint Archive, Report 2011/353, 2011. http://eprint.iacr.org/.
18. Louis Goubin and Nicolas T. Courtois. Cryptanalysis of the ttm cryptosystem. In Tatsuaki Okamoto, editor, *Advances in Cryptology — ASIACRYPT 2000*, pages 44–57, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg.
19. Cryptographic Technology Group. Submission requirements and evaluation criteria for the post-quantum cryptography standardization process. NIST CSRC, 2016. http://csrc.nist.gov/groups/ST/post-quantum-crypto/documents/call-for-proposals-final-dec-2016.pdf.
20. NIST Cryptographic Technology Group. Post quantum cryptography standardization (website), 2021. https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization.
21. Jintai Ding, Ming-Shing Chen, Albrecht Petzoldt, Dieter Schmidt, Bo-Yin Yang, Matthias Kannwischer, Jacques Patarin. Rainbow. available at `https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Rainbow-Round3.zip`, 2020. Technical report, National Institute of Standards and Technology.
22. Masao Kasahara and Ryuichi Sakai. A construction of public-key cryptosystem based on singular simultaneous equations. *IEICE Transactions*, 88-A(1):74–80, 2005.
23. A. Kipnis, J. Patarin, and L. Goubin. Unbalanced oil and vinegar signature schemes. *EUROCRYPT 1999. LNCS*, 1592:206–222, 1999.
24. A. Kipnis and A. Shamir. Cryptanalysis of the HFE public key cryptosystem by relinearization. *Advances in Cryptology - CRYPTO 1999, Springer*, 1666:788, 1999.
25. Tsutomu Matsumoto and Hideki Imai. Public Quadratic Polynominal-Tuples for Efficient Signature-Verification and Message-Encryption. In *EUROCRYPT*, pages 419–453, 1988.
26. National Institute of Standards and Technology. *NISTIR 8309: Status Report on the Second Round of the NIST Post-Quantum Cryptography Standardization Process*. NIST, 2020. https://doi.org/10.6028/NIST.IR.8309.

27. Jacques Patarin. Cryptoanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88. In Don Coppersmith, editor, *CRYPTO*, volume 963 of *Lecture Notes in Computer Science*, pages 248–261. Springer, 1995.
28. Jacques Patarin, Nicolas Courtois, and Louis Goubin. Quartz, 128-bit long digital signatures. In David Naccache, editor, *CT-RSA*, volume 2020 of *Lecture Notes in Computer Science*, pages 282–297. Springer, 2001.
29. Jacques Patarin, Louis Goubin, and Nicolas Courtois. C^*_{-+} and HM: Variations Around Two Schemes of T. Matsumoto and H. Imai. In Kazuo Ohta and Dingyi Pei, editors, *ASIACRYPT*, volume 1514 of *Lecture Notes in Computer Science*, pages 35–49. Springer, 1998.
30. Ray A. Perlner and Daniel Smith-Tone. Rainbow band separation is better than we thought. *IACR Cryptol. ePrint Arch.*, 2020:702, 2020.
31. Adi Shamir. Efficient signature schemes based on birational permutations. In Douglas R. Stinson, editor, *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, volume 773 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 1993.
32. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Sci. Stat. Comp.*, 26, 1484, 1997.
33. Daniel Smith-Tone. On the differential security of multivariate public key cryptosystems. In Bo-Yin Yang, editor, *PQCrypto*, volume 7071 of *Lecture Notes in Computer Science*, pages 130–142. Springer, 2011.
34. Daniel Smith-Tone. Practical cryptanalysis of k-ary c^*. In Jintai Ding and Jean-Pierre Tillich, editors, *Post-Quantum Cryptography - 11th International Conference, PQCrypto 2020, Paris, France, April 15-17, 2020, Proceedings*, volume 12100 of *Lecture Notes in Computer Science*, pages 360–380. Springer, 2020.
35. Daniel Smith-Tone and Cristina Tone. A multivariate cryptosystem inspired by random linear codes. *Finite Fields Their Appl.*, 69:101778, 2021.
36. Kohtaro Tadaki and Shigeo Tsujii. Two-sided multiplications are reduced to one-sided multiplication in linear piece in hand matrix methods. In *Proceedings of the International Symposium on Information Theory and its Applications, ISITA 2010, 17-20 October 2010, Taichung, Taiwan*, pages 900–904. IEEE, 2010.
37. Chengdong Tao, Albrecht Petzoldt, and Jintai Ding. Improved key recovery of the hfev- signature scheme. *IACR Cryptol. ePrint Arch.*, 2020:1424, 2020.
38. Enrico Thomae and Christopher Wolf. Solving underdetermined systems of multivariate quadratic equations revisited. In Marc Fischlin, Johannes Buchmann, and Mark Manulis, editors, *Public Key Cryptography - PKC 2012 - 15th International Conference on Practice and Theory in Public Key Cryptography, Darmstadt, Germany, May 21-23, 2012. Proceedings*, volume 7293 of *Lecture Notes in Computer Science*, pages 156–171. Springer, 2012.
39. Thomas Prest, Pierre-Alain Fouque, Jeffrey Hoffstein, Paul Kirchner, Vadim Lyubashevsky, Thomas Pornin, Thomas Ricosset, Gregor Seiler, William Whyte, Zhenfei Zhang. FALCON: Fast-Fourier Lattice-based Compact Signatures over NTRU. available at `https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Falcon-Round3.zip`, 2020. Technical report, National Institute of Standards and Technology.
40. Shigeo Tsujii, Masahito Gotaishi, Kohtaro Tadaki, and Ryou Fujita. Proposal of a signature scheme based on sts trapdoor. In Nicolas Sendrier, editor, *PQCrypto*, volume 6061 of *Lecture Notes in Computer Science*, pages 201–217. Springer, 2010.

41. Shigeo Tsujii, Toshiya Itoh, Atsushi Fujioka, Kaoru Kurosawa, and Tsutomu Matsumoto. A public-key cryptosystem based on the difficulty of solving a system of nonlinear equations. *Systems and Computers in Japan*, 19(2):10–18, 1988.
42. Vadim Lyubashevsky, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Peter Schwabe, Gregor Seiler, Damien Stehle, Shi Bai. CRYSTALS-Dilithium. available at `https://csrc.nist.gov/CSRC/media/Projects/post-quantum-cryptography/documents/round-3/submissions/Dilithium-Round3.zip`, 2020. Technical report, National Institute of Standards and Technology.
43. Christopher Wolf, An Braeken, and Bart Preneel. Efficient cryptanalysis of RSE(2)PKC and RSSE(2)PKC. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, volume 3352 of *Lecture Notes in Computer Science*, pages 294–309. Springer, 2004.
44. Christopher Wolf and Bart Preneel. Equivalent keys in multivariate quadratic public key systems. *J. Mathematical Cryptology*, 4(4):375–415, 2011.

A Toy Example

In this section we present a toy example of QSTS. We illustrate the selection of F, $\widetilde{F}$ and $\widehat{F}$ and then present a valid public key. Finally, we demonstrate inversion of the public key.

We randomly select a function F of STS shape:

$$\begin{aligned}
y_1 &= 5x_1^2\\
y_2 &= 6x_1^2 + 4x_1x_2\\
y_3 &= 6x_1^2 + 3x_1x_2 + 5x_2^2 + 3x_1x_3 + x_3^2\\
y_4 &= 5x_1^2 + 5x_1x_2 + 6x_1x_3 + x_2x_3 + x_1x_4 + 6x_2x_4 + 6x_3x_4 + x_4^2
\end{aligned}$$

We then construct the parametric family of STS functions, $\widetilde{F}$, by randomly multiplying monomials in F by random linear forms in the variables w_1, w_2:

$$\begin{aligned}
y_1 &= 3x_1^2w_1 + 3x_1^2w_2\\
y_1 &= 2x_1^2w_1 + 4x_1x_2w_1 + 5x_1^2w_2\\
y_1 &= 6x_1x_2w_1 + 5x_2^2w_1 + 5x_1x_3w_1 + 5x_3^2w_1 + x_1^2w_2 + 6x_1x_2w_2 + 6x_2^2w_2 + 3x_1x_3w_2\\
y_1 &= 2x_1^2w_1 + 6x_1x_3w_1 + x_2x_3w_1 + 6x_1x_4w_1 + 5x_2x_4w_1 + x_3x_4w_1 + 2x_1^2w_2\\
&\quad + 6x_1x_2w_2 + 5x_1x_3w_2 + x_2x_3w_2 + 5x_1x_4w_2 + 2x_2x_4w_2 + 5x_4^2w_2
\end{aligned}$$

Next, we do the final step of performing random replacements $x_iw_j = z_{ij}$ and adding random summands of the forms $ax_iz_{jk} - ax_jz_{ik}$ and $az_{ij}z_{rs} - az_{is}z_{rj}$ to

obtain $\widehat{F}$. In matrix form we have:

$$\widehat{\mathbf{F}}_1 = \begin{bmatrix} 0&0&0&0&5&5&3&0&3&6&2&4 \\ 0&0&0&0&4&0&0&0&1&0&1&1 \\ 0&0&0&0&4&1&6&0&0&0&6&1 \\ 0&0&0&0&5&3&6&6&1&6&0&0 \\ 5&4&4&5&0&0&0&6&0&2&0&4 \\ 5&0&1&3&0&0&1&0&5&0&3&0 \\ 3&0&6&6&0&1&0&0&0&4&0&3 \\ 0&0&0&6&6&0&0&0&3&0&4&0 \\ 3&1&0&1&0&5&0&3&0&0&0&6 \\ 6&0&0&6&2&0&4&0&0&0&1&0 \\ 2&1&6&0&0&3&0&4&0&1&0&0 \\ 4&1&1&0&4&0&3&0&6&0&0&0 \end{bmatrix}, \widehat{\mathbf{F}}_2 = \begin{bmatrix} 0&0&0&0&1&6&3&0&5&6&5&6 \\ 0&0&0&0&6&0&0&0&5&3&3&0 \\ 0&0&0&0&2&1&2&4&0&0&2&4 \\ 0&0&0&0&2&1&4&0&5&3&0&0 \\ 1&6&2&2&0&0&0&2&0&0&0&6 \\ 6&0&1&1&0&0&5&0&0&0&1&0 \\ 3&0&2&4&0&5&0&0&0&0&0&1 \\ 0&0&4&0&2&0&0&0&0&0&6&0 \\ 5&5&0&5&0&0&0&0&0&0&0&2 \\ 6&3&0&3&0&0&0&0&0&0&5&0 \\ 5&3&2&0&0&1&0&6&0&5&0&0 \\ 6&0&4&0&6&0&1&0&2&0&0&0 \end{bmatrix},$$

$$\widehat{\mathbf{F}}_3 = \begin{bmatrix} 0&0&0&0&0&4&4&1&0&0&6&1 \\ 0&0&0&0&6&2&6&3&2&1&2&6 \\ 0&0&0&0&6&5&5&6&6&0&2&1 \\ 0&0&0&0&1&6&5&1&5&6&0&0 \\ 0&6&6&1&0&0&0&5&0&4&0&1 \\ 4&2&5&6&0&0&2&0&3&0&6&0 \\ 4&6&5&5&0&2&0&0&0&2&0&0 \\ 1&3&6&1&5&0&0&0&5&0&0&0 \\ 0&2&6&5&0&3&0&5&0&0&0&3 \\ 0&1&0&6&4&0&2&0&0&0&4&0 \\ 6&2&2&0&0&6&0&0&0&4&0&0 \\ 1&6&1&0&1&0&0&0&3&0&0&0 \end{bmatrix}, \widehat{\mathbf{F}}_4 = \begin{bmatrix} 0&0&0&0&1&1&4&2&5&0&5&4 \\ 0&0&0&0&3&1&0&0&2&4&2&1 \\ 0&0&0&0&5&6&2&0&0&0&3&3 \\ 0&0&0&0&5&2&4&0&1&4&0&6 \\ 1&3&5&5&0&0&0&0&0&1&0&3 \\ 1&1&6&2&0&0&0&0&6&0&4&0 \\ 4&0&2&4&0&0&0&0&0&0&0&4 \\ 2&0&0&0&0&0&0&0&0&0&3&0 \\ 5&2&0&1&0&6&0&0&0&0&0&4 \\ 0&4&0&4&1&0&0&0&0&0&3&0 \\ 5&2&3&0&0&4&0&3&0&3&0&0 \\ 4&1&3&6&3&0&4&0&4&0&0&0 \end{bmatrix},$$

$$\mathbf{U} = \begin{bmatrix} 1&4&0&2&0&6&4&1&3&6&1&5 \\ 2&4&1&5&2&2&2&3&5&1&1&5 \\ 6&2&4&1&4&3&0&0&1&6&3&5 \\ 5&1&1&1&0&4&0&0&0&3&0&5 \\ 3&6&4&1&6&5&2&5&4&4&3&5 \\ 1&6&3&5&1&1&5&3&6&3&1&6 \\ 4&1&2&4&3&5&0&4&3&4&3&6 \\ 1&6&5&4&0&0&2&4&3&3&1&2 \\ 3&4&6&4&5&1&5&0&4&4&6&4 \\ 4&1&5&6&3&6&4&6&4&1&0&1 \\ 0&3&2&0&3&5&0&5&5&6&1&6 \\ 3&1&0&4&0&3&4&3&5&5&3&5 \end{bmatrix}, \text{ and } \mathbf{T} = \begin{bmatrix} 3&5&6&4 \\ 3&0&2&1 \\ 1&5&0&0 \\ 5&3&1&6 \end{bmatrix}.$$

$$\mathbf{P}_1 = \begin{bmatrix} 4&5&3&3&6&2&3&2&4&4&3&0 \\ 5&2&0&1&5&5&4&1&0&1&2&6 \\ 3&0&6&3&6&1&3&2&4&5&0&4 \\ 3&1&3&3&2&3&0&1&5&1&2&6 \\ 6&5&6&2&4&0&4&3&0&6&6&6 \\ 2&5&1&3&0&3&1&2&4&5&3&4 \\ 3&4&3&0&4&1&1&5&6&2&6&3 \\ 2&1&2&1&3&2&5&4&1&2&0&1 \\ 4&0&4&5&0&4&6&1&3&4&2&0 \\ 4&1&5&1&6&5&2&2&4&5&3&0 \\ 3&2&0&2&6&3&6&0&2&3&0&5 \\ 0&6&4&6&6&4&3&1&0&0&5&6 \end{bmatrix}, \mathbf{P}_2 = \begin{bmatrix} 3&3&1&6&3&6&5&2&2&0&1&6 \\ 3&3&3&6&1&2&4&0&0&0&6&3 \\ 1&3&4&6&1&4&5&1&1&4&3&0 \\ 6&6&6&0&1&2&1&3&2&6&6&3 \\ 3&1&1&1&3&0&0&3&3&4&2&5 \\ 6&2&4&2&0&2&1&1&0&1&0&4 \\ 5&4&5&1&0&1&0&0&4&4&0&6 \\ 2&0&1&3&3&1&0&3&1&4&1&5 \\ 2&0&1&2&3&0&4&1&0&0&0&0 \\ 0&0&4&6&4&1&4&4&0&5&5&2 \\ 1&6&3&6&2&0&0&1&0&5&4&1 \\ 6&3&0&3&5&4&6&5&0&2&1&3 \end{bmatrix},$$

$$\mathbf{P}_3 = \begin{bmatrix} 4&0&3&1&2&4&5&2&4&0&1&2 \\ 0&3&1&6&5&0&1&3&0&3&1&4 \\ 3&1&2&4&6&4&2&3&3&2&1&1 \\ 1&6&4&6&1&4&3&3&3&6&5&1 \\ 2&5&6&1&3&1&2&6&6&0&4&0 \\ 4&0&4&4&1&6&5&6&0&0&0&1 \\ 5&1&2&3&3&2&5&2&5&2&5&3&4 \\ 2&3&3&3&6&6&5&1&0&2&1&6 \\ 4&0&3&3&6&0&2&0&1&0&4&4 \\ 0&3&2&6&0&0&5&2&0&1&5&6 \\ 1&1&1&5&4&0&3&1&4&5&0&2 \\ 2&4&1&1&0&1&4&6&4&6&2&2 \end{bmatrix}, \mathbf{P}_4 = \begin{bmatrix} 5&2&4&3&4&6&5&6&4&1&0&0 \\ 2&5&5&5&3&4&3&5&0&4&5&2 \\ 4&5&6&5&0&0&6&6&4&2&6&4 \\ 3&5&5&5&4&6&2&3&6&2&4&4 \\ 4&3&0&4&3&1&2&1&0&5&0&2 \\ 6&4&0&6&1&2&3&0&5&2&2&3 \\ 5&3&6&2&2&3&1&6&2&1&0&1 \\ 6&5&6&3&1&0&6&2&3&6&1&0 \\ 4&0&4&6&0&5&2&3&6&6&4&5 \\ 1&4&2&2&5&2&1&6&6&5&5&1 \\ 0&5&6&4&0&2&0&1&4&5&4&2 \\ 0&2&4&4&2&3&1&0&5&1&2&3 \end{bmatrix}.$$

Finally, we choose input and output transformations U and T and derive the above public key.

We now demonstrate the inversion process for the public key. Given the ciphertext

$$\mathbf{y} = \begin{bmatrix} 3&2&2&5 \end{bmatrix},$$

we first randomly select the nonzero vector of auxiliary variables

$$\mathbf{w} = \begin{bmatrix} 6&3 \end{bmatrix}.$$

Then evaluating $\widetilde{F}$ at $\mathbf{w}$ we obtain the STS central map $\widetilde{F}(\cdot, \mathbf{w})$:

$$\begin{aligned}
y_1 &= 6x_1^2, \\
y_2 &= 6x_1^2 + 3x_1x_2, \\
y_3 &= 3x_1^2 + 5x_1x_2 + 6x_2^2 + 4x_1x_3 + 2x_3^2, \\
y_4 &= 4x_1^2 + 4x_1x_2 + 2x_1x_3 + 2x_2x_3 + 2x_1x_4 + x_2x_4 + 6x_3x_4 + x_4^2.
\end{aligned}$$

We then compute $\mathbf{y}\mathbf{T}^{-1} = \begin{bmatrix} 5 & 2 & 2 & 3 \end{bmatrix}$ and find the preimage under the above STS map:

$$\mathbf{u} = \begin{bmatrix} 3 & 2 & 5 & 6 \end{bmatrix}.$$

Next, we append

$$\mathbf{u} \otimes \mathbf{w} = \begin{bmatrix} 4 & 2 & 5 & 6 & 2 & 1 & 1 & 4 \end{bmatrix}$$

to $\mathbf{u}$. Finally we compute the plaintext

$$\mathbf{x} = (\mathbf{u} \oplus (\mathbf{u} \otimes \mathbf{w}))\, U^{-1} = \begin{bmatrix} 1 & 5 & 4 & 2 & 0 & 3 & 2 & 1 & 5 & 6 & 1 & 4 \end{bmatrix}.$$

We check that indeed $P(\mathbf{x}) = \mathbf{y}$.