Revisiting the Security of Salted UOV Signature

Sanjit Chatterjee[1], M. Prem Laxman Das[2], and Tapas Pandit[1(✉)]

[1] Department of Computer Science and Automation, Indian Institute of Science Bangalore,Bangalore, India
{sanjit,tapas}@iisc.ac.in
[2] Society for Electronic Transactions and Security, Chennai, India

Abstract. Due to the recent attack of Beullens on Rainbow, the crypto community looks back again at the unbalanced oil-and-vinegar (UOV) signature. The original UOV does not have any formal security reduction. It was Sakumoto et al. who added a random salt to the original UOV signature to give a reduction under the UOV-inversion (UOVI) problem in the classical random oracle model (CROM).

In this paper, we revisit the security of salted UOV signature. We start by identifying some issues related to programming the random oracle and the distribution of the salt. Then provide a security reduction of the salted UOV signature in the CROM that clearly addresses these issues. One crucial requirement of our reduction is that the field size needs to be asymptotically superpolynomial in the security parameter. We also give a security reduction of the salted UOV under the UOVI problem in the quantum random oracle model. This work is hoped to aid further concrete security analysis and thereby guide parameter choice of UOV-based schemes in the context of future standardization of post-quantum signature.

Keywords: Digital signature · Multivariate cryptography · UOV · Post-quantum security · QROM

1 Introduction

Multivariate quadratic polynomials (MQ) based signatures [DS05,PCY+15, CHR+16] are attractive candidates for post-quantum cryptography due to their fast verification and short signature. One of these is Rainbow which was a finalist in the recently concluded third round of NIST PQC Standardization competition. Rainbow [DS05] is a multilayered version of unbalanced oil-and-vinegar (UOV) signature scheme [KPG99]. Several variants of signatures, e.g., identity-based signature [CLND19,CDP21], blind signature [PSM17] and ring signature [MP17] have been designed in the MQ-setting using Rainbow (or UOV) as primary building block. Note that Rainbow as a multilayered extension of UOV [KPG99] was solely introduced to gain efficiency. For practical applications,

T. Isobe and S. Sarkar (Eds.): INDOCRYPT 2022, LNCS 13774, pp. 697–719, 2022.
https://doi.org/10.1007/978-3-031-22912-1_31

two-layered version of Rainbow is mainly recommended as further increasing the number of layers does not significantly improve its efficiency. However, the recent attack [Beu22] on the two-layered Rainbow basically works by peeling off the 2nd layer followed by an existing UOV attack in [KPG99] on the 1st layer with much smaller parameter size compared to the original UOV. The attack motivates the research community to look back at the UOV signature with renewed interest. Thus a rigorous security analysis would be useful in designing UOV-based signatures, which could be candidates for future standardization efforts.

Note that the original UOV proposal of [KPG99] does not have any formal security proof. It was Sakumoto et al. [SSH11] who first formally studied security of the UOV signature. They introduced a random salt to make the output signature somewhat uniform and then argued security in the classical random oracle model (CROM) from UOV-inversion (UOVI) problem[1] using the FDH-technique [BR93]. In [SSH11], the authors consider the hash function to be $\mathcal{H} : \mathcal{M} \times \mathsf{SaltSpac} \rightarrow \mathbb{F}^m$, where $\mathbb{F}$ is the underlying field. That is, the hash arguments have the form: (m, s), where m is the message and s is the salt (a binary string). A valid signature for m under the salted UOV is of the form: $\sigma = (\boldsymbol{x}, s)$ such that $\mathcal{P}(\boldsymbol{x}) = \mathcal{H}(\mathsf{m}, s)$, where $\mathcal{P} : \mathbb{F}^n \rightarrow \mathbb{F}^m$ is the UOV public map. In the security reduction, $\mathcal{H}$ is treated as a random oracle.

Our Result. In this paper, we revisit the security reduction of the salted UOV [SSH11] and identify some gaps pertaining to programming the random oracle and distribution of the salt (Sect. 3). In particular, when queried with (m, s) the random oracle involved in the signature oracle is programmed to return $\mathcal{P}(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathbb{F}^n$ is randomly chosen. That is, the authors implicitly assumed that for a random $\boldsymbol{x} \in \mathbb{F}^n$, $\mathcal{P}(\boldsymbol{x})$ is uniform. The paper also assumed that the salt part of the output signature is uniform, although the distribution of the salt actually depends on the size of the underlying field.

We then provide (Sect. 4) a security reduction of the salted UOV signature in the CROM that clearly addresses these issues. Here we consider the salted homogeneous UOV scheme, but through the subspace description [Beu21] of the UOV-trapdoor (Sect. 4.1). The main reason for using [Beu21] is that it improves secret key sizes (Sect. 4.2). For the reduction, we assume neither the uniformity of $\mathcal{P}(\boldsymbol{x})$ nor the uniformity of the salt involved in the output signature. We essentially show that both distributions deviate from the respective uniform distributions by at most $1/q$ (Proposition 1 and Corollary 2), where q is the size of the underlying field. One crucial implication of our result is that the field size q needs to be asymptotically superpolynomial in the security parameter. Suppose the upper bound on the numbers of signature queries and random oracle queries in practice are 2^{20} and 2^{60} respectively. Then, from a back-of-the-envelope calculation based on Theorem 1, one can see that the underlying field has to be chosen of size roughly 2^{88} for 128-bit security[2]. This will surely impact the efficiency of

[1] Given a random UOV public map $\mathcal{P} : \mathbb{F}^n \rightarrow \mathbb{F}^m$ and a random element $\boldsymbol{y} \in \mathbb{F}^m$, find an $\boldsymbol{x} \in \mathbb{F}^n$ such that $\mathcal{P}(\boldsymbol{x}) = \boldsymbol{y}$.

[2] Note that whatever the parameter choice of UOV, the unavoidable degradation due to the total number of random oracle queries will always be there.

the scheme. Thus deriving the parameter sizes and the consequent implication on efficiency based on a concrete analysis of our security reduction of salted UOV could be an interesting future work.

In principle, it is desirable to have security proof in the quantum random oracle model (QROM), rather than just in CROM. We achieve this for salted UOV by providing a security reduction (Sect. 5) from the UOVI problem in the QROM. Again based on this reduction, we do not provide any parameter choice, other than pointing out the fact that q needs to be asymptotically superpolynomial in the security parameter (Theorem 2).

2 Preliminaries

For $a \in \mathbb{N}\setminus\{0\}$, define $[a] = \{x \in \mathbb{N}\setminus\{0\} : x \leq a\}$. For a set X, we write $x \xleftarrow{\$} X$ to mean that x is drawn uniformly at random from X. For an algorithm A and its input x, the notation $y \leftarrow A(x)$ denotes that when A is run on x, it outputs y. We use bold-face lower case letters, e.g., $\boldsymbol{x}$ to denote column vectors. The i-th entry of $\boldsymbol{x}$ is denoted by $\boldsymbol{x}_i$. For a matrix A, the notation $A^\top$ is used to denote its transpose. The fixed finite field on which all the operations take place is denoted by $\mathbb{F}$. The notation q will denote the size of the field $\mathbb{F}$. We make no assumptions about the characteristic of $\mathbb{F}$.

We shall consider only homogeneous quadratic polynomials in n variables over $\mathbb{F}$. While discussing UOV scheme, the number of polynomials in the secret system $\mathcal{F}$ and that in the public system $\mathcal{P}$ will be m. This number would match the number of oil variables, as per the usual description of UOV scheme. The oil (vinegar) variables will be last m (respectively, the first $v = n - m$) of them among $\{X_1, \ldots, X_n\}$. With the secret and public systems one can associate polynomial maps $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ and $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$, respectively, which will just denote evaluation. All the polynomials which appear in this work are homogeneous. It is well-known in literature that the security of MQ-based systems depends mainly the quadratic part of the polynomials involved. The transformation used for mixing the variables in UOV is assumed to be an invertible matrix. Hence the public key obtained will be homogeneous whenever the secret key is so.

2.1 Quadratic Polynomials and Their Matrix Representation

We shall consider homogeneous quadratic polynomials in m variables over a finite field $\mathbb{F}$. Any such $\mathfrak{f}$ has a associated *polar form* $\mathfrak{f}'$, which is symmetric bilinear, satisfying

$$\mathfrak{f}'(X, Y) = \mathfrak{f}(X + Y) - \mathfrak{f}(X) - \mathfrak{f}(Y).$$

With every homogeneous quadratic polynomial one can associate a matrix. The matrix representing the polynomial is defined as follows.

Definition 1. *Let $\mathfrak{f}$ be a homogeneous quadratic polynomial over $\mathbb{F}$ in n variables. An $n \times n$ matrix $M_\mathfrak{f}$ is said to represent $\mathfrak{f}$ if*

$$\mathfrak{f}(X) = X^\top M_\mathfrak{f} X,$$

where $X = (X_1, \ldots, X_m)^\top$ *is a column vector of variables.*

Remark 1. The polar form of the quadratic form is bilinear. There is an obvious way (see [Beu21]) for obtaining the matrix representing the polar form, depending on the characteristic of the underlying field. If M'_f denotes this matrix, then $\mathfrak{f}'(X, Y) = X^\top M'_\mathfrak{f} Y$.

2.2 (Unbalanced) Oil-Vinegar Signature Schemes

We will be following the treatment of Kipnis *et al.* [KPG99]. Let $\mathbb{F}$ be a fixed finite field. As usual, let n and m be positive integers, and set $v = n - m$. Let $\{X_1, \ldots, X_v\}$ denote the (ordered) set of vinegar variables and $\{X_{v+1}, \ldots, X_n\}$ that of oil variables. The message (digest) space is $\mathbb{F}^m$ and the signature space is $\mathbb{F}^n$ (for plain-UOV scheme).

The central object in such schemes is a polynomial of the following special form. *The oil-vinegar type polynomial* is a quadratic polynomial over $\mathbb{F}$ in the variables described above, but without any quadratic terms involving only the oil variables. In other words, a oil variable is not allowed to mix with another oil variable in such a polynomial. The general form of such a polynomial is as follows:

$$\phi(X_1, \ldots, X_m) = \sum_{j=1}^{v} \sum_{k=j}^{n} \alpha_{jk} X_j X_k + \sum_{j=1}^{n} \beta_j X_j + \gamma. \tag{1}$$

For a given polynomial of the form in Eq. (1), if values are assigned to all vinegar variables, the resulting polynomial is linear in oil variables. This feature is the central theme of the trapdoor.

Let $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$ denote an invertible linear map. Such a transformation is used for mixing the input variables. The oil-vinegar trapdoor is described as follows.

Definition 2. *(Oil-Vinegar Trapdoor) Let* $\mathbb{F}$ *be a system of* m *polynomials of the form given in Eq. (1). Let* $\mathcal{T}$ *be a invertible linear transformation on* $\mathbb{F}^n$*. Let* $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$ *be the polynomial system obtained by composing each polynomial in* $\mathbb{F}$ *with* $\mathcal{T}$ *(i.e., transformed polynomials when* $\mathcal{T}$ *acts on the vector of variables). Given* $\mathcal{P}$ *and* $\boldsymbol{\tau} \in \mathbb{F}^m$*, the challenge is to solve* $\mathcal{P}(\cdot) = \boldsymbol{\tau}$*.*

Remark 2. Solving $\mathcal{P}(\cdot) = \boldsymbol{\tau}$ directly is assumed to be hard. But the knowledge of the trapdoor information, namely $\mathbb{F}$ and $\mathcal{T}$, can be used for solving such a system [KPG99]. Solving $\mathcal{F}(\cdot) = \boldsymbol{\tau}$ is easy. The strategy is to assign random values for vinegar variables and solving the linear system involving only the oil variables. The resulting assignment is then inverted under the affine transformation $\mathcal{T}$. The process of assigning values to vinegar variables and solving the resulting system of linear equations is repeated until one valid assignment for all variables is found.

Matrix Description of Homogeneous Quadratic System. Studying the matrices representing the $\mathcal{F}$ and $\mathcal{P}$ systems becomes useful from the analysis point of view. Let us consider the component polynomials $\mathfrak{f}$ involved in the system $\mathcal{F}$ to be homogeneous quadratic polynomials. Since every polynomial $\mathfrak{f}$ in the system $\mathcal{F}$ is devoid of oil-oil term and every polynomial $\mathfrak{g}$ in $\mathcal{P}$ is constructed as $\mathfrak{g} = \mathfrak{f} \circ \mathcal{T}$, the (block) form of their corresponding matrices will be

$$M_{\mathfrak{f}} = \begin{bmatrix} A & B \\ 0 & 0 \end{bmatrix} \text{ and } M_{\mathfrak{g}} = T^{\top} M_{\mathfrak{f}} \mathcal{T}, \tag{2}$$

where A is a $v \times v$ upper triangular matrix, B is a $v \times m$ matrix, and 0's are all zero matrices of suitable orders such that $M_{\mathfrak{f}}$ is an $n \times n$ matrix.

2.3 Linear Subspace Interpretation of Oil-Vinegar Trapdoor

Beullens [Beu21] takes a subspace approach to the oil-vinegar trapdoor description. The public key is a MQ system $\mathcal{P}$ (m MQ polynomials in n variables) which vanishes on a secret subspace $\mathsf{O} \subset \mathbb{F}^n$ of dimension m. The trapdoor is set as follows. First, the subspace O is chosen at random. Then a system $\mathcal{P}$, consisting of m multivariate quadratic polynomials in n variables, vanishing at this subspace O, is chosen uniformly at random. The trapdoor information is the description of O. For a target $\boldsymbol{\tau} \in \mathbb{F}^m$, solving $\mathcal{P}(\cdot) = \boldsymbol{\tau}$ is easy. Notice that

$$\mathcal{P}(\boldsymbol{v} + \boldsymbol{o}) = \mathcal{P}(\boldsymbol{v}) + \mathcal{P}(\boldsymbol{o}) + \mathcal{P}'(\boldsymbol{v}, \boldsymbol{o}) \tag{3}$$

holds for any $\boldsymbol{o}$ coming from the subspace O and any $\boldsymbol{v}$ coming from $\mathbb{F}^n$. Thus $\mathcal{P}(\cdot) = \boldsymbol{\tau}$ can be solved by solving

$$\mathcal{P}'(\boldsymbol{v}, \boldsymbol{o}) = \boldsymbol{\tau} - \mathcal{P}(\boldsymbol{v}),$$

where $\boldsymbol{o} \in \mathsf{O}$. The above system is linear in variable $\boldsymbol{o}$. For, the first term in the right hand side of Eq. (3) is fixed once $\boldsymbol{v}$ is fixed, the second term is zero since $\boldsymbol{o}$ is from the distinguished subspace O and the third term is linear in oil variables.

On the other hand, solving the MQ system $\mathcal{P}$, without the knowledge of the trapdoor information is assumed to be hard.

2.4 Syntax and Security of Signature Scheme

Definition 3 (Signature Scheme). *It consists of three PPT algorithms* - KeyGen, Sign *and* Ver.

- KeyGen*: It takes as input a security parameter κ and outputs a public and private key pair $(\mathcal{PK}, \mathcal{SK})$.*
- Sign*: It takes as input a message $\mathsf{m} \in \mathcal{M}$, where $\mathcal{M}$ is the message space, and the secret key $\mathcal{SK}$ and outputs a signature σ.*
- Ver*: It takes as input a message-signature pair (m, σ) and the public key $\mathcal{PK}$. It outputs a value 1 if (m, σ) is a valid message-signature pair else it outputs 0.*

Correctness: *For all* $(\mathcal{PK}, \mathcal{SK}) \leftarrow \mathsf{KeyGen}(1^\kappa)$ *and for all messages* $\mathsf{m} \in \mathcal{M}$*, it is required that*

$$\mathsf{Ver}(\mathsf{m}, \mathsf{Sign}(\mathsf{m}, \mathcal{SK}), \mathcal{PK}) = 1.$$

Next we define security model of the signature scheme. A security notion very useful in practice is called existentially unforgeable under chosen message attack (EUF-CMA).

Definition 4 (EUF-CMA). *A signature scheme is said to be EUF-CMA secure if for all quantum PPT algorithms* $\mathcal{A}$*, the advantage*

$$\mathsf{Adv}_{\mathcal{A}}^{EUF\text{-}CMA}(\kappa) = \mathsf{Pr}\left[\mathsf{Ver}(\mathsf{m}^*, \sigma^*, \mathcal{PK}) = 1 \;\middle|\; \begin{matrix}(\mathcal{PK}, \mathcal{SK}) \leftarrow \mathsf{KeyGen}(1^\kappa); \\ (\mathsf{m}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathcal{PK})\end{matrix}\right]$$

is a negligible function in κ*, where* $\mathcal{A}$ *is provided access to the sign oracle* $\mathcal{O}_{\mathsf{Sign}}$ *with a natural restriction that* $\mathsf{m}^* \neq \mathsf{m}$ *for all messages* m *queried to* $\mathcal{O}_{\mathsf{Sign}}$*.*

3 Revisiting the Security Reduction of Salted UOV

The unbalanced oil-and-vinegar (UOV) signature was proposed in [KPG99] to protect from the attack of [KS98] on the balanced oil-and-vinegar signature of Patarin [Pat97]. However, the authors of UOV-signature did not provide any formal security proof of their construction. The distribution of the signatures generated in the original UOV-signature [KPG99] is not uniform, even if the underlying hash function is treated as random oracle. Therefore, the FDH-technique [BR93] is not directly applicable in arguing security of the UOV-signature.

The signature scheme, salted UOV was presented in [SSH11, Section 4.1] (see Appendix B). The salt is appended to the message and hashed, thereby a system $\mathcal{P}(\cdot) = \mathcal{H}(\mathsf{m}||s)$ is set up. A solution is obtained by first putting values for the vinegar variables and solving (a linear system) for the oil variables. If the system does not have a solution, a fresh salt is chosen. The authors point out that, this way, the distribution of the signature will be uniform, and hence, the FDH-technique can be used to argue the security of the salted UOV signature.

3.1 On the Simulation of Random Oracle and Salt

First, we informally describe the FDH-style security reduction [BR93]. Let $(f : \mathfrak{D} \rightarrow \mathfrak{R}, y^* \in \mathfrak{R})$ be the given problem instance, where f is a trapdoor permutation and the goal is to find an $x^* \in \mathfrak{D}$ such that $f(x^*) = y^*$. Recall that the FDH-signature for a message m is of the form $\sigma = x$, where $y = \mathcal{H}(\mathsf{m})$, $x = f^{-1}(y)$ and $\mathcal{H} : \mathcal{M} \rightarrow \mathfrak{R}$ is the underlying hash function. A message-signature pair (m, σ) is valid, if $f(\sigma) = \mathcal{H}(\mathsf{m})$.

If an adversary can produce a valid forgery (m^*, σ^*) for this signature scheme, then a solver for the above problem can be constructed. Here the underlying hash function $\mathcal{H} : \mathcal{M} \to \mathfrak{R}$ is treated as a random oracle. That means, the adversary must have queried the corresponding message m^* to the random oracle $\mathcal{H}$. In the reduction, an index is guessed where the forgery message m^* could appear as a random oracle query and the corresponding random oracle value is appropriately programmed. In other words, pick an index i^* randomly as a guess and set $\mathcal{H}(\mathsf{m}_{i^*}) = y^*$. Note that for a correct guess, we have $\mathsf{m}^* = \mathsf{m}_{i^*}$. For a query on message $\mathsf{m} \in \mathcal{M}$ other than m_{i^*}, first pick a signature $\sigma \xleftarrow{\$} \mathfrak{D}$, then program the random oracle at m as $\mathcal{H}(\mathsf{m}) = y = f(\sigma)$ and store the tuple (m, σ, y) in a list List. So using the list, all the oracle queries can be answered. Note that $f(\sigma)$ will be uniform over $\mathfrak{R}$ as f is bijective[3]. If i^* is correctly guessed and (m^*, σ^*) is a valid forgery, then we have $f(\sigma^*) = \mathcal{H}(\mathsf{m}^*) = y^*$, which implies that $\boldsymbol{x}^* = \sigma^*$ is the required solution of the given problem instance.

In [SSH11], the authors showed a security reduction of salted UOV in the CROM under the hardness of UOVI-problem. Since a salt is involved as part of the signature, the FDH-style proof will be slightly different here. We summarize their security proof as follows. In the game between a simulator S and an adversary $\mathcal{A}$, S maintains a list $\mathsf{List}_{\mathsf{uov}}$ of three tuples $(\mathsf{m}, s, \boldsymbol{y})$, where $\boldsymbol{y}$ is the hash of $\mathsf{m}||s$. The random oracle query on challenge message is answered in a similar way as done above. The other queries are answered as follows. For an incoming random oracle query $\mathsf{m}||s$, if $(\mathsf{m}, s, \cdot) \in \mathsf{List}_{\mathsf{uov}}$, then the stored value is returned. Else a random value $\boldsymbol{y}$ is returned and $(\mathsf{m}, s, \boldsymbol{y})$ is appended to the list $\mathsf{List}_{\mathsf{uov}}$. If m is a signing oracle query, the simulator chooses a salt s at random. If $(\mathsf{m}, s, \cdot)$ is in the list, it aborts. Else, it chooses $\boldsymbol{x} \in \mathbb{F}^n$ uniformly at random and returns $(\boldsymbol{x}, s)$ as signature corresponding to m after appending $(\mathsf{m}, s, \boldsymbol{y})$, with $\boldsymbol{y} = \mathcal{P}(\boldsymbol{x})$, to the list $\mathsf{List}_{\mathsf{uov}}$. Similarly as above, when the index i^* is correctly guessed and $(\boldsymbol{x}^*, s^*)$ is a valid forgery for m^*, then $\boldsymbol{x}^*$ will be a solution of the given UOVI-problem instance.

Issue in Random Oracle Programming. Note that while answering the sign-queries, the random oracle $\mathcal{H}$ is programmed by assigning $\mathcal{P}(\boldsymbol{x})$ for random choice of $\boldsymbol{x}$ from $\mathbb{F}^n$. Since $\mathcal{P}$ is neither bijective nor known to be regular, it cannot be definitely said that $\mathcal{P}(\boldsymbol{x})$ is uniform over $\mathbb{F}^m$. Hence, $\mathcal{H}$ is treated as a random oracle in [SSH11] without any proper justification. This, in turn, opens up the possibility of a potential gap in the security claim.

Issue in Salt Distribution. A signature in the salted UOV [SSH11] consists of a salt and an element from $\mathbb{F}^n$. Note that only the salt generated in the last iteration of the loop in the sign algorithm (Algorithm 3) contributes to the final output signature. In other words, the salt in the output signature follows a distribution that samples a couple of salts in a row *without replacement* and outputs the final salt. More precisely, the distribution of the salt in the output signature depends on the rank of an $m \times m$ matrix, which further depends

[3] Note that the reduction also works, if f is considered to be a regular function. Here regular means the preimage sets of all the points in $\mathcal{M}$ under f are of same size.

on $q = |\mathbb{F}|$ (for details, see [SSH11, Section 3.1]). As described earlier, while answering the sign-queries in the reduction, the salts are always chosen uniformly at random. Essentially, this creates a difference between the distributions of salts, one involved in the actual signatures and the other in the simulated signatures. It seems the authors implicitly assume that a computational adversary cannot detect the difference.

4 A Clean Security Reduction of Salted UOV

For addressing the issues raised in the previous section, we consider the underlying maps involved in the public key $\mathcal{P}$ to be homogeneous quadratic polynomials. For general quadratic polynomial maps, closing the above gaps still remains an interesting research problem. Nonetheless, restricting to homogeneous quadratic maps does not weaken the security of the signature as the intractability of the underlying MQ-problem mainly relies on the quadratic part of the MQ-system. Using the result on the distribution of $\mathcal{P}(\boldsymbol{x})$ that we describe in Sect. 4.3, one can derive a clean security proof of the salted homogeneous UOV signature. However, in this paper, we argue the security of an alternative salted UOV signature (see Sect. 4.2) which is based on the subspace approach to UOV trapdoor [Beu21]. The reason for considering this alternative construction is that it improves upon the key sizes a bit. The remainder of this section is organized as follows. We start with the (plain) homogeneous UOV signature based on Beullens' subspace approach in Sect. 4.1. Then, present its salted version in Sect. 4.2. We analyze the distribution of $\mathcal{P}(\boldsymbol{x})$ in Sect. 4.3. Finally, provide a clean proof of the salted homogeneous UOV in Sect. 4.4.

4.1 Homogeneous UOV Signature Scheme Using the Subspace Interpretation

The trapdoor described in Sect. 2.3 can be used to design a signature scheme. We discuss the efficiency aspects of the key generation and signing modules (without salt) in this section. The public key system is an MQ-system, which vanishes on a subspace. The trapdoor information is the description of the subspace. The two major questions are the following. How does one sample a random subspace of $\mathbb{F}^n$ and a uniformly random MQ system which vanishes on this subspace? How does one represent the trapdoor information so that the MQ system can be solved, efficiently, using the trapdoor information? Next we discuss these two points based on [Beu21].

Efficient Setup for the UOV Trapdoor. The trapdoor can be efficiently setup using the matrix representation of the quadratic form. Let $\mathcal{F} = (\mathfrak{f}_1, \ldots, \mathfrak{f}_m)$, be the collection of secret UOV (homogeneous) polynomials. The matrix corresponding to $\mathfrak{f}_i$ has the form

$$M_{\mathfrak{f}_i} = \begin{array}{c} \\ v \\ m \end{array} \begin{array}{c} \begin{array}{cc} v & m \end{array} \\ \begin{bmatrix} A_i & B_i \\ 0 & 0 \end{bmatrix} \end{array} \tag{4}$$

where A_i is a random $v \times v$ upper triangular matrix, B_i is a random $v \times m$ matrix, and 0's are all zero matrices of suitable orders such that $M_{\mathfrak{f}_i}$ is an $n \times n$ matrix. The following subspace

$$\mathsf{O}' = \left\{(x_1, \ldots, x_n)^\top \in \mathbb{F}^n : x_1 = \cdots = x_{n-m} = 0\right\}$$

is called *oil subspace* of dimension m. Notice that every $\mathfrak{f}_i$ vanishes on O'. If $\mathcal{P} = (\mathfrak{g}_1, \ldots, \mathfrak{g}_m)$ is the public key system, where $\mathfrak{g}_i = \mathfrak{f}_i \circ \mathcal{T}$, then every quadratic form in $\mathcal{P}$ vanishes on $\mathsf{O} = \mathcal{T}^{-1}(\mathsf{O}')$, where $\mathcal{T}$ is an invertible matrix. Since $\mathcal{T}$ is invertible, the dimension of O is equal to m. So, O will be a random subspace when $\mathcal{T}$ is a random invertible matrix.

Note that the distribution of the public keys generated in both ways, one as discussed above and the traditional one are the same. Beullens also pointed out in [Beu21] that the public key generated using the subspace description (discussed in Sect. 2.3) and using the traditional description have the identical distribution.

Solving the MQ System Using Trapdoor, Efficiently. We discuss how a solution for $\mathcal{P}(\cdot) = \boldsymbol{\tau}$ can be obtained, efficiently, using the trapdoor information. Recall that the trapdoor information is a description of the subspace on which this system $\mathcal{P}$ vanishes. From Eq. (3), solving this system amounts to solving $\mathcal{P}'(\boldsymbol{v}, \boldsymbol{o}) = \boldsymbol{\tau}'$ for $\boldsymbol{o} \in \mathsf{O}$, where $\boldsymbol{\tau}' = \boldsymbol{\tau} - \mathcal{P}(\boldsymbol{v})$ and $\boldsymbol{v} \in \mathbb{F}^n$ is fixed. For a homogeneous quadratic polynomial $\mathfrak{g}$, the polar form is given by $\mathfrak{g}'(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{x}^\top M'_{\mathfrak{g}} \boldsymbol{y}$ for all $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{F}^n$ (see Remark 1).

We can describe the subspace $\mathsf{O} \subset \mathbb{F}^n$ of dimension m using column-span of a full-rank $n \times m$ matrix $\overline{M}$. For, if O is generated by a linearly independent set of vectors $\{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_m\}$ from $\mathbb{F}^n$, then i-th column of $\overline{M}$ will be $\boldsymbol{w}_i$. Thus $\overline{M}$ is a full-rank matrix and any element of the subspace O can be written as $\boldsymbol{o} = \overline{M}\boldsymbol{y}$ for some $\boldsymbol{y} \in \mathbb{F}^m$.

We now describe an effective procedure for solving the public key system. For each public key polynomial $\mathfrak{g}$, a row vector $\boldsymbol{c}_{\mathfrak{g}}$ is computed as $\boldsymbol{c}_{\mathfrak{g}} = \boldsymbol{v}^\top M'_{\mathfrak{g}} \overline{M}$, where $\boldsymbol{v}$ is a random vector chosen from $\mathbb{F}^n$. We then consider the following linear system: $C\boldsymbol{y} = \boldsymbol{\tau}'$ (recall that $\boldsymbol{\tau}' = \boldsymbol{\tau} - \mathcal{P}(\boldsymbol{v})$), where the $\mathfrak{g}$-th row of the matrix C is the row vector $\boldsymbol{c}_{\mathfrak{g}}$. If a solution $\boldsymbol{y} \in \mathbb{F}^m$ to the system exists, an element $\boldsymbol{o} \in \mathsf{O}$ can hence be obtained as $\boldsymbol{o} = \overline{M}\boldsymbol{y}$. The quantity $\boldsymbol{v} + \boldsymbol{o}$ is a solution for $\mathcal{P}(\cdot) = \boldsymbol{\tau}$.

Remark 3. Note that the matrix C can also be written as $C = C' \cdot \overline{M}$, where the row of C' corresponding to the public polynomial $\mathfrak{g}$ is given by $\boldsymbol{c}'_{\mathfrak{g}} = \boldsymbol{v}^\top M'_{\mathfrak{g}}$. When the map $\mathcal{P}'(\boldsymbol{v}, .) : \mathsf{O} \to \mathbb{F}^m$ is non-singular, then the rank of C' will be m,

which in turn implies that the matrix C will be non-singular as $\overline{M}$ is an $n \times m$ full-rank matrix. In [Beu21], Beullens uses the following fact:

$$\mathsf{Pr}[\mathcal{P}'(\boldsymbol{v}, .) : \mathsf{O} \to \mathbb{F}^m \text{ is non-singular} : \boldsymbol{v} \xleftarrow{\$} \mathbb{F}^n] \approx (1 - 1/q). \tag{5}$$

So, the procedure described above is expected to terminate after a few trials.

The algorithmic version of the above description is given in Appendix A. The signature scheme derived from the trapdoor is also described there.

4.2 Salted Homogeneous UOV

In this section, we illustrate a signature scheme, designed using the subspace description of the UOV trapdoor [Beu21]. The approach is similar to that used in [SSH11]. A salt is used for making the security reduction to go through in the random oracle model. Let us refer to this signature as salted homogeneous UOV (SHUOV) signature.

KeyGen. This takes the security parameter 1^κ as input and outputs the public and secret keys. The secret key $\mathcal{SK}$ is a description of the subspace $\mathsf{O} \subset \mathbb{F}^n$, that is, an $n \times m$ full-rank matrix $\overline{M}$ (as mentioned in Sect. 4.1). The public key $\mathcal{PK}$ is the system $\mathcal{P}$ consisting of m MQ-polynomials in n variables which vanishes at O. See Sect. 4.1 for a description. A hash function $\mathcal{H} : \mathcal{M} \times \mathsf{SaltSpac} \to \mathbb{F}^m$ for converting message into a fixed-length digest is known publicly, where $\mathcal{M}$ and $\mathsf{SaltSpac} = \{0,1\}^{\ell_s}$ are respectively the message space and the salt space. Note that the signature space of SHUOV-signature is $\Sigma = \mathbb{F}^n \times \mathsf{SaltSpac}$.

Sign. This takes message m and the secret key $\mathcal{SK}$ as input and outputs a signature σ. The procedure for computing the signature is described in Algorithm 1. The tuple $(\boldsymbol{z}, s)$ is returned as signature σ.

Ver. This module takes message m, the signature σ and the public key $\mathcal{PK}$ as input and outputs accept or reject. The steps are described below:

- Parse the signature as $(\boldsymbol{z}, s)$
- Compute $\boldsymbol{\tau} = \mathcal{H}(\mathsf{m}||s)$
- Accept the signature if $\mathcal{P}(\boldsymbol{z}) = \boldsymbol{\tau}$ holds; otherwise reject.

Correctness. The signature scheme is correct. If a message m, public key $\mathcal{P}$ and a signature $(\boldsymbol{z}, s)$, where $\boldsymbol{z}$ is obtained according to Algorithm 1 are given, then, we need to verify that $\mathcal{P}(\boldsymbol{z}) = \mathcal{H}(\mathsf{m}||s)$ holds. Let $\mathfrak{g}$ be any MQ-polynomial in the public key system $\mathcal{P}$. The following can be easily verified for each such $\mathfrak{g}$:

$$\begin{aligned} \mathfrak{g}(\boldsymbol{z}) &= \mathfrak{g}(\boldsymbol{v} + \boldsymbol{o}) \\ &= \mathfrak{g}(\boldsymbol{v}) + \mathfrak{g}(\boldsymbol{o}) + \mathfrak{g}'(\boldsymbol{v}, \boldsymbol{o}) \\ &= \mathfrak{g}(\boldsymbol{v}) + \mathfrak{g}(\boldsymbol{o}) + \mathfrak{g}'(\boldsymbol{v}, \overline{M}\boldsymbol{u}) \\ &= \mathfrak{g}(\boldsymbol{v}) + \mathfrak{g}(\boldsymbol{o}) + \boldsymbol{v}^\top M'_{\mathfrak{g}} \overline{M} \boldsymbol{u} \end{aligned} \tag{6}$$

Algorithm 1. Signing Module for Salted UOV

Require: The message m, secret key $\mathcal{SK}$ and the description of the salt space
Ensure: A signature $(\boldsymbol{z}, s)$ on m

1: Sample a vector $\boldsymbol{v} \xleftarrow{\$} \mathbb{F}^n$
2: Compute $\boldsymbol{c}_{\mathfrak{g}} = \boldsymbol{v}^\top M'_{\mathfrak{g}} \overline{M}$ ▷ $M'_{\mathfrak{g}}\overline{M}$ can be precomputed
3: Construct the $m \times m$ matrix C with $\boldsymbol{c}_{\mathfrak{g}}$ as rows
4: **repeat**
5: Sample $s \xleftarrow{\$} \{0,1\}^{\ell_s}$ ▷ a salt s is sampled
6: Compute $\boldsymbol{\tau} = \mathcal{H}(\mathsf{m}||s)$
7: Compute $\boldsymbol{\tau}' = \boldsymbol{\tau} - \mathcal{P}(\boldsymbol{v})$
8: **until** $\{\boldsymbol{y} \in \mathbb{F}^m : \ C \cdot \boldsymbol{y} = \boldsymbol{\tau}'\} \neq \emptyset$
9: Sample $\boldsymbol{u} \xleftarrow{\$} \{\boldsymbol{y} \in \mathbb{F}^m : \ C \cdot \boldsymbol{y} = \boldsymbol{\tau}'\}$
10: Compute $\boldsymbol{o} = \overline{M}\boldsymbol{u}$ ▷ column-span corresponding to $\boldsymbol{u}$
11: Compute $\boldsymbol{z} = \boldsymbol{v} + \boldsymbol{o}$
12: Output $\sigma = (\boldsymbol{z}, s)$

where $\mathfrak{g}'$ is the polar form of $\mathfrak{g}$. Since $\boldsymbol{o} \in \mathsf{O}$, the second term on the RHS of Eq. (6) is zero and the third term is equal to the $\mathfrak{g}$-th coordinate of the vector $\boldsymbol{\tau}' = \boldsymbol{\tau} - \mathcal{P}(\boldsymbol{v})$. Thus, combining the above observation for every such $\mathfrak{g}$, we obtain $\mathcal{P}(\boldsymbol{z}) = \mathcal{H}(\mathsf{m}||s)$. This proves that $(\boldsymbol{z}, s)$ is a valid signature on m.

Efficiency Comparison. One can easily check that both the versions, based on traditional approach [SSH11] and subspace approach (presented above) entertain more or less the same signing and verification time. However, the key sizes are improved in the subspace approach as only the basis information for the secret hidden subspace is required to store. In fact, the number of field elements required to store for both the approaches are presented in Table 1.

Table 1. Public and secret key sizes for UOV signature

Approach	Public key (# of field elements)	Secret key (# of field elements)
Traditional	$mn(n+1)/2$	$m(v(v+1)/2 + vm) + n^2$
Subspace	$mn(n+1)/2$	mn

We now discuss the distribution of the output signatures. Note that the output signature has two components $\boldsymbol{z}$ and the salt s. In the following, we first establish (in Proposition 1) that the statistical distance between the salt part of the output signature and the uniform distribution over $\{0,1\}^{\ell_s}$ is bounded by $1/q$. Then, we show (in Corollary 1) that the distribution of the signature deviates from the uniform distribution over $\mathbb{F}^n \times \{0,1\}^{\ell_s}$ by at most $1/q$. Let us define a good set and a bad set as follows:

$$\mathsf{Good} = \{\boldsymbol{v} \in \mathbb{F}^n : \mathcal{P}'(\boldsymbol{v}, .) : \mathsf{O} \to \mathbb{F}^m \text{ is non-singular}\}$$
$$\mathsf{Bad} = \{\boldsymbol{v} \in \mathbb{F}^n : \mathcal{P}'(\boldsymbol{v}, .) : \mathsf{O} \to \mathbb{F}^m \text{ is singular}\}.$$

Following Eq. (5), we have $|\mathsf{Good}| \approx q^n(1-1/q)$ and $|\mathsf{Bad}| \approx q^n \cdot \frac{1}{q}$, where $q = |\mathbb{F}|$. Sometimes, we refer to an element of Good (resp. Bad) as good (resp. bad) element. Let χ denote the random variable corresponding to the salt part of the output signature. Note that the distribution of χ depends on that of random variables $\boldsymbol{v}$ and s (involved in steps 1 and 5 respectively). Let U denote the uniform distribution over $\{0,1\}^{\ell_s}$. Then, the following proposition gives a bound on their statistical distance.

Proposition 1. *The statistical distance between χ and U is bounded by $1/q$.*

Proof. First observe that for any $\boldsymbol{a} \in \{0,1\}^{\ell_s}$, we have $\mathsf{Pr}[\chi = \boldsymbol{a} \mid \boldsymbol{v} \in \mathsf{Good}] = 1/2^{\ell_s}$, where the probability is taken over the random choice of $\boldsymbol{v} \in \mathbb{F}^n$ and $s \in \{0,1\}^{\ell_s}$. Then, calculate the following probability for any $\boldsymbol{a} \in \{0,1\}^{\ell_s}$.

$$\begin{aligned}\mathsf{Pr}[\chi = \boldsymbol{a}] &= \sum_{S \in \{\mathsf{Good},\mathsf{Bad}\}} \mathsf{Pr}[\chi = \boldsymbol{a} \mid \boldsymbol{v} \in S] \cdot \mathsf{Pr}[\boldsymbol{v} \in S] \\ &\approx \frac{1}{2^{\ell_s}} \cdot \left(1 - \frac{1}{q}\right) + p_{\boldsymbol{a}} \cdot \frac{1}{q} \end{aligned} \tag{7}$$

where $p_{\boldsymbol{a}} = \mathsf{Pr}[\chi = \boldsymbol{a} \mid \boldsymbol{v} \in \mathsf{Bad}]$. Then, the statistical distance between χ and U is given by

$$\begin{aligned}\Delta(\chi, U) &= \frac{1}{2} \cdot \sum_{\boldsymbol{a} \in \{0,1\}^{\ell_s}} |\mathsf{Pr}[\chi = \boldsymbol{a}] - \mathsf{Pr}[U = \boldsymbol{a}]| \\ &\approx \frac{1}{2} \cdot \sum_{\boldsymbol{a} \in \{0,1\}^{\ell_s}} \left| \frac{1}{2^{\ell_s}} \cdot \left(1 - \frac{1}{q}\right) + p_{\boldsymbol{a}} \cdot \frac{1}{q} - \frac{1}{2^{\ell_s}} \right| \qquad \text{[using Eq. (7)]} \\ &\le \frac{1}{2} \cdot \sum_{\boldsymbol{a} \in \{0,1\}^{\ell_s}} \left(\frac{1}{2^{\ell_s}} \cdot \frac{1}{q} + p_{\boldsymbol{a}} \cdot \frac{1}{q} \right) \\ &= \frac{1}{2 \cdot q} \cdot \left(1 + \sum_{\boldsymbol{a} \in \{0,1\}^{\ell_s}} p_{\boldsymbol{a}} \right) \\ &= \frac{1}{2 \cdot q} \cdot (1+1) = \frac{1}{q}. \end{aligned}$$

This completes the proof.

Corollary 1. *The distribution of the output signature deviates from the uniform distribution over Σ by at most $1/q$.*

Proof. Since $\boldsymbol{v}$ is chosen uniformly at random from $\mathbb{F}^n$, the $\boldsymbol{z}$-part of the signature is uniform over $\mathbb{F}^n$. Hence, the corollary follows from Proposition 1.

4.3 Uniformity of MQ-Systems

We now analyze the distribution of $\mathcal{P}(\boldsymbol{x})$, when $\boldsymbol{x} \in \mathbb{F}^n$ is chosen uniformly at random. In particular, we quantify the gap between this distribution and the

uniform distribution. This is essentially required for giving a concrete security reduction of the salted homogeneous UOV signature. Recall that $\mathcal{P}$ is a random MQ-system which vanishes on a random subspace O. We show (see Corollary 2) that the statistical distance between the distribution of $\mathcal{P}(\boldsymbol{x})$ and the uniform distribution over $\mathbb{F}^m$ is at most $1/q$, where $q = |\mathbb{F}|$. Since $|\mathbb{F}^n/\mathsf{O}| = q^{n-m}$, $\mathbb{F}^n$ can be written as a union of q^{n-m} disjoint cosets of O in $\mathbb{F}^n$, i.e.,

$$\mathbb{F}^n = \bigcup_{i=1}^{q^{n-m}} \mathsf{Coset}_i$$

where $\mathsf{Coset}_i = \boldsymbol{v}_i + \mathsf{O}$, $\boldsymbol{v}_i$ is called a coset representative and $\mathsf{Coset}_j \cap \mathsf{Coset}_k = \emptyset$ for distinct $j, k \in [q^{n-m}]$. We now study the behavior (basically, bijectivity) of $\mathcal{P}$ on each coset Coset_i which is independent of the choice of the representative.

Proposition 2. *When $\boldsymbol{v}_i \in \mathsf{Good}$, then the restricted map $\mathcal{P} : \mathsf{Coset}_i \to \mathbb{F}^m$ is bijective.*

Proof. It suffices to show $\mathcal{P} : \mathsf{Coset}_i \to \mathbb{F}^m$ is injective. Let $\boldsymbol{x}'_1, \boldsymbol{x}'_2 \in \mathsf{O}$ be two arbitrary distinct elements. Since $\mathcal{P}'(\boldsymbol{v}_i, .) : \mathsf{O} \to \mathbb{F}^m$ is injective, $\mathcal{P}'(\boldsymbol{v}_i, \boldsymbol{x}'_1) \neq \mathcal{P}'(\boldsymbol{v}_i, \boldsymbol{x}'_2)$, that means $\mathcal{P}(\boldsymbol{v}_i + \boldsymbol{x}'_1) \neq \mathcal{P}(\boldsymbol{v}_i + \boldsymbol{x}'_2)$.

When $\mathcal{P}$ is bijective on a coset, then we would refer to this coset as 'good coset', otherwise 'bad coset'. Note that given a coset, any element of it can be a representative. So, if a coset contains at least one good element, then $\mathcal{P}$ will be bijective on that coset. We now ask the following question. What is the probability that a randomly picked coset is good? To answer the question let us take a look at the worst case situation, although the likelihood of this is very low: Out of the total q^{n-m} cosets, roughly $\frac{1}{q} \cdot q^{n-m}$ many cosets contain only the bad elements. Therefore, if we pick up any coset randomly, then it will be good with probability roughly $(1 - 1/q)$ in the worst case. Let GSet be the union of all good cosets and BSet be the union of all bad cosets (i.e., $\mathsf{BSet} = \mathbb{F}^n \setminus \mathsf{GSet}$). So, $\mathsf{Pr}[\boldsymbol{x} \in \mathsf{GSet}] \approx 1 - 1/q$ and $\mathsf{Pr}[\boldsymbol{x} \in \mathsf{BSet}] \approx 1/q$, where the probability is taken over the uniform choice of $\boldsymbol{x} \in \mathbb{F}^n$. Note that the statistical distance between the distribution of $\mathcal{P}(\boldsymbol{x})$ and the uniform distribution over $\mathbb{F}^m$ will be maximum in the worst case situation mentioned above. The following corollary quantifies the gap of the two distributions.

Corollary 2. *Let $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ be a homogeneous UOV public map. When $\boldsymbol{x} \xleftarrow{\$} \mathbb{F}^n$, let χ denote the distribution of $\mathcal{P}(\boldsymbol{x})$ over $\mathbb{F}^m$. Let U be the uniform distribution over $\mathbb{F}^m$. Then $\Delta(\chi, U) \leq \frac{1}{q}$.*

Proof. When $\boldsymbol{x} \xleftarrow{\$} \mathsf{GSet}$, then $\boldsymbol{x}$ belongs to a random good coset; let us call it Coset. Then, $\boldsymbol{x}$ will be uniform over Coset. So, $\mathcal{P}(\boldsymbol{x})$ will be uniform over $\mathbb{F}^m$ thanks to Proposition 2. That is, for any $\boldsymbol{a} \in \mathbb{F}^m$, we have $\mathsf{Pr}[\chi = \boldsymbol{a} \mid \boldsymbol{x} \in \mathsf{GSet}] = 1/q^m$, where the probability is taken over the random choice of $\boldsymbol{x} \in \mathbb{F}^n$. Therefore, for any $\boldsymbol{a} \in \mathbb{F}^m$, we have

$$\begin{aligned}\mathsf{Pr}\left[\chi = \boldsymbol{a}\right] &= \sum_{S \in \{\mathsf{GSet}, \mathsf{BSet}\}} \mathsf{Pr}\left[\chi = \boldsymbol{a} \mid \boldsymbol{x} \in S\right] \cdot \mathsf{Pr}\left[\boldsymbol{x} \in S\right] \\ &\approx \frac{1}{q^m} \cdot \left(1 - \frac{1}{q}\right) + p_{\boldsymbol{a}} \cdot \frac{1}{q} \end{aligned} \tag{8}$$

where $p_{\boldsymbol{a}} = \mathsf{Pr}\left[\chi = \boldsymbol{a} \mid \boldsymbol{x} \in \mathsf{BSet}\right]$. Then, the statistical distance between χ and U is given by

$$\begin{aligned}\Delta(\chi, U) &= \frac{1}{2} \cdot \sum_{\boldsymbol{a} \in \mathbb{F}^m} \left|\mathsf{Pr}\left[X = \boldsymbol{a}\right] - \mathsf{Pr}\left[U = \boldsymbol{a}\right]\right| \\ &\approx \frac{1}{2} \cdot \sum_{\boldsymbol{a} \in \mathbb{F}^m} \left|\frac{1}{q^m} \cdot \left(1 - \frac{1}{q}\right) + p_{\boldsymbol{a}} \cdot \frac{1}{q} - \frac{1}{q^m}\right| \qquad \text{[using Eq. (8)]} \\ &\leq \frac{1}{2} \cdot \sum_{\boldsymbol{a} \in \mathbb{F}^m} \left(\frac{1}{q^m} \cdot \frac{1}{q} + p_{\boldsymbol{a}} \cdot \frac{1}{q}\right) \\ &= \frac{1}{2 \cdot q} \cdot \left(1 + \sum_{\boldsymbol{a} \in \mathbb{F}^m} p_{\boldsymbol{a}}\right) \\ &= \frac{1}{2 \cdot q} \cdot (1 + 1) = \frac{1}{q}. \end{aligned}$$

This completes the proof.

4.4 Security of Salted Homogeneous UOV Signature in CROM

In this section, we argue the security of SHUOV signature (presented in Sect. 4.2) in the classical random oracle model. Following the proof-style of [SSH11] and Corollaries 1 and 2, a security reduction can be easily shown from the UOVI-problem. For the shake of completeness, we give a proof-sketch in the CROM thereby resolving the issues raised in Sect. 3.1. Similarly, a security reduction for the traditional salted homogeneous UOV of [SSH11] can be derived.

Theorem 1. *If the UOVI-problem is intractable and q is superpolynomial in the security parameter κ, then the SHUOV-Signature is EUF-CMA secure in the CROM.*

Proof-sketch in CROM. The proof uses a hybrid argument over the following games.

0. Game_0. This is exactly the original EUF-CMA security game, where the hash function $\mathcal{H} : \mathcal{M} \times \mathsf{SaltSpac} \to \mathbb{F}^m$ is treated as random oracle. Note that the non-salt part of the output signature is distributed uniformly over $\mathbb{F}^n$. Let q_{uov} and q_{sign} be the number of hash queries and the number of sign-queries respectively. Let δ be the advantage of an adversary $\mathcal{A}_0$ in Game_0, i.e., $\mathsf{Adv}_{\mathcal{A}_0}^{\text{EUF-CMA}}(\kappa) = \delta$.

1. Game_1. This is same as Game_0, except[4] the salts involved in the answers of sign queries are chosen uniformly at random. That is, the output signature in Game_1 is distributed uniformly over Σ. Then, by Corollary 1, the advantage of an adversary $\mathcal{A}_1$ in Game_1 is given by $\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_1}(\kappa) \geq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_0}(\kappa) - q_{\mathsf{sign}} \cdot \frac{1}{q} = \delta - q_{\mathsf{sign}} \cdot \frac{1}{q}$.
2. Game_2. This is same as Game_1, except the q_{sign}-many random oracle queries are answered by $\mathcal{P}(\boldsymbol{x})$, where $\boldsymbol{x} \xleftarrow{\$} \mathbb{F}^m$. Then, by Corollary 2, the advantage of an adversary $\mathcal{A}_2$ in Game_2 is given by $\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_2}(\kappa) \geq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_1}(\kappa) - q_{\mathsf{sign}} \cdot \frac{1}{q} \geq \delta - 2 \cdot q_{\mathsf{sign}} \cdot \frac{1}{q}$.

We now show that using $\mathcal{A}_2$ in Game_2, we can break the UOVI-problem. An instance $(\mathcal{P}, \boldsymbol{y}^*) \in \mathcal{P}_{\mathsf{uov}}(\mathbb{F}^n, \mathbb{F}^m) \times \mathbb{F}^m$ of the UOVI-problem is given to a simulator S and the goal of S is to find $\boldsymbol{x}^* \in \mathbb{F}^n$ such that $\mathcal{P}(\boldsymbol{x}^*) = \boldsymbol{y}^*$. The simulator maintains a list $\mathsf{List}_{\mathsf{uov}}$ for keeping records of the form: $(\mathsf{m}, s, \mathcal{H}(\mathsf{m}||s))$. The adversary $\mathcal{A}_2$ may ask queries to hash oracle and sign-oracle in any order. The simulator S picks $i^* \xleftarrow{\$} [q_{\mathsf{uov}}]$ as a guess for the forgery message.

- **Hash-oracle**. When $\mathcal{A}_2$ asks the i-th $\mathcal{H}$-query on $\mathsf{m}_i||s_i$, it returns $\mathcal{H}(\mathsf{m}_i||s_i)$ if $(\mathsf{m}_i, s_i, \cdot) \in \mathsf{List}_{\mathsf{uov}}$. Otherwise, if $i = i^*$, then S updates $\mathsf{List}_{\mathsf{uov}}$ with the entry $(\mathsf{m}_i, s_i, \boldsymbol{y}^*)$ and returns $\boldsymbol{y}^*$, else it picks $\boldsymbol{y}_i \xleftarrow{\$} \mathbb{F}^m$, updates $\mathsf{List}_{\mathsf{uov}}$ with $(\mathsf{m}_i, s_i, \boldsymbol{y}_i)$ and returns $\boldsymbol{y}_i$.
- **Sign-oracle**. On the i-th query on message m_i, S picks $(\boldsymbol{x}_i, s_i) \xleftarrow{\$} \Sigma$. If $(\mathsf{m}_i, s_i, \cdot) \in \mathsf{List}_{\mathsf{uov}}$, it aborts, otherwise updates $\mathsf{List}_{\mathsf{uov}}$ with $(\mathsf{m}_i, s_i, \mathcal{P}(\boldsymbol{x}_i))$[5] and returns $\sigma_i = (\boldsymbol{x}_i, s_i)$.
- **Forgery**. When $\mathcal{A}_2$ produces a message-signature pair $(\mathsf{m}^*, \sigma^* = (\boldsymbol{x}^*, s^*))$, S submits $\boldsymbol{x}^*$ as a solution of the given instance of the UOVI-problem.

Note that all the queries of $\mathcal{A}_2$ are answered according to the description in Game_2. With probability $1/q_{\mathsf{uov}}$, S correctly guesses the message $\mathsf{m}^* = \mathsf{m}_{i^*}$, and $\boldsymbol{x}^*$ is a correct solution of the given problem instance if $(\mathsf{m}^*, \sigma^* = (\boldsymbol{x}^*, s^*))$ is a valid pair. So, the advantage of breaking the UOVI-problem is given by

$$\begin{aligned} \mathsf{Adv}^{\text{UOVI}}_{\mathsf{S}}(\kappa) &\geq \frac{1}{q_{\mathsf{uov}}} \cdot \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_2}(\kappa) \\ &= \frac{1}{q_{\mathsf{uov}}} \cdot \left(\delta - 2 \cdot q_{\mathsf{sign}} \cdot \frac{1}{q}\right) \\ &\approx \frac{1}{q_{\mathsf{uov}}} \cdot \delta \qquad [\text{as } q \text{ is superpolynomial in } \kappa\,] \end{aligned} \tag{9}$$

This ends the proof-sketch. □

[4] As mentioned earlier in Sect. 3.1, there is a gap between the distribution of salts involved in the construction and the security reduction of [SSH11]. That gap essentially depends on the size of the underlying field. But the authors implicitly assumed that a computational adversary cannot distinguish the difference. Unlike [SSH11], our security treatment takes into account this difference.

[5] Note that $\mathcal{H}(\mathsf{m}_i||s_i)$ is programmed by the value $\mathcal{P}(\boldsymbol{x}_i)$, instead of uniformly random value of $\mathbb{F}^m$ and this change is already captured in Game_2.

Remark 4. As mentioned earlier, we are able to resolve the issues in the security argument of [SSH11] (raised in Sect. 3) for the case of homogeneous salted UOV signature. While we utilize the subspace description of the scheme, one can easily check that the same strategy works for the case of conventional description (thanks to the identical distribution of keys in both the approaches). However, for general (not necessarily homogeneous) salted UOV signature, it is not known whether the corresponding key can be expressed through the subspace structure. Hence, one cannot directly apply Proposition 2 in this case.

Remark 5. Note that the above reduction makes sense, if q (that is, the size of the underlying field) involved in Eq. (9) is a superpolynomial in the security parameter. This q appears in Eq. (9) due to the bounds involved in Corollaries 1 and 2. Improving these bounds is an interesting future research problem as they have a direct bearing on the size of the underlying field.

5 Security of Salted Homogeneous UOV in QROM

In this section, we prove the security of SHUOV-signature in the quantum random oracle model. We start by recalling some notations and important results required for the security reduction. For two sets $\mathcal{X}$ and $\mathcal{Y}$, the notation $\mathcal{Y}^{\mathcal{X}}$ denotes the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$. For a distribution D on $\mathcal{Y}$, the notation $g \xleftarrow{} D^{\mathcal{X}}$ denotes sampling a function $g : \mathcal{X} \to \mathcal{Y}$ as follows: for $x \in \mathcal{X}$, $g(x)$ is sampled according to the distribution D. For a given function $f : \mathcal{X} \to \mathcal{Y}$, we can always handle on-the-fly simulation of the function by the following unitary (see [NC00]):

$$\begin{aligned} \mathcal{O}_f : \mathcal{X} \times \mathcal{Y} &\to \mathcal{X} \times \mathcal{Y} \\ |x, y\rangle &\mapsto |x, y \oplus f(x)\rangle \end{aligned} \tag{10}$$

So, for handling superposition queries to the random oracle $\mathcal{H}$, it suffices to give a function description of the oracle. Here, we will use the fact [Zha12b] that the advantage of a quantum algorithm in distinguishing a randomly chosen $2k$-wise independent function from a truly random function is 0, where the number of quantum queries is at most k. This means a quantum-accessible random oracle can be implemented by choosing a random $2k$-wise independent function.

We show a reduction in the QROM based on small-range distributions [Zha12a]. Here, we first give the definition and related results of small-range distribution.

Definition 5 (Small-range distributions [Zha12a]). *Given an integer $r \in \mathbb{N}$, two sets $\mathcal{X}$ and $\mathcal{Y}$, and a distribution D on $\mathcal{Y}$, a small-range distribution, denoted by $\mathsf{SR}_r^D(\mathcal{X})$, is defined to be the following distribution on $\mathcal{Y}^{\mathcal{X}}$:*

1. *For each $i \in [r]$, choose a random value y_i from $\mathcal{Y}$ according to the distribution D, i.e., sample a function, say, $g : [r] \to \mathcal{Y}$ according to $D^{[r]}$.*
2. *For each $x \in \mathcal{X}$, pick $i \xleftarrow{\$} [r]$ and set $\mathcal{O}(x) = y_i$.*

This distribution can be alternatively viewed as follows: choose $g \longleftarrow D^{[r]}$ and $f \xleftarrow{\$} [r]^{\mathcal{X}}$ and return the composition $\mathcal{O} = g \circ f$. Now, we state a result which is very important for arguing security of public-key schemes in the QROM. It essentially says that the difference between the output distributions of a quantum algorithm making k quantum queries to an oracle sampled either according to $\mathsf{SR}_r^D(\mathcal{X})$ or randomly from $\mathcal{Y}^{\mathcal{X}}$ is at most $27k^3/r$. The result is stated below.

Lemma 1 ([Zha12a, Corollary 7.5]). *Suppose a quantum algorithm asks k many quantum queries to an oracle either drawn from $\mathsf{SR}_r^D(\mathcal{X})$ or drawn randomly from $\mathcal{Y}^{\mathcal{X}}$. Then, the output distributions of the algorithm are $\ell(k)/r$-close, where $\ell(k) = \pi^2(2k)^3/3 < 27k^3$.*

Next, we describe another important result that can be used for programming random oracles. In particular, the result is useful in a situation, where oracle values were supposed to be assigned uniformly, but are assigned by sampling according to a distribution which is ϵ (negligible) distance apart from uniform distribution. Then, any quantum algorithm making k many queries to one of them can distinguishing them with probability at most $\mathcal{O}(k^{3/2}) \cdot \epsilon^{1/2}$. The result is stated below.

Lemma 2 ([BZ13, Lemma 2.5]). *Let $\mathcal{X}$ and $\mathcal{Y}$ be two sets. Suppose for each $x \in \mathcal{X}$, there are two distributions D_x and D'_x on $\mathcal{Y}$ with $|D_x - D'_x| \leq \epsilon$. Let two functions $\mathcal{O} : \mathcal{X} \to \mathcal{Y}$ and $\mathcal{O}' : \mathcal{X} \to \mathcal{Y}$ be defined as follows: for each $x \in \mathcal{X}$, $\mathcal{O}(x)$ and $\mathcal{O}'(x)$ are set by sampling from $\mathcal{Y}$ according to the distributions D_x and D'_x respectively. Then, any quantum algorithm making at most k quantum queries to $\mathcal{O}$ or $\mathcal{O}'$ can not distinguishing them, except with probability at most $\sqrt{8C_0k^3\epsilon}$, where $C_0 = 27$ (a universal constant).*

Let $\widetilde{\mathcal{M}} = \mathcal{M} \times \mathsf{SaltSpac}$. The sets $\mathcal{X}$ and $\mathcal{Y}$ that appear in the above lemma are considered to be $\widetilde{\mathcal{M}}$ and $\mathbb{F}^m$ in our context respectively. Further, we consider for all $x \in \mathcal{X}$, the distributions D_x (resp. D'_x) are to be the same and let us call it D (resp. D'). Now, we set D to be the uniform distribution over $\mathbb{F}^m$ and define the distribution D' over $\mathbb{F}^m$ as follows: Pick $\boldsymbol{x} \xleftarrow{\$} \mathbb{F}^n$ and output $\mathcal{P}(\boldsymbol{x})$, where $\mathcal{P} : \mathbb{F}^n \to \mathbb{F}^m$ is a random public key of SHUOV-scheme. Note that the statistical distance between D and D' is at most ϵ (thanks to Corollary 2), where $\epsilon = 1/q$ and $q = |\mathbb{F}|$. In the reduction, we use the following corollary.

Corollary 3. *Let $\widetilde{\mathcal{O}} : \widetilde{\mathcal{M}} \to \mathbb{F}^n$ and $\mathcal{O} : \widetilde{\mathcal{M}} \to \mathbb{F}^m$ be two quantum-accessible random oracles. Let $\mathcal{O}' : \widetilde{\mathcal{M}} \to \mathbb{F}^m$ be a quantum-accessible oracle defined as follows: for $\mathsf{m}||s \in \widetilde{\mathcal{M}}$, $\mathcal{O}'(\mathsf{m}||s) = \mathcal{P}(\widetilde{\mathcal{O}}(\mathsf{m}||s))$. Then, any quantum algorithm making at most k queries to $\mathcal{O}$ or $\mathcal{O}'$ can not distinguishing them, except with probability at most $\sqrt{8C_0k^3\epsilon}$.*

Proof. Let D be the uniform distribution over $\mathbb{F}^m$. Then, we define $D_{\mathsf{m}||s} = D$ for all $\mathsf{m}||s \in \widetilde{\mathcal{M}}$ and the computation of $\mathcal{O} : \widetilde{\mathcal{M}} \to \mathbb{F}^m$ can be thought of via sampling from $\mathbb{F}^m$ according to distribution D. The distribution D' picks $\boldsymbol{x} \xleftarrow{\$} \mathbb{F}^n$ and returns $\mathcal{P}(\boldsymbol{x})$. For each $\mathsf{m}||s \in \widetilde{\mathcal{M}}$, the distribution $D'_{\mathsf{m}||s}$ samples $\mathcal{P}(\boldsymbol{x})$,

where $\boldsymbol{x} = \widetilde{\mathcal{O}}(\mathsf{m}||s)$ is uniform over $\mathbb{F}^n$. Basically, D' and $D'_{\mathsf{m}||s}$ have identical distribution. The remainder of the proof immediately follows from Lemma 2 and Corollary 2.

Theorem 2. *If the UOVI-problem is intractable and q is exponential in the security parameter κ, then the SHUOV-Signature is EUF-CMA secure in the QROM.*

Proof. We adopt the proof strategy [Zha15] of signature from trapdoor permutations. The proof essentially follows from a hybrid argument over the following games.

0. Game_0. This is exactly the original EUF-CMA security game, where the hash function $\mathcal{H} : \widetilde{\mathcal{M}} \to \mathbb{F}^m$ is treated as random oracle. Note that the non-salt part of the output signature is distributed uniformly over $\mathbb{F}^n$. Let q_{uov} and q_{sign} be the number of hash queries and the number of sign-queries respectively. Let $q_{\mathsf{tot}} = q_{\mathsf{uov}} + q_{\mathsf{sign}} + 1$. Let δ be the advantage of an adversary $\mathcal{A}_0$ in Game_0, i.e., $\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_0}(\kappa) = \delta$.
1. Game_1. This is same as Game_0, except the random oracle is programmed as follows: Pick a quantum-accessible random oracle $\widetilde{\mathcal{O}} : \widetilde{\mathcal{M}} \to \mathbb{F}^n$. Then, for each $\mathsf{m}||s \in \widetilde{\mathcal{M}}$, define $\mathcal{H}(\mathsf{m}||s) = \mathcal{P}(\widetilde{\mathcal{O}}(\mathsf{m}||s))$. By Corollary 3, the advantage of an adversary $\mathcal{A}_1$ in Game_1 is
$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_1}(\kappa) \geq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_0}(\kappa) - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon} = \delta - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon}.$$
2. Game_2. This is same as Game_1, except the function $\widetilde{\mathcal{O}} : \widetilde{\mathcal{M}} \to \mathbb{F}^n$ is sampled according to the small-range distribution $\mathsf{SR}^D_r(\widetilde{\mathcal{M}})$, where D is the uniform distribution over $\mathbb{F}^n$, $r = \lceil 2 \cdot \ell(q_{\mathsf{tot}})/\delta \rceil$ and $\ell(q_{\mathsf{tot}}) = \pi^2 \cdot (2q_{\mathsf{tot}})^3/3 < 27 \cdot q_{\mathsf{tot}}^3$. Note that as mentioned earlier, $\widetilde{\mathcal{O}}$ can be viewed as $\widetilde{\mathcal{O}} = g \circ f$, where $g : [r] \to \mathbb{F}^n$ is described by the elements $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_r \xleftarrow{\$} \mathbb{F}^n$ and $f \xleftarrow{\$} [r]^{\widetilde{\mathcal{M}}}$, i.e., for $\mathsf{m}||s \in \widetilde{\mathcal{M}}$, $\widetilde{\mathcal{O}}(\mathsf{m}||s) = \boldsymbol{x}_i$, where $f(\mathsf{m}||s) = i$. Then, by Lemma 1, the advantage of an adversary $\mathcal{A}_2$ in Game_2 is
$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_2}(\kappa) \geq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_1}(\kappa) - \ell(q_{\mathsf{tot}})/r \geq \delta/2 - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon}.$$
3. Game_3. This is same as Game_2, except the salt computation in sign-oracle which is handled as follows: Let $\mathcal{O}_{\mathsf{salt}} : \widetilde{\mathcal{M}} \times [q_{\mathsf{sign}}] \to \mathsf{SaltSpac}$ be a classical[6] random oracle. A counter ctr (initially, set to 0) is maintained to keep track the index[7] of the current message queried to the sign-oracle. For a query message m, $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$ and the salt value for m is computed as $\mathcal{O}_{\mathsf{salt}}(\mathsf{m}||\mathsf{ctr})$. That is, the output signature in Game_3 is distributed uniformly over Σ. By Corollary 1, the advantage of an adversary $\mathcal{A}_3$ in Game_3 is
$$\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_3}(\kappa) \geq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_2}(\kappa) - q_{\mathsf{sign}} \cdot \epsilon \geq \delta/2 - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon} - q_{\mathsf{sign}} \cdot \epsilon.$$

[6] Since the salt generation in the security game is involved only in answering sign-oracle (classically), it is sufficient to have a salt generation random oracle $\mathcal{O}_{\mathsf{salt}}$ which is classical.

[7] The whole purpose of this counter is to generate different salts even for the same message queried multiple times to the sign-oracle.

4. Game_4. This is same as Game_3, except the following:
   (a) At the beginning of the game, pick $i^* \xleftarrow{\$} [r]$. (This is the guess where the forged message-salt appears to the oracle f, i.e., $f(\mathsf{m}^*||s^*) = i^*$.)
   (b) Abort, if $f(\mathsf{m}^*||s^*) \neq i^*$ or if for any sign-query on m, $f(\mathsf{m}||s) = i^*$, where s is computed as described in Game_3.

   The probability of not abort is

$$\Pr[\neg\mathsf{abort}] = \frac{1}{r} \cdot \left(1 - \frac{1}{r}\right)^{q_{\mathsf{sign}}} \geq \frac{1}{r} - \frac{q_{\mathsf{sign}}}{r^2} \geq \frac{1}{2 \cdot r} \text{ (as } r \geq 2 \cdot q_{\mathsf{sign}}).$$

   Then, the advantage of an adversary $\mathcal{A}_4$ in Game_4 is

$$\begin{aligned}\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_4}(\kappa) &\geq \frac{1}{2 \cdot r} \cdot \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_4}(\kappa) \\ &\geq \frac{1}{2 \cdot r} \cdot \left(\frac{\delta}{2} - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon} - q_{\mathsf{sign}} \cdot \epsilon\right).\end{aligned}$$

5. Game_5. This is same as Game_4, except the following change in answering hash queries: Pick $\boldsymbol{y} \xleftarrow{\$} \mathbb{F}^m$ and set $\mathcal{H}(\mathsf{m}||s) = \boldsymbol{y}$ (instead of defining $\mathcal{H}(\mathsf{m}||s) = \mathcal{P}(\boldsymbol{x}_{i^*})$) for all $\mathsf{m}||s \in \widetilde{\mathcal{M}}$ such that $f(\mathsf{m}||s) = i^*$. Then, the advantage of an adversary $\mathcal{A}_5$ in Game_5 (using Corollary 2) is

$$\begin{aligned}\mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_5}(\kappa) &\geq \mathsf{Adv}^{\text{EUF-CMA}}_{\mathcal{A}_4}(\kappa) - \epsilon \\ &\geq \frac{1}{2 \cdot r} \left(\frac{\delta}{2} - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon} - q_{\mathsf{sign}} \cdot \epsilon\right) - \epsilon.\end{aligned}$$

Now, we create a solver for the UOVI-problem using the adversary $\mathcal{A}_5$ (in Game_5). An instance $(\mathcal{P}, \boldsymbol{y}^*) \in \mathcal{P}_{\mathsf{uov}}(\mathbb{F}^n, \mathbb{F}^m) \times \mathbb{F}^m$ of the UOVI-problem is given to a simulator S and the goal of S is to find $\boldsymbol{x}^* \in \mathbb{F}^n$ such that $\mathcal{P}(\boldsymbol{x}^*) = \boldsymbol{y}^*$. The simulator will use $\mathcal{A}_5$ in the environment of Game_5 for breaking the problem instance. S picks $i^* \xleftarrow{\$} [r]$ and answers the following queries that may appear in any order:

- **Hash-oracle**. For answering quantum queries to hash-oracle, it suffices to describe only the classical description of the oracle function (without using any history) thanks to the on-the-fly simulation due to the unitary given in Eq. (10). For an input $\mathsf{m}||s \in \widetilde{\mathcal{M}}$, the function is defined as follows:

$$\mathcal{H}(\mathsf{m}||s) = \begin{cases} \boldsymbol{y}^* & \text{if } i = i^* \\ \mathcal{P}(\boldsymbol{x}_i) & \text{otherwise} \end{cases}$$

  where $f(\mathsf{m}||s) = i$. As mentioned earlier the quantum random oracle $f : \widetilde{\mathcal{M}} \to [r]$ can be implemented using random $2 \cdot q_{\mathsf{tot}}$-wise independent function.
- **Sign-oracle**. For a sign-query on m, S sets $\mathsf{ctr} \leftarrow \mathsf{ctr} + 1$ and computes $s = \mathcal{O}_{\mathsf{salt}}(\mathsf{m}||\mathsf{ctr})$ and $i = f(\mathsf{m}||s)$. If $i = i^*$, it aborts, otherwise returns the signature $\sigma = (\boldsymbol{x}, s)$, where $\boldsymbol{x} = \widetilde{\mathcal{O}}(\mathsf{m}||s)$.

– **Forgery**. When $\mathcal{A}$ produces a message-signature pair $(\mathsf{m}^*, \sigma^* = (\boldsymbol{x}^*, s^*))$, S checks whether $f(\mathsf{m}^*||s^*) = i^*$. If not, S aborts, otherwise it submits $\boldsymbol{x}^*$ as a solution of the given problem instance.

Note that when $(\mathsf{m}^*, \sigma^* = (\boldsymbol{x}^*, s^*))$ is a valid forgery, we have $\mathcal{P}(\boldsymbol{x}^*) = \mathcal{H}(\mathsf{m}^*||s^*) = \boldsymbol{y}^*$ and hence, $\boldsymbol{x}^*$ is a valid solution to the instance of the UOVI-problem. Therefore, we have

$$\begin{aligned}
\mathsf{Adv}_{\mathsf{S}}^{\text{UOVI}}(\kappa) &= \mathsf{Adv}_{\mathcal{A}_5}^{\text{EUF-CMA}}(\kappa) \\
&\geq \frac{1}{2 \cdot r} \left(\frac{\delta}{2} - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon} - q_{\mathsf{sign}} \cdot \epsilon \right) - \epsilon \\
&\geq \frac{\delta}{4 \cdot 27 \cdot q_{\mathsf{tot}}^3} \left(\frac{\delta}{2} - \sqrt{8 \cdot C_0 \cdot q_{\mathsf{tot}}^3 \cdot \epsilon} - q_{\mathsf{sign}} \cdot \epsilon \right) - \epsilon \\
&= \frac{\delta^2}{216 \cdot q_{\mathsf{tot}}^3} - \epsilon \cdot \left(1 + \frac{\delta \cdot q_{\mathsf{sign}}}{108 \cdot q_{\mathsf{tot}}^3} \right) - \sqrt{\epsilon} \cdot \sqrt{\frac{C_0}{54}} \cdot \frac{\delta}{\sqrt{q_{\mathsf{tot}}^3}} \\
&\approx \frac{\delta}{216 \cdot q_{\mathsf{tot}}^3} \qquad (11)
\end{aligned}$$

where the 2nd and the 3rd terms involved in Eq. (11) are ignored as $\epsilon = 1/q$ is negligible in κ. When δ is non-negligible in κ, then $\mathsf{Adv}_{\mathsf{S}}^{\text{UOVI}}(\kappa)$ is non-negligible – a contradiction.

6 Concluding Remark

In this paper, we have identified some issues related to the security reduction of the salted UOV signature in the CROM [SSH11] and then addressed these issues through the subspace description [Beu21] of the scheme. This alternative construction of salted UOV improves the signing key size a bit. We also have provided a security reduction of the same scheme in the QROM. Our security treatment is applicable only to the homogeneous salted UOV signature. A clean security reduction for general salted UOV signature remains an interesting research problem.

Acknowledgement. We would like to thank the anonymous reviewers of Indocrypt 2022 for their comments and suggestions that helped us in polishing the technical and editorial content of this paper. This work is supported by the Ministry of Electronics and Information Technology, Government of India through its grants for the Center of Excellence in Quantum Technology at IISc Bangalore, India.

A Signature Using Trapdoor Information

A.1 Algorithm for Solving the Public Key System Using Trapdoor Information

In this section, we give the method for solving the public key system using the trapdoor information as an algorithm. The procedure was described in Sect. 4.1.

Algorithm 2. Inverting Public Key System Using Trapdoor

Require: The matrices $M'_{\mathfrak{g}}$ for each public key polynomial $\mathfrak{g}$, the hidden subspace O and an image point $\boldsymbol{\tau} \in \mathbb{F}^m$, where $\mathcal{P}(\mathsf{O}) = \{\mathbf{0}\}$ and O is described as column-space of an $n \times m$ matrix $\overline{M}$.

Ensure: A solution $\boldsymbol{z} \in \mathbb{F}^n$ such that $\mathcal{P}(\boldsymbol{z}) = \boldsymbol{\tau}$.

repeat
  Sample a vector $\boldsymbol{v} \xleftarrow{\$} \mathbb{F}^n$
  Compute $\boldsymbol{\tau}' = \boldsymbol{\tau} - \mathcal{P}(\boldsymbol{v})$
  Compute $\boldsymbol{c}_{\mathfrak{g}} = \boldsymbol{v}^\top M'_{\mathfrak{g}} \overline{M}$
  Construct $m \times m$ matrix C with $\boldsymbol{c}_{\mathfrak{g}}$ as rows
until $\{\boldsymbol{y} \in \mathbb{F}^m : C \cdot \boldsymbol{y} = \boldsymbol{\tau}'\} \neq \emptyset$
Sample $\boldsymbol{u} \xleftarrow{\$} \{\boldsymbol{y} \in \mathbb{F}^m : C \cdot \boldsymbol{y} = \boldsymbol{\tau}'\}$
Compute $\boldsymbol{o} = \overline{M}\boldsymbol{u}$ ▷ column-span corresponding to $\boldsymbol{u}$
Compute $\boldsymbol{z} = \boldsymbol{v} + \boldsymbol{o}$
Output $\sigma = \boldsymbol{z}$

A.2 Signature Scheme

Let us write down the complete signature scheme based on this trapdoor.

KeyGen. This takes the security parameter 1^κ as input and outputs the public and secret keys. The secret key is a description of the subspace $\mathsf{O} \subset \mathbb{F}^n$ and the public key is the system $\mathcal{P}$ consisting of m MQ-polynomials in n variables which vanish at O. Note that O can be represented by an $n \times m$ matrix as described in Sect. 4.1. Thus $\mathcal{SK} = \mathsf{O}$ and $\mathcal{PK} = \mathcal{P}$. A hash function $\mathcal{H} : \mathcal{M} \to \mathbb{F}^m$ for converting message into a fixed-length digest is known publicly.

Sign. This takes message m and the secret key $\mathcal{SK}$ as input and outputs a signature σ. The signature σ is obtained by solving $\mathcal{P}(\cdot) = \mathcal{H}(\mathsf{m})$ using Algorithm 2.

Ver. This takes the message m, the signature σ and the public key $\mathcal{PK}$ as input and outputs accept or reject. If $\mathcal{P}(\sigma) = \mathcal{H}(\mathsf{m})$, holds, the signature is accepted. Otherwise, the signature is rejected.

B Signature of Sakumoto et al.

We reproduce the salted version of UOV signature given in [SSH11, Section 4.1]. The secret key is a UOV type MQ system $\mathcal{F}$ of m polynomials in n variables. The authors consider non-homogeneous polynomials. Then, as usual, an affine invertible transformation $\mathcal{T}$ is used for mixing the variables. The public key is obtained in the obvious way as $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$. The scheme uses a salt of length ℓ_s, which is a polynomial in the security parameter κ. The public and the secret keys contain a description of the salt space.

The verification follows the obvious procedure. We describe the signing algorithm in Algorithm 3. The variable list is parsed as $(\boldsymbol{x}_v, \boldsymbol{x}_m)$, where $\boldsymbol{x}_v$ denotes the vector of vinegar variables and $\boldsymbol{x}_m$ that of oil variables. There are v vinegar

variables and m oil variables. The notation $\mathcal{F}(\boldsymbol{x}'_v, \boldsymbol{x}_m)$ is used to denote the linear system in oil variables which is obtained after the vinegar variables have been specialized to the vector $\boldsymbol{x}'_v$.

Algorithm 3. Signing Algorithm of Sakumoto, Shirai and Hiwatari

Require: $\mathcal{F}$, $\mathcal{T}$ and the message m
Ensure: A signature σ on m such that $\mathcal{P}(\sigma) = \mathcal{H}(\mathsf{m}||s)$

1: Sample $\boldsymbol{x}'_v \xleftarrow{\$} \mathbb{F}^v$ ▷ uniform assignment for vinegar variables
2: **repeat**
3: Sample salt $s \xleftarrow{\$} \{0,1\}^{\ell_s}$ ▷ sampling random salt
4: Compute $\boldsymbol{y} = \mathcal{H}(\mathsf{m}||s)$
5: **until** $\{\boldsymbol{x}_m \in \mathbb{F}^m : \mathcal{F}(x'_v, x_m) = y\} \neq \emptyset$
6: Sample $\boldsymbol{x}'_m \xleftarrow{\$} \{\boldsymbol{x}_m \in \mathbb{F}^m : \mathcal{F}(\boldsymbol{x}'_v, \boldsymbol{x}_m) = \boldsymbol{y}\}$
7: Compute $\boldsymbol{x} = \mathcal{T}^{-1}(\boldsymbol{x}'_v, \boldsymbol{x}'_m)$ ▷ applying $\mathcal{T}^{-1}$ on a length n vector
8: Output $(\boldsymbol{x}, s)$ as signature

References

[Beu21] Beullens, W.: Improved cryptanalysis of UOV and rainbow. In: Canteaut, A., Standaert, F.-X. (eds.) EUROCRYPT 2021. LNCS, vol. 12696, pp. 348–373. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-77870-5_13

[Beu22] Beullens, W.: Breaking rainbow takes a weekend on a laptop. In: Dodis, Y., Shrimpton, T. (eds.) CRYPTO 2022. LNCS, vol. 13508, pp. 464–479. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15979-4_16

[BR93] Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: 1st ACM conference on Computer and communications security, pp. 62–73. SIAM (1993)

[BZ13] Boneh, D., Zhandry, M.: Secure signatures and chosen ciphertext security in a quantum computing world. In: Canetti, R., Garay, J.A. (eds.) CRYPTO 2013. LNCS, vol. 8043, pp. 361–379. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40084-1_21

[CDP21] Chatterjee, S., Dimri, A., Pandit, T.: Identity-based signature and extended forking algorithm in the multivariate quadratic setting. In: Adhikari, A., Küsters, R., Preneel, B. (eds.) INDOCRYPT 2021. LNCS, vol. 13143, pp. 387–412. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-92518-5_18

[CHR+16] Chen, M.-S., Hülsing, A., Rijneveld, J., Samardjiska, S., Schwabe, P.: From 5-pass $\mathcal{MQ}$-based identification to $\mathcal{MQ}$-based signatures. In: Cheon, J.H., Takagi, T. (eds.) ASIACRYPT 2016. LNCS, vol. 10032, pp. 135–165. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53890-6_5

[CLND19] Chen, J., Ling, J., Ning, J., Ding, J.: Identity-based signature schemes for multivariate public key cryptosystems. Comput. J. **62**(8), 1132–1147 (2019)

[DS05] Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005). https://doi.org/10.1007/11496137_12

[KPG99] Kipnis, A., Patarin, J., Goubin, L.: Unbalanced oil and vinegar signature schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_15

[KS98] Kipnis, A., Shamir, A.: Cryptanalysis of the oil and vinegar signature scheme. In: Krawczyk, H. (ed.) CRYPTO 1998. LNCS, vol. 1462, pp. 257–266. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055733

[MP17] Mohamed, M.S.E., Petzoldt, A.: RingRainbow – an efficient multivariate ring signature scheme. In: Joye, M., Nitaj, A. (eds.) AFRICACRYPT 2017. LNCS, vol. 10239, pp. 3–20. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57339-7_1

[NC00] Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press, New York (2000)

[Pat97] Patarin, J.: The oil and vinegar algorithm for signatures. In: Dagstuhl Workshop on Cryptography (1997)

[PCY+15] Petzoldt, A., Chen, M.-S., Yang, B.-Y., Tao, C., Ding, J.: Design principles for HFEv- based multivariate signature schemes. In: Iwata, T., Cheon, J.H. (eds.) ASIACRYPT 2015. LNCS, vol. 9452, pp. 311–334. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48797-6_14

[PSM17] Petzoldt, A., Szepieniec, A., Mohamed, M.S.E.: A practical multivariate blind signature scheme. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 437–454. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70972-7_25

[SSH11] Sakumoto, K., Shirai, T., Hiwatari, H.: On provable security of UOV and HFE signature schemes against chosen-message attack. In: Yang, B.-Y. (ed.) PQCrypto 2011. LNCS, vol. 7071, pp. 68–82. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25405-5_5

[Zha12a] Zhandry, M.: How to construct quantum random functions. In: FOCS, pp. 679–687. IEEE Computer Society (2012)

[Zha12b] Zhandry, M.: Secure identity-based encryption in the quantum random oracle model. In: Safavi-Naini, R., Canetti, R. (eds.) CRYPTO 2012. LNCS, vol. 7417, pp. 758–775. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32009-5_44

[Zha15] Zhandry, M.: Cryptography in the age of quantum computers. Ph.D. thesis, Stanford University (2015)