MQ Aggregate Signature Schemes with Exact Security Based on UOV Signature

Jiahui Chen[1], Jie Ling[1(✉)], Jianting Ning[2], Zhiniang Peng[3], and Yang Tan[4]

[1] School of Computer, Guangdong University of Technology, Guangzhou, China
csjhchen@gmail.com, jling@gdut.edu.cn
[2] School of Computing, National University of Singapore, Singapore, Singapore
jtning88@gmail.com
[3] Qihoo 360, Beijing, China
jiushigujiu@gmail.com
[4] Shenzhen Ringle.AI Technology Co. Ltd., Shenzhen, China
a7853z@qq.com

Abstract. Multivariate public key cryptography which relies on multivariate quadratic (MQ) problem is one of the main approaches to guarantee the security of communication in the post-quantum world. In this paper, we focus mainly on the yet unbroken (under proper parameter choice) Unbalanced Oil and Vinegar (UOV) scheme, and discuss the exact security of it. Then we propose a combined signature scheme which that (1) not only can reduce the public key size of the UOV signature scheme, and (2) but also can provide tighter security against chosen-message attack in the random oracle. On the other hand, we propose a novel aggregate signature scheme based on UOV signature scheme. Additionally, we give security proof for our aggregate signature scheme under the security of our proposed signature scheme.

Keywords: Multivariate cryptography · UOV signature scheme · Exact security · Aggregate signature

1 Introduction

Nowadays, the current major public-key cryptographic schemes are mainly based on the hardness of number theory such as integer factorization and discrete logarithm. Since according to the Shor's algorithm [1], these schemes will be broken in polynomial time after the emergence of quantum computers, which calls for doing research on the post-quantum cryptography [2].

According to the post quantum cryptography project submitted by the National Institute for Standards and Technology (NIST) [3], MPKC is popular for its efficiency in the post quantum cryptography aspect and signature schemes are promising. Also, multivariate signature schemes with special properties, such as proxy signature, ring signature and so on, are proposed. For example, Tang

Z. Liu and M. Yung (Eds.): Inscrypt 2019, LNCS 12020, pp. 443–451, 2020.
https://doi.org/10.1007/978-3-030-42921-8_26

et al. [4] proposed the first MPKC proxy signature scheme based on the problem of Isomorphisms of Polynomials (IP). Petzoldt et al. [5] proposed the first provable MPKC threshold ring signature scheme based on the result of [6]. Chen et al. [7] proposed the first online/offline signature based on UOV by utilizing the linear construction of the central map of UOV, so that the proposed scheme can be distributed in the wireless sensor networks. In addition, multivariate blind signature scheme by Petzoldt et al. [8] are proposed to enrich this area.

In this paper, we focus on this part, we firstly propose a combined signature scheme based on UOV signature, which can not only reduce the public key size of the UOV signature scheme but also can provide more tighter exact security proof. Thereafter, we propose a novel aggregate signature scheme based on the proposed signature scheme, which includes the stages of key generation, generation of signature, combination of signature and the verification of aggregate signature. We also give a strict security proof for our aggregate signature scheme under the security of our proposed signature scheme. We also give a toy example for our aggregate scheme. Finally, we propose parameters and comparisons for our proposed scheme.

The rest of the paper is organized as follows: In Sect. 2, we describe the schemes, the basic UOV signature scheme and the security models. In Sect. 3, we present our proposed signature scheme. Then the proposed aggregate signature scheme based on our signature scheme is described in Sect. 4. In Sect. 5, we present the analysis of our schemes. Finally we concludes the paper with a discussion in Sect. 6.

2 Preliminaries

2.1 Sequential Aggregate Signatures

Generally, a sequential aggregate signature scheme [10] $\mathcal{AS}$ is consisted with three algorithms :KeyGen, Sign, Verify:

- $\mathsf{AggGen}(1^\lambda)$: The algorithm inputs a security parameter 1^λ and outputs a signature key pair (sk, pk). In a sequential aggregation signature scheme, this algorithm is run by each user u_i and the corresponding key pair (sk_i, pk_i) is obtained.
- $\mathsf{AggSign}(m_i, sk_i, pk_1, ..., pk_{i-1}, \Sigma_{i-1})$: To generate a sequential aggregate signature, the algorithm is run by the user in this sequence using its secret key sk_i according to the message m_i. Then given the previous user's public key set $(pk_1, \ldots, pk_{i-1})$ and the previous aggregate signature Σ_{i-1}, the algorithm aggregates to generate an aggregate signature of Σ.
- $\mathsf{AggVerify}((m_1, ..., m_k), \Sigma, (pk_1, \ldots, pk_k))$: Given $((m_1, \ldots, m_k), \Sigma, (pk_1, \ldots, pk_k))$, if Σ is valid, the algorithm outputs TRUE, otherwise it outputs FALSE.

2.2 UOV Signature Scheme

The UOV scheme [11] is a single field construction, it work solely in the polynomial ring $\mathbb{F}_q[X]$, where $X = \{x_1, ..., x_n\}$. Let $|V| = v$, $|O| = o$ and $v + o = n$. We randomly choose o quadratic polynomials $q_k(X) = q_k(x_1, ..., x_n)$ the polynomial ring $\mathbb{F}_q[X]$ by

$$q_k(X) = \sum_{i \in O, j \in V} \alpha_{ij}^{(k)} x_i x_j + \sum_{i,j \in V, i \leq j} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in O \cup V} \gamma_i^{(k)} x_i + \eta^{(k)}, k = 1, ..., o, \tag{1}$$

To hide the structure of Q in the public key one concatenates it with an invertible affine map $T : F^n \to F^n$, then the public key of the UOV signature scheme is $P = Q \circ T$.

The key generation algorithm $\mathsf{UOVGen}(1^\lambda)$, takes 1^λ as input and outputs pk=P and sk= (T, Q).

Assume the document needs to be signed is $M = (y_1, y_2, ..., y_m)$, the signing algorithm $\mathsf{UOVSign}(M, T, Q))$ is as follows. Firstly, the user chooses the values of the v vinegar variables V=$(x_1, \ldots, x_v)$ at random, then it solves the equation: $M = Q(X, V)$, then it calculates $\sigma = T^{-1}(X, V)$ and get the signature σ.

Finally, the verification algorithm $\mathsf{UOVVerify}(\sigma, M, P)$ returns TRUE if $P(\sigma) = M$, otherwise returns FALSE.

2.3 Security Models

Exact Security Model for UOV Digital Signature. We quantify the security of UOV scheme as a uniform one-way function.

Definition 1. *We say that the UOV one-way function is $(t'(\lambda), \varepsilon'(\lambda)) - secure$ if there is no inverting algorithm that takes P, y as inputs and outputs a preimage x such that $P(x) = y$ at $t'(\lambda)$ processing time with probability at least $\varepsilon'(\lambda)$, where P is obtained by running $\mathsf{KeyGen}(1^\lambda)$ and y is randomly chosen from k^n. The standard asymptotic definition of security requests that the success probability of any PPT (probabilistic, polynomial time) algorithm is a negligible function of λ.*

Next, we quantify the exact security of UOV signature scheme. The exact security of the reduction which was used to prove the security of the full domain hash (FDH) signature scheme was first provides by Bellare and Rogaway [9] and analyzed in Theorem 1.

Similar to this work, we have

Definition 2. *We say that the UOV-based FDH signature scheme is $(t(\lambda), q_{sig}(\lambda), q_{hash}(\lambda), \varepsilon(\lambda)) - secure$ if there is no forger $\mathcal{A}$ who takes a public key pk generated via $(pk, \cdot) \leftarrow Gen(1^\lambda)$, after at most $q_{hash}(\lambda)$ queries to the random oracle, $q_{sig}(\lambda)$ signature queries, and $t(\lambda)$ processing time, then outputs a valid signature with probability at least $\varepsilon(\lambda)$.*

Security Model for Aggregated Signature. Similar to the work in [13], we formalize the sequential aggregation security under the selected message model in Definition 3.

Definition 3. *We say that a sequential aggregate signature scheme is* $(\varepsilon'', t'', q'_{sig}, q'_{hash})-$*secure if there is no forger* $\mathcal{A}$ *can win in the above game and satisfies that:* $\mathcal{A}$ *runs in time at most* t''*;* $\mathcal{A}$ *makes at most* q'_{hash} *queries to the hash function and at most* q'_{sig} *queries to the aggregate signing oracle;* $\mathrm{AdvAggSig_A}$ *is at least* ε''*.*

3 Our Proposed Signature Scheme

Our proposed signature scheme is consisted with three algorithms: Gen, Sig and Ver. The details are as follows.

The key generation algorithm Gen is described in Algorithm 1.

Algorithm 1. $\mathsf{Gen}(q, o, v, D)$

Input:
q: the underlying field (i.e. $\mathbb{F}_q = GF(2^5)$);
o,v: the number of Oil and Vinegar variables respectively;
n: $n = v + o$;
D: the number of non-trivial quadratic terms, $D = \frac{v\cdot(v+1)}{2} + o \cdot v$;
Output:
(T, Q): the private key to sign the message;
P: the public key corresponding to (T, Q);
1: Choose a vector $\mathbf{b} = (b_0, ..., b_{D-1})$ at random.
2: Choose an $n \times n$ invertible matrix T at random (given as a matrix $M_T = (t_{rs})_{r,s=1}^{n}$);
3: Set the entries of the first D columns of P to $p_{ij} = b_{(j-i) \bmod D}$;
4: Solve for $i = 0, ..., o-1$ and $j = 0, ..., D-1$ the linear systems given by $M' = Q \cdot A$ to get the non-zero coefficients of the quadratic terms of the central map Q, where the elements in A is

$$a_{kl}^{rs} = \begin{cases} t_{kr} \cdot t_{lr} (r = s) \\ t_{kr} \cdot t_{ls} + t_{ks} \cdot t_{lr} (r \neq s) \end{cases}$$

and the elements in M' is p_{ij};
5: Choose the coefficients of the linear and constant terms of the central map Q at random;
6: Compute the remaining coefficients of the public polynomials by composing Q and T using the equation $P = Q \circ T$;
7: **return** (Q, T, P);

Gen takes as inputs the underlying field, the number of Oil and Vinegar variables, and the number of non-zero quadratic terms, and returns the public/private key pairs. In fact in this part we use the strategy in [14]. We recommend to read more detail in [14].

In the signature generation part, we change the UOV signature into FDH-like signature scheme, so that we can make exact security proof. It firstly chooses a collision-resistant hash function $H : \{0,1\}^* \rightarrow \mathbb{F}_q^o$, and is described in Algorithm 2.

At last, the verification algorithm $\mathsf{Ver}(H,\sigma,m)$ returns 1 if $P(x) = H(m||r)$, otherwise it returns 0.

Algorithm 2. Sign(m, (T, Q))

Input:
m: the message to sign;
(T, Q): the private key to sign the message;
Output:
σ: the signature on message m;
1: $x_v{}' \in_R \mathbb{F}_q^v$;
2: **repeat**
3: $l \in_R \mathbb{Z}$; $r \in_R \{0,1\}^l$; $y \leftarrow H(m||r)$;
4: **until** $\{z_n | Q(z_n, x_v{}') = y\} \neq \emptyset$;
5: $x_n{}' \in_R \{z_n | Q(z_n, x_v{}') = y\}$;
6: $x \leftarrow T^{-1}(x_n{}', x_v{}')$;
7: **return** $\sigma = (x, r)$;

4 Our Proposed Sequential Aggregation Signature Scheme

In this section, we propose a novel sequential aggregation signature scheme based on our combined signature scheme. The main parts of this scheme are the key generation part, the aggregation signature generation part and the aggregation signature verification part.

4.1 Key Generation

Let $U = u_1, ..., u_k$ be the set of users. In the key generation algorithm of UOV-based sequential aggregation signature scheme, we assume each user u_i can generate a UOV key pair $((Q_i, T_i), P_i)$ through a given system parameter. Each user u_i then makes the public key P_i public and keeps the private key (Q_i, T_i). Also, we denote a split algorithm $\mathsf{Split}(*)$ that splits arbitrary message with length $O + iV, i = 1, ..., k$ into $k + 1$ small messages, where the length of the first message is equal to the size of O, the length of the other parts is equal to the size of V.

4.2 Signature Generation

Let $\mathcal{H} : \mathbb{F}^* \rightarrow \mathbb{F}^o$ be a hash function that can hash any message into a message with length o. Suppose each user u_i has a message m_i to be signed. To generate a message to aggregate signature Σ according to the message $m_1, ..., m_k$, each user of the user set $u_1, ..., u_k$ runs Algorithm 3 separately. The resulting aggregate signature Σ is the output of user u_k.

4.3 Signature Verification

To verify the correctness of an aggregate signature Σ_i, we split Σ_i into two parts, the first part being the block Σ_{i_1}, which is a vector of o elements. The second part is $(\Sigma_{i_2}, ..., \Sigma_{1_2})$, which is a collection of vectors consisting of v elements. Thereafter, the signature verification part is similar to the verification of the normal UOV. The process is shown in Algorithm 4.

Algorithm 3. AggSign($m_i, sk_i = (T_i, Q_i), pk_1, ..., pk_{i-1}, \Sigma_{i-1}$)

Input:
m_i: the messages that need to be signed;
$sk_i = (T_i, Q_i)$: the private key of user i;
$pk_1, ..., pk_{i-1}, \Sigma_{i-1}$: The previous public keys and aggregate signature
Output:
Σ_i: the aggregate signature corresponding to $m_1, ..., m_i$;

if $i = 1$ then
  $\Sigma_{i-1_1} = 0^o$, $\Sigma_{i-1_2} = \emptyset$
else if AggVerify$(pk_1, ..., pk_{i-1}, m_1, ..., m_{i-1}, \Sigma_{i-1})$=TURE then
  $(\Sigma_{i-1_1}, \Sigma_{i-1_2}, ..., \Sigma_{1_2}) =$ Split(Σ_{i-1})
else
  return FALSE
end if
$D = H(m_1, ..., m_i)$
$\Sigma_i =$ UOVSign$((D + \Sigma_{i-1_1}), (T_i, Q_i))$
$(\Sigma'_{i_1}, \Sigma'_{i_2}) =$ Split(Σ_i)
$\Sigma_{i_1} = \Sigma'_{i_1}$
$\Sigma_{i_2} = (\Sigma'_{i_2}||\Sigma_{i-1_2}||...||\Sigma_{1_2})$
$\Sigma_i = (\Sigma_{i_1}||\Sigma_{i_2})$
return Σ_i

Algorithm 4. AggVerify($pk_1, ..., pk_i, m_,..., m_i, \Sigma_i$)

Input:
$pk_1, ..., pk_i, m_,..., m_i, \Sigma_i$: The public key, message, and aggregate signature corresponding to the previous i user respectively
Output:
TRUE or FALSE: Determines whether the aggregate signature is valid;

$(\Sigma_{i_1}, \Sigma_{i_2}, ..., \Sigma_{1_2})$=Split$(\Sigma_i)$
for $j = i$ to 1 do
  $D_j = H(m_1, ..., m_j)$
  $\Sigma_{j-1_1} = pk_j(\Sigma_{j_1}, \Sigma_{j_2}) - D_j$
end for
if $\Sigma_{0_1} = 0^o$
return TRUE
end if
else
return FALSE
end if

4.4 A Toy Example

We propose a toy example to further illustrate our scheme. Let $k = 3$, $q = 4, o = 2, v = 4$.

When $i = 1$ (the first sequence), the scheme will generate the first aggregate signature Σ_1. The scheme sets $\Sigma_{0_1} = 0^o$, $\Sigma_{0_2} = \emptyset$, assume $D = H(m_1) = \{1, 3\}$, we have $D + \Sigma_{0_1}$=$\{1,3\}$, then it will use this $D + \Sigma_{0_1}$ to submit a regular UOV signature $\sigma_1 =$ UOVSign$((D + \Sigma_{0_1}), sk_1) = \{3, 1, 3, 0, 0, 1\}$. Thereafter, the scheme sets $\Sigma_{1_1} = \{3, 1\}, \Sigma_{1_2} = \{3, 0, 0, 1\}$ and the first aggregate signature $\Sigma_1 = (\Sigma_{1_1}, \Sigma_{1_2}) = (\{3, 1\}||\{3, 0, 0, 1\})$, the scheme will go to the second aggregate signature process.

When $i = 2$ (the second sequence), the scheme will firstly call for a verifying algorithm of the aggregate scheme to verify the first aggregation signature, where it splits and gets $\Sigma_{1_1} = \{3, 1\}$ and $\Sigma_{1_2} = \{3, 0, 0, 1\}$, and computes $pk_1(\Sigma_{1_1}, \Sigma_{1_2}) = \{1, 3\}$, $D_1 = H(m_1) = \{1, 3\}$, we have

$\Sigma_{0_1} = pk_1(\Sigma_{1_1}, \Sigma_{1_2}) - D_1 = \{0,0\}$, thus the verification is valid. Then the scheme will generate the second aggregate signature Σ_2. It first splits and gets $\Sigma_{1_1} = \{3,1\}, \Sigma_{1_2} = \{3,0,0,1\}$, assume $D_2 = H(m_1, m_2) = \{0,2\}$, we have $D_2 + \Sigma_{1_1}$=$\{3,3\}$, then we will use this $D_2 + \Sigma_{1_1}$ to submit a basic UOV signature $\sigma_2 = \mathsf{UOVSign}((D + \Sigma_{i_1}), sk_2) = \{0,2,0,3,1,3\}$ and $(\Sigma'_{2_1}, \Sigma'_{2_2}) = (\{0,2\}, \{0,3,1,3\})$. Then the scheme set $\Sigma_{2_1} = \Sigma'_{2_1} = \{0,2\}$ and $\Sigma_{2_2} = (\Sigma'_{2_2}||\Sigma_{1_2}) = \{0,3,1,3\}||\{3,0,0,1\}$. Then the second aggregate signature is $\Sigma_2 = (\Sigma_{2_1}||\Sigma_{2_2}) = (\{0,2\}||\{0,3,1,3\}||\{3,0,0,1\})$, the scheme will go to the third aggregation signature process.

When $i = 3$ (the third sequence), the scheme will firstly call for a verifying algorithm of the aggregate scheme and find the verification valid. Then the scheme will generate the third aggregate signature Σ_3. It first splits and gets $\Sigma_{2_1} = \{0,2\}$, $\Sigma_{2_2} = \{0,3,1,3\}$, $\Sigma_{1_2} = \{3,0,0,1\}$, assume $D = H(m_1, m_2, m_3) = \{3,2\}$, we have $D + \Sigma_{2-1_1} = \{3,0\}$, then we will use this $D + \Sigma_{2_1}$ to submit a signature $\sigma_3 = \mathsf{UOVSign}((D + \Sigma_{2_1}), sk_3) = \{2,1,3,3,1,1\}$. Thereafter, the scheme sets $\Sigma_{3_1} = \{2,1\}, \Sigma_{3_2} = \{3,3,1,1\}||\{0,3,1,3\}||\{3,0,0,1\}$, then the aggregate signature process is finished and the final aggregate signature is $\Sigma_3 = (\Sigma_{3_1}, \Sigma_{3_2}) = (\{2,1\}||\{3,3,1,1\}||\{0,3,1,3\}||\{3,0,0,1\})$.

5 Analysis

Our signature scheme can reduce the size of public key of UOV, more details about this property we recommend to read [14].

Proposition 1. *The trapdoor function of our proposed scheme is as secure as the basic function of UOV under the current attack techniques.*

Due to page limitation, we omit the proof here.

Proposition 2. *If the function of our scheme is* $(t', \varepsilon') - secure$*, our signature scheme is* $(\varepsilon, t, q_{sig}, q_{hash}) - secure$*, where* $\varepsilon(\lambda) \leq \frac{1}{(1-\frac{1}{q_{sig}+1})^{q_{sig}+1}} \cdot q_{sig} \cdot \varepsilon'(\lambda)$ *and* $t(\lambda) \geq t'(\lambda) - (q_{hash} + q_{sig} + 1)(t_{UOV} + O(1))$ *, where* t_{UOV} *is the time to compute the UOV function.*

Due to page limitation, we omit the proof here.

Proposition 3. *If UOV signature scheme is* $(\varepsilon, t, q_{sig}, q_{hash}) - secure$*, then our aggregation signature scheme is* $(\varepsilon'', t'', q'_{sig}, q'_{hash}) - secure$*, where* $\varepsilon''(\lambda) \leq 2(q'_{sig} + q'_{hash} + 1) \cdot \varepsilon(\lambda)$ *and* $t'' \leq t - (4kq'_{hash} + 4kq'_{sig} + 7k + 1)$.

Due to page limitation, we omit the proof here.

Finally, for the compression ratio, it is not difficult to calculate the size of our aggregate signature scheme is $|\Sigma| = o + n \cdot v$. Thus the compression ratio is $\tau = 1 - \frac{\Sigma}{n \cdot \sigma} = 1 - \frac{o+n \cdot v}{n \cdot (o+v)}$.

6 Conclusions

In this paper, we propose a new signature scheme based on UOV signature, which is shown that our proposed signature scheme can reduce the public key size and have better exact security bound. In addition, we propose an aggregated signature scheme based on the UOV signature scheme and also give security proof under the security of our proposed signature scheme. Finally, we find that the aggregate signature compression rate obtained by our aggregated signature scheme be $1-\frac{o+n\cdot v}{n\cdot(o+v)}$, indicating that our aggregate signature scheme is especially suitable for large-scale signature environments.

Acknowledgment. This work is supported by the Key Areas Research and Development Program of Guangdong Province (grant 2019B010139002), National Natural Science Foundation of China (grant 61902079) and the project of Guangzhou Science and Technology (grant 201902020006 & 201902020007).

References

1. Shor, P.W.: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. **26**, 1484–1509 (1997)
2. Bernstein, D.J.: Introduction to post-quantum cryptography. In: Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.) Post-quantum cryptography - PQCrypto 2009, LNCS, pp. 1–14. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88702-7_1
3. NIST CSRC: Cryptographic technology group: submission requirements and evaluation criteria for the post-quantum cryptography standardization process (2016)
4. Tang, S., Xu, L.: Towards provably secure proxy signature scheme based on isomorphisms of polynomials. Future Gener. Comput. Syst. **30**, 91–97 (2014)
5. Petzoldt, A., Bulygin, S., Buchmann, J.: A multivariate based threshold ring signature scheme. Appl. Algebra Eng. Commun. Comput. **24**(3–4), 255–275 (2013)
6. 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
7. Chen, J., Tang, S., He, D., Tan, Y.: Online/offline signature based on UOV in wireless sensor networks. Wirel. Netw. **23**(6), 1719–1730 (2017)
8. 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
9. Bellare, M., Rogaway, P.: The exact security of digital signatures-how to sign with RSA and Rabin. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 399–416. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_34
10. Boneh, D., Gentry, C., Lynn, B., Shacham, H.: Aggregate and verifiably encrypted signatures from bilinear maps. In: Biham, E. (ed.) EUROCRYPT 2003. LNCS, vol. 2656, pp. 416–432. Springer, Heidelberg (2003). https://doi.org/10.1007/3-540-39200-9_26
11. 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

12. Coron, J.-S.: On the exact security of Full Domain Hash. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 229–235. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-44598-6_14
13. Lysyanskaya, A., Micali, S., Reyzin, L., Shacham, H.: Sequential aggregate signatures from trapdoor permutations. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 74–90. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24676-3_5
14. Petzoldt, A., Bulygin, S., Buchmann, J.: A multivariate signature scheme with a partially cyclic public key. In: Proceedings of SCC, pp. 229–235. Springer, Cham (2010)