MQ-Sign: A New Post-Quantum Signature Scheme based on Multivariate Quadratic Equations: Shorter and Faster

Kyung-Ah Shim, Jeongsu Kim, and Youngjoo An

Cryptographic Technology Research Team, NIMS

Contents

- Design Principles and Features
- Digital Signature Algorithm Specification
- Security Proof and Security Analysis
- Secure Parameter Selection
- Implementation
- Future Plan

Design Principles and Features

- Single-layered Structure
  - Rainbow: multiple-layered structure additionally requires the hardness of the MinRank problem and causes the recent advanced attacks.
  - missing oil*oil structure and Oil-Vinegar function
- The Shortest Signature Length and Shorter Key Sizes.
  - The shortest among post-quantum signature schemes based on the other hard problems: 134 bytes, 200 bytes and 260 bytes at security categories I, II, and V, respectively.
  - Shorter secret key size than UOV by using sparse polynomials.

  ⇒ NIST has made a supplementary call for digital signatures: short signature and fast verification(not based on structured lattices)
- Fast Performance
  - The MQ-scheme with a single layer requires relatively large size of the matrix in Gaussian elimination. In order to resolve this inefficiency, we use the block inversion method that exploits the inversions of half-sized matrices.

Design Principles and Features

- Easy to Implement
  - Very simple and easy to understand and implement requiring basic linear algebra: matrix-vector products and solving linear systems over small finite fields
- Protection Side-Channel Attacks.
  - For resistance against side-channel attacks, UOV is secure against the correlation power analysis (CPA) by just using random affine maps instead of the equivalent keys without requiring an additional countermeasure.
  - All key dependent operations in our scheme are performed in a time-constant manner.
- Additional Performance Improvements.
  - Off-line and on-line signing: our scheme with precomputation is 15x to 60x faster than the original version without precomputation at the three security categories.
  - Despite fast signing and verification performance, the key generation of our scheme is inefficient. To speed up key generation, we exploit multiple cores for independent operations resulting in 2x to 3x faster than the performance on a single core.

Basic Buildingblocks

- UOV: Missing Oil*Oil Structure

Let $V = \{1, \cdots, v\}$ and $O = \{v+1, \cdots, v+o\}$ be sets of integers such that $|V| = v$, $|O| = o$, and $n = v + o$. We first describe the structure of UOV (Unbalanced Oil and Vinegar) [17]. A central map $\mathcal{F} : \mathbb{F}_q^n \to \mathbb{F}_q^o$ of UOV, $\mathcal{F} = (\mathcal{F}^{(1)}, \cdots, \mathcal{F}^{(o)})$ is o multivariate quadratic equations with n variables $x_1, \cdots, x_n$ defined by

$$\mathcal{F}^{(k)}(\mathrm{x}) = \sum_{i \in O, j \in V} \alpha_{ij}^{(k)} x_i x_j + \sum_{i,j \in V, i \leq j} \beta_{ij}^{(k)} x_i x_j + \sum_{i \in V \cup O} \gamma_i^{(k)} x_i + \eta^{(k)}. \quad (1)$$

Each central quadratic polynomial $\mathcal{F}^{(k)}$ is written as

$$\mathcal{F}^{(k)} = \mathcal{F}_V^{(k)} + \mathcal{F}_{OV}^{(k)} + \mathcal{F}_{L,C}^{(k)},$$

Block Matrix Inversion Method

- Solving linear system
  - Block matric inversion (BMI) method
    - ✓ After computing A^{-1}, [D-CA^{-1}B]^{-1}, compute $R^{-1} \cdot \alpha$.

$$R = \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} I & O \\ CA^{-1} & I \end{pmatrix} \begin{pmatrix} A & O \\ 0 & D - CA^{-1}B \end{pmatrix} \begin{pmatrix} I & A^{-1}B \\ 0 & I \end{pmatrix}$$

$$R^{-1} = U^{-1} \cdot D_{Sc}^{-1} \cdot L^{-1}$$

$$R^{-1} \cdot \alpha = \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} \begin{pmatrix} A^{-1} & O \\ 0 & [D - CA^{-1}B]^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix} \alpha$$

  - The larger the size of a matrix being inverted, the greater the performance improvement and the higher the security level, the greater the effect of the optimizations. We use the BMI method with depth 1 to solve the linear system.

Digital Signature Algorithm Specification

- Key Generation
  - Secret key: (F, T), Public key: P=F ∘ T
  - Central map: $\mathcal{F}^{(k)}(\mathbf{x}) = \sum_{i\in O, j\in V} \alpha_{ij}^{(k)} x_i x_j + \sum_{i,j\in V, i\le j} \beta_{ij}^{(k)} x_i x_j + \sum_{i\in V\cup O} \gamma_i^{(k)} x_i + \eta^{(k)}$, $\mathcal{F}^{(k)} = \mathcal{F}_V^{(k)} + \mathcal{F}_{OV}^{(k)} + \mathcal{F}_{L,C}^{(k)}$

- **[Selection of $\mathcal{F}_V^{(k)}$ using Sparse Polynomials.]** For the Vinegar × Vinegar quadratic parts, $\mathcal{F}_V^{(k)}$ for $k = 1, \cdots, o$,

$$\mathcal{F}_V^{(k)} = \mathcal{F}_{V,S}^{(k)} = \sum_{i=1}^{v} \alpha_i^k x_i x_{(i+k-1(mod\ v))+1},$$

where $\alpha_i^k \in_R \mathbb{F}_q^*$ $(i = 1, \cdots, v)$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_V^{(k)}$ has full rank and all the quadratic terms in each $\mathcal{F}_V^{(k)}$ don't overlap for $k = 1, \cdots, o$.

Digital Signature Algorithm Specification

- Key Generation
  - Secret key: (F, T), Public key: P=F ∘ T
  - Central map: $\mathcal{F}^{(k)}(\mathbf{x}) = \sum_{i\in O, j\in V} \alpha_{ij}^{(k)} x_i x_j + \sum_{i,j\in V, i\le j} \beta_{ij}^{(k)} x_i x_j + \sum_{i\in V\cup O} \gamma_i^{(k)} x_i + \eta^{(k)}$, $\mathcal{F}^{(k)} = \mathcal{F}_V^{(k)} + \mathcal{F}_{OV}^{(k)} + \mathcal{F}_{L,C}^{(k)}$

    - **[Selection of $\mathcal{F}_{OV}^{(k)}$ using Sparse Polynomials.]** For the Vinegar × Oil quadratic parts,

$$\mathcal{F}_{OV}^{(k)} = \mathcal{F}_{OV,S}^{(k)} = \sum_{i=1}^{v} \beta_i^k x_i x_{(i+k-2(mod\ o))+v+1},$$

      where $\beta_i^k \in_R \mathbb{F}_q^*$ $(i = 1, \cdots, v)$ so that the symmetric matrix of the quadratic part of $\mathcal{F}_{OV}^{(k)}$ has full rank and all the quadratic terms in each $\mathcal{F}_{OV}^{(k)}$ don't overlap for $k = 1, \cdots, o$.

      - The sparse polynomial $\mathcal{F}_{OV}^{(k)} = \mathcal{F}_{OV,S}^{(k)} = \sum_{i=}^{o} \beta_i^k x_{v+i} x_{(i+k-1(mod\ o))+v+1}$ in the document posted on the KpqC homepage should be changed to $\mathcal{F}_{OV}^{(k)} = \mathcal{F}_{OV,S}^{(k)} = \sum_{i=1}^{v} \beta_i^k x_i x_{(i+k-2(mod\ o))+v+1}$

Digital Signature Algorithm Specification

- Key Generation
  - Four types of central map
    - Sparse Vinegar $*$ Vinegar + Sparse Vinegar $*$ Oil:

$$\mathcal{F}_{SS}^{(k)} = \mathcal{F}_{V,S}^{(k)} + \mathcal{F}_{OV,S}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

    - Random Vinegar $*$ Vinegar + Sparse Vinegar $*$ Oil:

$$\mathcal{F}_{RS}^{(k)} = \mathcal{F}_{V,R}^{(k)} + \mathcal{F}_{OV,S}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

    - Sparse Vinegar $*$ Vinegar + Random Vinegar $*$ Oil:

$$\mathcal{F}_{SR}^{(k)} = \mathcal{F}_{V,S}^{(k)} + \mathcal{F}_{OV,R}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

    - Random Vinegar $*$ Vinegar + Random Vinegar $*$ Oil:

$$\mathcal{F}_{RR}^{(k)} = \mathcal{F}_{V,R}^{(k)} + \mathcal{F}_{OV,R}^{(k)} + \mathcal{F}_{L,C}^{(k)}.$$

Digital Signature Algorithm Specification

- Signing algorithm

  - **Sign**(SK, λ, M)**.** Given a message M and a collision-resistant hash function $\mathcal{H} : \{0,1\}^* \rightarrow \mathbb{F}_q^o$, do the followings:
    - Choose a λ-bit random salt r, compute $\mathbf{h} = \mathcal{H}(M||r) \in \mathbb{F}_q^o$.
    - Compute $\mathbf{a} = \mathcal{F}^{-1}(\mathbf{h})$, i.e. $\mathcal{F}(\mathbf{a}) = \mathbf{h}$ as follows:
      * Select Vinegar values $s_V = (s_1, \cdots, s_v) \in \mathbb{F}_q^v$ at random and obtain a linear system of o equations with o unknowns $x_{v+1}, \cdots, x_{v+o}$ by substituting s_V into o central polynomials $\mathcal{F}^{(k)}$ for $1 \leq k \leq o$. After that, find a solution $(s_{v+1}, \cdots, s_{v+o})$ of the linear system using the BMI method.
      * If the linear systems is not solvable, choose another vector of Vinegar values s'_V and try again.
    - Compute $\mathbf{z} = \mathcal{T}^{-1}(\mathbf{a})$, where $\mathbf{a} = (s_1, \cdots, s_v, s_{v+1}, \cdots, s_{v+o})$ and output $\sigma = (\mathbf{z}, r)$ as a signature on M.

Digital Signature Algorithm Specification

- Verification algorithm

  - **Verify**(PK, M, σ)**.** Given a signature $\sigma = (\mathbf{z}, r)$ on a message M and the public key $\mathcal{P}$, check the equality $\mathcal{P}(\mathbf{z}) = \mathcal{H}(M||r)$. If the equality holds, output *valid*.

Security Proof

- Hard problems

  - **MQ-Problem:** Given a system $\mathcal{P} = (P^{(1)}, \cdots, P^{(m)})$ of m qudratic equations defined over $\mathbb{F}_q$ in variables $x_1, \cdots, x_n$ and $\mathbf{y} = (y_1, \cdots, y_m) \in \mathbb{F}_q^m$, find values $(x'_1, \cdots, x'_n) \in \mathbb{F}_q^n$ such that $P^{(1)}(x'_1, \cdots, x'_n) = y_1, \cdots, P^{(m)}(x'_1, \cdots, x'_n) = y_m$.
  - **EIP (Extended Isomorphism of Polynomials) Problem:** Given a nonlinear multivariate system $\mathcal{P}$ such that $\mathcal{P} = S \circ \mathcal{F} \circ T$ for linear or affine maps S and T, and $\mathcal{F}$ belonging to a special class of nonlinear polynomial system $\mathcal{C}$, find a decomposition of $\mathcal{P}$ such that $\mathcal{P} = S' \circ \mathcal{F}' \circ T'$ for linear or affine maps S' and T', and $\mathcal{F}' \in \mathcal{C}$.

Security Analysis

- Existential unforgeability
  - EUF-CMA of UOV: security proof for the Full-Domain-Hash scheme by modifying the signing algorithm to provide uniform distribution of the signatures: H(M) -> H(M, r) for a random salt. The existential unforgeability of our scheme follows the security proof of the modified UOV.
- Security analysis of known algebraic attacks
  - Direct attack: Grobner basis, F4, F5, hybrid F5, XL, Polynomial XL
  - Kipnis-Shamir attack, key recovery attack using good keys, Intersection attack

Attack	Complexity
Direct Attack	$C_{MQ}(q,\ o,\ n)$
UOV-Reconciliation Attack	$C_{MQ}(q,\ o,\ v)$
Kipnis-Shamir Attack	$q^{v-o-1} \cdot o^4$
Intersection Attack	$C_{MQ}(1,\ ok(k+1)/2 - k(k-1),\ vk - o(k-1))$

Parameter Selection

- Parameter (F_q, o, v), q=256
  - o: the number of equations, even
    - ✓ The selection of o depends on their security against the direct attacks.
    - ✓ o>=46, 72, 96 (security level I, III, V)
  - v: the number of Vinegar values
    - ✓ The selection of v depends on their security against the Kipnis-Shamir attack, key recovery attack using good keys and Intersection attack.
    - ✓ The intersection attack is the most powerful attack among the above attacks.
    - ✓ v >=1.5 o, v>=72, 112, 146 (security level I, III, V)

Parameter Selection

- Suggested parameters at three security levels
  - Concrete parameters
  - Conservative parameters
    - ✓ Polynomial XL: complexity of (F_2^8, 46, 72) is 131.25.
    - ✓ (F_2^8, 48, 76): 138.19

Security level	1	3	5
(q, o, v)	$(\mathbb{F}_{2^8}, 46, 72)$	$(\mathbb{F}_{2^8}, 72, 112)$	$(\mathbb{F}_{2^8}, 96, 148)$
Direct(HF5)	135.5	202.4	262.3
Intersection attack	171.883	242.9	304.5

Security Category I Parameter	$(\mathbb{F}_{2^8}, 48, 76)$
Direct Attack (Polynomila XL)	138.19
Intersection Attack	180.48

Key Sizes and Signature Lengths

- Key sizes and signature lengths
  - The same sizes of public keys and signature lengths, different sizes of secret keys
  - The shortest among post-quantum signature schemes based on the other hard problems
    - ✓ **Falcon 666 byte, 1280 byte, 21%, 15.6%**

Scheme	Security Category	I	III	V
Parameter	$(\mathbb{F}_q, o, v)$	$(\mathbb{F}_{2^8}, 46, 72)$	$(\mathbb{F}_{2^8}, 72, 112)$	$(\mathbb{F}_{2^8}, 96, 148)$
MQ-Sign-SS	Sig. Size	134	200	260
	PK	328,441	1,238,761	2,892,961
	SK	15,561	37,729	66,421
MQ-Sign-RS	Sig. Size	134	200	260
	PK	328,441	1,238,761	2,892,961
	SK	133,137	485,281	1,110,709
MQ-Sign-SR	Sig. Size	134	200	260
	PK	328,441	1,238,761	2,892,961
	SK	164,601	610,273	1,416,181
MQ-Sign-RR	Sig. Size	134	200	260
	PK	328,441	1,238,761	2,892,961
	SK	282,177	1,057,825	2,460,469

Scheme	I	III	V
MQ-Sign	134	200	260
Dilithium	2,420	3,293	4,595
Falcon	666	1,280	

Optimal Implementation

- Implementation Specification
  - Target platform: Intel(R) Core(TM) i7-6700X CPU, 3.40GHz, Ubuntu 20.04LTS
  - Random Number Generation and Hashing: AES CTR DRBG, SHA-2 hash function family, ㄴ HA256, SHA384, SHA512 with output lengths of 256, 384, and 512 bits, respectively.
  - Finite field: F_q = F_256
  - Secret key: the form of equivalent key $\mathcal{T} = \begin{pmatrix} \hat{I} & T' \\ 0 & I \end{pmatrix}$ linear map
  - Linear terms
    - ✓ SS, SR, RS have the linear terms.
    - ✓ RR has no linear terms.
  - Solving the linear system
    - ✓ Use the BMI method with depth 1.
    - ✓ $A^{-1} \cdot \underline{\alpha} = \underline{\beta},\ C \cdot \underline{\beta},\ \ [D - CA^{-1}B]^{-1} \cdot \underline{\gamma},\ (A^{-1}B) \cdot \underline{\gamma'}.$ $\quad R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$
  - Constant-time implementation

AVX2-Optimized Implementation

- AVX2-optimized implementation
  - Each result of signing and verification (resp. key generation) is an average of 100,000 (resp. 10,000) measurements using the C programming language with GNU GCC version 9.4.0 compiler
  - Due to different selection of secret keys in KeyGen, different results for key generation and signing, but the same results for verification.

Scheme	Security Category	I	III	V
MQ-Sign-SS	KeyGen.	6,046,385	25,506,351	62,972,759
	Sign	154,645	378,634	471,085
	Verify	71,267	232,377	401,412
MQ-Sign-RS	KeyGen.	9,026,556	38,892,327	95,016,980
	Sign	174,790	447,656	626,373
	Verify	71,267	232,377	401,412
MQ-Sign-SR	KeyGen.	10,222,889	43,634,459	104,441,512
	Sign	166,987	417,445	630,000
	Verify	71,267	232,377	401,412
MQ-Sign-RR	KeyGen.	13,493,778	56,071,342	138,481,524
	Sign	184,761	491,738	708,415
	Verify	71,267	232,377	401,412

Additional Improvement and Future Plan

- Additional improvements
  - Parallel computation on multiple cores
    - ✓ 2x faster key generation
  - Precomputation
    - ✓ Off-line sign/on-line sign
    - ✓ 50x faster signing

Thanks.