



# MQ-Sign: 다변수 이차식 기반 전자서명 알고리즘

NIMS 암호기술연구팀  
심경아, 권혁동



## 목 차

- 설계 방향/원칙
- MQ-Sign RR, MQ-Sign-LR
- 안전성 증명 및 안전성 분석
- 파라미터 설정
- 최적 구현
- 향후 계획



## 설계 방향/원칙

### ■ 단일 레이어 UOV 기반 구조의 최소화/짧은 길이의 전자서명

- 기존 다변수 이차식 기반 잠재적 위협 제거
  - ✓ Multiple-layer 구조 관련 공격의 위험성 제거 => Single-layer 구조 사용
- 전자서명 값의 길이가 짧다는 측면에서 대체 불가 원천기술
  - ✓ 안전도 1, 3, 5에서 150, 216, 276바이트

### ■ UOV 보다 짧은 비밀키 제공

- MQ-Sign-RR: 랜덤 Vinegar\*Vinegar 이차항, 랜덤 Vinegar\*Oil 이차항, linear map  $\mathcal{T}_E = \begin{pmatrix} I & T \\ 0 & I \end{pmatrix}$
- MQ-Sign-LR: 선형식 기반 Vinegar\*Vinegar 이차항, 동치키 형태의 linear map
  - ✓ Vinegar\*Vinegar 이차항의 compact한 표현으로 UOV 보다 비밀키 길이 42% 이상 축소

### ■ 공개키 변형을 이용한 잠재적 대한 안전성 보장

- $H(M)$  대신  $H(M||H(P))$  사용함으로써 공개키 변형을 통한 잠재적 위조 공격 방어
  - ✓ 공개키 전체 대신 부분을 활용 가능
- 공개키와 메시지에 연결된 전자서명 생성



## 설계 방향/원칙

### ▪ 빠른 서명 생성과 사전 계산이 용이한 구조

- 블록 행렬 을 이용한 고속화 기법으로 서명 생성 효율성 향상
- MQ-Sign-LR Vinegar\* Vinegar 이차항 Vinegar value 대비 부분 순환행렬-벡터 곱
  - ✓ RR 보다 키생성 34%-42% 향상, 서명 생성 27%-37% 향상
- 서명 생성에서 대부분의 계산이 메시지와 독립적으로 이루어지도록 설계
  - ✓ off-line 계산 후 on-line 서명 생성은 해시 계산과 행렬과 벡터 계산 한번으로 구성
  - ✓ 4-6배 이상 서명 속도 향상

### ▪ S/W, H/W 구현 용이

- 작은 크기의 유한체 사용, 행렬 곱, 행렬-벡터 곱, 구현 용이
- $F_{2^8}$  유한체 사용 8-비트 AVR microprocessor에서 우수한 성능

### ▪ 부채널 공격 대응

- 시간차 공격 대응 constant-time 구현
- 단일 레이어 사용 -> 특정 부채널 공격에 안전



## Oil-Vinegar 함수

### 키생성 알고리즘

➤ 비밀키:  $(F, T)$ , 공개키:  $P = F \circ T$ ,  $\mathcal{P} = (\mathcal{P}^{(1)}, \dots, \mathcal{P}^{(m)})$ ,  $\mathcal{F} = (\mathcal{F}^{(1)}, \dots, \mathcal{F}^{(o)})$

➤ Index set :  $\{1, \dots, v, v+1, \dots, v+o=n\}$

$v$ 개 Vinegar variable     $o$ 개 Oil variable

▪ Central map:  $\mathcal{F}^{(k)} = \mathcal{F}_V^{(k)} + \mathcal{F}_{OV}^{(k)} + \mathcal{F}_L^{(k)} + \mathcal{F}_C^{(k)}$

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

Missing Oil\*Oil structure

	$V$	$O$
$V$		
$O$		

➤ 이차식의 시스템 invert 방법

- ✓  $v$ 개의 Vinegar value 선택 후 central map의 Vinegar 변수에 대입
- ✓ 이차식의 시스템이 linear system으로 변환 -> 가우스 소거법으로 해를 구함



## MQ-Sign

- MQ-Sign 키생성: MQ-Sign-RR
  - Vinegar\*Vinegar 이차항 생성
    - ✓ Vinegar\*Vinegar 이차항 부분에 대응되는 symmetric 행렬의 full rank를 가지도록 선택

$$\mathcal{F}_V^{(k)} = \mathcal{F}_{V,R}^{(k)} = \sum_{i \in O, j \in V} \alpha_{ij}^{(k)} x_i x_j,$$



## MQ-Sign

### MQ-Sign 키생성: MQ-Sign-LR

- Vinegar\*Vinegar 이차항 생성:  $L_i = \sum_{j=1}^v \delta_j x_j$  ( $1 \leq i \leq v$ )
- Vinegar\*Vinegar 이차항 부분에 대응되는 symmetric 행렬의 full rank를 가지도록 선택

$$\begin{aligned}\mathcal{F}_V^{(1)} &= \mathcal{F}_{V,LR}^{(1)} = x_1 \cdot L_1 + x_2 L_2 + \cdots + x_v L_v, \\ \mathcal{F}_V^{(2)} &= \mathcal{F}_{V,LR}^{(2)} = x_v \cdot L_1 + x_1 L_2 + \cdots + x_{v-1} L_v, \\ &\quad \cdots, \\ \mathcal{F}_V^{(o)} &= x_{v-o_1+2} \cdot L_1 + x_{v-o_1+3} L_2 + \cdots + x_{v-o_1+1} L_v.\end{aligned}$$

- Compact representation: 키길이  $v \cdot v \cdot o/2$  개의 유한체 원소에서  $v \cdot o$  개 유한체 원소로 축소

$$\begin{pmatrix} x_1 & x_2 & \cdots & x_v \\ x_v & x_1 & \cdots & x_{v-1} \\ \cdots & \cdots & \cdots & \cdots \\ x_{v-o_1+2} & x_{v-o_1+3} & \cdots & x_{v-o_1+1} \end{pmatrix} \cdot \begin{pmatrix} L_1 \\ L_2 \\ \cdots \\ L_v \end{pmatrix}$$



## MQ-Sign

➤ Vinegar value 대입 연산: 순환 행렬-벡터 곱, 효율적 연산 가능

✓ 랜덤 Vinegar value  $s_V = (s_0, \dots, s_v) \in \mathbb{F}_q^v$

$$\begin{pmatrix} \mathcal{F}_{V,LR}^{(1)}(s_V) \\ \mathcal{F}_{V,LR}^{(2)}(s_V) \\ \dots \\ \mathcal{F}_{V,LR}^{(o)}(s_V) \end{pmatrix} = \begin{pmatrix} s_1 & s_2 & \dots & s_v \\ s_v & s_1 & \dots & s_{v-1} \\ \dots & \dots & \dots & \dots \\ s_{v-o_1+2} & s_{v-o_1+3} & \dots & s_{v-o_1+1} \end{pmatrix} \cdot \begin{pmatrix} L_1(s_V) \\ L_2(s_V) \\ \dots \\ L_v(s_V) \end{pmatrix}$$

▪ Vinegar\*Oil 이차항 생성

➤ Vinegar\*Oil 이차항 부분에 대응되는 symmetric 행렬의 full rank를 가지도록 선택

$$\mathcal{F}_{OV}^{(k)} = \mathcal{F}_{OV,R}^{(k)} = \sum_{i,j \in V, i \leq j} \beta_{ij}^{(k)} x_i x_j$$





## MQ-Sign

- MQ-Sign 키생성

- MQ-Sign-RR, MQ-Sign-LR: 일차항과 상수항 없음

- Random  $\mathcal{F}_V^{(k)}$ , random  $\mathcal{F}_{OV}^{(k)}$ , and the linear map  $\mathcal{T}_E$  for MQ-Sign-RR:

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

- Line-based  $\mathcal{F}_{V,LR}^{(k)}$ , random  $\mathcal{F}_{OV,R}^{(k)}$ , and the linear map  $\mathcal{T}_E$  for MQ-Sign-LR:

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



## MQ-Sign

- MQ-Sign 키생성: linear map  $T$

➤ 동치키 형태의 linear map 사용  $\mathcal{T}_E = \begin{pmatrix} I & T \\ 0 & I \end{pmatrix}$

➤ 공개키  $\mathcal{P} = (\mathcal{P}^{(1)}, \dots, \mathcal{P}^{(m)})$  계산  $\rightarrow \mathcal{P}^{(k)} = \mathcal{T}_E^T \mathcal{F}^{(k)} \mathcal{T}_E, \mathcal{F} = (\mathcal{F}^{(1)}, \dots, \mathcal{F}^{(o)})$

$$\begin{pmatrix} I & 0 \\ T & I \end{pmatrix} \begin{pmatrix} \mathcal{F}_1^{(k)} & \mathcal{F}_2^{(k)} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} I & T \\ 0 & I \end{pmatrix} \\ = \begin{pmatrix} \mathcal{F}_1^{(k)} & \mathcal{F}_1^{(k)} T + \mathcal{F}_2^{(k)} \\ T^T \mathcal{F}_1^{(k)} & T^T \mathcal{F}_1^{(k)} T + T^T \mathcal{F}_2^{(k)} \end{pmatrix}$$

$$\mathcal{P}^{(k)} = \begin{pmatrix} \mathcal{P}_1^{(k)} & \mathcal{P}_2^{(k)} \\ 0 & \mathcal{P}_3^{(k)} \end{pmatrix} = \begin{pmatrix} \mathcal{F}_1^{(k)} & (\mathcal{F}_1^{(k)} + \mathcal{F}_1^{(k)})^T T + \mathcal{F}_2^{(k)} \\ 0 & \text{Upper}(T^T \mathcal{F}_1^{(k)} T + T^T \mathcal{F}_2^{(k)}) \end{pmatrix}$$



## MQ-Sign

- Linear system의 해 계산: BMI 방법으로  $R \cdot \mathbf{x} = \xi$  계산

- R의 역행렬 계산없이  $R^{-1} \cdot \xi$  을 직접 계산: Half-size 행렬-벡터 곱, 식의 개수가 클수록 효과 큼
- $A^{-1}$ ,  $A^{-1}B$ ,  $D-CA^{-1}$ ,  $[D-CA^{-1}B]^{-1}$  계산: Half-size 행렬의 역행렬 계산

$$R^{-1} \cdot \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_o \end{pmatrix} = \begin{pmatrix} I & -A^{-1}B \\ 0 & I \end{pmatrix} \begin{pmatrix} A^{-1} & 0 \\ 0 & [D-CA^{-1}B]^{-1} \end{pmatrix} \begin{pmatrix} I & 0 \\ -CA^{-1} & I \end{pmatrix} \begin{pmatrix} \xi_1 \\ \vdots \\ \xi_o \end{pmatrix}$$

$A_{Sc} = [D-CA^{-1}B]$ : A의 Schur complement

$$CA^{-1} \cdot (\xi_1, \dots, \xi_{o/2})^T + (\xi_{o/2+1}, \dots, \xi_o)^T = (\alpha_{o/2+1}, \dots, \alpha_o)^T,$$

$$A^{-1} \cdot (\xi_1, \dots, \xi_{o/2})^T = (\beta_1, \dots, \beta_{o/2})^T,$$

$$A_{Sc}^{-1} \cdot (\alpha_{o/2+1}, \dots, \alpha_o)^T = (\beta_{o/2+1}, \dots, \beta_o)^T,$$

$$(\beta_1, \dots, \beta_{o/2})^T + (-A^{-1}B) \cdot (\beta_{o/2+1}, \dots, \beta_o)^T = (\gamma_1, \dots, \gamma_{o/2})^T.$$

- $R \cdot \mathbf{x} = \xi$  의 해:  $s_O = (\gamma_1, \dots, \gamma_{o/2}, \beta_{o/2+1}, \dots, \beta_o)$



## 사전 계산에 용이한 구조

- 사전 계산: 메시지에 독립적인 부분 off-line 사전 계산
  - Vinegar 값 대입 후 linear system의 해를 구하는 부분 사전 계산: 10-50배 이상 향상

### [Offline Signing]

- Choose random Vinegar values  $s_V = (s_1, \dots, s_v) \in \mathbb{F}_q^v$ .
- Substitute  $s_V$  into  $o$  the secret polynomials  $\mathcal{F}^{(k)}$  ( $1 \leq k \leq o$ ), and get a  $o \times o$  coefficient matrix  $R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$  and a constant vector  $\mathbf{c}_V = (c_1, \dots, c_o) = (\mathcal{F}_V^{(1)}(s_V), \dots, \mathcal{F}_V^{(o)}(s_V))$ .
- Compute  $A^{-1}$ ,  $CA^{-1}$ ,  $CA^{-1}B$ ,  $A_{sc}^{-1}$  and  $A^{-1}B$ . If  $A$  or  $A_{sc}$  is not invertible then go back to the first step.
- Choose a random salt  $r$ .
- Store the precomputed values  $\langle s_V, \mathbf{c}_V, A^{-1}, CA^{-1}, A_{sc}^{-1}, A^{-1}B, r \rangle$ .



## 사전 계산이 용이한 구조

- On-line 계산: 메시지 해시, 한 번의 행렬-벡터 곱
  - Vinegar 값의 노출과 재사용이 없도록 안전하게 사용

### [Online Signing]

- Compute  $\mathbf{h} = H(M||r||ph)$  ( $H(M||r)$  in LR) for a message  $M$ .
- Compute  $R^{-1} \cdot \xi = s_O$  by computing four block matrix-vector products from the precomputed values in the BMI method, where  $\xi = \mathbf{h} - c_V = (h_1 - c_1, \dots, h_o - c_o)$  and  $\mathbf{h} = (h_1, \dots, h_o)$ .
- Compute  $\mathbf{z} = \begin{pmatrix} s_V^T + T \cdot s_O^T \\ s_O \end{pmatrix}$ .
- Output  $\sigma = (\mathbf{z}, \mathbf{r})$  as a signature on  $M$ .

행렬-벡터 곱 한번



## MQ-Sign

- 서명 생성에 공개키와 서명을 묶는 binding technique 추가
  - 두개의 공개키  $\mathcal{P} = \mathcal{F} \circ T$ ,  $\mathcal{P}' = (\mathcal{F} \circ T) \circ T'$ .
  - 메시지  $M$ , 공개키  $\mathcal{P}$  에 대한 서명  $\sigma = (z, r)$  이용
    - >  $\mathcal{P}'$  에 대한 서명  $\sigma' = (z', r)$ ,  $z' = (T')^{-1}(z)$  생성
  - 서명 생성/검증 알고리즘에서 메시지와 공개키의 해시 값 추가:  $H(M || r || H(\mathcal{P}))$ 
    - ✓ 공개키와 메시지에 정확하게 연결된 서명 생성
- $H(\mathcal{P})$  계산: 큰 공개키 사이즈로 비효율적
  - MQ-Sign-RR: 키생성에서  $H(\mathcal{P})$  계산한 후 공개키 비밀키에 추가
  - $H(\mathcal{P})$  대신  $H(\mathcal{P}_2 || \mathcal{P}_3)$ ,  $\mathcal{P}_2 = \{\mathcal{P}_2^{(k)}\}_{k=1}^o$ ,  $\mathcal{P}_3 = \{\mathcal{P}_3^{(k)}\}_{k=1}^o$  계산
    - ✓ 위 공격에서  $\mathcal{P}_1^{(k)} = \mathcal{P}'_1^{(k)}$  ( $k = 1, \dots, o$ )
    - ✓ 두 공개키의 central map은  $\mathcal{F}$ 로 동일, 서로 다른 linear map,  $\mathcal{T}$ 와  $\mathcal{T} \circ \mathcal{T}'$ ,  $\mathcal{P}_1^{(k)}$  값은 linear map에 의존하지 않음.



## MQ-Sign

- MQ-Sign 키생성

- MQ-Sign-RR, MQ-Sign-LR

- Random  $\mathcal{F}_V^{(k)}$ , random  $\mathcal{F}_{OV}^{(k)}$ , and the linear map  $\mathcal{T}_E$  for MQ-Sign-RR:

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

- Line-based  $\mathcal{F}_{V,LR}^{(k)}$ , random  $\mathcal{F}_{OV,R}^{(k)}$ , and the linear map  $\mathcal{T}_E$  for MQ-Sign-LR:

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

- 비밀키 (F, T, ph=H(P)), 공개키 (P, ph)

- ✓ 공개키 생성:  $\mathcal{P} = \mathcal{F} \circ \mathcal{T}$

- ✓ 비밀키 (F, 동치키 형태의 linear map T)



## MQ-Sign

### ■ 전자서명 생성 알고리즘

- **Sign**( $SK, \lambda, M$ ). Given a message  $M$  and a collision-resistant hash function  $H : \{0, 1\}^* \rightarrow \mathbb{F}_q^o$ , compute  $\mathbf{a} = \mathcal{F}^{-1}(\xi)$ , i.e.  $\mathcal{F}(\mathbf{a}) = \xi$  as follow:

- **[Vinegar Value Substitution.]** Select Vinegar values  $s_V = (s_1, \dots, s_v) \in \mathbb{F}_q^v$  at random and do the followings:

- \* Substitute  $s_V$  into  $o$  central polynomials  $\mathcal{F}^{(k)}$  ( $1 \leq k \leq o$ ) and get  $o$  polynomials of unknowns  $x_{v+1}, \dots, x_{v+o}$  with an  $o \times o$  coefficient matrix  $R = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ .
- \* Compute  $A^{-1}$ ,  $CA^{-1}$ ,  $CA^{-1}B$ ,  $A_{sc}^{-1}$ ,  $A^{-1}B$ .
- \* If  $A$  or  $A_{sc}$  is not invertible, then choose another vector of Vinegar values  $s'_V$  and try again.

메시지에 무관한 계산





## MQ-Sign

### ■ 전자서명 생성 알고리즘

- **[Solving Linear System.]** Choose a  $l$ -bit random salt  $r$ ,

- \* MQ-Sign-RR: compute  $\mathbf{h} = H(M||r||ph) \in \mathbb{F}_q^o$ .

- \* MQ-Sign-LR: compute  $\mathbf{h} = H(M||r) \in \mathbb{F}_q^o$ .

Find a solution  $s_O = (s_{v+1}, \dots, s_{v+o})$  of the linear system  $R \cdot \mathbf{x} = \xi$  by using the block matrices of the BMI method, where  $\xi = \mathbf{h} - c_V$  and  $c_V = (\mathcal{F}_V^{(1)}(s_V), \dots, \mathcal{F}_V^{(o)}(s_V))$ .

- **[Output.]** Compute  $\mathbf{z} = \begin{pmatrix} s_V^T + T \cdot s_O^T \\ s_O \end{pmatrix}$ . Output  $\sigma = (\mathbf{z}, r)$  as a signature on  $M$ .

} 메시지에 의존



## MQ-Sign

- 전자서명 검증 알고리즘

- **Verify**( $PK, M, \sigma$ ). Given a signature  $\sigma = (\mathbf{z}, r)$  on a message  $M$  and the public key  $\mathcal{P}$ ,
  - MQ-Sign-RR: check the equality  $\mathcal{P}(\mathbf{z}) = H(M || r || ph)$ .
  - MQ-Sign-LR: check the equality  $\mathcal{P}(\mathbf{z}) = H(M || r)$ .



- Hard problems

**Definition 1 (MQ-Problem).** Given a system  $\mathcal{P} = (\mathcal{P}^{(1)}, \dots, \mathcal{P}^{(m)})$  of  $m$  quadratic equations defined over  $\mathbb{F}_q$  in variables  $x_1, \dots, x_n$  and  $\mathbf{y} = (y_1, \dots, y_m) \in \mathbb{F}_q^m$ , find values  $\mathbf{x}' = (x'_1, \dots, x'_n) \in \mathbb{F}_q^n$  such that  $\mathcal{P}(\mathbf{x}') = \mathbf{y}$ :  $\mathcal{P}^{(1)}(\mathbf{x}') = y_1, \dots, \mathcal{P}^{(m)}(\mathbf{x}') = y_m$ .

**Definition 2 (Extended Isomorphism of Polynomials (EIP) 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  $\mathcal{F}' \in \mathcal{C}$ .



## 안전성 증명 및 안전성 분석

**Definition 3 (UOV Problem).** Given  $(\mathcal{P}, y)$ , find a preimage  $z \in \mathbb{F}_q^n$  such that  $\mathcal{P}(z) = y$ , where  $\mathcal{P}$  is derived from  $(\mathcal{P}, \mathcal{F}, \mathcal{T}) \leftarrow \text{GenUOVfunc}(1^\lambda)$  and a challenge  $y \in \mathbb{F}_q^m$ .

**Definition 4.** An algorithm  $\mathcal{A}$  has advantage  $\epsilon$  to solve the UOV problem if

$$\text{Adv}_{\mathcal{A}}(t) = \Pr [\mathcal{A}(\mathcal{P}, y) = z \mid \mathcal{P} \leftarrow \text{GenUOVfunc}, y \leftarrow \mathbb{F}_q^m] \geq \epsilon.$$

If there is no algorithm  $\mathcal{A}(t, \epsilon)$  that solves the UOV problem then we define that the UOV problem is  $(t, \epsilon)$ -hard.

**Definition 5 (UOV Assumption).** The UOV function generator  $\text{GenUOVfunc}$  is  $(t(\lambda), \epsilon(\lambda))$ -secure if there is no inverting algorithm that takes as input  $\mathcal{P}$  generated  $\mathcal{P} \leftarrow \text{GenUOVfunc}$  and a challenge  $y \in \mathbb{F}_q^m$  to find a preimage  $z \in \mathbb{F}_q^n$  such that  $\mathcal{P}(z) = y$  at  $t(\lambda)$  time with probability at least  $\epsilon(\lambda)$ .

**Theorem 1.** If the UOV problem is  $(\epsilon', t')$ -hard then the modified UOV is  $(\epsilon, t, q_H, q_s)$ -secure in the EUF-CMA game, where

$$\epsilon' \geq \epsilon \cdot \frac{1 - (q_H + q_s)q_s 2^{-l}}{q_H + q_s + 1}, \quad t' \geq t + (q_H + q_s + 1)(t_{\mathcal{P}} + \mathcal{O}),$$

$t_{\mathcal{P}}$  is running time to evaluate  $\mathcal{P}$  and  $l$  is the length of a salt.



## 안전성 증명 및 안전성 분석

- Direct attack

➤ MQ-문제:  $\mathcal{P}(x) = H(M||r||ph)$  ( $H(M||r)$  in LR) , XL, Polynomial XL, Grobner basis 알고리즘

Algorithms	44	46	48	50	52
Hybrid F5	131.86	137.43	142.99	148.56	154.12
Wiedemann XL	133.40	138.98	144.55	150.13	155.70
Polynomial XL	125.50	131.25	138.19	142.66	146.99

**Table 1.** Complexity Estimates against Direct Attacks at the Security Level 1.

Algorithms	68	70	72	74	76
Hybrid F5	195.37	200.92	203.58	209.03	214.59
Wiedemann XL	196.93	202.51	204.97	210.41	216.01
Polynomial XL	189.41	194.50	199.39	203.04	209.49

**Table 2.** Complexity Estimates against Direct Attacks at the Security Level 3.

Algorithms	94	96	98	100	102
Hybrid F5	261.18	266.50	272.10	277.32	279.90
Wiedemann XL	262.50	267.76	273.38	278.54	281.23
Polynomial XL	253.98	260.24	267.35	271.57	275.31

**Table 3.** Complexity Estimates against Direct Attacks at the Security Level 5.



## 안전성 증명 및 안전성 분석

- key recovery attack using good keys

- $(\mathcal{F}', \mathcal{T}')$ 가  $(\mathcal{F}, \mathcal{T})$ 의 동치키,  $\mathcal{P} = \mathcal{F} \circ \mathcal{T} = \mathcal{F}' \circ \mathcal{T}'$ ,  $\mathcal{F}'$ 가  $\mathcal{F}$ 의 모든 zero의 위치를 유지

$$\mathcal{P} = (\mathcal{F} \circ \Omega) \circ (\Omega^{-1} \circ \mathcal{T})$$

$$\mathcal{T}'^{-1} = \mathcal{T}^{-1} \cdot \Omega = \begin{pmatrix} I_{v \times v} & \widetilde{T'}_{v \times o} \\ 0_{o \times v} & I_{o \times o} \end{pmatrix}, \quad \Omega = \begin{pmatrix} \Omega_{v \times v}^{(1)} & 0_{v \times o} \\ \Omega_{o \times v}^{(3)} & \Omega_{o \times o}^{(4)} \end{pmatrix}$$

- Good key 이용:  $v$ 개의 변수를 갖는  $o$ 개의 이차 시스템을 푸는 것과 동일

$$Complexity_{KRA}(q, o, v) = C_{MQ}(q, o, v)$$



## 안전성 증명 및 안전성 분석

- Kipnis-Shamir attack

- Invariant subspace를 찾는 공격:  $Complexity_{KS}(q, o, v) = q^{v-o-1} \cdot o^4$

- intersection attack

- Kipnis-Shamir공격의 향상된 버전

$$Complexity_{Inter}(q, o, v) = C_{MQ}(q, ok(k+1)/2 - k(k-1), vk - o(k-1)), \quad k < v/(v-o)$$

- Replacement attack

$$\mathcal{F}_V^{(1)} = \overline{L}_1 \cdot y_1 + \overline{L}_2 y_2 + \cdots + \overline{L}_v y_v,$$

$$\mathcal{F}_V^{(2)} = \overline{L}_v \cdot y_1 + \overline{L}_1 y_2 + \cdots + \overline{L}_{v-1} y_v,$$

$\cdots,$

$$\mathcal{F}_V^{(o)} = \overline{L}_{v-o+2} \cdot y_1 + \overline{L}_{v-o+3} y_2 + \cdots + \overline{L}_{v-o+1} y_v,$$



## 안전성 증명 및 안전성 분석

- 안전성 분석: 알려진 대수적인 공격 분석
  - Direct attack
  - 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	$C_{MQ}(1, ok(k+1)/2 - k(k-1), vk - o(k-1))$





## 파라미터 설정

- 파라미터  $(F_q, o, v)$ ,  $q=256$ 
  - $o$ : 식의 개수,  $v$ : Vinegar 변수의 개수,  $n=v+o$ : 변수의 개수
  - $o$ 의 선택: direct attack에 의존
    - ✓  $o=46, 72, 96$
  - $v$ 의 선택: intersection attack, UOV attack, key recovery attack using good keys
    - ✓ 가장 강력한 공격: intersection attack
    - ✓  $v > 1.5 o$ ,  $v=72, 112, 146 \dots$
  - 안전한 파라미터 결정

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



## 키길이와 서명 길이

- 키길이, 서명 길이 비교
  - 공개키 길이, 비밀키 길이는 다름, 서명 길이는 동일
    - ✓ MQ-Sign-LR의 비밀키 길이가 RR 보다 42% 축소
  - 양자내성 전자서명 중 중 가장 짧은 서명 길이
    - ✓ **Falcon 666 byte, 1280 byte 대비 20%, 16% 정도**

Scheme	Security Level	1	3	5
MQ-Sign-RR	Public Key	328,505	1,238,825	2,893,025
	Secret Key	276,649	1,044,385	2,436,769
	Signature	150	216	276
MQ-Sign-LR	Public Key	328,441	1,238,761	2,892,961
	Secret Key	160,881	601,249	1,400,113
	Signature	150	216	276



## AVX2를 이용한 최적 구현

### ■ AVX2를 이용한 구현 결과

- Intel Xeon(R) Gold 6234 CPU, 3.3GHz
- 키생성: 1,000번 평균, 서명 생성/검증: 10,000번 중간 값
- Hyperthreading and Turbo Boost are switched off
- MQ-Sign-LR이 RR 보다 키생성 34%-42% 향상, 서명 생성 27%-37% 향상, 서명 검증은 동일

Scheme	Security Level	1	3	5
Performance (Reference Code, median cycles)				
MQ-Sign-RR	KeyGen	122,046,651	438,023,770	994,466,810
	Sign	861,724	1,752,258	3,053,560
	Verify	755,522	1,339,244	2,218,340
MQ-Sign-LR	KeyGen	89,401,545	312,936,150	701,261,588
	Sign	451,262	1,004,830	2,026,304
	Verify	774,652	1,414,666	2,202,376
Performance (AVX2-optimized, median cycles)				
MQ-Sign-RR	KeyGen	9,454,708	40,250,626	102,775,550
	Sign	90,480	268,866	524,030
	Verify	50,460	185,086	363,611
MQ-Sign-LR	KeyGen	5,451,597	25,605,484	67,485,424
	Sign	65,300	168,684	360,636
	Verify	51,744	191,986	381,019



## 향후 계획

- 최적 구현
  - 느린 키생성 향상 가능
  - 서명 생성/검증 향상의 여지

감사합니다.