Improvements of MQ-Sign and NCC-Sign

NIMS 암호기술연구팀

목 차

MQ-Sign 개선 사항

설계 방향/원칙

- **단일 레이어 기반 구조의 최소화/가장 짧은 길이의 전자서명**
  - 기존 다변수 이차식 기반 잠재적 위협 제거
    - ✓ Multiple-layer 구조, MinRank 공격의 위험성 제거 => Single-layer 구조 사용
  - 전자서명 값의 길이가 가장 짧다는 측면에서 대체 불가 원천기술
- **다양한 변형의 키 생성 알고리즘 선택**
  - 비밀키 축소를 위한 변형 알고리즘
  - 변형된 키 생성 알고리즘을 사용, 동일한 서명 생성/검증 알고리즘 사용 가능
- **고속화 기법 사용으로 효율성 증대**
  - 서명 생성에서 가우스 소거법 대신 블록 행렬을 이용하는 방법 사용
  - off-line/on-line 서명 생성 (precomputation)으로 수십 배 서명 생성 속도 향상

MQ-Sign 키생성 알고리즘

- MQ-Sign
  - MQ-Sign 키생성 알고리즘 4가지 종류의 변형으로 구성

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

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

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

MQ-Sign 키생성 알고리즘

- **[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 $k = 1, \cdots, o$.

- **[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 $k = 1, \cdots, o$.

MQ-Sign 키생성 알고리즘

- MQ-Sign 4종류의 변형

  - 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:

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

MQ-Sign sparse version공격

- MQ-Sign sparse version
  - ➢ MQ-Sign: RR, SR, RS, SS spare polynomial을 이용한 변형, Kpqc 제출
  - ➢ RS, SS 버전 공격: 비밀키를 찾을 수 있는 특별한 관계식 발견
    - ✓ Aulbach et al 공격
      - Sparse 성질과 동치키 형태 $S = \begin{bmatrix} I_{v\times v} & \mathbf{0}_{v\times o} \\ * & I_{o\times o} \end{bmatrix}$ 이용
    - ✓ Ikematsu et al 공격
      - Sparse 성질만 이용
      - 특별한 관계식으로부터 공개키 null space의 basis를 찾음->linear system의 해
    - ✓ 복잡도를 높일 수 있는 이차항 추가로 공격을 막을 수 있음.
- MQ-Sign
  - ➢ MQ-Sign: 공격에 대한 개선 없이 RR, SR 버전으로 유지

MQ-Sign 개선 사항

- 서명 생성에 공개키와 서명을 묶는 binding technique 추가
  - 공개키와 메시지에 정확하게 연결된 서명 생성
  - 서명 생성/검증 알고리즘에서 메시지와 공개키의 해시 값 추가: $H(M||r||H(\mathcal{P}))$

Security Guarantee against Potential Attacks. In order to certain potential attacks, we use a binding technique so that a signature is identified with a unique public key and message. For given two public keys $\mathcal{P}$ and $\mathcal{P}'$ such that $\mathcal{P}' = \mathcal{P} \circ T'$, if $\tau = (\sigma, r)$ is a signature on a message M under the public key $\mathcal{P}$ then one who knows T' can generate a valid signature $\tau' = (\sigma', r)$ on the same message M under the public key $\mathcal{P}'$ by computing $\sigma' = (T')^{-1}(\sigma)$. To prevent this type of attacks, one needs to bind a message being signed with the public key, i.e. $H(M||r||H(\mathcal{P}))$. So, we use $H(M||r||H(\mathcal{P}))$ in the signing and verification algorithms.

MQ-Sign 개선 사항

- Reference 구현: Intel i7-13700K 3.40GHz
  - 기존

Scheme	Security Level	1	3	5
MQ-Sign-RR	KeyGen.	79,864,302	302,322,971	755,934,235
	Sign	1,303,024	3,333,303	6,577,958
	Verify	1,243,091	3,125,277	5,545,017
MQ-Sign-SR	KeyGen.	76,237,178	288,902,825	717,203,934
	Sign	201,834	707,959	1,486,775
	Verify	1,243,091	3,125,277	5,545,017

  - 개선

Scheme	Security Level	1	3	5
MQ-Sign-RR	KeyGen.	49,920,471	193.215,572	485,890,975
	Sign	805,276	2,000,408	3,916,850
	Verify	775,744	1,872,550	3,493,739
MQ-Sign-SR	KeyGen.	47,331,471	183,855,056	466,702,514
	Sign	435,128	1,209,312	2,599,835
	Verify	777,002	1,875,890	3,581,903

NCC-Sign 개선 사항

설계 방향/원칙

- NCC-Sign: Non-cyclotomic과 cyclotomic 모두 지원하는 Ring-LWE 기반 전자서명
  - Module lattice 전용 BKZ 알고리즘 존재
  - power-of-2 cyclotomic polynomial ring이 아닌 파라미터 점프가 없는 polynomial ring 사용
  - 키/서명 길이의 축소보다는 안전성/속도에 초점을 둔 설계
- [안전성 강화] 최초의 non-cyclotomic 격자 기반 전자서명 알고리즘
  - Intermediate Security Guarantee : Standard lattice 기반 > Non-cyclotomic > power-of-2 cyclotomic
  - 구조의 최소화: prime-degree large Galois group, inert modulus, $\phi(X) = X^p - X + 1$
- [효율성 증대] trinomial, $\phi(X) = X^n - X^{n/2} - 1$, $n = 2^a \cdot 3^b$
- Fiat-Shamir with aborts 기반 전자서명, 검증된 방법론, 공개키 압축 방법 이용 공개키 축소
- 보수적인 파라미터 선택
  - Core-SVP estimate 복잡도 128, 192, 256에 밀접하게 선택
  - NCC-Sign trinomial은 Dilithium 보다 더 높은 복잡도를 제공, 성능 우위 (Reference 구현 기준)

NCC-Sign 개선 사항

- NCC-Sign non-cyclotomic version
  - ➢ 보수적인 파라미터 선택
    - ✓ Core-SVP estimate 복잡도 128, 192, 256에 밀접하게 선택
    - ✓ NTT unfriendly ring $Z_q[X]/(X^p - X + 1)$ 에서 NTT 사용

Parameter/Security Level	1^c	3^c	5^c
p	1201	1607	2039
q	17279291	17305741	17287423
d [dropped bits from t] $(2^d\tau < \gamma_2)$	12	13	13
τ [# of ± 1's in c]	32	32	32
challenge entropy $[\log \binom{p}{\tau} + \tau]$	241	254	265
γ_1 [y coefficient range]	2^{19}	2^{19}	2^{19}
γ_2 [low-order rounding range]	$(q-1)/70$ $(= 246847)$	$(q-1)/60$ $(= 288429)$	$(q-1)/58$ $(= 298059)$
η [secret key range]	2	2	2
β	128	128	128
ω [max # of 1's in hint]	80	80	80
Exp. reps. $[\approx e^{(p_1\beta_2+p_2\beta_1)(1/\gamma_1+1/\gamma_2)}]$	2.5	3.02	3.95
Key/Signature Size			
Public key size	1984	2443	3091
Secret key size	2800	3914	4940
Signature size	3186	4251	5385
SIS Hardness (Core-SVP)			
BKZ block size b to break SIS	463	666	895
Best known classical bit cost	135	194	261
Best known quantum bit cost	122	176	237
LWE Hardness (Core-SVP)			
BKZ block size b to break LWE	491	711	956
Best known classical bit cost	143	207	279
Best known quantum bit cost	130	188	253
LWE Estimator			
Cost to SIS (BKZ b)	155.5 (484)	218.1 (697)	289.7 (941)
Quantum cost to SIS	135.3	192.0	256.8
Cost to LWE (BKZ b)	167.3 (483)	229.3 (704)	298.1 (949)
Quantum cost to LWE	141.1	198.4	262.0

NCC-Sign 개선 사항

- NCC-Sign non-cyclotomic version
  - ➢ 보수적인 파라미터 선택
    - ✓ Core-SVP estimate 복잡도 128, 192, 256에 밀접하게 선택
    - ✓ NTT unfriendly ring $Z_q[X]/(X^p - X + 1)$ 에서 NTT 사용

Parameter/Security Level	$1^{c,1}$	$3^{c,1}$	$5^{c,1}$
p	1201	1607	2039
q	17279291	17305741	17287423
d [dropped bits from t] $(2^d\tau < \gamma_2)$	12	13	13
τ [# of ± 1's in c]	32	32	32
challenge entropy $[\log \binom{p}{\tau} + \tau]$	241	254	265
γ_1 [y coefficient range]	2^{19}	2^{19}	2^{19}
γ_2 [low-order rounding range]	$(q-1)/70 = 246847$	$(q-1)/60 = 288429$	$(q-1)/58 = 298059$
η [secret key range]	1	1	1
β	64	64	64
ω [max # of 1's in hint]	80	80	80
Exp. reps. $[\approx e^{(p_1\beta_2+p_2\beta_1)(1/\gamma_1+1/\gamma_2)}]$	1.58	1.74	1.98
pk size	1984	2443	3091
sk size	2703	3817	4843
sig size	3936	5255	6659
BKZ block-size b to break SIS	463	666	895
Best Known Classical bit-cost	135	194	261
Best Known Quantum bit-cost	122	176	237
Best Plausible bit-cost	96	138	185
BKZ block-size b to break LWE	450	656	884
Best Known Classical bit-cost	131	191	258
Best Known Quantum bit-cost	119	174	234
Core-SVP cost by Lattice estimator			
BKZ block-size b to break LWE	442	642	863
Classical bit-cost (method)	129.1 (usvp)	187.8 (dual hybrid)	252.2 (dual hybrid)
Hybrid-decoding attack cost			
BKZ block-size b to break LWE	445	655	890
Classical bit-cost	168.6	231.4	301.5
Hybrid-dual attack cost			
BKZ block-size b to break LWE	430	621	842
Classical bit-cost	156.1	213.2	277.1

NCC-Sign 개선 사항

- NCC-Sign trinomial version
  - ➢ 보수적인 파라미터 선택
    - ✓ 안전도 마진 확보
    - ✓ NTT friendly ring

      $Z_q[X]/(X^n - X^{n/2} - 1)$

      에서의 NTT 사용

Parameter/Security Level	1	3	5
n	1152	1536	2304
q	8401537	8397313	8404993
d [dropped bits from t] $(2^d\tau < \gamma_2)$	12	12	13
τ [# of ± 1's in c]	25	29	32
challenge entropy $[\log\binom{p}{\tau} + \tau]$	195	232	271
γ_1 [y coefficient range]	2^{18}	2^{18}	2^{19}
γ_2 [low-order rounding range]	131274	131208	262656
η [secret key range]	1	1	1
β	50	58	64
ω [max # of 1's in hint]	80	80	80
Exp. reps. $[\approx e^{n\beta(1/\gamma_1+1/\gamma_2)}]$	1.93	2.76	2.32
Key/Signature Size			
pk size	1760	2336	3200
sk size	2400	3168	4992
sig size	2912	3872	6080
SIS Hardness (Core-SVP)			
BKZ block-size b to break SIS	462	671	1005
Best Known Classical bit-cost	135	196	293
Best Known Quantum bit-cost	122	177	266
LWE Hardness (Core-SVP)			
BKZ block-size b to break LWE	451	652	1078
Best Known Classical bit-cost	131	190	315
Best Known Quantum bit-cost	119	172	285
Lattice estimator (Core-SVP)			
BKZ block-size b to break LWE	452	652	1072
Classical bit-cost (method)	132 (usvp)	190.7 (dual hybrid)	313.3 (dual hybrid)

NCC-Sign 개선 사항

- NCC-Sign vs Dilithium

Scheme	Core-SVP/Size	1(128)	3(192)	5(256)
Dilithium	SIS	123	186	265
	LWE	123	182	252
	Repetitions	4.25	5.1	3.85
	Public key size	1312	1952	2592
	Signature size	2420	3293	4595
NCC-Sign Non-Cyclototmic ($\eta = 2$)	SIS	135	194	261
	LWE	143	207	279
	Repetitions	2.27	2.7	3.43
	Public key size	1984	2443	3091
	Signature size	3186	4251	5385
NCC-Sign Non-Cyclototmic ($\eta = 1$)	SIS	135	194	261
	LWE	131	191	258
	Repetitions	1.58	1.74	1.98
	Public key size	1984	2443	3091
	Signature size	3936	5255	6659
NCC-Sign Trinomial	SIS	135	196	293
	LWE	131	190	315
	Repetitions	2.27	2.7	3.43
	Public key size	1760	2336	3200
	Signature size	2912	3872	6080

NCC-Sign 개선 사항

- Reference 구현: Intel i9-10980XE 3.0GHz
  - ➢ NCC-Sign non-cyclotomic
    - ✓ Toom-Cook

Algorithm/Security Level	I^c	III^c	V^c
KeyGen	1,727,508	2,965,942	4,700,228
Sign	11,768,076	20,816,964	42,227,652
Verify	3,400,702	5,876,246	9,324,876

    - ✓ NTT

Algorithm/Security Level	1^c	3^c	5^c
KeyGen	979,979	1,001,022	1,034,193
Sign	7,269,506	8,752,038	10,719,700
Verify	1,863,350	1,884,647	1,926,235

  - ➢ NCC-Sign trinomial: NTT, Dilithium 보다 성능 우위 -> Dilithium reference 구현

Algorithm/Security Level	1	3	5
KeyGen	139,507	180,869	272,344
Sign	440,334	742,644	1,033,312
Verify	168,804	213,435	347,790

Algorithm/Security Level	1	3	5
KeyGen	150,321	280,158	407,743
Sign	520,136	832,706	1,049,686
Verify	162,721	252,928	413,280

감사합니다.