What is going on in the on ramp call?

Violetta Weger

Young Cryptographers in Genova 2024

November 28, 2024

Post-quantum Cryptography

Asymmetric

Public-key

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Public-key

→ Integer factorization

→ Discrete logarithm over F_p

→ Discrete logarithm over ell. curves

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

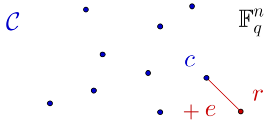
Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Code-based



- $C = \quad G \quad \mathbb{F}_q^n$ linear subspace
- Decode: $r = mG + e$ find closest $c = mG$
- $\mathrm{wt}_H(e) = |\{i : e_i = 0\}|$

Post-quantum Cryptography

Asymmetric

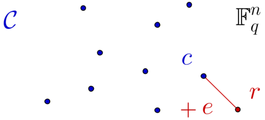
- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

Public-key

→ Integer factorization
→ Discrete logarithm over $\mathbb{F}_p$
→ Discrete logarithm over ell. curves

Code-based



- $C = \quad G \quad \mathbb{F}_q^n$ linear subspace
- Decode: $r = mG + e$ find closest $c = mG$
- $\mathrm{wt}_H(e) = |\{i : e_i = 0\}|$

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

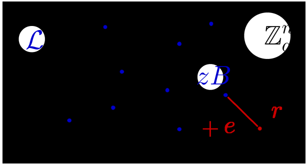
Public-key

→ Integer factorization

→ Discrete logarithm over F_p

→ Discrete logarithm over ell. curves

Lattice-based



- $L = \{\sum z_i b_i \mid z_i \in Z\} = \langle B \rangle \subset Z_q^n$
- SVP: $r = zB + e$ find closest zB
- $||e||_2 = \sqrt{\sum e_i^2}$, $||e||_\infty = \max\{|e_i|\}$

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

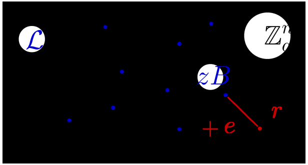
Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Lattice-based



- $L = \{\sum z_i b_i \mid z_i \in \mathbb{Z}\} = \langle B \rangle \subseteq \mathbb{Z}_q^n$
- SVP: $r = zB + e$ find closest zB
- $||e||_2 = \sqrt{\sum e_i^2}$, $||e||_\infty = \max\{|e_i|\}$

Post-quantum Cryptography

Asymmetric

Public-key

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Multivariate

- $P = (p_1, \ldots, p_m) \quad \mathbb{F}_q[x_1, \ldots, x_n]$
- Given $P(m) = c$ find m
- $P = S \circ F \circ T$, F quadr., S, T affine

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Multivariate

- $P = (p_1, \dots, p_m) \quad \mathbb{F}_q[x_1, \dots, x_n]$
- Given $P(m) = c$ find m
- $P = S \circ F \circ T$, F quadr., S, T affine

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Isogeny-based

- E, E ell. curves over $\mathbb{F}_q$
- find isogeny $\phi : E \to E$

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Quantum computer

Isogeny-based

- E, E ell. curves over $\mathbb{F}_q$
- find isogeny $\phi : E \to E$

Post-quantum Cryptography

Asymmetric

- RSA signature, encryption
- DH, DSA
- ECDH, ECDSA

Quantum computer

Public-key

→ Integer factorization

→ Discrete logarithm over $\mathbb{F}_p$

→ Discrete logarithm over ell. curves

Post-quantum crypto

Code-based

- Multivariate

Hash-based

- Lattice-based
- Isogeny-based

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures
	Standardized KEM:	KYBER
	4th round:	BIKE, Classic McEliece, HQC
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures
	Standardized KEM:	KYBER
	4th round:	BIKE, Classic McEliece, HQC
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+
	On ramp announcement	only signatures

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures
	Standardized KEM:	KYBER
	4th round:	BIKE, Classic McEliece, HQC
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+
	On ramp announcement	only signatures

lattice-based: need to outperform DILITHIUM, FALCON

non-lattice-based: need one advantage over SPHINCS+

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures
	Standardized KEM:	KYBER
	4th round:	BIKE, Classic McEliece, HQC
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+
	On ramp announcement	only signatures

lattice-based: need to outperform DILITHIUM, FALCON

non-lattice-based: need one advantage over SPHINCS+

necessary: EUF-CMA, attackers $\geq 2^{64}$ signatures, security levels breaking AES

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures
	Standardized KEM:	KYBER
	4th round:	BIKE, Classic McEliece, HQC
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+
	On ramp announcement	only signatures

lattice-based: need to outperform DILITHIUM, FALCON

non-lattice-based: need one advantage over SPHINCS+

necessary: EUF-CMA, attackers $\geq 2^{64}$ signatures, security levels breaking AES

Example: Level 1: AES-128: 2^{157} quantum / 2^{143} classical gates

Timeline

2016 — NIST standardization call — for post-quantum PKE/KEM and signatures

Standardized KEM: KYBER

4th round: BIKE, Classic McEliece, HQC

2022 — Standardized signatures: DILITHIUM, FALCON, SPHINCS+

On ramp announcement — only signatures

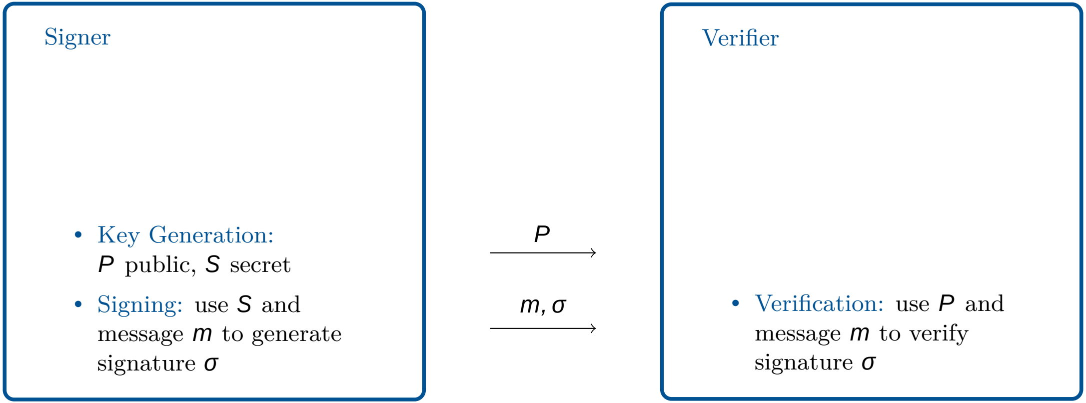
lattice-based: need to outperform DILITHIUM, FALCON

non-lattice-based: need one advantage over SPHINCS+

necessary: EUF-CMA, attackers $\geq 2^{64}$ signatures, security levels   breaking AES

nice to haves: side-channel resistant, BUFF, multi-key attacks, well-understood math

Idea of Signature Schemes



Idea of Signature Schemes

Signer

- Key Generation: P public, S secret
- Signing: use S and message m to generate signature σ

EUF-CMA

small P

small σ

Verifier

fast verification

- Verification: use P and message m to verify signature σ

Idea of Signature Schemes

Signer

- Key Generation:
  P public, S secret
- Signing: use S and message m to generate signature σ

EUF-CMA

small P

small σ

Verifier

fast verification

- Verification: use P and message m to verify signature σ

Approaches for signatures:

- Hash-and-Sign
- ZK Protocol
- ZK + MPC

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures
	Standardized KEM:	KYBER
	4th round:	BIKE, Classic McEliece, HQC
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+
	On ramp announcement	
2023	1st round candidates:	40 submissions

1st round Candidates

Code-based: 6

- CROSS
- Enhan. pqsigRM
- FuLeeca
- LESS
- MEDS
- Wave

Lattice-based: 7

- EagleSign
- EHT
- HAETAE
- Hawk
- HuFu
- Raccoon
- Squirrels

MPCitH: 7

- Biscuit
- MIRA
- MiRitH
- MQOM
- PERK
- RYDE
- SDitH

Other: 5

- ALTEQ
- eMLE-Sig
- KAZ-SIGN
- Preon
- Xifrat1-Sign.I

Isogeny: 1

- SQISign

Multivariate: 10

- 3wise
- DME-Sign
- HPPC
- MAYO
- PROV
- QRUOV
- SNOVA
- TUOV
- UOV
- VOX

Symmetric: 4

- AIMer
- Ascon-Sign
- FAEST
- SPHINCS

1st round Candidates

Code-based: 6

- CROSS
- Enhan. pqsigRM
- FuLeeca
- LESS
- MEDS
- Wave

Lattice-based: 7

- EagleSign
- EHT
- HAETAE
- Hawk
- HuFu
- Raccoon
- Squirrels

MPCitH: 7

- Biscuit
- MIRA
- MiRitH
- MQOM
- PERK
- RYDE
- SDitH

Other: 5

- ALTEQ
- eMLE-Sig
- KAZ-SIGN
- Preon
- Xifrat1-Sign.I

Isogeny: 1

- SQISign

Multivariate: 10

- 3wise
- DME-Sign
- HPPC
- MAYO
- PROV
- QRUOV
- SNOVA
- TUOV
- UOV
- VOX

Symmetric: 4

- AIMer
- Ascon-Sign
- FAEST
- SPHINCS

1st round Candidates

Code-based: 9

- CROSS
- LESS
- MEDS
- MIRA
- MiRitH
- PERK
- RYDE
- SDitH
- Wave

Other: 1

- Preon

Lattice-based: 5

- HAETAE
- Hawk
- HuFu
- Raccoon
- Squirrels

Symmetric: 4

- AIMer
- Ascon-Sign
- FAEST
- SPHINCS

Multivariate: 9

- Biscuit
- MAYO
- MQOM
- PROV
- QRUOV
- SNOVA
- TUOV
- UOV
- VOX

Isogeny: 1

- SQISign

Basics

Code $C \quad \mathbb{F}_q^n$ linear subspace

G generator matrix ! $c = mG$

Basics

Code C $\mathbb{F}_q^n$ linear subspace

H parity-check matrix ! $cH^{>} = 0$

Basics

Code $\mathcal{C} \subseteq \mathbb{F}_q^n$ linear subspace

H parity-check matrix $\rightarrow$ $rH^\top = eH^\top = s$

Hamming weight: $\mathrm{wt}_H(e) = |\{i \mid e_i \neq 0\}|$

Basics

algebraic structure

e.g. RS, Goppa codes

! e cient decoders

Basics

random code

decoding is NP-hard

! Information set decoding

Syndrome Decoding Problem (SDP)

Given H, s, weight t, nd e s.t.

1. $s = eH^{>}$
2. $wt_H(e) = t$

Basics

Code $C \subseteq \mathbb{F}_{q^m}^n$ linear subspace

H parity-check matrix $\rightarrow$ $rH^\top = eH^\top = s$

Rank weight: $\mathrm{wt}_R(e) = \dim(\langle e_1, \ldots, e_n \rangle_{\mathbb{F}_q})$

Rank SDP

Given H, s, weight t, find e s.t.

1. $s = eH^\top$
2. $\mathrm{wt}_R(e) = t$

$\mathrm{wt}_R(e) = \dim_{\mathbb{F}_q}(E)$

Basics

Code C $\mathbb{F}_q^{m}$ ^{n} linear subspace

$G_1; \ldots; G_k$! $C = \mathrm{P}\ {}_i G_i$;

Rank weight: $\mathrm{wt}_R(E) = \mathrm{rk}(E)$

MinRank

Given C $\mathbb{F}_q^{m}$ ^{n}, R, t, nd E s.t.

1. R $E \in C$
2. $\mathrm{rk}(E) = t$

basis of $\mathbb{F}_{q^m} / \mathbb{F}_q$: $\mathrm{wt}_R(e) = \mathrm{rk}((e))$ basis

Classical Approach: Hash and Sign

structured code

e cient decoding

random code

hard to decode

Idea McEliece: use Goppa code as secret code

trapdoor

encryption

messages

ciphertexts

Classical Approach: Hash and Sign

structured code

e cient decoding

random code

hard to decode

Idea McEliece: use Goppa code as secret code

trapdoor

signature

signatures

messages

Classical Approach: Hash and Sign

structured code

e cient decoding

random code

hard to decode

Idea McEliece: use Goppa code as secret code

trapdoor

Hash

signature

signatures

messages

Classical Approach: Hash and Sign

structured code

e cient decoding

random code

hard to decode

Idea McEliece: use Goppa code as secret code

trapdoor

Hash

signature

signatures

messages

repeat

Classical Approach: Hash and Sign

structured code

e cient decoding

random code

hard to decode

Idea McEliece: use Goppa code as secret code

trapdoor

Disadvantage: slow signing, large public key

Advantage: small signatures, fast veri cation

Classical Approach: Hash and Sign

structured code

e cient decoding

random code

hard to decode

Idea McEliece: use Goppa code as secret code

trapdoor

Disadvantage: slow signing, large public key

Advantage: small signatures, fast veri cation

Wave: $(u; u + v)$ ternary code and t large

Zero-Knowledge Protocol

Signature Scheme

Signer

Veri er

secret

public

$\longrightarrow$

Zero-Knowledge Protocol

ZK Protocol

Prover		Veri er
secret		public
	⟵	
Commitments c	Interaction	
	⟶	Challenge
		recover c X
Responser		

Zero-Knowledge Protocol

Signature Scheme

Signer

Veri er

secret

public

Commitments c

Fiat-Shamir

Challenge= Hash(m; c) $\longrightarrow$

X

Responser

Zero-Knowledge Protocol

Signature Scheme

Impersonator

Veri er

secret

public

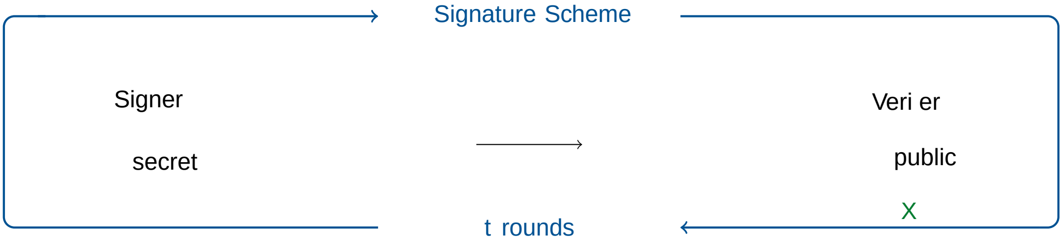
cheating prob.

Fiat-Shamir

$\longrightarrow$

X

Zero-Knowledge Protocol



Zero-Knowledge Protocol

ZK Protocol

Prover

secret

$\longleftarrow$

Interaction

$\longrightarrow$

Verifier

public

X

Isomorphism Problems Given O, O^0, find φ s.t. $\varphi(O) = O^0$

φ

O, O^0

$1: \varphi_1(O) = \tilde{O}$ X /

$2: \varphi_2(O^0) = \tilde{O}$ X

Zero-Knowledge Protocol

ZK Protocol

Prover

secret

Veri er

public

X

$\longleftarrow$

Interaction

$\longrightarrow$

Isomorphism Problems Given O, O^0, nd ' s.t. ' (O) = O^0

'

O, O^0

1: ' $_1$(O) = Ơ X /

2: ' $_2$(O^0) = Ơ X

! MEDS, LESS

Code Equivalence

Code equivalence

Given $G, G' \in \mathbb{F}_q^{k \times n}$ find isometry φ s.t.

$$\varphi(\langle G \rangle) = \langle G' \rangle$$

Hamming isometries $\varphi \in (\mathbb{F}_q^{\star})^n \rtimes S_n$

$\rightarrow$ LESS

Rank isometries $\varphi \in GL_m(\mathbb{F}_q) \times GL_n(\mathbb{F}_q)$

$\rightarrow$ MEDS

Disadvantages: medium/large public keys

Advantages: medium/small signatures

Zero-Knowledge Protocol

ZK Protocol

Prover		Veri er
	⟵	
	Interaction	
secret	⟶	public
		X

SDP Given H, s, t, nd e s.t. 1. $s = eH^{\top}$, 2. $\mathrm{wt}_H(e) = t$

Zero-Knowledge Protocol

ZK Protocol

Prover	Interaction	Veri er
secret		public
		X

SDP Given H, s, t, nd e s.t. 1. $s = eH^{>}$, 2. $\mathrm{wt}_H(e) = t$

e of $\mathrm{wt}_H(e) = t$

$H; s, t$

1. X / 2. X

Zero-Knowledge Protocol

ZK Protocol

Prover ⟵ Veri er

secret Interaction public

⟶

X

SDP	Given H, s, t, nd e s.t.	1. $s = eH^{>}$,	2. $wt_H(e) = t$

e of $wt_H(e) = t$

$H; s, t$

' : 1. X / ' (e): 2. X

Zero-Knowledge Protocol

ZK Protocol

Prover ⟵ Veri er

Interaction

secret ⟶ public

X

SDP	Given H , s, t, nd e s.t.	1. $s = eH^{>}$,	2. $wt_H(e) = t$

e of $wt_H(e) = t$

H; s , t

' : 1. X / ' (e): 2. X

1. Problem   cheating prob. $\frac{1}{2}$! many rounds

Zero-Knowledge Protocol

ZK Protocol

Prover ← Veri er

secret Interaction public

→ X

SDP Given H, s, t, nd e s.t. 1. $s = eH^{>}$, 2. $\mathrm{wt}_H(e) = t$

e of $\mathrm{wt}_H(e) = t$

$H; s$, t

' : 1. X / ' (e): 2. X

1. Problem cheating prob. $\frac{1}{2}$! many rounds

! Solution MPCitH: change protocol

MPC in-the-head

ZK Protocol

Prover		Verifier
secret S		public
$(N \quad 1)$-private MPC:	$c_i ; \quad _i$ $\longrightarrow$	
Split S into N shares: s_i	ℓ $\longleftarrow$	Challenge $\ell \in \{1, \ldots, N\}$
Commitments c_i for s_i	s_i for $i \notin \ell$ $\longrightarrow$	Check $c_i ; \quad _i$ for $i \notin \ell$ ✓
Broadcasts $\quad _i = f(s_i)$		

$(N \quad 1)$-private MPC: Secret S split into N shares s_i

$N \quad 1$ many s_i $\Rightarrow$ no info. on S — broadcasts $\quad _i$ to check validity of S

Example $e = \sum_{i=1}^{N} e^{(i)}$, $f(e^{(i)}) = e^{(i)} H^{\top} = s^{(i)}$ $\Rightarrow$ can check $\sum_{i=1}^{N} s^{(i)} = s$

MPC in-the-head

Prover | ZK Protocol | Verifier

secret S

public

$c_i ; \ {}_i$ $\longrightarrow$

ℓ $\longleftarrow$

s_i for $i \neq \ell$ $\longrightarrow$

Challenge $\ell \in \{1, \dots, N\}$

Check $c_i ; \ {}_i$ for $i \neq \ell$ X

MPC in-the-head

Prover

ZK Protocol

Verifier

secret S

public

$c_i ;\ \ _i$

ℓ

Challenge $\ell \in \{1, \dots, N\}$

s_i for $i \neq \ell$

Check $c_i ;\ \ _i$ for $i \neq \ell$ ✓

MPC in-the-head

Prover	ZK Protocol	Verifier
secret S		public
	c_i ; $\;_i$ →	
	← ℓ	Challenge $\ell \in \{1, \ldots, N\}$
	s_i for $i \neq \ell$ →	Check c_i ; $\;_i$ for $i \neq \ell$ ✓

→ New cheating probability: $1/N$

MPC in-the-head

ZK Protocol

Prover		Veri er
secret S		public
m-private MPC:	c_i ; $_i$ →	
Split S into N shares: s_i	← I	Challenge $|I| = m$
Commitments c_i for s_i	s_i for $i \in I$ →	Check c_i ; $_i$ for $i \in I$ ✓
Broadcasts $_i = f(s_i)$		

! New cheating probability: $1/\binom{N}{m}$

MPC in-the-head

ZK Protocol

Prover		Verifier
secret S		public
$(N - 1)$-private MPC:	$c_i; \ {}_i$ $\longrightarrow$	
Split S into N shares: s_i	ℓ $\longleftarrow$	Challenge $\ell \in \{1, \ldots, N\}$
Commitments c_i for s_i	s_i for $i \neq \ell$ $\longrightarrow$	Check $c_i; \ {}_i$ for $i \neq \ell$ ✓
Broadcasts ${}_i = f(s_i)$		

$\rightarrow$ New cheating probability: $1/N$

t/N rounds, but more computations

MPC in-the-head

ZK Protocol

Prover		Verifier
secret S		public
$(N-1)$-private MPC:	c_i, α_i →	
Split S into N shares: s_i	← ℓ	Challenge $\ell \in \{1, \dots, N\}$
Commitments c_i for s_i	s_i for $i \neq \ell$ →	Check c_i, α_i for $i \neq \ell$ ✓
Broadcasts $\alpha_i = f(s_i)$		

$\rightarrow$ New cheating probability: $1/N$

t/N rounds, but more computations

Disadvantages: slow

Advantages: small sizes

MPC in-the-head

ZK Protocol

Prover		Verifier
secret S		public
$(N - 1)$-private MPC:	$c_i, {}_i$ $\longrightarrow$	
Split S into N shares: s_i	ℓ $\longleftarrow$	Challenge $\ell \in \{1, \dots, N\}$
Commitments c_i for s_i	s_i for $i \neq \ell$ $\longrightarrow$	Check $c_i, {}_i$ for $i \neq \ell$ ✓
Broadcasts ${}_i = f(s_i)$		

$\to$ New cheating probability: $1/N$

t/N rounds, but more computations

Disadvantages: slow

Advantages: small sizes

Using rank SDP $\to$ RYDE

Using MinRank $\to$ MIRA, MiRitH

More novel problems

d-split SDP

Given H, s, t, find $(e_1; e_2)$ s.t.

1. $s = eH^\top$
2. $\mathrm{wt}_H(e_i) = t/2$

Subcode equivalence

Given $G \in \mathbb{F}_q^{k \times n}$; $G' \in \mathbb{F}_q^{k' \times n}$ find P s.t.

$$\langle GP \rangle \subseteq \langle G' \rangle$$

$\rightarrow$ SDitH

$\rightarrow$ PERK

More novel problems

d-split SDP

Given H, s, t, nd $(e_1; e_2)$ s.t.

1. $s = eH^{>}$
2. $wt_H(e_i) = t{=}2$

Permuted Kernel

Given G 2 $F_q^{k \quad n}$; H^0 2 $F_q^{n \quad k^0 \quad n}$ nd P s.t.

$$H^0(GP)^{>} = 0$$

! SDitH

! PERK

More novel problems

d-split SDP

Given H, s, t, nd $(e_1; e_2)$ s.t.

1. $s = eH^{>}$
2. $wt_H(e_i) = t=2$

Relaxed permuted kernel problem

Given G 2 $F_q^{k \; n}$; H^0 2 $F_q^{n \; k^0 \; n}$ nd x; P :

$$H^0(xGP)^{>} = 0$$

! SDitH

! PERK

Zero-Knowledge Protocol

SDP Given H, s, t, nd e s.t. 1. $s = eH^{>}$, 2. $\mathrm{wt}_H(e) = t$

e of $\mathrm{wt}_H(e) = t$

$H; s$, t

' : 1. X / ' (e): 2. X

Zero-Knowledge Protocol

SDP Given H, s, t, nd e s.t. 1. $s = eH^{>}$, 2. $\mathrm{wt}_H(e) = t$

e of $\mathrm{wt}_H(e) = t$

$H; s, t$

' : 1. X / ' (e): 2. X

2. Problem

1 round: large commun. cost

Zero-Knowledge Protocol

SDP Given H , s, t, nd e s.t. 1. $s = eH^{>}$, 2. $wt_H(e) = t$

e of $wt_H(e) = t$

H; s , t

' : 1. X / ' (e): 2. X

2. Problem

1 round: large commun. cost

$S = f\, wt_H(e) = t g$

' : S ! S linear, transitive

! j ' j large

' 2 $(F_q^?)^n$ o S_n

j' j t $\log_2(n(q \quad 1))$

Zero-Knowledge Protocol

SDP Given H, s, t, nd e s.t. 1. $s = eH^{>}$, 2. $\mathrm{wt}_H(e) = t$

e of $\mathrm{wt}_H(e) = t$

$H; s, t$

$' :$ 1. X / $'(e)$: 2. X

2. Problem

1 round: large commun. cost

$S = f\,\mathrm{wt}_H(e) = t g$

$'$: S ! S linear, transitive

! j ' j large

$' \; 2 \; (\mathbb{F}_q^?)^n$ o S_n

$j' j \quad t \log_2(n(q \quad 1))$

! Solution

change underlying problem

! CROSS

Hard Problems

Syndrome Decoding Problem Given p.c. matrix H, syndrome s, weight t, nd e s.t.

lin. constraint 1. $s = eH^{>}$ 2. $\mathrm{wt}_H(e) = t$ non-lin. constraint

Hard Problems

Restricted SDP (R-SDP) Given p.c. matrix H, syndrome s, restriction E, find e s.t.

lin. constraint 1. $s = eH^\top$ 2. $e \in E^n$ non-lin. constraint

$E = \{g^i \mid i \in \{1, \ldots, z\}\} < \mathbb{F}_q^*$ $g \in \mathbb{F}_q^*$ of prime order z

Hard Problems

Restricted SDP (R-SDP) Given p.c. matrix H, syndrome s, restriction E, find e s.t.

lin. constraint 1. $s = eH^\top$ 2. $e \in E^n$ non-lin. constraint

$E = \{g^i \mid i \in \{1, \ldots, z\}\} < \mathbb{F}_q^\star$ $g \in \mathbb{F}_q^\star$ of prime order z

$\mathbb{F}_q^\star$ $\mathbb{F}_q^\star$ $\mathbb{F}_q^\star$ $\rightarrow$ g^{i_1} g^{i_2} g^{i_n}

NP-hard

adaption of ISD: exponential cost

R-SDP

Benefits

restriction $\mathbb{E} = \{g^i \mid i \in \{1, \ldots, z\}\}$

rest. vectors $e = (g^{i_1}, \ldots, g^{i_n}) \in \mathbb{F}_q^n$

R-SDP

Bene ts

restriction $\textcolor{green}{E} = \{ g^i \mid i \in \{1, \ldots, z\}\}$	$\xrightarrow{\ell}$	exponents $\mathbb{F}_z^n$
rest. vectors $e = (g^{i_1}, \ldots, g^{i_n}) \in \mathbb{F}_q^n$		$\ell(e) = (i_1, \ldots, i_n)$

R-SDP

Benefits

restriction $E = \{g^i \mid i \in \{1, \dots, z\}\}$	$\xrightarrow{\ell}$	exponents $\mathbb{F}_z^n$
rest. vectors $e = (g^{i_1}, \dots, g^{i_n}) \in \mathbb{F}_q^n$		$\ell(e) = (i_1, \dots, i_n)$
secret space $S = E^n$; $\varphi : S \to S$	$\xrightarrow{\ell}$	$\lvert e \rvert = \lvert \varphi \rvert = n \log_2(z)$
$\varphi(e) = e' \star e$, $e' = (g^{j_1}, \dots, g^{j_n})$		

R-SDP

Benefits

restriction $E = \{g^i \mid i \in \{1,\dots,z\}\}$	$\xrightarrow{\ell}$	exponents $\mathbb{F}_z^n$
rest. vectors $e = (g^{i_1},\dots,g^{i_n}) \in \mathbb{F}_q^n$		$\ell(e) = (i_1,\dots,i_n)$
secret space $S = E^n$; $\varphi : S \to S$	$\xrightarrow{\ell}$	$\lvert e\rvert = \lvert\varphi\rvert = n\log_2(z)$
$\varphi(e) = e' \star e$; $e' = (g^{j_1},\dots,g^{j_n})$		$\ell(\varphi(e)) = \ell(e) + \ell(e')$

R-SDP

Benefits	restriction $E = \{g^i \mid i \in \{1, \dots, z\}\}$	$\xrightarrow{\ell}$	exponents $\mathbb{F}_z^n$
	rest. vectors $e = (g^{i_1}, \dots, g^{i_n}) \in \mathbb{F}_q^n$		$\ell(e) = (i_1, \dots, i_n)$
	secret space $S = E^n$; $\varphi : S \to S$	$\xrightarrow{\ell}$	$\lvert e \rvert = \lvert \varphi \rvert = n \log_2(z)$
	$\varphi(e) = e' \star e$; $e' = (g^{j_1}, \dots, g^{j_n})$		$\ell(\varphi(e)) = \ell(e) + \ell(e')$
Example	$E = \{1, 3, 9\} \subseteq \mathbb{F}_{13}$	$\xrightarrow{\ell}$	exponents in $\mathbb{F}_3^4$
	$e = (1, 9, 3, 3)$		$\ell(e) = (0, 2, 1, 1)$
	$\downarrow \star(3, 3, 9, 1)$		$\downarrow +(1, 1, 2, 0)$
	$\tilde{e} = (3, 1, 1, 3)$		$\ell(\tilde{e}) = (1, 0, 0, 1)$

R-SDP(G)

R-SDP Given H, s, E, nd e s.t. 1. $s = eH^{>}$ 2. $e \, 2 \, E^n$ $(E^n ; ?) \, ' \, (F_z^n ; +)$

R-SDP(G)

R-SDP(G) Given H, s, G, find e s.t. 1. $s = eH^{\top}$ 2. $e \in G$ $(G, \star) < (E^n, \star)$

Benefits

$$x_1 = (g^{i_1}, \ldots, g^{i_n})$$

$$\vdots$$

$$x_m = (g^{j_1}, \ldots, g^{j_n})$$

R-SDP(G)

R-SDP(G) Given H, s, G, nd e s.t. 1. $s = eH^{>}$ 2. $e \, 2 \, G$ $(G; ?) < (E^n; ?)$

Bene ts

$$x_1 = (g^{i_1}; : : : ; g^{i_n})$$
$$\vdots$$
$$x_m = (g^{j_1}; : : : ; g^{j_n})$$

$$\longrightarrow \quad M = \begin{pmatrix} i_1 & & i_n \\ \vdots & & \vdots \\ j_1 & & j_n \end{pmatrix} \, 2 \, F_z^{m \; n}$$

R-SDP(G)

R-SDP(G) Given H, s, G, find e s.t. 1. $s = eH^\top$ 2. $e \in G$ $\quad G \cong C \quad \mathbb{F}_z^n$

Benefits

$$
\begin{matrix}
x_1 = (g^{i_1}, \ldots, g^{i_n}) \\
\vdots \\
x_m = (g^{j_1}, \ldots, g^{j_n})
\end{matrix}
\quad \xrightarrow{\ \ell\ } \quad
M = \begin{pmatrix} i_1 & \cdots & i_n \\ \vdots & & \vdots \\ j_1 & \cdots & j_n \end{pmatrix} \in \mathbb{F}_z^{m \times n}
$$

$$
e = x_1^{u_1} \star \cdots \star x_m^{u_m} \qquad\qquad \ell(e) = (u_1, \ldots, u_m) M
$$

$$
\varphi : G \to G;\ \varphi(e) = e' \star e \quad \xrightarrow{\ \ell\ } \quad |e| = |\varphi| = m \log_2(z) < 1.5
$$

R-SDP(G)

R-SDP(G) Given H, s, G, find e s.t. 1. $s = eH^\top$ 2. $e \in G$ $\quad G \cong C \subseteq \mathbb{F}_z^n$

Benefits

$$x_1 = (g^{i_1}, \dots, g^{i_n}), \quad \vdots \quad x_m = (g^{j_1}, \dots, g^{j_n}) \xrightarrow{\ell} M = \begin{pmatrix} i_1 & \cdots & i_n \\ \vdots & & \vdots \\ j_1 & \cdots & j_n \end{pmatrix} \in \mathbb{F}_z^{m \times n}$$

$$e = x_1^{u_1} \star \cdots \star x_m^{u_m} \qquad \ell(e) = (u_1, \dots, u_m)M$$

$$\varphi : G \to G, \ \varphi(e) = e' \star e \xrightarrow{\ell} |E| = |\varphi| = m \log_2(z) < 1.5$$

Example

$$E = \{1, 3, 9\} \subseteq \mathbb{F}_{13} \xrightarrow{\ell} \text{exponents in } \mathbb{F}_3^4$$

$$x_1 = (3, 1, 1, 3) \qquad M = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 2 & 0 \end{pmatrix}$$

$$x_2 = (1, 3, 9, 1)$$

$$e = x_1^{2} \star x_2^{1} = (9, 3, 9, 9) \qquad \ell(e) = (2, 1, 2, 2) = (2, 1)M$$

Summary

Hash & Sign

Large weight SDP	→	WAVE	large public key

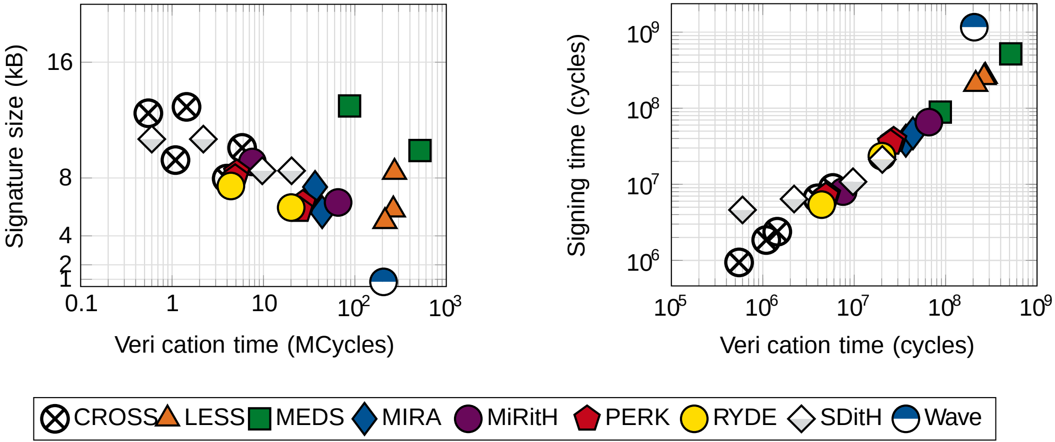
ZK Protocol

Restricted SDP	→	CROSS	
CEP	→	LESS	
Matrix CEP	→	MEDS	large signature

ZK + MPC

d-split SDP	→	SDitH	
Rank SDP	→	RYDE	
MinRank	→	MIRA/MiRitH	
PKP	→	PERK	slow

Comparison



Timings taken from https://pqshield.github.io/nist-sigs-zoo/

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures	
	Standardized KEM:	KYBER	
	4th round:	BIKE, Classic McEliece, HQC	
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+	
2023	On ramp announcement		
	1st round candidates:	29 survivors	9 code-based
2024			

Timeline

2016	NIST standardization call	for post-quantum PKE/KEM and signatures	
	Standardized KEM:	KYBER	
	4th round:	BIKE, Classic McEliece, HQC	
2022	Standardized signatures:	DILITHIUM, FALCON, SPHINCS+	
2023	On ramp announcement		
	1st round candidates:	29 survivors	9 code-based
2024	2nd round announced	14 schemes	6 code-based

2nd Round Candidates

Code-based: 9

- CROSS
- LESS
- MEDS
- MIRA
- MiRitH
- PERK
- RYDE
- SDitH
- Wave

Other: 1

- Preon

Lattice-based: 5

- HAETAE
- Hawk
- HuFu
- Raccoon
- Squirrels

Symmetric: 4

- AIMer
- Ascon-Sign
- FAEST
- SPHINCS

Multivariate: 9

- Biscuit
- MAYO
- MQOM
- PROV
- QRUOV
- SNOVA
- TUOV
- UOV
- VOX

Isogeny: 1

- SQISign

2nd Round Candidates

Code-based: 6

- CROSS
- LESS
- MEDS
- MiRatH
- PERK
- RYDE
- SDitH
- Wave

Other: 0

- Preon

Lattice-based: 1

- HAETAE
- Hawk
- HuFu
- Raccoon
- Squirrels

Symmetric: 1

- AIMer
- Ascon-Sign
- FAEST
- SHPINCS

Multivariate: 5

- Biscuit
- MAYO
- MQOM
- PROV
- QRUOV
- SNOVA
- TUOV
- UOV
- VOX

Isogeny: 1

- SQISign

2nd Round Candidates

NIST.IR.8528 Status report

1) security   2) cost and performance   3) implementation

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

2nd Round Candidates

NIST.IR.8528 Status report

1) security 2) cost and performance 3) implementation

a) simplicity b) uniqueness c) elegance

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

2nd Round Candidates

NIST.IR.8528 Status report

1) security 2) cost and performance 3) implementation

a) simplicity b) uniqueness c) elegance

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

non-lattice, better performance than SPHINCS

new, improve performance

2nd Round Candidates

NIST.IR.8528 Status report

1) security 2) cost and performance 3) implementation

a) simplicity b) uniqueness c) elegance

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

non-lattice, better performance than SPHINCS

new, improve performance: threshold, VOLE

2nd Round Candidates

NIST.IR.8528 Status report

1) security 2) cost and performance 3) implementation

a) simplicity b) uniqueness c) elegance

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

non-lattice, better performance than SPHINCS

complex, technical

2nd Round Candidates

NIST.IR.8528 Status report

1) security 2) cost and performance 3) implementation

a) simplicity b) uniqueness c) elegance

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

no oating points

new

2nd Round Candidates

NIST.IR.8528 Status report

1) security 2) cost and performance 3) implementation

a) simplicity b) uniqueness c) elegance

Code-based: 6

- CROSS
- LESS
- MiRatH
- PERK
- RYDE
- SDitH

Lattice-based: 1

- Hawk

Symmetric: 1

- FAEST

Isogeny: 1

- SQISign

Multivariate: 5

- MAYO
- MQOM
- QRUOV
- SNOVA
- UOV

non-lattice, better performance than SPHINCS

new, recent attacks

How will the 2nd round go?

Timeline

- Submission deadline: Jan. 17
- 3rd round decision?
- How many schemes?

How will the 2nd round go?

Timeline

Submission deadline: Jan. 17

3rd round decision? 2026

How many schemes? nal?

How will the 2nd round go?

Timeline

- Submission deadline: Jan. 17
- 3rd round decision? 2026
- How many schemes? nal?

What's next?

- Will MPC ! VOLE?
- Will SQISign reduce times?
- New attacks?

How will the 2nd round go?

Timeline

- Submission deadline: Jan. 17
- 3rd round decision? 2026
- How many schemes? nal?

What's next?

- Will MPC ! VOLE?
- Will SQISign reduce times?
- New attacks?

Open Problems

- Cost of d-split SDP
- Cost of restricted SDP
- Cost of rank SDP
- Cost of q-ary SDP

How will the 2nd round go?

Timeline

- Submission deadline: Jan. 17
- 3rd round decision? 2026
- How many schemes? final?

What's next?

- Will MPC $\rightarrow$ VOLE?
- Will SQISign reduce times?
- New attacks?

Open Problems

- Cost of d-split SDP
- Cost of restricted SDP
- Cost of rank SDP
- Cost of q-ary SDP
- How hard is code equivalence?

Abhi's talk!

How will the 2nd round go?

Timeline

- Submission deadline: Jan. 17
- 3rd round decision? 2026
- How many schemes? final?

What's next?

- Will MPC → VOLE?
- Will SQISign reduce times?
- New attacks?

Open Problems

- Cost of d-split SDP
- Cost of restricted SDP
- Cost of rank SDP
- Cost of q-ary SDP
- How hard is code equivalence?

Slides

Stay tuned!

Thank you

VOLE

Vector Oblivious Linear Transfer

ZK Protocol

Prover	Verifier
secret s	public
v random	Δ eval. point
$f(x) = sx + v$	
	$q = f(\Delta)$

VOLE

Vector Oblivious Linear Transfer

ZK Protocol

Prover			Verifier
	$\xleftarrow{s, v}$	$\xleftarrow{\Delta}$	
secret s			public
v random	$\xleftarrow{\Delta}$	$\xrightarrow{\text{seed}_i}$	Δ eval. point
$f(x) = sx + v$			seed_i for $i = \Delta$
$q = f(\Delta)$			$q = f(\Delta)$

VOLE correlation $q = s\Delta + v = f(\Delta)$

dishonest prover needs to guess Δ before committing to GGM tree: $\mathsf{P} \quad 1/p$

VOLE

Vector Oblivious Linear Transfer

ZK Protocol

Prover			Verifier
	s, v $\longleftarrow$	Δ $\longleftarrow$	
secret s			public
v random	Δ $\longleftarrow$	seed_i $\longrightarrow$	Δ eval. point
$f(x) = sx + v$			seed_i for $i = \Delta$
$q = f(\Delta)$			$q = f(\Delta)$

MPC

$s = \square\, s_i$ MPC $\longleftarrow$ $N - 1$ views

VOLE

$s = \square\, s_i$ GGM $\overset{\Delta}{\longleftarrow}$ $N - 1$ seeds

$v = \square\, i s_i$ $q = \square\, s_i(\Delta - i) = s\Delta + v$

VOLE

Vector Oblivious Linear Transfer

ZK Protocol

Prover				Verifier
	s, v $\longleftarrow$		Δ $\longleftarrow$	
secret s				public
v random	Δ $\longleftarrow$		seed_i $\longrightarrow$	Δ eval. point
$f(x) = sx + v$				seed_i for $i = \Delta$
$q = f(\Delta)$				$q = f(\Delta)$

$$f(x) = \sum_{i=0}^{d} f_i x^i,$$

$$s = f_d$$

$$f_1(x), f_2(x)$$

$$f_1(\Delta) + f_2(\Delta) = (f_1 + f_2)(\Delta)$$

$$f_1(\Delta) f_2(\Delta) = (f_1 f_2)(\Delta)$$

VOLE

Vector Oblivious Linear Transfer

ZK Protocol

Prover				Verifier
	$\xleftarrow{s, v}$		$\xleftarrow{\Delta}$	
secret s				public
v random	$\xleftarrow{\Delta}$		$\xrightarrow{\mathsf{seed}_i}$	Δ eval. point
$f(x) = sx + v$				seed_i for $i = \Delta$
$q = f(\Delta)$				$q = f(\Delta)$

Disadvantages: slow

Advantages: small sizes

Main Features

Implementation

- optimized AVX2 — fast < 1 MCycle (NIST cat. I)
- memory-optimized — fits on Cortex-M4 microcontroller
- constant worst-case runtime — no signature rejection
- available on lib open quantum safe

Ingredients

- Restricted Syndrome Decoding
  - → compact objects & efficient arithmetic
  - → NP-hard problem
- Zero-Knowledge protocol
  - → simple and well-studied
  - → EUF-CMA security
  - → BUFF security
  - → standard optimizations

Future of CROSS

What's next?

- Hardware implementation
- Side-channel protection
- Worst-case to average-case reduction
- Smaller signatures: VOLE

Website

CROSS

Codes & Restricted Objects Signature Scheme
http://cross-crypto.com/

Attacks

- E, G have multiplicative structure

  $e = (g^{i_1}, \ldots, g^{i_n})$

- $s = eH^\top$ has additive structure

  $s_j = \sum_{\ell=1}^{n} h_{j,\ell} g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$

Attacks

- E, G have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$
- Take E with no additive structure

- $s = eH^{\top}$ has additive structure
  $s_j = \sum_{\ell=1}^{n} h_{j,\ell}\, g^{i_\ell}$ for $j \in \{1, \ldots, n-k\}$

Attacks

- E, G have multiplicative structure
  $e = (g^{i_1}, \ldots, g^{i_n})$
- $s = eH$ has additive structure
  $s_j = \sum_{i=1}^{n} h_{j,i} g^{i}$ for $j \in \{1, \ldots, n-k\}$
- Take E with no additive structure
- good: $q = 13, g = 3, \mathsf{E} = \{1, 3, 9\}$
- bad: $q = 13, g = 5, \mathsf{E} = \{1, 5, -1, -5\}$

Attacks

- E, G have multiplicative structure

  $e = (g^{i_1}, \dots, g^{i_n})$

- $s = eH^\top$ has additive structure

  $s_j = \sum_{i=1}^{n} h_{j,i}\, g^{i}$ for $j \in \{1, \dots, n-k\}$

- Take E with no additive structure

- good: $q = 13, g = 3, \mathsf{E} = \{1, 3, 9\}$

- bad: $q = 13, g = 5, \mathsf{E} = \{1, 5, -1, -5\}$

- combinatorial:

  ISD algorithms

S. Bitzer, A. Pavoni, V. Weger, P. Santini, M. Baldi, and A. Wachter-Zeh. "Generic Decoding of Restricted Errors", ISIT, 2023.

M. Baldi, S. Bitzer, A. Pavoni, P. Santini, A. Wachter-Zeh, and V. Weger. "Zero knowledge protocols and signatures from the restricted syndrome decoding problem", PKC, 2024.

Attacks

- E, G have multiplicative structure

  $e = (g^{i_1}, \dots, g^{i_n})$

- $s = eH$ has additive structure

  $s_j = \sum_{=1}^{n} h_{j,} g^{i}$ for $j \in \{1, \dots, n-k\}$

- Take E with no additive structure

- good: $q = 13, g = 3, \mathsf{E} = \{1, 3, 9\}$

- bad: $q = 13, g = 5, \mathsf{E} = \{1, 5, -1, -5\}$

- combinatorial:

  ISD algorithms

- algebraic attacks:

  $e_i^z = 1$ Gröbner basis

S. Bitzer, A. Pavoni, V. Weger, P. Santini, M. Baldi, and A. Wachter-Zeh. "Generic Decoding of Restricted Errors", ISIT, 2023.

M. Baldi, S. Bitzer, A. Pavoni, P. Santini, A. Wachter-Zeh, and V. Weger. "Zero knowledge protocols and signatures from the restricted syndrome decoding problem", PKC, 2024.

M. Baldi, et al. "CROSS", NIST PQC round 1, 2023.

W. Beullens, P. Briaud, M. Øygarden. "A Security Analysis of Restricted Syndrome Decoding Problems", 2024.

Performance

NIST cat. I

Problem	q, z	**Type**	(n, k, m)	**rounds**	**|Sign.|** (kB)	**Sign** (MCycles)	**Verify** (MCycles)
R-SDP	$(127, 7)$	fast	$(127, 76, -)$	163	19.1	1.28	0.78
		balanced		252	12.9	2.38	1.44
		short		960	10.1	8.96	5.84
R-SDP(G)	$(509, 127)$	fast	$(55, 36, 25)$	153	12.5	0.94	0.55
		balanced		243	9.2	1.85	1.09
		short		871	7.9	6.54	3.96

private and public keys < 0.1 kB   key gen. < 0.1 MCycle

Measurements collected on an AMD Ryzen 5 Pro 3500U, clocked at 2.1GHz. The computer was running Debian GNU/Linux 12

CVE

PROVER		VERIFIER
KEY GENERATION		
Choose e with $\mathrm{wt}_H(e) = t$		
H parity-check matrix		
Compute $s = eH$	$\xrightarrow{P=(H,s,t)}$	
		VERIFICATION
Choose u $\mathbb{F}_q^n$, ϕ M_n		
Set $c_1 = \mathrm{Hash}(\phi, uH \quad)$		
Set $c_2 = \mathrm{Hash}(\phi(u), \phi(e))$	$\xrightarrow{c_1, c_2}$	
	$\xleftarrow{z}$	Choose z $\mathbb{F}_q^\times$
Set $y = \phi(u + ze)$	$\xrightarrow{y}$	
$r_1 = \phi$	$\xleftarrow{b}$	Choose b $\{1, 2\}$
$r_2 = \phi(e)$	$\xrightarrow{r_b}$	$b = 1$: $c_1 = \mathrm{Hash}(\phi, \phi^{-1}(y)H \quad - zs)$
		$b = 2$: $\mathrm{wt}_H(\phi(e)) = t$
		and $c_2 = \mathrm{Hash}(y - z\phi(e), \phi(e))$

CVE

PROVER		VERIFIER
KEY GENERATION		
Choose e with $\text{wt}_H(e) = t$		
H parity-check matrix		
Compute $s = eH$	$\xrightarrow{P=(H,s,t)}$	
		VERIFICATION
Choose $u \quad \mathbb{F}_q^n$, $\phi \quad M_n$		
Set $c_1 = \text{Hash}(\phi, uH \quad)$		
Set $c_2 = \text{Hash}(\phi(u), \phi(e))$	$\xrightarrow{c_1, c_2}$	
	$\xleftarrow{z}$	Choose $z \quad \mathbb{F}_q^\times$
Set $y = \phi(u + ze)$	$\xrightarrow{y}$	
$r_1 = \phi$	$\xleftarrow{b}$	Choose $b \quad \{1, 2\}$
$r_2 = \phi(e)$	$\xrightarrow{r_b}$	$b = 1$: $c_1 = \text{Hash}(\phi, \phi^{-1}(y)H \quad - zs)$
		$b = 2$: $\text{wt}_H(\phi(e)) = t$
		and $c_2 = \text{Hash}(y - z\phi(e), \phi(e))$

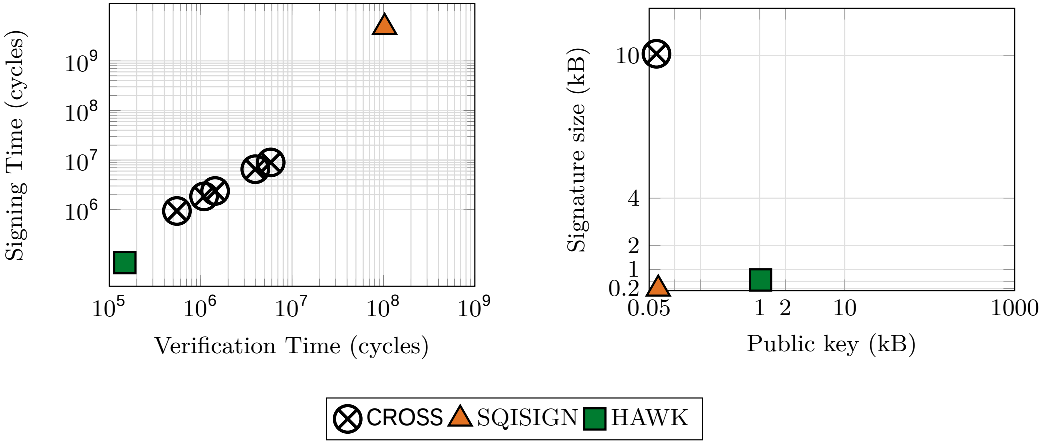
Recall SDP: (1) $s = eH$ (2) $\text{wt}_H(e) = t$

CVE

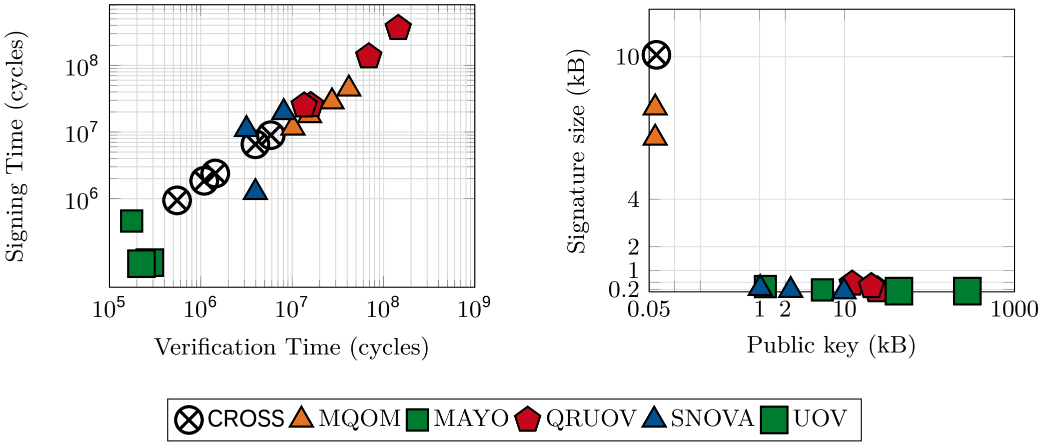
PROVER		VERIFIER
KEY GENERATION		
Choose e with $\mathrm{wt}_H(e) = t$		
H parity-check matrix		
Compute $s = eH$	$\xrightarrow{P=(H,s,t)}$	
		VERIFICATION
Choose $u \quad \mathbb{F}_q^n,\ \phi \quad M_n$		
Set $c_1 = \mathrm{Hash}(\phi, uH \quad)$		
Set $c_2 = \mathrm{Hash}(\phi(u), \phi(e))$	$\xrightarrow{c_1, c_2}$	
	$\xleftarrow{z}$	Choose $z \quad \mathbb{F}_q^\times$
Set $y = \phi(u + ze)$	$\xrightarrow{y}$	
$r_1 = \phi$	$\xleftarrow{b}$	Choose $b \quad \{1, 2\}$
$r_2 = \phi(e)$	$\xrightarrow{r_b}$	$b = 1$: $c_1 = \mathrm{Hash}(\phi, \phi^{-1}(y)H \quad - zs)$
		$b = 2$: $\mathrm{wt}_H(\phi(e)) = t$
		and $c_2 = \mathrm{Hash}(y - z\phi(e), \phi(e))$

Problem: big signature sizes

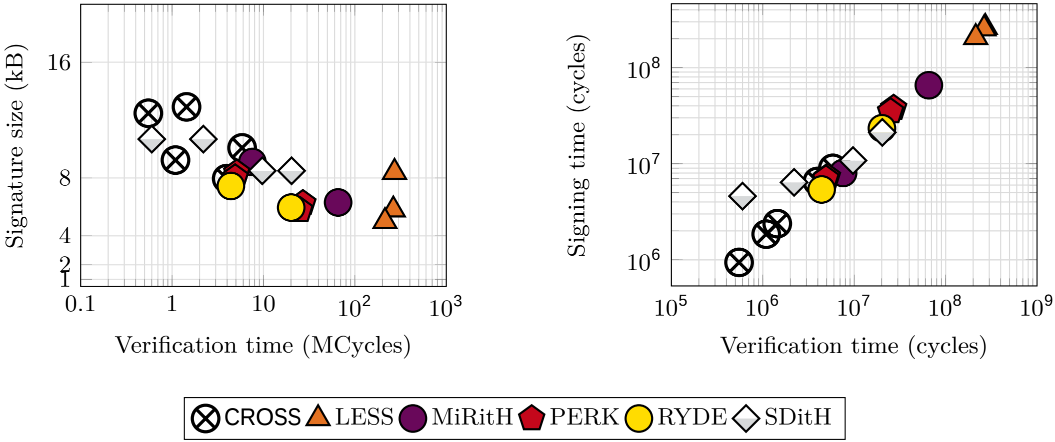
vs: Isogenies and lattices



vs: Multivariate



Comparison



Signature size (kB)
16
8
4
2
1
0.1
1
10
10^2
10^3
Verification time (MCycles)
Signing time (cycles)
10^8
10^7
10^6
10^5
10^6
10^7
10^8
10^9
Verification time (cycles)
CROSS
LESS
MiRitH
PERK
RYDE
SDitH