Recent Advances in Code-based Signatures

Violetta Weger

Rudolf Mößbauer Tenure Track Professorship: Symposium ”Selected Topics in Science and Technology”

March 22, 2023
Outline

1. Code-based Cryptography
   • Introduction to Coding Theory
   • Hard Problems from Coding Theory
   • Previous Work

2. Code-based Signature Schemes
   • Idea and Previous Work
   • FuLeeca
   • Restricted Errors

3. Future Research
   • Rank-metric Decoding
   • Quantum Codes
   • Further Research Directions
Outline

1. Code-based Cryptography
   • Introduction to Coding Theory
   • Hard Problems from Coding Theory
   • Previous Work

2. Code-based Signature Schemes
   • Idea and Previous Work
   • FuLeeca
   • Restricted Errors

3. Future Research
   • Rank-metric Decoding
   • Quantum Codes
   • Further Research Directions
Motivation

- Quantum computers: break all currently used asymmetric cryptosystems
  → Need quantum-secure alternatives
- Candidates for post-quantum cryptography: Systems based NP-hard problems
Motivation

- Quantum computers: break all currently used asymmetric cryptosystems
  \[\rightarrow\] Need quantum-secure alternatives
- Candidates for post-quantum cryptography: Systems based NP-hard problems

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

- Quantum computers: break all currently used asymmetric cryptosystems
  → Need quantum-secure alternatives
- Candidates for post-quantum cryptography: Systems based NP-hard problems

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

- PKE/KEM: 1 lattice-based, round 4: 3 code-based
- Signature schemes: 1 hash-based and 2 based on ideal lattices
Motivation

- Quantum computers: break all currently used asymmetric cryptosystems
  → Need quantum-secure alternatives
- Candidates for post-quantum cryptography: Systems based NP-hard problems

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

- PKE/KEM: 1 lattice-based, round 4: 3 code-based
- Signature schemes: 1 hash-based and 2 based on ideal lattices

2022 NIST reopened standardization call for signature schemes
Coding Theory

Set Up

- Code $C \subseteq \mathbb{F}_q^n$ linear $k$-dimensional subspace
- $c \in C$ codeword
- $G \in \mathbb{F}_q^{k \times n}$ generator matrix
  \[ C = \{ xG \mid x \in \mathbb{F}_q^k \} \]
- $H \in \mathbb{F}_q^{(n-k) \times n}$ parity-check matrix
  \[ C = \{ c \in \mathbb{F}_q^n \mid cH^\top = 0 \} \]
- $s = eH^\top$ syndrome
Set Up

- Code $C \subseteq \mathbb{F}_q^n$ linear $k$-dimensional subspace
- $c \in C$ codeword
- $G \in \mathbb{F}_q^{k \times n}$ generator matrix
  \[ C = \{ xG \mid x \in \mathbb{F}_q^k \} \]
- $H \in \mathbb{F}_q^{(n-k) \times n}$ parity-check matrix
  \[ C = \{ c \in \mathbb{F}_q^n \mid cH^\top = 0 \} \]
- $s = eH^\top$ syndrome
- Decode: find closest codeword
Coded Theory

**Set Up**

- **Code** $\mathcal{C} \subseteq \mathbb{F}_q^n$ linear $k$-dimensional subspace
- $c \in \mathcal{C}$ codeword
- $G \in \mathbb{F}_q^{k \times n}$ generator matrix
  \[ \mathcal{C} = \{ xG \mid x \in \mathbb{F}_q^k \} \]
- $H \in \mathbb{F}_q^{(n-k) \times n}$ parity-check matrix
  \[ \mathcal{C} = \{ c \in \mathbb{F}_q^n \mid cH^\top = 0 \} \]
- $s = eH^\top$ syndrome
- **Decode**: find closest codeword
- **Hamming metric**: For $x, y \in \mathbb{F}_q^n$
  \[ d_H(x, y) = | \{ i \mid x_i \neq y_i \} | \]
## Set Up

- **Code** $\mathcal{C} \subseteq \mathbb{F}_q^n$ linear $k$-dimensional subspace
- $c \in \mathcal{C}$ codeword
- $G \in \mathbb{F}_q^{k \times n}$ generator matrix
  
  $\mathcal{C} = \{xG \mid x \in \mathbb{F}_q^k\}$

- $H \in \mathbb{F}_q^{(n-k) \times n}$ parity-check matrix
  
  $\mathcal{C} = \{c \in \mathbb{F}_q^n \mid cH^\top = 0\}$

- $s = eH^\top$ syndrome

- **Decode**: find closest codeword

- **Hamming metric**: For $x, y \in \mathbb{F}_q^n$
  
  $d_H(x, y) = |\{i \mid x_i \neq y_i\}|$

- **minimum distance of a code**:
  
  $d(\mathcal{C}) = \min\{d_H(x, y) \mid x \neq y \in \mathcal{C}\}$
Coding Theory

Set Up

- Code \( \mathcal{C} \subseteq \mathbb{F}_q^n \) linear \( k \)-dimensional subspace
- \( c \in \mathcal{C} \) codeword
- \( G \in \mathbb{F}_q^{k \times n} \) generator matrix
  \[ \mathcal{C} = \{ xG \mid x \in \mathbb{F}_q^k \} \]
- \( H \in \mathbb{F}_q^{(n-k) \times n} \) parity-check matrix
  \[ \mathcal{C} = \{ c \in \mathbb{F}_q^n \mid cH^\top = 0 \} \]
- \( s = eH^\top \) syndrome
- Decode: find closest codeword
- Hamming metric: For \( x, y \in \mathbb{F}_q^n \)
  \[ d_H(x, y) = | \{ i \mid x_i \neq y_i \} | \]
- minimum distance of a code:
  \[ d(C) = \min \{ d_H(x, y) \mid x \neq y \in \mathcal{C} \} \]
- error-correction capacity: \( t = (d(C) - 1)/2 \)
Hard Problems from Coding Theory

Algebraic structure
(Reed-Solomon, Goppa, …)
→ efficient decoders

random code

→ how hard to decode?


A. Becker, A. Joux, A. May, A. Meurer “Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding”, Eurocrypt, 2012.
Hard Problems from Coding Theory

Algebraic structure
(Reed-Solomon, Goppa, ... )
→ efficient decoders

random code

• Decoding random linear code is NP-hard

Hard Problems from Coding Theory

Algebraic structure
(Reed-Solomon, Goppa, ... )\[ \rightarrow \text{efficient decoders} \]
\[ \langle G \rangle \rightarrow \varphi \rightarrow \langle \tilde{G} \rangle \rightarrow \text{Seemingly random code} \]

- Decoding random linear code is NP-hard
- First code-based cryptosystem based on this problem


Hard Problems from Coding Theory

Algebraic structure (Reed-Solomon, Goppa, ...)
→ efficient decoders

scrambling

Semingly random code
→ how hard to decode?

- Decoding random linear code is NP-hard
- First code-based cryptosystem based on this problem
- Fastest solvers: ISD, exponential time


A. Becker, A. Joux, A. May, A. Meurer “Decoding random binary linear codes in $2^{n/20}$: How $1+1=0$ improves information set decoding”, Eurocrypt, 2012.
Previous Work

Lee Metric

For $x, y \in \mathbb{Z}/p^s\mathbb{Z}^n$

- **Lee weight:**
  \[
  wt_L(x) = \sum_{i=1}^{n} wt_L(x_i) = \sum_{i=1}^{n} \min\{x_i, |p^s - x_i|\}
  \]

- **Lee distance:**
  \[
  d_L(x, y) = wt_L(x - y).
  \]

$\rightarrow$ $d_L(C)$ much larger than $d_H(C)$
Previous Work

Lee Metric

For \( x, y \in \mathbb{Z}/p^s\mathbb{Z}^n \)

- **Lee weight:**
  \[
  \text{wt}_L(x) = \sum_{i=1}^{n} \text{wt}_L(x_i) = \sum_{i=1}^{n} \min\{x_i, |p^s - x_i|\}
  \]
- **Lee distance:**
  \[
  d_L(x, y) = \text{wt}_L(x - y).
  \]

\( d_L(C) \) much larger than \( d_H(C) \)

- Decoding random linear code in Lee-metric is NP-hard
- Fastest solvers: Lee-metric ISD, exponential time
- Behaviour of random ring-linear codes


Outline

1. Code-based Cryptography
   • Introduction to Coding Theory
   • Hard Problems from Coding Theory
   • Previous Work

2. Code-based Signature Schemes
   • Idea and Previous Work
   • FuLeeca
   • Restricted Errors

3. Future Research
   • Rank-metric Decoding
   • Quantum Codes
   • Further Research Directions
Idea of Signature Schemes

**Signer**

- **Key Generation**
  - Secret key $S$, public key $P$

- **Signing**
  - Message $m$, signature $\sigma$

**Verifier**

- **Verification**
  - Verify $\sigma$

Two approaches to get a code-based signature scheme:
- **Hash-and-sign** → large public key sizes → our solution: FuLeeca
- **Through ZK protocol** → large signature sizes → our solution: restricted errors
Idea of Signature Schemes

Signer

Key Generation

Secret key $S$, public key $P$

Signing

Message $m$, signature $\sigma$

Verifier

Two approaches to get a code-based signature scheme:

- Hash-and-sign
- Through ZK protocol
Idea of Signature Schemes

**Signer**

**Key Generation**
- Secret key $S$, public key $P$

**Signing**
- Message $m$, signature $\sigma$

**Verifier**

$P \rightarrow m, \sigma \rightarrow$ Verify $\sigma$

Two approaches to get a code-based signature scheme:

- **Hash-and-sign**
  - large public key sizes
  - our solution: FuLeeca

- **Through ZK protocol**
  - large signature sizes
  - our solution: restricted errors
Idea of Signature Schemes

**Signer**

Key Generation

- Secret key \( S \), public key \( P \)

Signing

- Message \( m \), signature \( \sigma \)

**Verifier**

\[ P \]

\[ m, \sigma \]

Verification

- Verify \( \sigma \)

Two approaches to get a code-based signature scheme:

- **Hash-and-sign**
  - large public key sizes
  - our solution: FuLeeca

- **Through ZK protocol**
  - large signature sizes
  - our solution: restricted errors
Hash-and-Sign


- Following idea of McEliece:
  - start with structured code
  - publish scrambled code
  - Hash($m$) = $eH^T$, $wt_H(e) \leq t$
  - Signature is scrambled $e$

  → large public key sizes
  → slow signing

How to reduce public key sizes/thwart statistical attacks?

How to speed-up signing?
Hash-and-Sign


- Following idea of McEliece:
  - start with structured code
  - publish scrambled code
- Hash\(m\) = \(eH^T\), \(wt_H(e) \leq t\)
- Signature is scrambled \(e\)

Reduce key sizes:
- use quasi-cyclic codes
- use low density generators

→ large public key sizes
→ slow signing
→ statistical attacks
Hash-and-Sign


- Following idea of McEliece:
  → start with structured code
  → publish scrambled code
  → large public key sizes
- Hash(m) = eHᵀ, wt_H(e) ≤ t
  → slow signing
- Signature is scrambled e
- Reduce key sizes:
  → use quasi-cyclic codes
  → statistical attacks
  → use low density generators

How to reduce public key sizes/ thwart statistical attacks?
How to speed-up signing?
<table>
<thead>
<tr>
<th>Secret key</th>
<th>Quasi-cyclic, low Lee weight generators</th>
</tr>
</thead>
<tbody>
<tr>
<td>Public key</td>
<td>Systematic form, scrambled generator matrix</td>
</tr>
<tr>
<td>Signature</td>
<td>Codeword $\sigma$ with low Lee weight and full Hamming weight, $\sigma$ and $\text{Hash}(m)$ have many signs matching</td>
</tr>
</tbody>
</table>

---

Violetta Weger — Recent Advances in Code-based Signatures
Secret key
Quasi-cyclic, low Lee weight generators

Public key
Systematic form, scrambled generator matrix

Signature
Codeword \( \sigma \) with low Lee weight and full Hamming weight, \( \sigma \) and Hash(\(m\)) have many signs matching

<table>
<thead>
<tr>
<th></th>
<th>public key size</th>
<th>signature size</th>
<th>total size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Falcon</td>
<td>897 B</td>
<td>666 B</td>
<td>1563 B</td>
</tr>
<tr>
<td>Dilithium</td>
<td>1312 B</td>
<td>2420 B</td>
<td>3732 B</td>
</tr>
<tr>
<td>Sphincs+</td>
<td>32 B</td>
<td>7856 B</td>
<td>7888 B</td>
</tr>
<tr>
<td>FuLeeca</td>
<td>389 B</td>
<td>276 B</td>
<td>665 B</td>
</tr>
</tbody>
</table>

\(\rightarrow\) Can beat all standardized signature schemes in total size
Code-based ZK Protocols

ZK protocol $\xrightarrow{\text{Fiat-Shamir}}$ Signature scheme

 Syndrome Decoding Problem

Given parity-check matrix $H$, syndrome $s$, weight $t$, find $e$ s.t.

1. $s = eH^\top$
2. $\text{wt}_H(e) \leq t$


- Random $H, e$ of weight $t$, compute $s = eH^\top \rightarrow$ small public key sizes
- Verifier challenges either 1. or 2. by asking for transformation $\varphi$ or transformed secret $\varphi(e)$
Code-based ZK Protocols

- **ZK protocol** \[\xRightarrow{\text{Fiat-Shamir}}\] Signature scheme

### Syndrome Decoding Problem

Given parity-check matrix $H$, syndrome $s$, weight $t$, find $e$ s.t.

1. $s = eH^\top$
2. $\text{wt}_H(e) \leq t$

---


- Random $H, e$ of weight $t$, compute $s = eH^\top$ \(\rightarrow\) small public key sizes
- Verifier challenges either 1. or 2. by asking for transformation $\varphi$ or transformed secret $\varphi(e)$
- Large cheating probability \(\rightarrow\) many rounds, large signature size, CVE: 40 KB
- Recent improvements through in the head computations \(\rightarrow\) smaller signature sizes, 10 KB


Restricted Errors

Syndrome Decoding Problem

Given $H \in \mathbb{F}_q^{(n-k) \times n}$, $s \in \mathbb{F}_q^{n-k}$, weight $t$, find $e \in \mathbb{F}_q^n$ such that $s = eH^\top$ and $\text{wt}(e) \leq t$.

Can we avoid permutations - but keep the hardness of the problem?
Restricted Errors

Syndrome Decoding Problem

Given $H \in \mathbb{F}_q^{(n-k) \times n}$, $s \in \mathbb{F}_q^{n-k}$, weight $t$, find $e \in \mathbb{F}_q^n$ such that $s = eH^\top$ and $\text{wt}(e) \leq t$.

Can we avoid permutations - but keep the hardness of the problem?

↓

Restricted Syndrome Decoding Problem

Given $H \in \mathbb{F}_q^{(n-k) \times n}$, syndrome $s \in \mathbb{F}_q^{n-k}$, $E \subseteq \mathbb{F}_q^*$, find $e \in E^n$ such that $s = eH^\top$.
Restricted Errors


Restricted SDP: Given $H \in \mathbb{F}_q^{(n-k) \times n}$, $s \in \mathbb{F}_q^{n-k}$, $E \subseteq \mathbb{F}_q^*$, find $e \in E^n$ such that $s = eH^\top$. 

Can replace SDP with Restricted SDP in any code-based ZK protocol: 10 KB $\rightarrow$ 7.2 KB

Open Question
Can we exploit the commutativity of the restricted transformations?
Restricted Errors


Restricted SDP: Given $H \in \mathbb{F}_q^{(n-k) \times n}$, $s \in \mathbb{F}_q^{n-k}$, $E \subseteq \mathbb{F}_q^*$, find $e \in E^n$ such that $s = eH^\top$.

Idea

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

M. Baldi, S. Bitzer, A. Pavoni, P. Santini, A. Wachter-Zeh, **V.W.** “Zero Knowledge Protocols and Signatures from the Restricted Syndrome Decoding Problem”, Preprint, 2023

---

**Restricted SDP:** Given $H \in \mathbb{F}_q^{(n-k) \times n}$, $s \in \mathbb{F}_q^{n-k}$, $E \subseteq \mathbb{F}_q^*$, find $e \in E^n$ such that $s = eH^\top$.

<table>
<thead>
<tr>
<th>$e$</th>
<th>$g^i$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$e'$</td>
<td>$g^j$</td>
</tr>
<tr>
<td>$e \ast e'$</td>
<td>$g^{i+j}$</td>
</tr>
</tbody>
</table>

**Idea**

- $g \in \mathbb{F}_q^*$ of order $z$, $E = \{g^i \mid i \in \{1, \ldots, z\}\}$
- transf. $\phi : E^n \rightarrow E^n$, $e \mapsto e \ast e'$ for $e' \in E^n$
- size of $\phi$ is $n \log_2(z)$ (instead of $n \log_2((q-1)n)$)

---

Violetta Weger — Recent Advances in Code-based Signatures 11/14
Restricted SDP: Given \( H \in \mathbb{F}_q^{(n-k) \times n} \), \( s \in \mathbb{F}_q^{n-k} \), \( E \subseteq \mathbb{F}_q^* \), find \( e \in E^n \) such that \( s = eH^\top \).

**Idea**

- \( g \in \mathbb{F}_q^* \) of order \( z \), \( E = \{g^i \mid i \in \{1, \ldots, z\}\} \)
- transf. \( \varphi : E^n \to E^n, e \mapsto e \ast e' \) for \( e' \in E^n \)
- size of \( \varphi \) is \( n \log_2(z) \) (instead of \( n \log_2((q-1)n) \))

Can replace SDP with Restricted SDP in any code-based ZK protocol: 10 KB \( \rightarrow \) 7.2 KB
Restricted SDP: Given $H \in \mathbb{F}_q^{(n-k) \times n}$, $s \in \mathbb{F}_q^{n-k}$, $E \subseteq \mathbb{F}_q^*$, find $e \in E^n$ such that $s = eH^\top$.

**Idea**

- $g \in \mathbb{F}_q^*$ of order $z$, $E = \{g^i \mid i \in \{1, \ldots, z\}\}$
- transf. $\varphi : E^n \rightarrow E^n$, $e \mapsto e \ast e'$ for $e' \in E^n$
- size of $\varphi$ is $n \log_2(z)$ (instead of $n \log_2((q-1)n)$)

Can replace SDP with Restricted SDP in any code-based ZK protocol: 10 KB $\rightarrow$ 7.2 KB

**Open Question**

Can we exploit the commutativity of the restricted transformations?
Outline

1. Code-based Cryptography
   - Introduction to Coding Theory
   - Hard Problems from Coding Theory
   - Previous Work

2. Code-based Signature Schemes
   - Idea and Previous Work
   - FuLeeca
   - Restricted Errors

3. Future Research
   - Rank-metric Decoding
   - Quantum Codes
   - Further Research Directions
Future Research: Rank-metric Decoding

- For $x \in \mathbb{F}_q^n$:
  \[ \text{Rank metric: } wt_R(x) = \dim(\langle x_1, \ldots, x_n \rangle_{\mathbb{F}_q}) \]

- Rank Syndrome Decoding Problem: no NP-hard reduction

- Hamming-metric decoders have cost in $O(q^{nc})$ for some constant $c$

- Rank-metric decoders have cost in $O(q^{n^2c'})$ for some constant $c'$
  \[ \rightarrow \text{ Small key sizes} \]

\[ \rightarrow \text{Goal: Improve decoders} \]

- Error support $E = \langle e_1, \ldots, e_n \rangle_{\mathbb{F}_q}$

- Candidate supersupports $F, F'$

TUM   Antonia Wachter-Zeh
International   Alberto Ravagnani (TU/e)
Future Research: Quantum Codes

- Quantum error-corrections:
  1. depolarizing channel,
  2. dephasing channel
- Introduced errors:
  1. $Z$ and $X$-errors,
  2. only $Z$-errors
- $X$-errors are in $\mathbb{F}_{q^2} \setminus \mathbb{F}_q$
  $Z$-errors are in $\mathbb{F}_q \setminus \{0\}$

$\rightarrow$ Errors in base field more likely
$\rightarrow$ New metric:
\[
\text{wt}_\lambda(x) = \lambda \text{ if } x \in \mathbb{F}_{q^2} \setminus \mathbb{F}_q \\
\text{wt}_\lambda(x) = 1 \text{ if } x \in \mathbb{F}_q \setminus \{0\}
\]

$\rightarrow$ Goal: New bounds and constructions

TUM Robert König
International Markus Grassl (ICTQT)
Further Research Directions

- **Quantum-Private Information Retrieval**
  - Retrieve file from database managed by untrusted server
  - without revealing to the server which file was requested
  - single server: only number-theoretic solutions: not quantum-secure
  → **Goal:** code-based quantum-private information retrieval
    
    TUM Antonia Wachter-Zeh
    International Camilla Hollanti (Aalto University)

- **Locally Recoverable Codes**
  → **Goal:** New constructions
    
    TUM Gregor Kemper

- **Isogeny-based Cryptography**
  → **Goal:** New systems
    
    TUM Christian Liedtke
Questions?

Thank you!
# Hash-and-Sign: CFS

<table>
<thead>
<tr>
<th>PROVER</th>
<th>VERIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEY GENERATION</strong></td>
<td></td>
</tr>
<tr>
<td>( S = H ) parity-check matrix</td>
<td></td>
</tr>
<tr>
<td>( \mathcal{P} = (t, HP) ) permuted ( H )</td>
<td></td>
</tr>
<tr>
<td><strong>SIGNING</strong></td>
<td></td>
</tr>
<tr>
<td>Choose message ( m )</td>
<td></td>
</tr>
<tr>
<td>( s = Hash(m) )</td>
<td></td>
</tr>
<tr>
<td>Find ( e: s = eH^\top = eP(HP)^\top ), and ( wt(e) \leq t )</td>
<td></td>
</tr>
<tr>
<td>( m, eP )</td>
<td></td>
</tr>
<tr>
<td><strong>VERIFICATION</strong></td>
<td></td>
</tr>
<tr>
<td>Check if ( wt(eP) \leq t ) and ( eP(HP)^\top = Hash(m) )</td>
<td></td>
</tr>
</tbody>
</table>
# Hash-and-Sign: CFS

<table>
<thead>
<tr>
<th>PROVER</th>
<th>VERIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEY GENERATION</strong></td>
<td></td>
</tr>
<tr>
<td>$S = H$ parity-check matrix</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P} = (t, HP)$ permuted $H$</td>
<td></td>
</tr>
<tr>
<td><strong>SIGNING</strong></td>
<td></td>
</tr>
<tr>
<td>Choose message $m$</td>
<td></td>
</tr>
<tr>
<td>$s = \text{Hash}(m)$</td>
<td></td>
</tr>
<tr>
<td>Find $e$: $s = eH^\top = eP(HP)^\top$, and $\text{wt}(e) \leq t$</td>
<td></td>
</tr>
<tr>
<td>$m,eP$</td>
<td></td>
</tr>
<tr>
<td><strong>VERIFICATION</strong></td>
<td></td>
</tr>
<tr>
<td>Check if $\text{wt}(eP) \leq t$ and $eP(HP)^\top = \text{Hash}(m)$</td>
<td></td>
</tr>
</tbody>
</table>

**Problem:** Distinguishability
# Hash-and-Sign: CFS

<table>
<thead>
<tr>
<th>PROVER</th>
<th>VERIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEY GENERATION</strong></td>
<td></td>
</tr>
<tr>
<td>$S = H$ parity-check matrix</td>
<td></td>
</tr>
<tr>
<td>$\mathcal{P} = (t, HP)$ permuted $H$</td>
<td></td>
</tr>
<tr>
<td><strong>SIGNING</strong></td>
<td></td>
</tr>
<tr>
<td>Choose message $m$</td>
<td></td>
</tr>
<tr>
<td>$s = \text{Hash}(m)$</td>
<td></td>
</tr>
<tr>
<td>Find $e$: $s = eH^\top = eP(HP)^\top$, and $\text{wt}(e) \leq t$</td>
<td></td>
</tr>
<tr>
<td>$m, eP$</td>
<td></td>
</tr>
<tr>
<td><strong>VERIFICATION</strong></td>
<td></td>
</tr>
<tr>
<td>Check if $\text{wt}(eP) \leq t$ and $eP(HP)^\top = \text{Hash}(m)$</td>
<td></td>
</tr>
</tbody>
</table>

Not any $s$ is syndrome of low weight $e$
PROVER VERIFIER

commitments $c_0, c_1$ $\xrightarrow{c_0, c_1}$

response $r_b$ $\xrightarrow{r_b}$

$\leftarrow b \in \{0, 1\}$

Verify $c_b$ using $r_b, \mathcal{P}$

SIGNING

Choose message $m$

Construct signature $s$ from $\mathcal{S}, m$

$\xrightarrow{m,s}$

VERIFICATION

Verify signature $s$ using $\mathcal{P}, m$
### Signature Scheme

<table>
<thead>
<tr>
<th>PROVER</th>
<th>VERIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td>commitments $c_0, c_1$</td>
<td>$c_0, c_1$</td>
</tr>
<tr>
<td>$b$</td>
<td>$b \in {0, 1}$</td>
</tr>
<tr>
<td>response $r_b$</td>
<td>$r_b$</td>
</tr>
<tr>
<td>Verify $c_b$ using $r_b, \mathcal{P}$</td>
<td></td>
</tr>
</tbody>
</table>

#### Fiat-Shamir

<table>
<thead>
<tr>
<th>SIGNING</th>
<th>VERIFICATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choose message $m$</td>
<td>Verify signature $s$ using $\mathcal{P}, m$</td>
</tr>
<tr>
<td>Construct signature $s$ from $\mathcal{S}, m$</td>
<td></td>
</tr>
</tbody>
</table>
Fiat-Shamir

PROVER

KEY GENERATION
Given $\mathcal{P}, \mathcal{S}$ of some ZKID and message $m$

SIGNING
Choose commitment $c$
$b = \text{Hash}(m, c)$
Compute response $r_b$
Signature $s = (b, r_b)$

VERIFICATION
Using $r_b$, $\mathcal{P}$ construct $c$
check if $b = \text{Hash}(m, c)$
# PROVER

## KEY GENERATION

Choose $e$ with $\text{wt}(e) \leq t$

$H$ parity-check matrix

Compute $s = eH^\top$

\[ P = (H, s, t) \rightarrow \]

## VERIFIER

## VERIFICATION

Choose $u \in \mathbb{F}_q^n$, $\sigma \in S_n$

Set $c_1 = \text{Hash}(\sigma, uH^\top)$

Set $c_2 = \text{Hash}(\sigma(u), \sigma(e))$

Choose $z \in \mathbb{F}_q^\times$

Choose $b \in \{1, 2\}$

$\begin{align*}
    r_1 &= \sigma \\
    r_2 &= \sigma(e)
\end{align*}$

If $b = 1$: $c_1 = \text{Hash}(\sigma, \sigma^{-1}(y)H^\top - zs)$

If $b = 2$: $\text{wt}(\sigma(e)) = t$

and $c_2 = \text{Hash}(y - z\sigma(e), \sigma(e))$
**CVE**

<table>
<thead>
<tr>
<th>PROVER</th>
<th>VERIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEY GENERATION</strong></td>
<td><strong>VERIFICATION</strong></td>
</tr>
<tr>
<td>Choose $e$ with $\text{wt}(e) \leq t$</td>
<td>Recall SDP: (1) $s = eH^\top$ (2) $\text{wt}(e) \leq t$</td>
</tr>
<tr>
<td>$H$ parity-check matrix</td>
<td></td>
</tr>
<tr>
<td>Compute $s = eH^\top$</td>
<td>$\mathcal{P} = (H, s, t)$</td>
</tr>
</tbody>
</table>

Choose $u \in \mathbb{F}_q^n$, $\sigma \in S_n$
Set $c_1 = \text{Hash}(\sigma, uH^\top)$
Set $c_2 = \text{Hash}(\sigma(u), \sigma(e))$

Choose $z \in \mathbb{F}_q^\times$
Choose $b \in \{1, 2\}$

Set $y = \sigma(u + ze)$
$r_1 = \sigma$
$r_2 = \sigma(e)$
$b = 1$: $c_1 = \text{Hash}(\sigma, \sigma^{-1}(y)H^\top - zs)$
$b = 2$: $\text{wt}(\sigma(e)) = t$
and $c_2 = \text{Hash}(y - z\sigma(e), \sigma(e))$
<table>
<thead>
<tr>
<th><strong>PROVER</strong></th>
<th><strong>VERIFIER</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEY GENERATION</strong></td>
<td><strong>VERIFICATION</strong></td>
</tr>
<tr>
<td>Choose ( e ) with ( \text{wt}(e) \leq t )</td>
<td>(&lt;P=(H,s,t)&gt;)</td>
</tr>
<tr>
<td>( H ) parity-check matrix</td>
<td> </td>
</tr>
<tr>
<td>Compute ( s = eH^\top )</td>
<td> </td>
</tr>
</tbody>
</table>

Choose \( u \in \mathbb{F}_q^n \), \( \sigma \in S_n \)
Set \( c_1 = \text{Hash}(\sigma, uH^\top) \)
Set \( c_2 = \text{Hash}(\sigma(u), \sigma(e)) \)

Choose \( z \in \mathbb{F}_q \)

Set \( y = \sigma(u + ze) \)

\( r_1 = \sigma \)

\( r_2 = \sigma(e) \)

Choose \( b \in \{1, 2\} \)

\( b = 1: \ c_1 = \text{Hash}(\sigma, \sigma^{-1}(y)H^\top - zs) \)

\( b = 2: \ \text{wt}(\sigma(e)) = t \)

and \( c_2 = \text{Hash}(y - z\sigma(e), \sigma(e)) \)

Problem: big signature sizes
Cheating Probability

- Cheating probability = Probability of impersonator getting accepted
- For security level $2^\lambda$ want cheating probability $2^{-\lambda}$
- If cheating probability $\delta$, with $N$ rounds $\rightarrow$ cheating probability $\delta^N$
Cheating Probability

- Cheating probability = Probability of impersonator getting accepted
- For security level $2^\lambda$ want cheating probability $2^{-\lambda}$
- If cheating probability $\delta$, with $N$ rounds $\rightarrow$ cheating probability $\delta^N$
- might need many rounds: large communication cost
Cheating Probability

- Cheating probability = Probability of impersonator getting accepted
- For security level $2^\lambda$ want cheating probability $2^{-\lambda}$
- If cheating probability $\delta$, with $N$ rounds $\rightarrow$ cheating probability $\delta^N$
- might need many rounds: large communication cost
- solution: compression technique
  - do not send $c_0^i, c_1^i$ in each round $i$
  - before 1. round send $c = \text{Hash}(c_0^1, c_1^1, \ldots, c_0^N, c_1^N)$
  - $i$th round: receiving challenge $b$ prover sends $r_b^i, c_{1-b}^i$
  - end: verifier checks $c = \text{Hash}(c_0^1, c_1^1, \ldots, c_0^N, c_1^N)$

Cheating Probability

- Cheating probability = Probability of impersonator getting accepted
- For security level $2^\lambda$ want cheating probability $2^{-\lambda}$
- If cheating probability $\delta$, with $N$ rounds $\rightarrow$ cheating probability $\delta^N$
- might need many rounds: large communication cost
- other solution: MPC in the head
- third party: trusted helper sends commitments $\rightarrow \delta = 0$
- instead prover sends seeds of commitment: not ZK $\rightarrow$ cut and choose
- $x < N$ times send response, $N - x$ times send the seed of commitment
- to compress: use Merkle root or seed tree

## Comparison

<table>
<thead>
<tr>
<th></th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduction to NP-hard</td>
<td></td>
<td></td>
</tr>
<tr>
<td>low public key size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>low signature size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fast verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison

<table>
<thead>
<tr>
<th></th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduction to NP-hard</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>low public key size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>low signature size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fast verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparison

<table>
<thead>
<tr>
<th></th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduction to NP-hard</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>low public key size</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>low signature size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>fast verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduction to NP-hard</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>Low public key size</td>
<td>CVE: 70 B</td>
<td>WAVE: 3 MB</td>
</tr>
<tr>
<td>Low signature size</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fast verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduction to NP-hard</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>low public key size</td>
<td>CVE: 70 B</td>
<td>WAVE: 3 MB</td>
</tr>
<tr>
<td>low signature size</td>
<td>~</td>
<td>✓</td>
</tr>
<tr>
<td>fast verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduction to NP-hard</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>low public key size</td>
<td>CVE: 70 B</td>
<td>WAVE: 3 MB</td>
</tr>
<tr>
<td>low signature size</td>
<td>CVE: 43 KB</td>
<td>WAVE: 1 KB</td>
</tr>
<tr>
<td>fast verification</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>ZKID</th>
<th>Hash-and-Sign</th>
</tr>
</thead>
<tbody>
<tr>
<td>reduction to NP-hard</td>
<td>✓</td>
<td>×</td>
</tr>
<tr>
<td>low public key size</td>
<td>CVE: 70 B</td>
<td>WAVE: 3 MB</td>
</tr>
<tr>
<td>low signature size</td>
<td>CVE: 43 KB</td>
<td>WAVE: 1 KB</td>
</tr>
<tr>
<td>fast verification</td>
<td>~</td>
<td>✓</td>
</tr>
</tbody>
</table>
For $x \in \mathbb{F}_p$: $\text{wt}_L(x) = \min\{x, |p - x|\}$.  
For $x \in \mathbb{F}_p^n$: $\text{wt}_L(x) = \sum_{i=1}^{n} \text{wt}_L(x_i)$.  
Representing $\mathbb{F}_p = \{-\frac{p-1}{2}, \ldots, 0, \ldots, \frac{p-1}{2}\}$, $\text{wt}_L(x) = |x|$.


\begin{itemize}
  \item For $x \in \mathbb{F}_p$: $\text{wt}_L(x) = \min\{x, | p - x | \}$.
  For $x \in \mathbb{F}_p^n$: $\text{wt}_L(x) = \sum_{i=1}^{n} \text{wt}_L(x_i)$.
  \item Representing $\mathbb{F}_p = \{-\frac{p-1}{2}, \ldots, 0, \ldots, \frac{p-1}{2}\}$, \\
  $\text{wt}_L(x) = |x|$.
  \item Number of matches between $x, y \in \mathbb{F}_p^n$
  $\text{mt}(x, y) = |\{i \mid \text{sgn}(x_i) = \text{sgn}(y_i)\}|$.
\end{itemize}
Statistical Attacks

- Low Hamming weight generators will produce low Hamming weight signatures
- Observing many signatures reveals the support of the secret low Hamming weight generators
Statistical Attacks

- Low Hamming weight generators will produce low Hamming weight signatures.
- Observing many signatures reveals the support of the secret low Hamming weight generators.

Low Lee weight generators:
\[ \text{supp}_L(x) = (\text{wt}_L(x_1), \ldots, \text{wt}_L(x_n)) \]
- Signatures have low Lee weight.
- Recovering Lee support of secret generators: much harder.
<table>
<thead>
<tr>
<th>PROVER</th>
<th>VERIFIER</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>KEY GENERATION</strong></td>
<td></td>
</tr>
<tr>
<td>Secret key: $G = [A \ B]$, quasi-cyclic matrix, with low Lee weight</td>
<td></td>
</tr>
<tr>
<td>Public key: $G' = [\text{Id} \ A^{-1}B]$</td>
<td>$(G',t,\mu)$</td>
</tr>
<tr>
<td><strong>SIGNING</strong></td>
<td></td>
</tr>
<tr>
<td>Choose message $m$</td>
<td></td>
</tr>
<tr>
<td>$c = \text{Hash}(m) \in {\pm 1}^n$</td>
<td></td>
</tr>
<tr>
<td>Iteratively use $G$ to construct code-word $\sigma$ with</td>
<td></td>
</tr>
<tr>
<td>$\text{wt}_L(\sigma) \leq t$,</td>
<td></td>
</tr>
<tr>
<td>$\text{mt}(\sigma, c) \geq \mu$</td>
<td></td>
</tr>
<tr>
<td><strong>VERIFICATION</strong></td>
<td></td>
</tr>
<tr>
<td>Verify that: (1) $\sigma H^\top = 0$,</td>
<td></td>
</tr>
<tr>
<td>(2) $\text{wt}_L(\sigma) \leq t$,</td>
<td></td>
</tr>
<tr>
<td>(3) $\text{mt}(c, \sigma) \geq \mu$</td>
<td></td>
</tr>
</tbody>
</table>