A Tale of VOLEs, Zero-Knowledge Proofs and Post-Quantum Signatures

*Peter Scholl*

LatinCrypt 2023
Standardization of Post-Quantum Signatures

Dilithium
- Security: Structured lattices
- Speed: Fast
- Size: 2.4 kB

Falcon
- Security: Structured lattices
- Speed: Fast
- Size: 0.7 kB

SPHINCS+
- Security: Hash-based
- Speed: Slow signing
- Size: 8-17 kB

FAEST
- Security: AES/hash-based
- Speed: Fast-ish
- Size: 5-7 kB

2023: new algorithms submitted to diversify candidates
FAEST: Design and Inspiration

MPC-in-the-head signatures

VOLE-based ZK

Secure 2-Party Computation

BBQ

Picnic

MPC-in-the-head signatures

Banquet

Line-Point ZK

Mac’n’Cheese

QuickSilver

SoftSpokenOT

QuickSilver

Line-Point ZK

Mac’n’Cheese

Banquet

BBQ

Picnic

FAEST

Peter Scholl

3
Overview of today

Vector oblivious linear evaluation (VOLE)

Zero-knowledge proofs

VOLE-in-the-head

FAEST
Based on

Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head with Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Kloß, Emmanuela Orsini, Lawrence Roy
CRYPTO 2023 (ePrint 2023/996)

FAEST Digital Signature Scheme
+ Christian Majenz, Shibam Mukherjee, Sebastian Ramacher, Christian Rechberger
Submission to NIST PQC Standardization process
https://faest.info
Zero-knowledge proofs

• A proof where the verifier learns nothing
  • Except the truth of the statement

Proof should be correct, sound and zero-knowledge
Zero-knowledge proofs

• A proof where the verifier learns nothing
  • Except the truth of the statement: $C(w) = 0$
  • $C : \mathbb{F}^n \rightarrow \mathbb{F}$ (arithmetic circuit)

Proof should be correct, sound and zero-knowledge
Families of ZK Proofs

Size: < 1 field elem. per mult.

designated verifier
Families of ZK Proofs

- **Linear**
  - MPC-in-the-head
  - VOLE-in-the-head

- **Succinct**
  - Ligero
  - STARKs
  - Groth16

**Proof size**

Size: 1 – 10 field elem. per mult.
publicly verifiable
Vector Oblivious Linear Evaluation

Today: $\mathbf{v}$ always uniform

Variant: random VOLE where $\mathbf{w}$ also uniform
What is VOLE good for?

Fundamental *building block* in many cryptographic protocols:

- General-purpose secure computation
- Oblivious transfer
  - Implied by variant of VOLE
- Private set intersection
  - Contact discovery; online advertising
Linearly homomorphic commitments from VOLE

To commit to $\overrightarrow{w}$:
- Alice inputs $(\overrightarrow{w}, \overrightarrow{v})$ to VOLE, for random $\overrightarrow{v}$

To open $w$:
- Alice sends $(w, v)$, Bob checks if $q = w\Delta + v$
- Hiding: since $v$ is random
- Binding: opening to $w' \neq w$ requires guessing $\Delta$, prob. $1/|\mathbb{F}|$

Commitments are linearly homomorphic

$\text{BMRS 21, WYKW 21}$
VOLE-ZK: Zero Knowledge Proofs with VOLE
Proving circuits with linear commitments

**Goal:** prove knowledge of $x$ such that $C(x) = z$

- Commit to extended witness $\overrightarrow{w}$
  - inputs, + output wire of every mult.

- Evaluate linear gates
  - Using linear homomorphism

- Prove correctness of multiplications

[Cramer-Damgård 97]
Checking multiplication gates

• Multiply two lines $\Rightarrow$ quadratic polynomial
  $p_{ab}(x) = p_a(x)p_b(x)$
  $= abx^2 + \cdots$

• Compute:
  • $p_{ab}(x) - xp_c(x) = (ab - c)x^2 + dx + e$
    $= dx + e$

• Send $(d, e)$ to Bob
  • Masked with random VOLE
  • Bob checks $d\Delta + e = q_{ab} - \Delta q_c$
ZK proof from VOLE: Initial Protocol

\[ \vec{d}_i, \vec{e}_i \] for \( i \)-th mult. gate

\[ \vec{v}, \vec{w} \in \mathbb{F}^n \]

\[ \Delta \]

\[ \vec{q} = \vec{w} \Delta + \vec{v} \]

Soundness error:
- \( \frac{2}{|\mathbb{F}|} \)

Cost for \( m \) multiplications:
- VOLE + \( 2m \) field elements

[DIO 21]
Optimization: batching multiplications

\[ \tilde{v}, \tilde{w} \in \mathbb{F}^n \]

\[ \Delta \]

\[ \tilde{q} = \tilde{w}\Delta + \tilde{v} \]

Soundness error:
• \( \frac{2}{|\mathbb{F}|} + \frac{m}{|\mathbb{F}|} \)

Cost for \( m \) multiplications:
• Length-\( m \) VOLE

(\( d_i, e_i \)) for \( i \)-th mult. gate

\[ \sum_i d_i r^i, \sum_i e_i r^i \]

[YSWW 21]
Improvements/extensions

• Circuits over $\mathbb{F}_2$: [YSWW 21]
  • Let $w \in \mathbb{F}_2$, but use subfield VOLE $q = w\Delta + v$ in $\mathbb{F}_{2^k}$

• Higher-degree checks: [YSWW 21]
  • Keep adding/multiplying VOLE commitments
  • Commit to every $k$-th mult. gate $\Rightarrow$ poly degree up to $2^k$

• Mixed Boolean/arithmetic circuits [BBMRS 21, YYXKW 21]
  • VOLE in $\mathbb{F}_2$ and $\mathbb{F}_p$, prove consistency
Building VOLE

• Linearly homomorphic encryption
  ➢ Fairly slow
  ➢ $O(m)$ communication

• Pseudorandom correlation generators ("Silent" VOLE)
  • Learning parity with noise
  • Random, length-$m$ VOLE: $O(\log m)$ communication (+$m$ field elem. for chosen $\vec{w}$)

• With oblivious transfer ("SoftSpokenVOLE")
  • Mainly symmetric primitives, fast
  • $O(\log m)$ communication in small fields
Building VOLE in $\mathbb{F}_n$ with oblivious transfer (OT)

(SoftSpokenOT [Roy 22])

Convert to VOLE

$\vec{v}, \vec{w}$

$\vec{q} = \vec{w}\Delta + \vec{v} \in \mathbb{F}^m$
Conversion to VOLE

Key observation: \((n - 1)\)-out-of-\(n\) secret sharing \(\Rightarrow\) VOLE in \(\mathbb{F}_n\)

\[w = w_1 + \cdots + w_n\]
\[\nu = -1 \cdot w_1 - \cdots - n \cdot w_n \text{ (in } \mathbb{F}_n\text{)}\]

\[\Delta \in \mathbb{F}_n\]

\[q = \sum_{i=1}^{n} w_i \cdot (\Delta - i)\]

\[= w\Delta + \nu\]

[Roy 22]
Conversion to VOLE

Key observation: \((n - 1)\)-out-of-\(n\) secret sharing ⇒ VOLE in \(\mathbb{F}_n\)

[Roy 22]

\[
\begin{align*}
\vec{w}_1 &= \text{PRG}(s_1) \\
\vdots \\
\vec{w}_n &= \text{PRG}(s_1) \\
\vec{w} &= \vec{w}_1 + \cdots + \vec{w}_n \\
\vec{v} &= -1 \cdot \vec{w}_1 - \cdots - n \cdot \vec{w}_n \quad \text{(in } \mathbb{F}_n^m) \\
\Delta &\in \mathbb{F}_n \\
\vec{w}_i &\quad \text{for } i \neq \Delta \\
\tilde{q} &= \sum_{i=1}^{n} \vec{w}_i \cdot (\Delta - i) \\
&= \vec{w} \Delta + \vec{v}
\end{align*}
\]
VOLE-in-the-head: from designated verifier to publicly verifiable ZK
Public-Receiver VOLE (aka VOLE-in-the-head)

\[ \vec{q} = \vec{w} \Delta + \vec{v} \]
How to do VOLE-in-the-head? Just commit!

[BBdGKORS 23]

All-but-one vector commitment

Commit to $n$ random strings

Challenge $\Delta$

Open $n - 1$

Convert to VOLE

$\vec{w}, \vec{v}$

Convert to VOLE

$\vec{q} = \vec{w}\Delta + \vec{v}$
VOLE-in-the-head: Summary

• If \( \overrightarrow{w} \) is random, can succinctly commit to arbitrarily long VOLE
  • With PRG/hash

• Communication cost:
  • \( O(\log n) \) with PRG tree optimization

• For non-random \( w \):
  • Send extra \( |w| \) field elements
ZK from VOLE-in-the-head: putting things together

VOLE: “commit” to extended witness

\[ r \leftarrow \mathbb{F}_n \]
\[ d, e \]
\[ \Delta \leftarrow \mathbb{F}_n \]

“open” VOLE

Soundness error:
• \(3/|\mathbb{F}|\) (small fields)
• Improve via parallel repetition

Communication cost:
• \(\mathbb{F}_2\): \(\approx 10\) bits per AND
• \(F_p\): 1-2 field elements per mult
The Curse of Parallel Repetitions with >3 Rounds

• Problem: Fiat-Shamir can worsen security for >3-round protocols
  ➢ Adversary can attack each round independently

• Solution: more rounds!
  ➢ Consistency check: prove same witness is committed in small-field VOLEs
  ➢ Allows to combine multiplication checks into one check
Final Protocol: Overview

VOLE: “commit” to extended witness

- $c_h_1$
  - mult check
  - $c_h_2$
  - consistency check
  - $\Delta$
  - “open” VOLE
PQ Signatures From VOLE-in-the-Head
FAEST: high-level overview

• **Public key**: AES encryption of known message under secret key

• **Signature on m**:  
  • Zero-knowledge proof that key is valid  
  • Using VOLE-in-the-head
AES: a ZK-friendly block cipher?

ShiftRows, MixColumns, AddRoundKey:

• All linear over $\mathbb{F}_2$

S-Box:

• Inversion in $\mathbb{F}_{2^8}$
• Prove in ZK as 1 multiplication check
FAEST: example performance

<table>
<thead>
<tr>
<th></th>
<th>Sign/Verify</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAEST-128s</td>
<td>≈ 8ms</td>
<td>5 006 B</td>
</tr>
<tr>
<td>FAEST-128f</td>
<td>≈ 1ms</td>
<td>6 336 B</td>
</tr>
<tr>
<td>FAEST-256s</td>
<td>≈ 27ms</td>
<td>22 100 B</td>
</tr>
<tr>
<td>FAEST-256f</td>
<td>≈ 3ms</td>
<td>28 400 B</td>
</tr>
</tbody>
</table>

- **Signature sizes:**
  - Smaller than SPHINCS+ and most code-based candidates
  - Faster signing, slower verification

- **Possible variants:**
  - Fixed-key AES (Even-Mansour): 10% smaller
  - MQ instead of AES: size ≈ 3 kB
Conclusion

VOLE-in-the-head ZK proofs:
• Lightweight, fast and powerful
• Proof size:
  • \( \approx 10 \) bits or 1 field element per mult.

Application: FAEST PQ signature:
• Conservative security
• Reasonable performance

Resources:
• https://ia.cr/2023/996
• https://faest.info

Thank you!