The Syndrome Decoding in the Head (SD-in-the-Head) Signature Scheme

Carlos Aguilar Melchor, Slim Bettaieb, Loïc Bidoux,
Thibauld Feneuil, Philippe Gaborit, Nicolas Gama,
Shay Gueron, James Howe, Andreas Hülsing, David Joseph,
Antoine Joux, Mukul Kulkarni, Edoardo Persichetti,
Tovohery H. Randrianarisoa, Matthieu Rivain, Dongze Yue

Sixth PQC Standardization Conference

September 25, 2025, *NIST*

Table of Contents

Round-2 Design Updates

Choice of the Framework

SDitH v2 - Design Rationale

- **Conservative security**: signature scheme for which the security relies on the hardness of solving ***fully random unstructured*** instances of the syndrome decoding (SD) problem.

  SD Problem: Given a matrix H and a syndrome y, find x such that

  $Hx = y$ and x has w non-zero coordinates.

- **Design Choice**: Signature scheme built upon the ***MPC-in-the-Head*** paradigm, which provides a generic way to build a secure scheme from a hard problem.

SDitH v2 - Design Rationale

- **SD Parameters**:

  - The hardest instances:

    Unique solution, close to the Gilvert-Varshamov frontier

  - choice of the SD field

    - SDitH v1: $GF(251)$ and $GF(256)$
    - SDitH v2: $GF(2)$

🔎 *Motivation to choose GF(2)*:

More conservative security assumption

Hard problem easier to arithmetize

SDitH v2 - Design Rationale

- **MPC-in-the-Head paradigm**: possible existing frameworks
  - SDitH v1: rely on the framework using linear broadcast-based MPC
  - SDitH v2: ***two new MPCitH frameworks*** since the previous NIST deadline:

VOLE-in-the-Head	and	**TC-in-the-Head**
(summer 2023)		(fall 2023)

[BBD+23] Baum, Braun, Delpech, Klooß, Orsini, Roy, Scholl. Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head. Crypto 2023.

[FR25] Feneuil, Rivain. *Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments*. Journal of Cryptology, 2025.

SDitH v2 - Design Rationale

Instance	*Trade-off*	*VOLEitH*	*TCitH*
L1 - SD over GF(2)	Short	3 705 B	4 271 B (+15%)
	Fast	4 484 B	5 509 B (+23%)
L3 - SD over GF(2)	Short	7 964 B	8 426 B (+6%)
	Fast	9 916 B	11 374 B (+15%)
L5 - SD over GF(2)	Short	14 121 B	15 618 B (+11%)
	Fast	17 540 B	19 968 B (+14%)

For SD: Between 5-25% of difference, in favor of VOLEitH

Trade-off definition:
- « Short » uses trees of 2048-4096 leaves,
- « Fast » uses trees of 256 leaves.

Tree Optimisation: One-tree technique

SDitH v2 - Design Rationale

- **MPC-in-the-Head paradigm**: possible existing frameworks
  - SDitH v1: rely on the framework using linear broadcast-based MPC
  - SDitH v2: ***two new MPCitH frameworks*** since the previous NIST deadline:

VOLE-in-the-Head and **TC-in-the-Head**

(summer 2023) (fall 2023)

VOLE-in-the-Head (summer 2023)

- **Exponentially large fields:** e.g. $GF(2^{128})$ for L1
- **Only one protocol execution** (over the large field)
- **Based on 7-round protocol (or 5-round protocol)**
  - Rely on a **consistency check**
- Better signature sizes

TC-in-the-Head (fall 2023)

- **Small fields:** typically $GF(256), GF(2048), \dots$
- **Several parallel repetitions** (over the small field)
- **Based on 5-round protocol (or 3-round protocol)**

SDitH v2 - Design Rationale

- <u>**MPC-in-the-Head paradigm**</u>: possible existing frameworks
  - SDitH v1: rely on the framework using linear broadcast-based MPC
  - SDitH v2: ***two new MPCitH frameworks*** since the previous NIST deadline:

VOLE-in-the-Head (summer 2023) and **TC-in-the-Head** (fall 2023)

VOLE-in-the-Head (summer 2023)

- **Exponentially large fields:** e.g. $GF(2^{128})$ for L1
- **Only one protocol execution** (over the large field)
- **Based on 7-round protocol (or 5-round protocol)**
  - Rely on a **consistency check**
- ***Better signature sizes***

SDitH v2

TC-in-the-Head (fall 2023)

- **Small fields:** typically $GF(256), GF(256^2), \ldots$
- **Several parallel repetitions** (over the small field)
- **Based on 5-round protocol (or 3-round protocol)**

SDitH v2 - Design Rationale

- **Formalism**:
    - TCitH: sharing-based formalism
    - VOLEitH: VOLE-based formalism
    - SDitH v2: PIOP-based formalism (polynomial-based formalism)

🔎 *Motivation to choose the PIOP formalism:*

Simpler description of the scheme, that does not depend on MPC technology. Easier-to-understand scheme for those who do not already know those two frameworks.

[Fen24] Feneuil. *The Polynomial-IOP Vision of the Latest MPCitH Framework for Signature Schemes*. PQ Algebraic Cryptography Workshop, IHP 2024.

SDitH v2 - Underlying 3-Round Identification Scheme

Secret Key: $w \in \mathbb{F}_2^n$

Public Key: some degree-d multivariate polynomials $f_1, \ldots, f_m$ such that $f_1(w) = \ldots = f_m(w) = 0$.

① Sample n random degree-1 polynomials $P_1, \ldots, P_n$ such that $P_1(0) = x_1, \ldots, P_n(0) = x_n$.
Sample m random degree-$(d-1)$ polynomials $M_1, \ldots, M_m$.

② Commit to those polynomials.

SDitH v2 - Underlying 3-Round Identification Scheme

Secret Key: $w \in \mathbb{F}_2^n$

Public Key: some degree-d multivariate polynomials $f_1, \ldots, f_m$ such that $f_1(w) = \ldots = f_m(w) = 0$.

① Sample n random degree-1 polynomials $P_1, \ldots, P_n$ such that $P_1(0) = x_1, \ldots, P_n(0) = x_n$.
Sample m random degree-$(d-1)$ polynomials $M_1, \ldots, M_m$.

② Commit to those polynomials.

③ Send the polynomials $Q_1, \ldots, Q_m$ of degree at most $d-1$ defined such as

$$\begin{aligned} X \cdot Q_1(X) &:= X \cdot M_1(X) + f_1(P_1(X), \ldots, P_n(X)) \\ &\vdots \\ X \cdot Q_m(X) &:= X \cdot M_m(X) + f_m(P_1(X), \ldots, P_n(X)) \,. \end{aligned}$$

SDitH v2 - Underlying 3-Round Identification Scheme

Secret Key: $w \in \mathbb{F}_2^n$

Public Key: some degree-d multivariate polynomials $f_1, \ldots, f_m$ such that $f_1(w) = \ldots = f_m(w) = 0$.

① Sample n random degree-1 polynomials $P_1, \ldots, P_n$ such that $P_1(0) = x_1, \ldots, P_n(0) = x_n$.
Sample m random degree-$(d-1)$ polynomials $M_1, \ldots, M_m$.

② Commit to those polynomials.

③ Send the polynomials $Q_1, \ldots, Q_m$ of degree at most $d-1$ defined such as

$$\begin{aligned} X \cdot Q_1(X) &:= X \cdot M_1(X) + f_1(P_1(X), \ldots, P_n(X)) \\ &\vdots \\ X \cdot Q_m(X) &:= X \cdot M_m(X) + f_m(P_1(X), \ldots, P_n(X)) . \end{aligned}$$

④ Get a random evaluation point $r \in \mathscr{C}$ from the verifier.

⑤ Reveal the evaluations $P_1(r), \ldots, P_n(r)$ and $M_1(r), \ldots, M_m(r)$.

Soundness: $\dfrac{d}{|\mathscr{C}|}$ from the Schwartz-Zippel Lemma

SDitH v2 - Underlying 5-Round Identification Scheme

Secret Key: $w \in \mathbb{F}_2^n$

Public Key: some degree-d multivariate polynomials $f_1, \ldots, f_m$ such that $f_1(w) = \ldots = f_m(w) = 0$.

① Sample n random degree-1 polynomials $P_1, \ldots, P_n$ such that $P_1(0) = x_1, \ldots, P_n(0) = x_n$.
Sample a random degree-$(d-1)$ polynomial M.

② Commit to those polynomials.

③ Get random coefficients $\gamma_1, \ldots, \gamma_n \in \mathbb{K}$ from the verifier.

④ Send the polynomial Q of degree at most $d-1$ defined such as

$$X \cdot Q(X) := X \cdot M(X) + \sum_{j=1}^{m} \gamma_j \cdot f_j(P_1(X), \ldots, P_n(X)).$$

⑤ Get a random evaluation point $r \in \mathscr{C}$ from the verifier.

⑥ Reveal the evaluations $P_1(r), \ldots, P_n(r)$ and $M(r)$.

Soundness: $\dfrac{d}{|\mathscr{C}|} + \dfrac{1}{|\mathbb{K}|}$

Round-2 Design Updates

Syndrome Decoding Arithmetization

SDitH v2 - Design Rationale

System of polynomial constraints:

given the constraints $f_1, \ldots, f_m$*, it is hard to find* w *such that* $f_1(w) = \ldots = f_m(w) = 0$.

$\Big\downarrow$ *VOLEitH (or TCitH)*

Signature scheme - SDitH

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

SD Arithmetization

System of polynomial constraints:

given the constraints $f_1, \ldots, f_m$, it is hard to find w such that $f_1(w) = \ldots = f_m(w) = 0$.

VOLEitH (or TCitH)

Signature scheme - SDitH

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

Using the SDitH v1 arithmetization:
- $|w| \approx 2100$
- $m \approx 14000$
- $d = 2$

👉 We would obtain sizes around 5.3 KB for short L1.

SD Arithmetization

System of polynomial constraints:

given the constraints $f_1, \ldots, f_m$, it is hard to find w such that $f_1(w) = \ldots = f_m(w) = 0$.

VOLEitH (or TCitH)

Signature scheme - *SDitH*

Parameters of the constraint system:

- *The witness size $|w|$*
- *The number m of constraints*
- *The degree d of the constraints*

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

SD Arithmetization

System of polynomial constraints:

given the constraints $f_1, \ldots, f_m$, it is hard to find w such that $f_1(w) = \ldots = f_m(w) = 0$.

VOLEitH (or TCitH)

Signature scheme - *SDitH*

Parameters of the constraint system:

- *The witness size $|w|$*
- *The number m of constraints*
- *The degree d of the constraints*

Using [BBGK24]:

- $|w| \approx 550$
- $m \approx 4100$
- $d = 12$

👉 We would obtain sizes around 4.9 KB for short L1.

[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni. *Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures*. ePrint 2024.

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

Provable reduction: SD → RSD
loss of $d \cdot \log_2 \frac{n}{w_H} - \log_2 \binom{n}{w_H}$ bits

Regular Syndrome Decoding Problem:

given H and y, it is hard to find $x := (v_1 \parallel v_2 \parallel \dots \parallel v_{w_H})$
such that $y = Hx$ and $\{v_i\}_i$ are elementary vectors
(have only one non-zero coordinate).

RSD Arithmetization

System of polynomial constraints:

given the constraints $f_1, \dots, f_m$, it is hard to
find w such that $f_1(w) = \dots = f_m(w) = 0$.

VOLEitH (or TCitH)

Signature scheme - SDitH

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

Provable reduction: SD → RSD
loss of $d \cdot \log_2 \frac{n}{w_H} - \log_2 \binom{n}{w_H}$ bits

Regular Syndrome Decoding Problem:

given H and y, it is hard to find $x := (v_1 \parallel v_2 \parallel \dots \parallel v_{w_H})$
such that $y = Hx$ and $\{v_i\}_i$ are elementary vectors
(have only one non-zero coordinate).

↓ *RSD Arithmetization*

System of polynomial constraints:

given the constraints $f_1, \dots, f_m$, it is hard to
find w such that $f_1(w) = \dots = f_m(w) = 0$.

↓ *VOLEitH (or TCitH)*

Signature scheme - SDitH

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

Provable reduction: SD → RSD
loss of $d \cdot \log_2 \frac{n}{w_H} - \log_2 \binom{n}{w_H}$ bits

Regular Syndrome Decoding Problem:

given H and y, it is hard to find $x := (v_1 \parallel v_2 \parallel \ldots \parallel v_{w_H})$
such that $y = Hx$ and $\{v_i\}_i$ are elementary vectors
(have only one non-zero coordinate).

RSD Arithmetization

System of polynomial constraints:

given the constraints $f_1, \ldots, f_m$, it is hard to
find w such that $f_1(w) = \ldots = f_m(w) = 0$.

VOLEitH (or TCitH)

Signature scheme - *SDitH*

⚠️ The security assumption in SDitH is still the **unstructured** SD problem! The regular SD problem is *just* a proof artifact.

SDitH v2 - Design Rationale

Syndrome Decoding Problem:

given H and y, it is hard to find x such that
$y = Hx$ and x has at most w_H non-zero coordinates.

Provable reduction: SD → RSD
loss of $d \cdot \log_2 \frac{n}{w_H} - \log_2 \binom{n}{w_H}$ bits

Regular Syndrome Decoding Problem:

given H and y, it is hard to find $x := (v_1 \parallel v_2 \parallel \ldots \parallel v_{w_H})$
such that $y = Hx$ and $\{v_i\}_i$ are elementary vectors
(have only one non-zero coordinate).

RSD Arithmetization

System of polynomial constraints:

given the constraints $f_1, \ldots, f_m$, it is hard to
find w such that $f_1(w) = \ldots = f_m(w) = 0$.

VOLEitH (or TCitH)

Signature scheme - SDitH

⚠️ The security assumption in SDitH is still the **unstructured** SD problem! The regular SD problem is *just* a proof artifact.

(R)SD Arithmetization

[OTX24] Ouyang, Tang, Xu. *Code-Based Zero-Knowledge from VOLE-in-the-Head and Their Applications: Simpler, Faster, and Smaller*. Asiacrypt 2024.

[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni. *Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures*. ePrint 2024.

We need to have a **compact representation for an elementary vector**

$$v = (0,0,\ldots,0,1,0,\ldots,0) \in \mathbb{F}_2^n$$

where the i^{th} coordinate is the non-zero one.

Let us denote $e_0 = (1,0)$ and $e_1 = (0,1)$. We have that

$$v = e_{b_0} \otimes e_{b_1} \otimes \ldots \otimes e_{b_{\ell-1}}$$

where $i := b_0 + 2 \cdot b_1 + \ldots + 2^{\ell-1} \cdot b_{\ell-1}$.

For example: when $n = 4$,

$$(1,0,0,0) = e_0 \otimes e_0 \qquad (0,0,1,0) = e_1 \otimes e_0$$

$$(0,1,0,0) = e_0 \otimes e_1 \qquad (0,0,0,1) = e_1 \otimes e_1$$

(R)SD Arithmetization

[OTX24] Ouyang, Tang, Xu. *Code-Based Zero-Knowledge from VOLE-in-the-Head and Their Applications: Simpler, Faster, and Smaller*. Asiacrypt 2024.

[BBGK24] Bettaieb, Bidoux, Gaborit, Kulkarni. *Modelings for generic PoK and Applications: Shorter SD and PKP based Signatures*. ePrint 2024.

We need to have a **compact representation for an elementary vector**

$$v = (0,0,\ldots,0,1,0,\ldots,0) \in \mathbb{F}_2^n$$

where the i^{th} coordinate is the non-zero one.

Let us denote $e_0 = (1,0)$ and $e_1 = (0,1)$. We have that

$$v = e_{b_0} \otimes e_{b_1} \otimes \ldots \otimes e_{b_{\ell-1}}$$

where $i := b_0 + 2 \cdot b_1 + \ldots + 2^{\ell-1} \cdot b_{\ell-1}$.

RSD Arithmetization: Constraints in $y = Hx$ for $x := (v_1 \parallel \ldots \parallel v_{w_H})$, when writing all v_i as a tensorial product of elementary vectors.

Constraint Degree = 8

(R)SD Arithmetization

Relaxed arithmetization used in SDitH v2:
Better signature sizes, better computation performance

We need to have a **compact representation for an elementary vector**

$$v = (0,0,\ldots,0,1,0,\ldots,0) \in \mathbb{F}_2^n$$

where the i^{th} coordinate is the non-zero one.

Let us denote $e_j^{(\mu)} \in \mathbb{F}_2^{\mu}$ the j^{th} elementary vector. We have that

$$v = e_{c_0}^{(\mu_0)} \otimes e_{c_1}^{(\mu_1)} \otimes e_{c_2}^{(\mu_2)} \otimes e_{c_3}^{(\mu_3)}$$

where $i := c_0 + \mu_0 \cdot c_1 + (\mu_1\mu_0) \cdot c_2 + (\mu_3\mu_1\mu_0) \cdot c_2$, with $c_j \in \{0,\ldots,\mu_j - 1\}$.

RSD Arithmetization: Constraints in $y = Hx$ for $x := (v_1 \parallel \ldots \parallel v_{w_H})$,
when writing all v_i as a tensorial product of elementary vectors.

Constraint Degree = 4

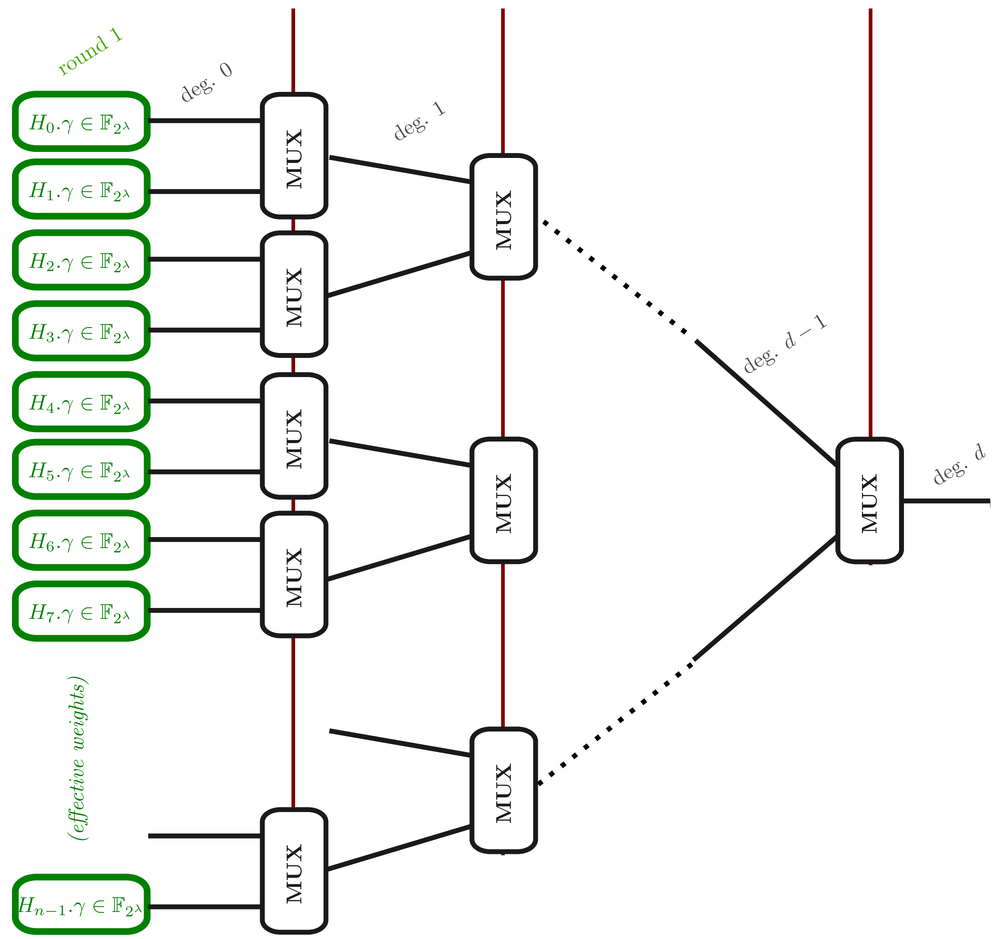
Round-2 Design Updates

Optimizations

SDitH v2 - Design Rationale

Used Optimizations (Arithmetic):

- Computation Speed-Up using Selection/Mux Tree



- Folding: using Gray code

👉 *See SDitH's specifications for details.*

SDitH v2 - Design Rationale

Used Optimizations (Symmetric primitives):

- Tree optimisation: one-tree technique [BBM$^+$24]

  [BBM$^+$24] Baum, Beullens, Mukherjee, Orsini, Ramacher, Rechberger, Roy, Scholl. *One tree to rule them all: optimizing GGM trees and OWFs for post-quantum signatures.* Asiacrypt 2024.

- PRG: based on AES-128 and Rijndael-256, instead of SHAKE
- Seed Commitment: based on AES-128 and Rijndael-256, instead of SHAKE

👉 *See SDitH's specifications for details.*

Round-2 Performance Updates

Performance on AVX-based CPU

We propose an optimized implementations for AVX2-based CPU.

SDitHv2 Instance		***PK Size***	***Sig. Size***	***Sig. Running time***	***Verif. Running time***
NIST I	Short	70 B	3 705 B	≈ 31.7 Mcycles	≈ 27.2 Mcycles
	Fast		4 484 B	≈ 9.3 Mcycles	≈ 8.2 Mcycles
NIST III	Short	98 B	7 964 B	≈ 189.7 Mcycles	≈ 176.6 Mcycles
	Fast		9 916 B	≈ 28.9 Mcycles	≈ 25.9 Mcycles
NIST V	Short	132 B	14 121 B	≈ 271.6 Mcycles	≈ 254.4 Mcycles
	Fast		17 540 B	≈ 43.4 Mcycles	≈ 39.5 Mcycles

Benchmark from PQ-Sort (https://pqsort.tii.ae)

Signature size: saving between 56% and 61%

Advantages & Limitations

Advantages & Limitations

- **Advantages**:
  - Conservative assumption: based on the oldest code-based hard problem

    The ***unstructured binary*** syndrome decoding problem
  - Adaptive and tunable parameters
  - Competitive code-based signatures: 3.7 KB for L1
  - Very small public keys: around 120~240 bytes
  - Competitive signature + public key size: for L1 short, 3.8 KB
    - 3.7 KB for ML-DSA, and 7.8 for SLH-DSA

- **Limitations**:
  - Quadratic growth w.r.t. to the security level
  - Limited performance: slow compared to lattice-based schemes, but competitive with many other post-quantum signature schemes.