Cross-Signature Signing-Key Recovery and Domain-Separation Repair for SDitH v2

José Luis Delgado

Independent Researcher

Abstract. SDitH is a code-based signature scheme built from the regular syndrome decoding (RSD) problem. Each v2 signature masks the long-term witness with pseudorandom leaf-seed shares. In each repetition, cancelling the shares reconstructed by the verifier leaves the witness XORed with the hidden seed's PRG output. The VOLEitH PRG runs AES or Rijndael in counter mode from zero, so every signature and leaf uses the same input. One encryption therefore tests a candidate seed against every exposed value. RSD elementary-vector constraints discard false candidates before the seed commitment confirms a match. A matching hidden seed reveals the long-term witness and yields a signing key.

With $q = 2^{12}$ chosen-message signatures, the attack costs 11.23–11.67 bits less than AES-128/192/256 exhaustive key search across all six parameter sets. Both use the same Boolean-circuit model, and the attack cost extends through fresh forgery. Reduced-keyspace executions against the official C implementation recover witnesses of 616, 876, and 1,144 bits and produce accepted fresh signatures for every parameter set. Using the signature salt as the IV leaves one signature's repetitions under one input and retains 1.95–3.29 bits of separation. Combining salt, BAVC leaf index, and block position assigns a separate counter range to each PRG call without changing signature size or the number of block-cipher calls.

Artifact. The code for reproducing the attack, cost estimates, and domain-separation repair is available at

https://github.com/hypergalois/SDitHCrossSignature.

1 Introduction

Every SD-in-the-Head-2 (SDitH v2) signature uses the same long-term witness. The signer masks it with pseudorandom shares generated from fresh leaf seeds and opens every leaf except one in each repetition. Once the verifier reconstructs and cancels the opened shares, the remaining public value is the witness XORed with the PRG output of the hidden seed.

The released VOLEitH PRG uses the leaf seed as an AES or Rijndael key and starts counter mode from the all-zero initialization vector (IV). Because the input contains no signature or leaf identifier, a candidate seed produces the same output block wherever it is tested. One encryption under that seed can be XORed with the public values from every signature and repetition, and the correct seed exposes the same long-term witness in each case.

The structure of the witness supplies an inexpensive test for this search. It encodes a solution to the public regular syndrome decoding (RSD) instance, with each RSD chunk represented as a tensor product of elementary vectors. The Hamming-weight constraints

E-mail: jose@delgado.fyi (José Luis Delgado)

on a few consecutive chunks reject almost every wrong candidate before the published seed commitment confirms a survivor. The recovered witness fits directly into the secret-key format of the reference implementation and produces a valid signature on a fresh message.

Each chosen-message signature contributes one hidden seed per repetition, and reaching any one of these seeds recovers the witness. With q signatures and τ repetitions, a single PRG evaluation therefore tests $q\tau$ targets, reducing the search range for success probability $1/2$ by approximately the same factor. The public tests preserve this saving by reusing common Boolean terms and public prefixes, while Gray-code order reduces the work between consecutive candidate seeds.

This construction gives the first cross-signature signing-key recovery attack on SDitH v2. With $q = 2^{12}$ chosen-message signatures, recovery and fresh forgery cost 11.23–11.67 bits less than AES-128, AES-192, or AES-256 exhaustive key search across all six parameter sets, a factor of 2,403–3,250. The attack and AES baseline use the same circuit library, key schedules, candidate order, controller, early filtering, and delayed validation. The attack total also includes signature acquisition, target identification, commitment checks, witness reconstruction, and fresh signing.

SDitH is a code-based signature scheme that combines the RSD relation with a vector oblivious linear evaluation (VOLE)-in-the-Head proof and the Fiat–Shamir transform [MBB+25]. NIST selected it for the third round of the Additional Digital Signatures process [ABC+26]. Our analysis covers version 2.0, released in February 2025, including the short and fast variants in all three security categories.

Reduced-keyspace experiments exercise the path from serialized signatures to forgery while keeping transcript parsing, joint filtering, commitment checking, witness reconstruction, signing-key construction, and verification unchanged. The official C implementation accepts the resulting signature on a fresh message in every parameter set.

The source of the attack also determines the required domain separation. Initializing the PRG with the signature salt separates different signatures but leaves the repetitions of one signature together, giving a residual attack 1.95–3.29 bits below the AES baseline. Adding the BAVC leaf index and block position gives each PRG call a separate counter range while preserving the signature format and the number of block-cipher calls. Under the same circuit comparison, the adapted witness test then costs 0.107–0.127 bits more than generic multi-target search.

1.1 Contributions

1. We give a cross-signature attack that recovers the long-term witness from SDitH v2 signatures and produces an accepted signature on a fresh message. At $q = 2^{12}$, its circuit cost is 11.23–11.67 bits below AES exhaustive key search in every parameter set under the same circuit model.

2. We derive the attack from the serialized transcript, analyze its success probability and Boolean-circuit cost, and implement the path from transcript parsing to forgery against all six official reference parameter sets.

3. We analyze the target reuse left by a salt-only change and separate the PRG inputs by signature, leaf, and block position. The resulting construction preserves the interface and removes the attack's advantage over generic multi-target key search in the ideal-cipher model.

2 Attack overview

2.1 A four-leaf example

Consider a repetition with four leaf seeds s_0, s_1, s_2, s_3. On the witness coordinates, seed s_a generates a pseudorandom share $r_a = G_{\text{wit}}(s_a)$. Suppose that the challenge hides s_2 and opens the other three seeds. The signature also publishes the witness masked by the XOR of all four shares:

$$B = \mathsf{wit} \oplus r_0 \oplus r_1 \oplus r_2 \oplus r_3.$$

The opened seeds reveal r_0, r_1, r_3. XORing these known shares with B cancels them and leaves the public value

$$A = B \oplus r_0 \oplus r_1 \oplus r_3 = \mathsf{wit} \oplus G_{\text{wit}}(s_2).$$

This public masked-witness value becomes the target: for a candidate seed s, the attacker computes $A \oplus G_{\text{wit}}(s)$, which equals the secret witness when $s = s_2$. The elementary-vector constraints in a short witness prefix reject most wrong candidates, and the published commitment to s_2 identifies the correct seed among the survivors. The public RSD instance checks the recovered witness, which is then inserted into the signing-key format used by the official implementation.

2.2 From one repetition to many signatures

Each repetition can contain more than four leaves, but the cancellation is unchanged: all opened shares disappear and leave one hidden share. SDitH stores the masked witness for repetition zero and publishes a BLC auxiliary value for every later repetition. Applying this value before cancelling the opened shares gives the same form of public value for every hidden seed.

The PRG uses its seed as the block-cipher key and starts counter mode at zero, without incorporating the signature salt or leaf position. A candidate seed therefore produces the same $G_{\text{wit}}(s)$ wherever it is tested.

Two signatures with two repetitions each therefore yield four target values $A_1, \ldots, A_4$ and four hidden seeds. One evaluation of $G_{\text{wit}}(s)$ tests the candidate against all four equations

$$A_t = \mathsf{wit} \oplus G_{\text{wit}}(s_t), \qquad 1 \le t \le 4.$$

Finding any one of the four seeds reveals the same witness. With q signatures and τ repetitions, the same computation tests $q\tau$ targets and reduces the search range for success probability $1/2$ by approximately that factor. The recovery algorithm evaluates the public tests jointly so that their cost stays below this gain.

2.3 SDitH data used by the attack

The public key defines an RSD instance, and the secret witness is an encoded solution reused in every signature. A signing execution creates fresh seed trees, expands the leaves into VOLE shares, and uses their XOR to mask the witness in the public circuit input of the VOLE-in-the-Head proof.

Among the τ parallel repetitions, repetition e contains 2^κ leaf seeds $s_{e,a}$ and PRG outputs $r_{e,a} = G_{\text{full}}(s_{e,a})$. Write $u_e = \bigoplus_a r_{e,a}$. The signature publishes $\mathsf{wit} \oplus u_0$ on the witness coordinates and the auxiliary value $u_e \oplus u_0$ for $e > 0$. Fiat–Shamir selects one hidden leaf in every repetition. A one-tree batched all-but-one vector commitment (BAVC) opens the siblings needed to reconstruct the other leaves and publishes a seed commitment for each hidden leaf.

The Goldreich–Goldwasser–Micali (GGM) parent-to-child expansion includes the signature salt and node position, and the seed commitment includes the salt and absolute node index. The VOLEitH PRG G_{full} uses the leaf seed as the block-cipher key with a zero IV. The GGM expansion and seed commitment therefore depend on their positions, while the PRG input that masks the witness is common to all signatures and leaf positions.

2.4 Elementary-vector encoding

The verification circuit evaluates the elementary vectors of the RSD solution through a multiplexer tree (mux tree). A mux input of arity a is encoded by $a-1$ bits constrained to Hamming weight at most one: the all-zero word selects the first branch, and the $a-1$ unit vectors select the others. The encoding accepts a strings among the 2^{a-1} possibilities.

The elementary vectors within one RSD chunk use arities $(4,4,4,3)$ in Categories I and V, giving an 11-bit chunk with 192 valid strings and density $3/32$. Category III uses four arity-4 vectors, giving a 12-bit chunk with 256 valid strings and density $1/16$. The witnesses contain 56, 73, and 104 RSD chunks in Categories I, III, and V.

XORing a candidate PRG output with a public target produces a witness candidate whose first chunks can be checked before the syndrome equation. An honest seed always passes, while a false candidate passes r chunks with marginal density ρ^r. The filter also accepts the unused mixed-radix indices beyond the RSD chunk length m, which only increases the false-positive count. Survivors are checked against the seed commitment and then the public syndrome.

2.5 Attack setting

Let $(\mathsf{pk},\mathsf{sk})$ be produced by the released SDitH v2 KeyGen. The adversary chooses distinct messages $M_1,\dots,M_q$ and receives

$$\sigma_j \leftarrow \mathsf{Sign}(\mathsf{sk}, M_j), \qquad 1 \le j \le q.$$

After verifying the collected signatures, it constructs a signing-capable key $\widehat{\mathsf{sk}}$, chooses a fresh message $M^\star \notin \{M_1,\dots,M_q\}$, and outputs

$$\sigma^\star \leftarrow \mathsf{Sign}(\widehat{\mathsf{sk}}, M^\star).$$

It wins when $\mathsf{Verify}(\mathsf{pk}, M^\star, \sigma^\star) = 1$. The source messages are fixed before the signatures are returned, so the attack is nonadaptive and fits the standard existential-unforgeability under chosen-message attack (EUF-CMA) game. Its $q = 2^{12}$ queries lie within NIST's evaluation regime of at most 2^{64} chosen-message signatures [Nat22].

For one parameter set, $\lambda \in \{128, 192, 256\}$ is the seed length, τ is the number of repetitions, and 2^κ is the number of leaves per repetition. The one-tree BAVC has $n = \tau 2^\kappa$ leaves per signature. Set

$$N = 2^\lambda, \qquad M = q\tau.$$

The ideal-cipher block size is $b = 128$ in Category I and $b = 256$ in Categories III and V.

For signature j and repetition e, let $d_{j,e}$ be the hidden position, $i_{j,e}$ its absolute GGM-tree node index, $s_{j,e}$ its seed, and $C_{j,e} \in \{0,1\}^{2\lambda}$ its commitment. The witness projection of $G_{\text{full}}(s)$ is $G_{\text{wit}}(s)$, and wit denotes the long-term encoded witness.

If an RSD chunk contains elementary vectors of arities a, define

$$\ell = \sum_a (a-1), \qquad \rho = \frac{\prod_a a}{2^\ell}. \tag{1}$$

Thus $(\ell,\rho) = (11, 3/32)$ in Categories I and V and $(\ell,\rho) = (12, 1/16)$ in Category III. Table 1 collects the remaining dimensions and the number M of targets at $q = 2^{12}$.

Table 1: SDitH v2 parameters and targets from 2^{12} signatures.

Set	λ	τ	κ	witness bits	chunks	M
CAT1-short	128	11	11	616	56	45,056
CAT1-fast	128	16	8	616	56	65,536
CAT3-short	192	16	12	876	73	65,536
CAT3-fast	192	24	8	876	73	98,304
CAT5-short	256	21	12	1,144	104	86,016
CAT5-fast	256	32	8	1,144	104	131,072

All XOR equations use the serialized coordinates of the implementation, including the category-specific partial final byte. An accepted signature is an honest signing-oracle output accepted by the released verifier.

3 Transcript extraction

The four-leaf example in Section 2 omitted the tree indices and BLC auxiliary values. Including these fields yields the same masked-witness target and its seed commitment directly from an accepted serialized SDitH v2 signature.

3.1 Hidden seeds and commitments

Signature j contains a salt salt_j, a one-tree opening, and 2λ commitment bits for each of its τ hidden leaves. Its proof hash and public 32-bit counter determine the hidden positions $d_{j,e}$. The released function `hidden_leaves_indexes2` maps repetition e and position $d_{j,e}$ to the absolute node index

$$i_{j,e} = \tau 2^{\kappa} + \tau d_{j,e} + e, \qquad 0 \le e < \tau. \tag{2}$$

The opening routine sorts these indices, but the residue modulo τ recovers the repetition. The verifier can therefore associate each hidden commitment $C_{j,e}$ with salt_j and $i_{j,e}$.

3.2 Target value

Let $s_{j,e,a}$ denote leaf a in repetition e of signature j, and write

$$r_{j,e,a} = G_{\text{full}}(s_{j,e,a}), \qquad u_{j,e} = \bigoplus_{a} r_{j,e,a}.$$

The signature contains

$$\mathsf{aux}_{j,0} = 0, \qquad \mathsf{aux}_{j,e} = u_{j,e} \oplus u_{j,0} \quad (e > 0), \tag{3}$$

and the public circuit input satisfies, on the witness coordinates,

$$B_j = \texttt{circuit_in_pub}_j = \mathsf{wit} \oplus u_{j,0}. \tag{4}$$

The one-tree opening reconstructs every leaf except $s_{j,e} = s_{j,e,d_{j,e}}$. The witness coordinates of the opened PRG outputs give

$$o_{j,e} = \bigoplus_{a \neq d_{j,e}} G_{\text{wit}}(s_{j,e,a}).$$

Starting from B_j, the auxiliary value replaces the baseline sum $u_{j,0}$ by $u_{j,e}$, and $o_{j,e}$ cancels all opened shares. The remaining public value is

$$A_{j,e} = B_j \oplus o_{j,e} \oplus \mathsf{aux}_{j,e} \tag{5}$$
$$= \mathsf{wit} \oplus G_{\mathrm{wit}}(s_{j,e}). \tag{6}$$

For $e > 0$, the serialized auxiliary value used here is row $e - 1$.

3.3 Common PRG input

The function producing $G_{\mathrm{full}}(s)$ is initialized solely from the seed. Category I calls

`aes128_ctrle_init_key_ivzero(seed)`,

while Categories III and V use the corresponding `rijndael256_ctrle` initialization with an all-zero IV. Category III embeds the 192-bit seed as $0^{64}\|s$. Every signature and repetition thus uses the same map $s \mapsto G_{\mathrm{full}}(s)$.

The first witness coordinates lie in one PRG block: byte 2 of counter 4 in Categories I and V, and byte 2 of counter 3 in Category III. One encryption under a candidate seed supplies the 55, 60, or 66 bits used by the filter for all $M = q\tau$ target values.

3.4 Seed commitment

The seed commitment confirms a candidate that passes the witness test. Using the little-endian counter convention of the implementation, write $\boxplus$ for addition modulo 2^b and let $\overline{\mathsf{salt}}_j = \mathsf{salt}_j$ in Categories I and V and $\overline{\mathsf{salt}}_j = 0^{64}\|\mathsf{salt}_j$ in Category III. For absolute index i, the two commitment plaintexts are

$$P_{j,i,0} := \overline{\mathsf{salt}}_j \oplus \langle 2i \rangle_b, \qquad P_{j,i,1} := P_{j,i,0} \boxplus 1,$$

where $\langle 2i \rangle_b$ places the 64-bit integer $2i$ in the low word. The seed is the variable cipher key. Table 2 lists the primitive and published output bytes in each category.

Table 2: Seed-commitment map for a hidden leaf. Byte ranges follow the C source-memory representation of the two cipher outputs.

Category	Variable key	Primitive	Published bytes
I	$s \in \{0,1\}^{128}$	AES-128, two blocks	block 0: 0–15; block 1: 0–15
III	$0^{64}\|s$, $s \in \{0,1\}^{192}$	Rijndael-256-256, two blocks	block 0: 8–31; block 1: 0–23
V	$s \in \{0,1\}^{256}$	Rijndael-256-256, two blocks	block 0: 0–31; block 1: 0–31

Every hidden leaf supplies the public test

$$\mathsf{CommitSeed}_\lambda(s; \mathsf{salt}_j, i_{j,e}) = C_{j,e}. \tag{7}$$

In Category III, the published 48 bytes start at offset 8 of the 64-byte output buffer. With least-significant-bit-first numbering within each byte, these are bits 64–255 of the first block and bits 0–191 of the second.

3.5 Extraction and signing-key recovery

Theorem 1 (Cross-signature extraction)**.** *For q honest signatures accepted by the released SDitH v2 verifier, the public key and signatures determine $(i_{j,e}, C_{j,e}, A_{j,e})$ for all $q\tau$ hidden leaves. The function G_{wit} is common to all of them. If $s = s_{j,e}$ is an honest hidden seed, then*

$$\widehat{\mathsf{wit}}_{j,e}(s) = A_{j,e} \oplus G_{\mathrm{wit}}(s)$$

is the encoded witness, including its final partial byte.

Proof. Equation (2) and the BAVC fields determine $(i_{j,e}, C_{j,e})$. The opening supplies every non-hidden seed and hence $o_{j,e}$. The public circuit input and auxiliary value determine $A_{j,e}$ through Equation (6). Substituting the honest seed cancels its PRG output and leaves wit. □

The released secret-key buffer has the form

$$\mathsf{sk} = \mathsf{skey_seed} \parallel \mathsf{pkey_seed} \parallel \mathsf{skey_encoded_solution} \parallel \mathsf{pkey_y}. \tag{8}$$

The last three fields are the public matrix seed, the recovered witness, and the public syndrome. The attacker sets `skey_seed` to 0^λ, copies the public fields, inserts $\widehat{\mathsf{wit}}$, and calls the official signer, whose working state is determined by the remaining fields.

Corollary 1 (Signing-key recovery)**.** *Reaching any honest hidden seed among the $q\tau$ targets gives the adversary a secret-key encoding accepted by the reference implementation for the original public key and a valid signature on a fresh message.*

Figure 1 traces the path from accepted signatures to a fresh forgery.

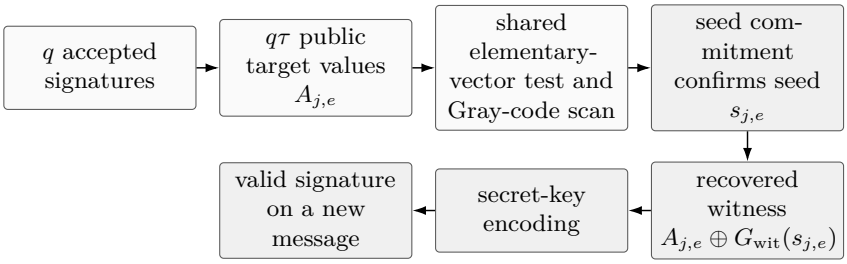


Figure 1: Cross-signature recovery. One candidate PRG output is tested against all $q\tau$ target values; the seed commitment confirms a survivor before witness and signing-key reconstruction.

4 Signing-key recovery

A candidate seed passes through two public tests. XORing its first PRG output block with a target value checks the beginning of the encoded witness; a candidate that passes this inexpensive filter is then checked against the commitment to the corresponding hidden leaf. Expansion of the remaining PRG output and reconstruction of the signing key follow only after a commitment match.

4.1 Recovery algorithm

The adversary obtains and verifies q signatures on distinct messages, computes the $M = q\tau$ target values from Equation (6), and then applies Algorithm 1.

Algorithm 1 Cross-signature signing-key recovery

Require: Public key, q accepted signatures, candidate set $S \subseteq \{0,1\}^\lambda$ of size P
Ensure: A signing key and an accepted signature on a fresh message
Parse each signature and reconstruct the $M = q\tau$ target values and their seed commitments.
for all $s \in S$ in reflected Gray-code order **do**
  Compute the PRG block containing the first witness coordinates.
  Test the selected RSD chunks against all target values.
  for all targets that pass the elementary-vector test **do**
    Check the two-block seed commitment under s.
    if the commitment matches **then**
      Recover $\widehat{\mathsf{wit}} = A_{j,e} \oplus G_{\mathrm{wit}}(s)$ and verify the public RSD relation.
      Insert $\widehat{\mathsf{wit}}$ into the secret-key format, sign a fresh message, and return after verification succeeds.
    end if
  end for
end for

The loop evaluates one PRG block for each candidate seed and expands the full PRG output only after the commitment matches.

4.2 Elementary-vector filter

Let $W_r \subseteq \{0,1\}^{r\ell}$ be the set of strings accepted by the elementary-vector constraints in the first r RSD chunks. From Equation (1), its density is

$$\frac{|W_r|}{2^{r\ell}} = \rho^r = \begin{cases} (3/32)^r, & \text{Category I, Category V,} \\ (1/16)^r, & \text{Category III.} \end{cases} \tag{9}$$

The calculation uses five chunks in Categories I and III and six chunks in Category V. These chunks occupy 55, 60, and 66 bits, respectively, all within one PRG block.

Index the public target values by t. If $A_t^{(r)}$ and $X^{(r)}(s)$ denote the selected coordinates of the target and candidate PRG output, the initial test is

$$F_t(s) = 1 \quad \Longleftrightarrow \quad A_t^{(r)} \oplus X^{(r)}(s) \in W_r. \tag{10}$$

The honest hidden seed always passes because the XOR is the true witness prefix. A false candidate passes one target with probability at most ρ^r in the analysis model. Section 5 explains how all M tests are evaluated jointly.

4.3 Search range

Let $N = 2^\lambda$ be the seed domain and let a fixed set S contain P candidates. For M uniform target seeds, the probability that at least one lies in S is

$$p_{\mathrm{cap}}(P) = 1 - \left(1 - \frac{P}{N}\right)^M. \tag{11}$$

The leading search range for success probability $1/2$ is therefore

$$P \simeq \frac{\ln 2}{M}\, 2^\lambda. \tag{12}$$

Every collected signature multiplies M by τ, so the main $q = 2^{12}$ experiment searches about $q\tau$ times fewer candidate seeds than a single-target search over half the seed space.

The formal analysis uses a slightly larger integer value of P to cover seed collisions, list sizes, false commitment matches, transcript processing, and fresh signing. Appendix A gives the model and probability bounds, and Appendix B gives the corresponding work calculation.

4.4 Recovery result

Theorem 2 (Cross-signature key recovery)**.** *In the ideal-cipher and random-oracle model of Appendix A, Algorithm 1 with $q = 2^{12}$ chosen-message signatures recovers a signing key and produces an accepted signature on a fresh message with probability at least one half. Across the six SDitH v2 parameter sets, its Boolean-circuit cost is 11.23–11.67 bits below the AES key-search baseline defined in Section 5.5.*

Proof. A target seed in the search range passes the elementary-vector test by Theorem 1, matches its public commitment, and yields the signing key of Corollary 1. The search range gives capture probability greater than one half by a 2^{-64} margin. The seed-collision, capacity, false-match, and final-signing bounds in Appendix A have total probability below that margin. The circuit derivation in Appendix B produces the costs reported in Section 6. □

5 Joint evaluation of the public tests

The $M = q\tau$ hidden seeds shorten the search range by approximately a factor of M, but testing every target with a separate circuit would spend the same factor again. The attack shares this work in two stages: for one candidate seed, the elementary-vector tests reuse their Boolean terms and common public prefixes; between candidates, Gray-code order changes one seed bit at a time, so only gates that depend on that bit are reevaluated.

5.1 Candidate PRG block

The only variable inputs to the initial test are the λ bits of the candidate seed. The plaintext is the public counter whose output contains the first witness coordinates. Category I uses AES-128, while Categories III and V use Rijndael-256-256. Category III embeds its 192-bit seed as $0^{64} \| s$, following the SDitH v2 implementation [MBB+25]. The circuits use the released key schedules, round counts, and ShiftRows variants. The AES path follows FIPS 197 [Nat23]; its S-box and MixColumns networks use published circuits [BP11, Max19, Nat26].

One encryption supplies the selected PRG bits for every target. Each target changes only the public mask XORed into those bits. Scheme constants and transcript values are fixed circuit inputs, and all gates in the relevant dependency paths are included in the cost.

5.2 Shared elementary-vector tests

Consider first a three-bit vector $y = (y_0, y_1, y_2)$ from an elementary-vector encoding. Validity requires at most one set bit and is computed by

$$U_3(y) = 1 \oplus y_0y_1 \oplus y_0y_2 \oplus y_1y_2. \tag{13}$$

For a target with public mask a_t, the test substitutes $y = x \oplus a_t$. Across all eight possible masks, the quadratic terms in x are computed once. The eight shifted tests together use three AND gates and twelve XOR or NOT gates.

Categories I and V also contain a two-bit selector, whose predicate is

$$U_2(y) = 1 \oplus y_0y_1. \tag{14}$$

All four public shifts of this predicate use seven gates together. Exhaustive truth tables for both selector sizes agree with direct Hamming-weight tests.

The same sharing extends from one elementary vector to an RSD chunk and then across the selected chunks. Public masks that begin with the same vector values reuse the conjunction for that prefix. A prefix tree stores these common computations, and a second prefix tree combines the selected RSD chunks. The root output says whether any target passed. When it is true, following the true branches identifies the corresponding target indices for commitment validation. Appendix B gives the number of terms at each depth, the traversal bound, and the resulting identification cost.

5.3 Gray-code updates

Let $g(x) = x \oplus (x \gg 1)$ be reflected Gray code. Across the candidate sequence $g(0), \ldots, g(P-1)$, logical coordinate j flips

$$f_j(P) = \left\lfloor \frac{P-1}{2^j} \right\rfloor - \left\lfloor \frac{P-1}{2^{j+1}} \right\rfloor . \tag{15}$$

Let C contain the gates needed for the final target flag, and let D_i be the gates in C that depend on physical seed bit i. Set $c_i = |D_i|$. A public permutation π maps the most frequently changing Gray coordinates to the seed bits with the smallest values of c_i.

Lemma 1 (Gray-code update cost)**.** *After one evaluation of C, the flags for a Gray-code prefix of P candidates can be evaluated with*

$$|C| + \sum_{j=0}^{\lambda-1} f_j(P) c_{\pi(j)} \tag{16}$$

fan-in-two gate evaluations and one stored bit per gate in C.

Proof. One Gray-code transition changes a single input bit. Gates that do not depend on that bit keep their values. Reevaluating the affected gates in topological order produces the circuit state for the next candidate. Summing these updates over all transitions gives Equation (16). Pairing the largest flip counts with the smallest dependency sets minimizes this sum by the rearrangement inequality. □

A ripple incrementer and binary-to-Gray conversion cost at most $3\lambda - 3$ gates per transition. The calculation charges 4λ controller gates and one match-accumulator gate for every candidate, including the first.

5.4 Commitment validation

Only targets that pass the elementary-vector test reach commitment validation. For each surviving candidate and target, the circuit expands the variable-key schedule, evaluates the two public commitment plaintexts, projects the published output coordinates, and compares all 2λ bits. The two-block costs are 51,992 gates in Category I, 142,928 in Category III, and 145,440 in Category V. A full PRG output is expanded only after this comparison succeeds.

Appendix A bounds the number of candidates that reach this stage, including the keys used while producing the collected signatures. Appendix B combines the candidate scan, target identification, commitment validation, transcript processing, witness reconstruction, and fresh signing.

5.5 AES exhaustive key-search baseline

The reference calculation uses ordinary AES-128, AES-192, and AES-256 key expansion together with the same circuit library, Gray-code updates, 4λ-gate controller, and delayed validation. Three public plaintext/ciphertext pairs determine one key. An eight-bit equality test on the first pair filters candidates before full comparison; the second and third encryptions are evaluated only for survivors of the preceding comparisons. The filter width and location are selected after evaluating every width from 1 through 32 at every legal output position.

Table 3: AES key search under the same Boolean-circuit model.

Primitive	filter start	filter width	$\log_2 G_{\mathrm{AES}}$
AES-128	0	8	141.259
AES-192	80	8	205.379
AES-256	80	8	269.734

A wrong candidate satisfies all three pairs with probability low enough that the total false-key probability is below 2^{-128} in every category. For a multi-target reference, M independent AES keys encrypt one common plaintext and any matching key is accepted. This search uses the same $(\ln 2)2^{\lambda}/M$ leading search range as the SDitH attack and tests the candidate ciphertext against the public set. Comparing with this reference separates the saving from having many targets from the cost of recognizing an SDitH witness.

Published full-round biclique tradeoffs with at most 2^{64} data save at most 1.02 bits over half-domain single-key search, while full-codebook variants save less than 1.7 bits [BCGS14]. Reducing each AES reference here by an additional factor of 2^{10} leaves 1.23–1.67 bits of separation across the six parameter sets.

5.6 Primitive-call comparison

The primitive-call comparison uses the following query bound for single-key search.

Lemma 2 (Single-key classical query bound)**.** *Let K be uniform in a key domain of size N, and give an adversary any fixed number $1 \le p \le 3$ of distinct plaintext/ciphertext pairs under the ideal permutation indexed by K. If it makes Q classical forward or inverse ideal-cipher queries and outputs K with probability at least one half, then $Q \ge N/2 - 1$.*

Proof. Reveal, with the first query under a candidate key k, whether $k = K$, which can only help the adversary. After Q queries, membership has been tested for at most $d \le Q$ keys. A tested key is correct with probability d/N. If all tests fail, K is uniform among the $N - d$ untested keys, so a final guess has probability at most $1/(N - d)$. The total success probability is at most

$$\frac{d}{N} + \left(1 - \frac{d}{N}\right)\frac{1}{N-d} = \frac{d+1}{N} \le \frac{Q+1}{N}.$$

□

Single-key AES search with success probability $1/2$ therefore needs at least $2^{\lambda-1} - 1$ candidate-key queries. At $q = 2^{12}$, the SDitH attack has a 13.78–16.09-bit advantage when only primitive calls are counted. Its larger Boolean-circuit cost includes the work needed to recognize a witness prefix, identify the matching target, and validate the seed commitment.

6 Attack cost

6.1 Main result at $q = 2^{12}$

Table 4 gives the cost of recovering the signing key and producing a fresh signature for every SDitH v2 parameter set. The circuit separation subtracts the attack cost from the AES baseline cost in Table 3. The query separation compares the number of block-cipher calls with the $2^{\lambda-1}$ lower bound of Lemma 2.

Table 4: Cross-signature attack with $q = 2^{12}$. All logarithms are base two.

Set	targets	chunks	$\log_2 G$	circuit sep.	query sep.	MiB
CAT1-short	45,056	5	130.029	11.231	13.783	6.66
CAT1-fast	65,536	5	129.957	11.302	13.993	9.66
CAT3-short	65,536	5	194.037	11.342	15.207	12.76
CAT3-fast	98,304	5	193.909	11.470	15.654	19.09
CAT5-short	86,016	6	258.146	11.588	15.618	21.43
CAT5-fast	131,072	6	258.068	11.666	16.089	32.59

The circuit separation is at least 11.23 bits in all six cases, corresponding to a factor of 2,403 for CAT1-short and 3,250 for CAT5-fast. Counting only primitive calls gives a larger separation because the circuit total also includes the Boolean work required to test the witness encoding and identify the target whose commitment must be checked.

The largest memory use is 32.59 MiB for CAT5-fast. This storage contains the circuit state for the current candidate, the public target data, the prefix trees, and the commitment data. Candidate seeds are generated as a stream, so their storage does not grow with the search range.

An independent ten-bit reduction of every AES circuit cost leaves a separation of 1.23–1.30 bits in Category I, 1.34–1.47 bits in Category III, and 1.59–1.67 bits in Category V. This independent ten-bit reduction exceeds the published full-codebook biclique saving discussed in Section 5.5 by more than eight bits.

6.2 Number of chosen-message signatures

The number of targets grows linearly with the number of chosen-message signatures. The $q = 2^{12}$ point is the main result, providing more than eleven bits of circuit separation with less than 33 MiB of memory. Table 5 gives two additional query counts. The $q = 2^{10}$ rows show that the separation is already close to the main result, while $q = 2^{20}$ increases it to 11.71–12.14 bits at a maximum memory cost below 8.15 GiB.

Table 5: Circuit separation as the number of chosen-message signatures changes.

q	smallest separation	largest separation
1,024	10.521	11.268
4,096	11.231	11.666
1,048,576	11.708	12.135

All rows use the same attack and comparison. Five RSD chunks minimize the cost in Categories I and III, giving 55 and 60 filter bits, while six chunks minimize Category V with 66 filter bits. Appendix B reports the candidate counts, individual cost components, memory calculation, and filter selection.

7 Domain separation for the VOLEitH PRG

7.1 Salt-only domain separation

The attack uses one candidate encryption for every signature and repetition because all VOLEitH PRG calls begin at the same counter. Using the signature salt as the IV creates a separate PRG input for each signature. The τ hidden leaves within that signature still share the input and form the targets for one search.

For a primitive input shared by h uniform hidden seeds, the search range for success probability $1/2$ in a seed space of size $N = 2^\lambda$ is

$$P_h = \left\lceil N\left(1 - 2^{-1/h}\right) \right\rceil . \tag{17}$$

The three relevant choices of PRG input are summarized in Table 6.

Table 6: Target multiplicity under three choices of VOLEitH PRG input.

PRG input	domains	targets per input	search range ($p = 1/2$)
zero IV	1	$q\tau$	$P_{q\tau}$
signature salt	q	τ	P_τ
salt and BAVC leaf index	$q\tau$	1	$N/2$

With the salt-only IV, the attacker selects one collected signature and tests its τ repetitions together. The smaller search range survives the cost of the witness and commitment tests, leaving the separations in Table 7.

Table 7: Attack cost when the VOLEitH PRG uses the signature salt as its IV.

Set	chunks	$\log_2 G$	circuit separation
CAT1-short	4	138.505	2.754
CAT1-fast	4	137.973	3.287
CAT3-short	4	203.424	1.954
CAT3-fast	4	202.843	2.535
CAT5-short	5	267.045	2.689
CAT5-fast	5	266.443	3.291

The seed commitment already contains the salt and absolute node index, but it is evaluated only after the shared elementary-vector test. Eliminating the remaining target multiplicity requires the PRG input itself to identify the leaf.

7.2 Per-leaf counter ranges

Let $a = i - \tau 2^\kappa = \tau d + e$ be the zero-based BAVC leaf index used by the VOLE loop, let c be the block position in that leaf's PRG output, and let b be the cipher block size. Every parameter set uses at most nine blocks per leaf, so sixteen consecutive counters are reserved for each leaf.

Write T_b for the block whose most significant bit is one. Let $\boxplus c$ denote the little-endian counter addition used by the released implementation, let clr_4 clear the four least significant salt bits, and let $\langle x \rangle_\lambda$ encode a 64-bit integer in the low word of a λ-bit IV. The initial counter and subsequent PRG inputs are

$$I_{j,a} := \overline{\mathsf{clr}_4(\mathsf{salt}_j) \oplus \langle 16a \rangle_\lambda} \oplus T_b, \tag{18}$$

$$X_{j,a,c} := I_{j,a} \boxplus c, \qquad 0 \le c < 16, \tag{19}$$

$$G_{j,a}(s) := \mathsf{Enc}_s(X_{j,a,0}) \| \mathsf{Enc}_s(X_{j,a,1}) \| \cdots . \tag{20}$$

The overline is the block-sized salt representation from Section 3. Clearing four bits aligns the sixteen-counter ranges and leaves at least 124 salt bits. The largest one-tree instance has fewer than 2^{17} leaves.

The embedding is the identity in Categories I and V. Category III places a zero 64-bit low word before the 192-bit IV in the 256-bit little-endian counter; the top bit of IV byte 23 becomes T_b in byte 31 of the embedded block.

Within one signature, consecutive leaf bases differ by a multiple of sixteen. In Categories I and V, the four cleared low bits keep additions $0 \le c < 16$ within the assigned range. Category III uses the low zero word for the counter and the following word for the leaf label. The bit T_b lies outside the counter, leaf, node, and commitment-tweak fields. It also distinguishes VOLEitH PRG inputs from the GGM and commitment inputs used under the same leaf seed.

Both signer and verifier know the salt, BAVC leaf index, and block position. Replacing zero-IV initialization by $I_{j,a}$ preserves the signature format and the number of block-cipher calls, and it uses the existing PRG output interface. The BLC auxiliary values and the remaining polynomial interactive oracle proof (PIOP) computation are unchanged.

For the hidden BAVC leaf index $a_{j,e} = \tau d_{j,e} + e$, cancellation of the opened shares now gives

$$A_{j,e} = \mathsf{wit} \oplus \left(G_{j,a_{j,e}}(s_{j,e})\right)_{\mathrm{wit}}. \tag{21}$$

The witness relation is preserved, but testing a candidate against several leaves requires a separate encryption for each leaf input. The key schedule can still be shared across these encryptions.

7.3 Cost after domain separation

We compare the adapted witness test with a generic multi-target key search that shares the same candidate key schedule across all labelled inputs. Both circuits use the same Gray-code updates, controller, eight-bit initial equality test, and delayed validation. Table 8 gives the costs at $q = 2^{12}$.

Table 8: Witness filtering and generic multi-target key search after domain separation at $q = 2^{12}$.

Set	witness test $\log_2 G$	generic search $\log_2 G$	difference
CAT1-short	141.545	141.419	0.127
CAT1-fast	141.545	141.419	0.127
CAT3-short	207.046	206.938	0.107
CAT3-fast	207.046	206.938	0.107
CAT5-short	271.050	270.938	0.111
CAT5-fast	271.050	270.938	0.111

The adapted witness test is 0.107–0.127 bits more expensive than the generic key search. At the primitive level, both evaluate approximately $(\ln 2)2^{\lambda}$ filter-block encryptions across the labelled inputs. This is 0.471 bits above the $2^{\lambda-1}$ cost of selecting one target and searching its domain. Appendix C analyzes adaptive queries, pairwise combinations of targets, the one-tree opening, and the shared key schedule after this change.

8 Implementation and experiments

The implementation follows the attack from serialized SDitH signatures to an accepted forgery. A C harness links directly to the official SDitH v2 reference objects for parsing, tree and share expansion, VOLE reconstruction, signing, and verification. Independent Python

code constructs the Boolean circuits and evaluates the probability and cost formulas. Tests against the C objects fix the byte, bit, counter, and key-embedding conventions shared by the two implementations.

8.1 Recovery and forgery in reduced key spaces

For each parameter set, the harness generates one key pair and 64 valid signatures on distinct messages. It parses the public key and serialized signatures, then reconstructs all 64τ target values and their two-block seed commitments.

To make exhaustive execution feasible, one honest target is placed in an eight-bit seed subspace by fixing the other $\lambda - 8$ seed bits. The filter widths are unchanged from the full-keyspace calculation. For each candidate, one PRG block is tested against the entire target table. A candidate that passes the prefix test triggers the two-block commitment check, and the full PRG output is expanded only after that check succeeds. The harness rebuilds the secret-key buffer, signs a fresh message, and passes the signature to the official verifier.

Table 9 reports the six executions. In every row, one candidate reaches witness expansion and the resulting key produces a fresh accepted signature.

Table 9: Reduced-keyspace recovery against the official C implementation. The targets and chunk counts correspond to the 64-signature experiments.

Set	signatures	targets	chunks	outcome
CAT1-short	64	704	5	key recovery and forgery
CAT1-fast	64	1,024	5	key recovery and forgery
CAT3-short	64	1,024	5	key recovery and forgery
CAT3-fast	64	1,536	5	key recovery and forgery
CAT5-short	64	1,344	6	key recovery and forgery
CAT5-fast	64	2,048	6	key recovery and forgery

The recovered witness agrees byte for byte with the witness generated by KeyGen. After each forgery, the harness changes the fresh message and one proof byte in separate tests; the official verifier rejects both modifications.

8.2 Repair experiments

The artifact contains a salt-only source patch and the per-leaf repair from Section 7. The first initializes the VOLEitH PRG from the serialized salt. The second passes the salt to the VOLE expansion, constructs the IV in Equation (18), and invokes the existing PRG initializer and output routine. The per-leaf patch is compiled separately in all six reference source trees; the salt-only patch modifies their shared implementation.

The reduced-keyspace harness runs the original, salt-only, and per-leaf variants with two signatures and 256 candidate keys. One target seed lies in the searched subspace. Each candidate requires one filter-block encryption in the original PRG input, one per signature after adding the salt, and one per hidden leaf after adding the BAVC leaf index. The candidate key schedule is expanded once in all three variants. Table 10 reports the measured counts.

Table 10: Filter-block encryptions in the reduced-keyspace repair experiments. Each of the 18 runs recovers the witness and produces a valid forgery.

Set	targets	zero IV	salt only	per-leaf input
CAT1-short	22	256	512	5,632
CAT1-fast	32	256	512	8,192
CAT3-short	32	256	512	8,192
CAT3-fast	48	256	512	12,288
CAT5-short	42	256	512	10,752
CAT5-fast	64	256	512	16,384

All three variants preserve the serialized signature length. Patched signing, patched verification, transcript reconstruction, and the corresponding attack code agree in every run. Table 8 gives the full-keyspace circuit comparison.

8.3 Primitive and encoding tests

For each category, the official implementation evaluates eight deterministic seed, salt, and index cases with two primitive inputs. The independent Python circuits agree in all 144 byte-for-byte comparisons. Category III separately tests the $0^{64}\|s$ key embedding and its 192-bit commitment projections. A FIPS-197 AES-128 known-answer vector fixes the state, key, and bit-order conventions [Nat23].

The test suite evaluates every input to the shifted two- and three-bit elementary-vector predicates and compares each target prefix tree with direct language membership. It also checks the witness offsets and final-byte masks for every parameter set, the candidate-range inequality, Gray-code flip counts, dependency sets, and the values in the result tables.

9 Related work

The original SDitH construction applies multiparty computation (MPC)-in-the-Head techniques to syndrome decoding [FJR22]. Version 2 uses the VOLE-in-the-Head framework of Baum et al. [BBdSG+23]. Its RSD component uses the compact models developed for that setting [OTX25, BBGK24], and its correlated seed material follows the one-tree organization studied by Baum et al. [BBM+25]. Our analysis uses the algorithms and parameters of the v2 specification [MBB+25].

Previous attacks on SDitH use physical access or faults. Godard et al. recover the signing key of the first-round TCitH implementation from single-trace leakage in finite-field multiplication [GAG+25], while Jendral and Dubrova recover keys from faulted signatures in the second-round VOLE-in-the-Head implementations of SDitH and FAEST [JD26]. Our attack uses public, correctly generated signatures in the chosen-message model.

Dinur and Nadler recover an ephemeral PRG seed from multiple Picnic transcripts and derive the associated long-term signing key [DN19]. Their multi-target search recognizes a correct PRG output from partial transcript information. For SDitH, the elementary-vector encoding of the RSD witness supplies this test, while shared predicate terms and prefix trees evaluate it across all transcripts.

FAEST analyzes exhaustive search against its vector-commitment tree and its AES counter-mode conversion to VOLE [BBB+25]. Within one signature, several missing keys may be tested against one candidate block. Across signatures, FAEST samples a fresh IV for each transcript. SDitH v2 salts its GGM tree and seed commitments but starts the VOLEitH PRG from the all-zero IV, creating the common input used here.

Our AES key-search baseline separates the generic benefit of many winning keys from the additional cost of recognizing SDitH witnesses. A game with M independent AES keys

on one public plaintext has the same leading candidate range as the attack. The comparison in Section 5.5 then measures the Boolean work required by the elementary-vector predicates, target identification, commitment validation, and signing-key recovery.

10 Conclusion

We give the first cross-signature signing-key recovery attack on SDitH v2. Because the VOLEitH PRG starts from the all-zero IV, every hidden leaf uses the same PRG input. Cancelling the opened shares from a signature exposes the long-term witness XORed with the PRG output of its hidden seed, so one candidate encryption can be tested across every signature and repetition. The elementary-vector constraints of the RSD witness filter wrong candidates, the seed commitment identifies a match, and the recovered witness produces a signing key and an accepted signature on a fresh message. With $q = 2^{12}$ chosen-message signatures, the attack is 11.23–11.67 bits cheaper than AES exhaustive key search across all six parameter sets, a factor of 2,403–3,250. Reduced-keyspace executions against the official C implementation reproduce the path from serialized signatures to fresh forgery.

The same analysis determines the domain separation needed to remove this gain. Using the signature salt alone leaves the repetitions of one signature as targets for a common search with 1.95–3.29 bits of separation. Adding the BAVC leaf index and block position assigns a separate counter range to every VOLEitH PRG call while preserving the signature format and the number of primitive calls. After this change, the adapted witness test is 0.107–0.127 bits more expensive than generic multi-target key search, and the ideal-cipher bound reduces witness recovery to ordinary block-cipher key search or solution of the public RSD instance.

Acknowledgements

Huge thanks to Thibauld Feneuil and the whole SDitH team for their careful analysis and scrutiny; their work has been a constant source of inspiration to me. They confirmed that the attack affects the security of SDitH. The Round-3 specification acknowledges this work and implements the same domain-separation principle by initializing the VOLEitH PRG with the signature salt and repetition index.

This paper is dedicated to the victims of the Burgohondo wildfire and to the people of Ávila, my hometown, who are living with its aftermath as I write. It is the largest wildfire ever recorded in Spain.

References

[ABC+26] Gorjan Alagic, Maxime Bros, Pierre Ciadoux, Quynh Dang, Thinh Dang, John Kelsey, Jacob Lichtinger, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, Ray Perlner, Angela Robinson, Hamilton Silberg, Daniel Smith-Tone, and Noah Waller. Status report on the second round of the additional digital signature schemes for the NIST post-quantum cryptography standardization process. Technical Report NIST IR 8610, National Institute of Standards and Technology, May 2026. URL: https://csrc.nist.gov/pubs/ir/8610/final, doi:10.6028/NIST.IR.8610.

[BBB+25] Carsten Baum, Ward Beullens, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Christian Majenz, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. FAEST v2: Algorithm specifications, February 2025. Version 2.0, NIST Additional Digital Signatures Round-2 submission. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/faest-spec-round2-web.pdf.

[BBdSG+23] Carsten Baum, Lennart Braun, Cyprien Delpech de Saint Guilhem, Michael Klooß, Emmanuela Orsini, Lawrence Roy, and Peter Scholl. Publicly verifiable zero-knowledge and post-quantum signatures from VOLE-in-the-head. In *Advances in Cryptology—CRYPTO 2023, Part V*, volume 14085 of *Lecture Notes in Computer Science*, pages 581–615. Springer, Cham, 2023. doi:10.1007/978-3-031-38554-4_19.

[BBGK24] Slim Bettaieb, Loïc Bidoux, Philippe Gaborit, and Mukul Kulkarni. Modelings for generic PoK and applications: Shorter SD and PKP based signatures. Cryptology ePrint Archive, Paper 2024/1668, 2024. URL: https://eprint.iacr.org/2024/1668.

[BBM+25] Carsten Baum, Ward Beullens, Shibam Mukherjee, Emmanuela Orsini, Sebastian Ramacher, Christian Rechberger, Lawrence Roy, and Peter Scholl. One tree to rule them all: Optimizing GGM trees and OWFs for post-quantum signatures. In *Advances in Cryptology—ASIACRYPT 2024, Part I*, volume 15484 of *Lecture Notes in Computer Science*, pages 463–493. Springer, Singapore, 2025. doi:10.1007/978-981-96-0875-1_15.

[BCGS14] Andrey Bogdanov, Donghoon Chang, Mohona Ghosh, and Somitra Kumar Sanadhya. Bicliques with minimal data and time complexity for AES (extended version). Cryptology ePrint Archive, Paper 2014/932, 2014. URL: https://eprint.iacr.org/2014/932.

[BP11] Joan Boyar and Rene Peralta. A depth-16 circuit for the AES S-box. Cryptology ePrint Archive, Paper 2011/332, 2011. URL: https://eprint.iacr.org/2011/332.

[DN19] Itai Dinur and Niv Nadler. Multi-target attacks on the Picnic signature scheme and related protocols. In *Advances in Cryptology—EUROCRYPT 2019, Part III*, volume 11478 of *Lecture Notes in Computer Science*, pages 699–727. Springer, Cham, 2019. doi:10.1007/978-3-030-17659-4_24.

[FJR22] Thibauld Feneuil, Antoine Joux, and Matthieu Rivain. Syndrome decoding in the head: Shorter signatures from zero-knowledge proofs. In *Advances in Cryptology—CRYPTO 2022, Part II*, volume 13508 of *Lecture Notes in*

Computer Science, pages 541–572. Springer, Cham, 2022. doi:10.1007/978-3-031-15979-4_19.

[GAG+25] Julie Godard, Nicolas Aragon, Philippe Gaborit, Antoine Loiseau, and Julien Maillard. Single trace side-channel attack on the MPC-in-the-head framework. In *Post-Quantum Cryptography—PQCrypto 2025*, volume 15578 of *Lecture Notes in Computer Science*, pages 267–293. Springer, Cham, 2025. doi:10.1007/978-3-031-86602-9_10.

[JD26] Sönke Jendral and Elena Dubrova. Fault attacks on VOLEitH signature schemes. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, 2026(1):225–249, 2026. doi:10.46586/tches.v2026.i1.225-249.

[Max19] Alexander Maximov. AES MixColumn with 92 XOR gates. Cryptology ePrint Archive, Paper 2019/833, 2019. URL: https://eprint.iacr.org/2019/833.

[MBB+25] 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, and Dongze Yue. The syndrome decoding in the head (SD-in-the-head) signature scheme: Algorithm specifications and supporting documentation, February 2025. Version 2.0, NIST Additional Digital Signatures Round-2 submission. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-2/spec-files/sdith-spec-round2-web.pdf.

[Nat22] National Institute of Standards and Technology. Call for additional digital signature schemes for the post-quantum cryptography standardization process, October 2022. URL: https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf.

[Nat23] National Institute of Standards and Technology. Advanced encryption standard (AES). Technical Report FIPS 197-upd1, National Institute of Standards and Technology, May 2023. URL: https://csrc.nist.gov/pubs/fips/197/final, doi:10.6028/NIST.FIPS.197-upd1.

[Nat26] National Institute of Standards and Technology. Circuit complexity: List of circuits, 2026. Updated August 7, 2026. URL: https://csrc.nist.gov/projects/circuit-complexity/list-of-circuits.

[OTX25] Ying Ouyang, Deng Tang, and Yanhong Xu. Code-based zero-knowledge from VOLE-in-the-head and their applications: Simpler, faster, and smaller. In *Advances in Cryptology—ASIACRYPT 2024, Part V*, volume 15488 of *Lecture Notes in Computer Science*, pages 436–470. Springer, Singapore, 2025. doi:10.1007/978-981-96-0935-2_14.

A Probability calculations

The success bound in Theorem 2 is evaluated with integer and rational arithmetic. Logarithms are used only to display the resulting values.

A.1 Analysis model

Each block-cipher key indexes an independent uniform permutation on b bits. Category III embeds its 192-bit seed as $0^{64}\|s$ in the 256-bit key space. The Fiat–Shamir hashes and proof-counter challenge are random oracles on their absorbed byte strings. Root seeds and salts are independent uniform samples from the entropy source prescribed by the scheme. The circuit costs evaluate the released AES and Rijndael algorithms directly.

The challenge selects the hidden leaves after the signer has fixed the tree commitment. The signer accepts index tuples whose one-tree opening fits the public bound `target_topen`. This condition depends on the indices and not on the leaf-seed values, so the selected leaves are a sample from the seeds generated by the signer.

A.2 Seeds generated by the signer

One signature expands $n = \tau 2^{\kappa}$ leaves and generates at most $2n$ seed values. Across q signatures, let

$$V = 2qn, \qquad \epsilon_{\mathrm{src}} = \frac{V(V-1)}{2^{\lambda}}. \tag{22}$$

The numerator covers the switch from the keyed tree permutations to independent random functions and all collisions among generated seeds. The primitive block size is at least λ, so its switching denominator is no smaller than the one used above. Conditioned on the complement of this event, the hidden leaves are distinct uniform λ-bit seeds.

Category III embeds a seed as $0^{64}\|s$ in the Rijndael-256-256 key. Projecting an independent 256-bit random-function output to 192 bits preserves uniformity, so Equation (22) applies there as well.

Table 11: Seed-generation event bounds for $q = 2^{12}$.

Set	target seeds	$\log_2 \epsilon_{\mathrm{src}}$
CAT1-short	45,056	−73.081
CAT1-fast	65,536	−78.000
CAT3-short	65,536	−134.000
CAT3-fast	98,304	−140.830
CAT5-short	86,016	−197.215
CAT5-fast	131,072	−204.000

A.3 Search range

Let $N = 2^{\lambda}$, $M = q\tau$, and let S contain P fixed candidates. For independent uniform target seeds,

$$p_{\mathrm{cap}}(P) = 1 - \frac{(N-P)^M}{N^M}. \tag{23}$$

When the targets are sampled without replacement, their miss probability is

$$\frac{(N-P)_M}{(N)_M} \le \left(1 - \frac{P}{N}\right)^M. \tag{24}$$

Equation (23) is therefore a lower bound for the attack experiment.

Set $c = \epsilon_{\mathrm{src}} + 2^{-64}$ and $L_2 = 693148/10^6 > \ln 2$. The search range is

$$P = \left\lceil \frac{N}{M} \left(L_2 + \frac{2c}{1-2c} \right) \right\rceil . \tag{25}$$

Since $-\ln(1-2c) \le 2c/(1-2c)$,

$$\left(1 - \frac{P}{N}\right)^M \le e^{-MP/N} \le \frac{1-2c}{2}.$$

The capture probability exceeds one half by $\epsilon_{\mathrm{src}} + 2^{-64}$. When negligible terms are omitted, P/N approaches $1 - 2^{-1/M}$ and then $(\ln 2)/M$ as M grows.

A.4 Surviving-pair capacity

The honest signers use at most

$$H = q(2n+1) \tag{26}$$

block-cipher keys. Each of the corresponding HM key/target pairs receives a full witness and syndrome check. At 2^{24} gates per pair, this contributes less than 2^{70} gates in every parameter set.

For a candidate s outside this set, let $Y_s \in [0, M]$ count the accepted witness-prefix predicates. The candidate permutation is independent of the transcript, so

$$\mathbb{E}Y_s \le M\rho^r. \tag{27}$$

Distinct remaining candidate keys index independent ideal-cipher permutations. Define

$$\mu = \lceil PM\rho^r \rceil, \qquad K = 2(\mu + HM). \tag{28}$$

Hoeffding's inequality gives

$$\Pr\left[\sum_s Y_s > K - HM \right] \le \exp\!\left(-\frac{2(K - HM - \mu)^2}{PM^2} \right), \tag{29}$$

which is below 2^{-512} for all six parameter sets.

A.5 Target identification

The union flag detects a survivor, while the commitment check also needs the corresponding target index. Let U_i count distinct prefix-tree terms after i RSD chunks and set

$$\overline{U}_i := \min(M, 2^{i\ell}), \qquad B_{\mathrm{tr}} := \sum_{i=1}^{r} \overline{U}_i. \tag{30}$$

Every transcript satisfies $U_i \le \overline{U}_i$. The traversal inspects every first-level term and descends only when the current prefix is true. A term at depth $i+1$ is reached with probability at most ρ^i, so a candidate outside the set of keys used during signature generation inspects at most

$$J = \overline{U}_1 + \sum_{i=1}^{r-1} \overline{U}_{i+1} \rho^i \tag{31}$$

terms in expectation.

Each candidate contributes a value in $[0, B_{\mathrm{tr}}]$, and the candidates are independent. Charging B_{tr} for each key used during signature generation gives

$$K_{\mathrm{tr}} = 2\big(\lceil PJ \rceil + HB_{\mathrm{tr}}\big). \tag{32}$$

Hoeffding's inequality yields

$$\Pr[\text{more than } K_{\mathrm{tr}} \text{ term inspections}] \le \exp\left(-\frac{2(K_{\mathrm{tr}} - HB_{\mathrm{tr}} - \lceil PJ \rceil)^2}{PB_{\mathrm{tr}}^2}\right). \tag{33}$$

This probability is below 2^{-512} for every parameter set. The gate total also charges 8λ dispatch gates for every surviving candidate/target pair.

A.6 Commitment and terminal events

For a cipher with b-bit blocks, one published λ-bit commitment projection has $L = 2^{b-\lambda}$ compatible full-block outputs. The two commitment plaintexts are distinct, so a false key outside the set used during signature generation matches both projections with probability at most

$$p_{\mathrm{com}} \le \frac{L^2}{2^b(2^b - 1)}. \tag{34}$$

A union bound covers the at most PM false candidate/target pairs. If a commitment input equals the selected PRG counter for the witness coordinates, the pair still has to satisfy both commitment projections, and the same marginal bound applies.

Two 2^{-512} terms cover surviving-pair and target-identification overflow. A 2^{-160} term covers witness reconstruction and fresh signing. Their sum is smaller than the 2^{-64} margin in Equation (25) for every parameter set. A captured honest seed passes the witness filter, matches the commitment, reconstructs the witness, and produces the key of Corollary 1. The resulting success probability is at least one half.

The surviving-pair and target-identification overflow probabilities are below 2^{-512}. The remaining failure probability is dominated by false commitment matches in Category I and by the 2^{-160} final term in Categories III and V. The base-two logarithms of their sums are -128.529 for both Category-I variants and -160.000 for the other four parameter sets.

B Boolean-circuit cost derivation

The circuit costs in Tables 4 and 3 use the following conventions and arithmetic.

B.1 Circuit conventions

Every counted gate has fan-in at most two. XOR, AND, OR, and NOT each cost one gate, while fan-out and wiring cost zero. AES and Rijndael states follow source byte order and least-significant-bit-first order within each byte. The count includes key expansion, S-boxes, round-key additions, MixColumns, target predicates, the controller, and the match accumulator. ShiftRows and public wire selection cost no gates.

Each gate stores a mask of the candidate-key bits on which it depends. A reverse traversal from the selected outputs produces the ancestor cone C. For input bit i, the update cost c_i is the number of gates in C whose mask contains i. The Gray-code assignment pairs frequent logical flips with small physical update costs.

B.2 Elementary-vector predicates and target prefix trees

The selector predicates are

$$U_2(x_0, x_1) = 1 \oplus x_0 x_1, \tag{35}$$

$$U_3(x_0, x_1, x_2) = 1 \oplus x_0 x_1 \oplus x_0 x_2 \oplus x_1 x_2. \tag{36}$$

After the public shift $x \mapsto x \oplus a$, all masks share the quadratic terms. The four shifted U_2 predicates cost seven gates together, and the eight shifted U_3 predicates cost fifteen. Their outputs are checked over every input against direct Hamming weight.

Let $\nu_1 \le \cdots \le \nu_d$ be the vector-mask alphabet sizes and $R = 2^\ell$ the chunk-mask domain. Shifted predicates for one RSD chunk cost $G_{\text{digit}} = 52$ gates in Categories I and V and 60 gates in Category III. The prefix trees within a chunk and across r chunks together cost at most

$$r\left(G_{\text{digit}} + \sum_{i=2}^{d} \min\left(M, \prod_{j=1}^{i} \nu_j\right)\right) + \sum_{i=1}^{r} \min(M, R^i) - 1. \tag{37}$$

The vector-mask alphabet sizes are $(4, 8, 8, 8)$ in Categories I and V and $(8, 8, 8, 8)$ in Category III. The corresponding per-chunk bounds are 2,388 and 4,732 gates. A bottom-up construction with U_i distinct prefixes after i chunks uses $\sum_{i=1}^{r-1} U_i$ AND gates and $U_r - 1$ OR gates, with $U_i \le \min(M, R^i)$. The maximum number of public prefixes is used at every depth, so the bound holds for every transcript table.

B.3 Scan, identification, and validation

For a Gray prefix of P candidates, logical bit j flips

$$f_j(P) = \left\lfloor \frac{P-1}{2^j} \right\rfloor - \left\lfloor \frac{P-1}{2^{j+1}} \right\rfloor. \tag{38}$$

The resulting scan cost is

$$G_{\text{scan}} = |C| + \sum_j f_j(P) c_{\pi(j)} + P(4\lambda + 1), \tag{39}$$

where 4λ charges the Gray controller and one gate records whether any candidate has matched.

The stored prefix-tree terms identify the surviving target indices. Write

$$\overline{U}_i = \min(M, 2^{i\ell}), \qquad B_{\text{tr}} = \sum_i \overline{U}_i, \qquad J = \overline{U}_1 + \sum_{i=1}^{r-1} \overline{U}_{i+1} \rho^i.$$

One candidate inspects at most B_{tr} terms and, outside the set of keys used during signature generation, at most J terms in expectation. The calculation uses K_{tr} from Equation (32) and charges 8λ gates for every surviving pair. The overflow probability is below 2^{-512}. Thus the target-identification cost is

$$G_{\text{id}} = K_{\text{tr}} + 8\lambda K. \tag{40}$$

Each of the at most K surviving pairs receives a two-block commitment check. The circuit costs are 51,992 gates in Category I, 142,928 in Category III, and 145,440 in Category V. The 2^{120} auxiliary term covers target construction, signature acquisition, terminal witness expansion, and fresh signing.

For the target-construction term, let B be the primitive block length in bytes and L the VOLEitH PRG output length. Reconstructing every opened leaf in the $q = 2^{12}$ transcripts costs at most

$$2^{12}\left(2(n-1) + 2n + n\left\lceil \frac{L}{B} \right\rceil\right) 2^{18} < 2^{51}. \tag{41}$$

The three terms count tree expansion, two-block seed commitments, and leaf-seed PRG expansions; 2^{18} bounds one primitive block circuit. One fresh-signing counter trial costs

less than 2^{24} gates and one message batch less than 2^{28}. The 2^{69} counter trials, at most 2^{37} message batches, recovery work below 2^{116}, and Equation (41) sum to less than 2^{117} gates. Public-syndrome checks for every key/target pair used during signature generation add less than 2^{70} gates. The total is

$$G = G_{\text{scan}} + G_{\text{id}} + KG_{\text{com}} + 2^{120} + 2^{24}HM. \tag{42}$$

Table 12 applies this decomposition to the selected filter in each parameter set.

Table 12: Components of the 2^{12}-signature attacks. Scan, target ID, and commitment entries are base-two logarithms.

Set	chunks	scan	target ID	commitments	memory (bytes)
CAT1-short	5	129.739	125.794	127.062	6,986,843
CAT1-fast	5	129.660	125.656	127.062	10,130,523
CAT3-short	5	193.901	189.535	189.596	13,382,208
CAT3-fast	5	193.771	189.284	189.596	20,017,728
CAT5-short	6	258.039	253.513	253.131	22,474,459
CAT5-fast	6	257.959	253.418	253.131	34,172,123

The memory bound contains the current circuit state, controller state, and public target table. Candidate/target survivors are streamed to commitment validation, and one full PRG output is materialized after a commitment match. Across the six parameter sets, the attack uses 1.23–2.90 primitive calls per candidate. Commitment checks and false witness prefixes add at most 1.54 bits over the ideal multi-target scan.

B.4 Reference implementation

The experiments use the SDitH v2 submission archive with SHA-256

`deabd02323e811b94b4f2bf5ee940da05cb95bd1b3b9342316dad8b8599d2354.`

The compiled reference objects provide parameter and signature parsing, GGM and pseudorandom generator expansion, VOLE reconstruction, signing, and verification. Differential vectors compare the AES-128 and Rijndael-256-256 outputs of these objects with the Python circuit implementation.

B.5 AES comparator

The AES circuits use the standard AES-128, AES-192, and AES-256 key schedules and three public plaintext/ciphertext pairs under one key. The first pair provides the short filter and full equality. A second encryption is charged for survivors of the first equality, and a third for survivors of both earlier equalities. This identifies the intended key with false-key probability below 2^{-128} even when the key space exceeds the block space.

The search range is $2^{\lambda-1}$. Later-stage capacities are the larger of 128 candidates and twice the ceiling of the ideal-cipher mean. A Chernoff bound places each capacity overflow below 2^{-840}. For three distinct pairs, the final false-key probability is at most

$$\frac{2^{\lambda-1}}{2^{128}(2^{128}-1)(2^{128}-2)}.$$

For each primitive, the calculation considers every contiguous filter of width 1 through 32 and every start position, then applies the same dependency-mask ordering, Gray controller, and delayed validation as the attack. The selected eight-bit filters give Table 3. Subtracting the attack totals gives the 11.23–11.67-bit separations in Table 4; an additional 2^{10} reduction of the AES totals leaves 1.23–1.67 bits.

C Analysis after domain separation

Index the public target values by t and write their repaired form as

$$A_t = \mathsf{wit} \oplus G_t(s_t),$$

where each G_t uses the PRG input assigned to its signature and leaf by Equation (19). The witness encoding and public syndrome still recognize a correct seed, but the primitive input changes with t.

In the ideal-cipher model, a PRG output is an independent pad until the adversary queries its seed and input. The adversary receives the public RSD instance, the target values, the sibling openings, the salted seed commitments, and access to the ideal block cipher used by the signer.

Let d be the number of PRG blocks per leaf, let $r = d + 2$ include the two commitment blocks, and mark the M hidden leaf seeds. The value V from Equation (22) bounds the seed values generated by the signer. With $W = rM + V$, define

$$\epsilon_{\mathrm{lab}} := \frac{W(W-1)}{2^{\lambda-3}}, \qquad \epsilon_{\mathrm{perm}} := \frac{M\binom{r}{2}}{2^b}. \tag{43}$$

The first term bounds collisions among PRG, commitment, and tree inputs after accounting for the four cleared salt bits. The second bounds the statistical distance between the r outputs of one marked-key permutation and r independent uniform strings. At $q = 2^{12}$, their largest values across the parameter sets are below $2^{-70.07}$ and $2^{-106.21}$.

For an adversary making Q primitive queries, set

$$\epsilon_{\mathrm{qry}} := \frac{WQ}{N2^b}. \tag{44}$$

This term covers a query under a marked leaf or tree key at an input programmed for another use. At $Q = N$, its largest value for $q = 2^{12}$ is below $2^{-100.5}$. For the additional $q = 2^{20}$ regime, the largest seed-generation, input-collision, permutation, and query terms are below $2^{-57.08}$, $2^{-54.07}$, $2^{-98.21}$, and $2^{-92.53}$.

Theorem 3 (Recovery after domain separation)**.** *Let an adversary make $Q \le N$ adaptive ideal-cipher queries using the inputs of Equation (19). Its probability of returning a signing witness is at most*

$$\mathsf{Adv}_{\mathrm{RSD}}(T') + \frac{Q}{N} + \epsilon_{\mathrm{src}} + \epsilon_{\mathrm{lab}} + \epsilon_{\mathrm{perm}} + \epsilon_{\mathrm{qry}}. \tag{45}$$

Here $\mathsf{Adv}_{\mathrm{RSD}}(T')$ is the probability of solving the public RSD instance in the time T' left after the ideal-cipher queries and polynomial-time transcript simulation.

Proof. Remove first the seed-generation event of Equation (22) and the input-collision event of Equation (43). The marked seeds and their assigned inputs are then distinct. For one marked key, its r permutation outputs form a uniform sample without replacement. Replacing them by independent uniform strings changes the distribution by at most $\binom{r}{2}/2^b$; summing over the M marked keys gives ϵ_{perm}.

Each target value now contains an independent pad. The GGM tree contributes at most V other programmed outputs. Within one signature, the salt and node labels of the tree, commitment inputs, and VOLEitH PRG inputs are disjoint. The term ϵ_{lab} covers collisions across signatures.

For a query under a marked key at another input, couple the answer to a fresh random string until it collides with a programmed output. Marked keys retain uniform marginals

in this game, including when sampled without replacement. Weighting leaf keys by their r programmed outputs and tree keys by their at most V outputs gives

$$\mathbb{E}[\text{same-key query/programmed-input pairs}] \le \frac{WQ}{N}.$$

Each pair collides in its b-bit output with probability 2^{-b}, so the coupling fails with probability at most ϵ_{qry}. This calculation also covers queries directed at the one-tree opening.

Until a query reaches an assigned input for a target leaf or an unknown GGM frontier node, the pads hide the planted witness. One primitive query can reach at most one such input, whose marked key is uniform, and the adaptive union bound is Q/N. If no query reaches one, any valid witness returned by the adversary solves the public RSD instance. Adding the four game-transition terms gives Equation (45). □

The bound includes pairwise combinations of target values. For two targets t and u, cancelling the witness gives

$$A_t \oplus A_u = G_t(s_t) \oplus G_u(s_u),$$

a planted claw between functions with different inputs. Sampling K keys for each function reaches both planted seeds with probability at most $K^2/(N(N-1))$ after conditioning on distinct seeds. Other truncated-output collisions do not identify the planted pair until both seeds have been reached and the public witness test succeeds.

The one-tree opening has the same property. Every unknown frontier point has its own salt and node label. Starting from a disclosed sibling requires inverting the labelled edge from its unknown parent, while guessing a root requires reaching the first labelled edge before deriving the challenged leaves. Both paths first reach an assigned point with probability at most Q/N and then pay for the remaining tree expansion. The commitment inputs use the same salt and node labels.

For the circuit comparison, let S be the variable-key schedule cost and E the cost of one encryption data path. The values are $(S, E) = (5{,}880, 22{,}800)$ in Category I and $(16{,}352, 64{,}032)$ in Categories III and V. A batch of h labelled inputs has work for success probability $1/2$, relative to a one-target search, equal to

$$2\big(1 - 2^{-1/h}\big)\frac{S + hE}{S + E}. \tag{46}$$

For $h = q\tau$ in the six parameter sets, this ratio is 1.102–1.104. Searching all inputs together is therefore 0.140–0.143 bits more expensive than choosing one input and performing ordinary key search.

Transcript extraction, witness recognition, commitment validation, and conversion of a witness into a signing key still apply after the change. The separate PRG inputs reduce witness recovery in the ideal-cipher model to primitive inversion or solution of the public RSD instance.