# Improving Skipping Fault Correction Attacks on Randomized Dilithium via MILP

Haobo Ouyang Chaoran Wang Guowei Liu
Lixuan Wu Meiqin Wang Yanhong Fan

School of Cyber Science and Technology, Shandong University

August 26–28, 2026

# Contents

# Randomization and Fault Recovery

**Core signing computation**

$$\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{s}_1$$

- $\boldsymbol{y}$ is freshly randomized for every signature.
- $c$ is a sparse challenge with exactly $\tau$ nonzero coefficients.
- $\boldsymbol{s}_1$ is secret, but each coefficient is small: $s[j] \in [-\eta, \eta]$.

Randomized signing does not fully preclude fault-based key recovery.

**Relevant parameters** ($N = 256$)

| Level | $\ell$ | $\eta$ | $\tau$ | $\ell N$ |
|---|---|---|---|---|
| L2 | 4 | 2 | 39 | 1024 |
| L3 | 5 | 4 | 49 | 1280 |
| L5 | 7 | 2 | 60 | 1792 |

**Key property:** the coefficients of $\boldsymbol{s}_1$ lie in the bounded small-integer domain $[-\eta, \eta]$.

# One Fault, One Equation

1. **Inject an instruction-skipping fault** while computing $\boldsymbol{z} = \boldsymbol{y} + c\boldsymbol{s}_1$.
2. **Use public verification as a correction oracle.** At the faulted coefficient, try $\alpha \in [-\beta, \beta]$ until the modified signature verifies.
3. **Record the recovered relation**

$$\langle c_i, \boldsymbol{s} \rangle = \alpha,$$

where $c_i$ is a rotation of the public challenge and $\boldsymbol{s}$ is one $N$-coefficient component of $\boldsymbol{s}_1$.

Krahmer et al.: collect $N = 256$ equations for every component to obtain a full-rank system.

# Why Full Rank Is Unnecessary

**Baseline: full-rank equations**

- $M = N = 256$ equations per component
- Collect a full-rank system for each component
- Total number of injected faults: $\ell N$
- Does not use $\boldsymbol{s} \in [-\eta, \eta]^N$ in recovery

**Our approach: underdetermined + bounded**

- Collect only $M < N$ equations
- Retain the coefficient bound explicitly
- Restrict recovery to the bounded small-integer domain
- Recover with an MILP solver

**Question.** How small can $M$ be while

$$\mathcal{S} = \{\boldsymbol{s} \in \mathbb{Z}^N : C\boldsymbol{s} = \boldsymbol{x},\ \|\boldsymbol{s}\|_\infty \le \eta\}$$

still contains exactly one element with high probability?

# Online Collection, Offline Recovery

**Online: collect corrected faults**

- Each usable fault returns $(c_i, x_i)$.
- Every row has $\tau$ entries in $\{\pm 1\}$.
- Stack $M$ samples into $C \in \{-1, 0, 1\}^{M \times N}$ and $\boldsymbol{x} \in \mathbb{Z}^M$.

**Offline: solve a bounded system**

$$C\boldsymbol{s} = \boldsymbol{x}, \qquad \boldsymbol{s} \in [-\eta, \eta]^N.$$

- The equation system is underdetermined when $M < N$.
- The coefficient-wise bound limits the search domain.
- The bounded feasible set can still contain exactly one element.

The online phase collects equations; the offline phase invokes the MILP solver.

# Coverage and Uniqueness

**Step 1: coverage**

- Let $H_j$ count rows involving coefficient $j$.
- If $H_j = 0$, coefficient $j$ is unconstrained; uniqueness is impossible.
- Under the random-row model,

$$\Pr[\forall j : H_j \geq 1] \geq 1 - N\left(1 - \frac{\tau}{N}\right)^M.$$

**Step 2: uniqueness**

- Two bounded solutions differ by a nonzero $\boldsymbol{d} \in [-2\eta, 2\eta]^N$.
- Non-uniqueness is equivalent to $C\boldsymbol{d} = 0$ for an admissible $\boldsymbol{d}$.
- A first-moment bound gives

$$\Pr[\text{non-unique}] \leq \left((2\eta+1)^N - 1\right) p_{\text{gauss}}^M.$$

Full coverage is necessary; Theorem 1 bounds the probability of a second bounded solution.

# Sufficient Fault Thresholds

For target failure probability $2^{-\lambda}$, it suffices to choose

$$M \geq \frac{\log\left((2\eta+1)^N - 1\right) + \lambda}{-\log p_{\mathrm{gauss}}}.$$

Here

$$p_{\mathrm{gauss}} = \frac{1}{\sqrt{2\pi}\,\sigma_d}, \qquad \sigma_d^2 = \frac{2\tau\eta(\eta+1)}{3}.$$

This sufficient condition is derived under the random-row model and is not intended to be tight.

| Level | $p_{\mathrm{gauss}}$ | $M_{\min}$ | $M_{\min}/N$ |
|---|---|---|---|
| L2 | 0.03194 | 122 | 47.7% |
| L3 | 0.01561 | 137 | 53.5% |
| L5 | 0.02575 | 114 | 44.5% |

Values are per polynomial component and target at least 99% uniqueness probability.

# High-Probability Coverage

**Lemma 1**

- Each row contains $\tau$ nonzero positions.
- Position $j$ is absent from all $M$ rows with probability $\left(1 - \frac{\tau}{N}\right)^M$.
- A union bound over all $N$ positions gives

$$\Pr[\exists j : H_j = 0] \le N\left(1 - \frac{\tau}{N}\right)^M.$$

**Dilithium-L2 example**

$$N = 256, \qquad \tau = 39, \qquad M = 128.$$

$$256\left(1 - \frac{39}{256}\right)^{128} \approx 1.7 \times 10^{-7}.$$

Therefore, at $M = N/2$, all secret coefficients are covered with high probability.

Coverage is necessary but does not by itself guarantee a unique bounded solution.

# MILP Recovery Model

| Variables | Constraints | Solver setup |
| --- | --- | --- |
| $s[j] \in \mathbb{Z}, \quad -\eta \le s[j] \le \eta$ | $C_i \boldsymbol{s} = x_i, \quad i = 0, \ldots, M-1$ | $\min f(\boldsymbol{s})$ |
| $N = 256$ bounded integers | $M < N$ sparse equalities | MILP solver (Gurobi) objective function $f(\boldsymbol{s})$ |

**Adaptive retry strategy**

1. Start below $N$ with $M_{\mathrm{try}}$ collected equations.
2. Solve one component and check the candidate by reconstructing signatures and running public verification.
3. If the check fails or solving times out, collect four more equations and retry.

# Plain and Shuffling Settings

**Plain setting**

- Fault location is controlled and known.
- Repeatedly target a fixed coefficient in each component.
- Append each corrected equation to that component's set.
- Invoke MILP after $M_{\mathrm{try}}$ samples; retry adaptively.

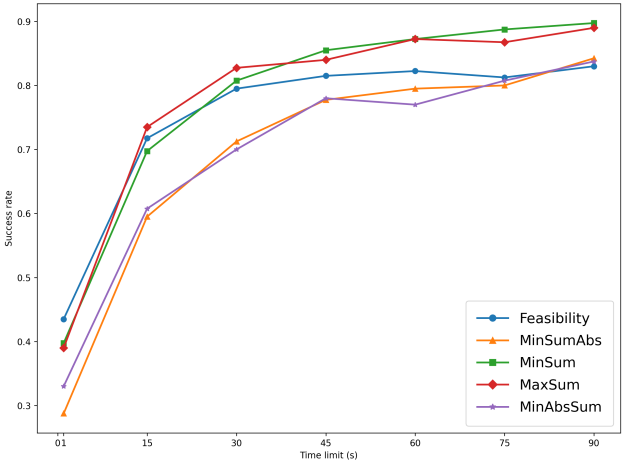
The target position is known

**Shuffling setting**

- Addition order is randomized; the fault location is unknown.
- Search all $\ell N$ candidate locations during correction.
- Route a recovered equation to the corresponding component.
- Stop solving components once their candidates verify.

The fault location must be resolved by correction search

# Objective and Time Limit



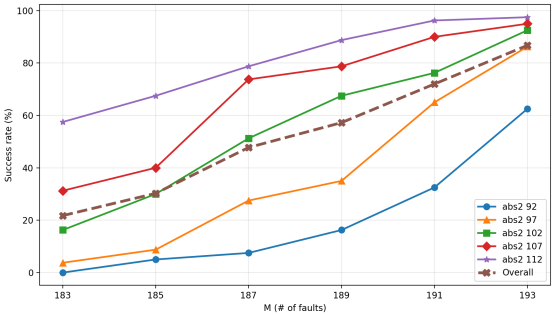

**Setup**

- 11th Gen Intel i7 at 2.5 GHz
- Gurobi MILP solver
- Dilithium implementation
- L2; $M = 193$; 400 components

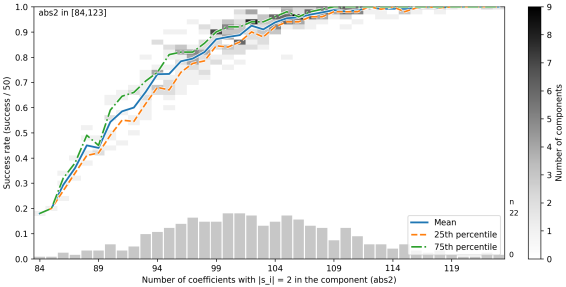
**Observation**

- Feasibility leads for short budgets.
- MinSum/MaxSum lead after 30 seconds.
- Curves largely plateau by $T = 60$ s.

# Recovery Success Factors




At fixed $\mathrm{abs}2$, success rises with $M$.




At fixed $M = 193$, components with larger $\mathrm{abs}2$ are easier.

**Explanation:** larger $\|\boldsymbol{s}\|_2^2$ lowers the single-row collision probability, making the system more constrained.

# Fault Reduction: 13–26%

**Plain setting**

| Level | Baseline | Ours | Reduction |
|---|---|---|---|
| L2 | 1024 | 759 | 25.9% |
| L3 | 1280 | 1073 | 16.2% |
| L5 | 1792 | 1334 | 25.6% |

200 independent trials; 60-second MILP budget.

**Shuffling setting**

| Level | Baseline | Ours | Reduction |
|---|---|---|---|
| L2 | 1157 | 861 | 25.6% |
| L3 | 1374 | 1189 | 13.5% |
| L5 | 2075 | 1529 | 26.3% |

10 independent trials; exhaustive location search.

Under shuffling, each usable equation requires correction search over all $\ell N$ possible coefficient positions.

# Theory vs. Experiments

$M_{\min} = 122$ (L2 sufficient uniqueness bound) vs. $M = 193$ (97.5% at $\mathrm{abs2} = 112$)

**What the theorem controls**

- An idealized independent random-row model
- Existence of a unique bounded solution
- A conservative first-moment/union bound

**What the experiment adds**

- Generic branch-and-bound search
- A finite 60-second time budget
- Private-key feature $\mathrm{abs2}$
- Matrix rows obtained from rotated challenge vectors

**Interpretation:** uniqueness is necessary for recovery, but not sufficient for fast recovery by a generic solver.

# Additional Countermeasures

**Sign-then-verify**

- Verification runs inside the signing device.
- A signature that fails internal verification is not returned.
- The attacker cannot obtain the faulty signature required by EqFromSig.
- Public verification can no longer be used as the correction oracle.

**Double computation**

- The signing computation is performed twice.
- Results are compared before releasing the signature.
- A successful attack requires compatible faults in both executions.
- Such faults typically have to affect the same coefficient position.

The paper does not evaluate the attack under these two countermeasures; quantitative analysis is left for future work.

# Key Takeaways

1. **Theory:** coverage plus bounded-solution uniqueness gives $(M_{\min})_{\mathrm{L2,L3,L5}} = (122, 137, 114)$ per component for a 99% target.
2. **Method:** MILP makes the coefficient bound explicit and supports adaptive online collection with offline retry.
3. **Attack settings:** the adaptive MILP attack is described for both plain and shuffling settings.
4. **Experimental result:** the measured number of injected faults drops by 13–26% across all three Dilithium security levels.

Collecting a full-rank system of equations is not required for key recovery.

# Questions & Discussion

Haobo Ouyang

hbouyang@mail.sdu.edu.cn

Code: https://anonymous.4open.science/r/sfcafDvM-H73B