Key-Recovery Attacks on TALUS: A Cryptanalytic Note

Guilhem Niot

July 7, 2026

Abstract

We present key-recovery attacks on the constructions of TALUS (Kao and Chang), a threshold ML-DSA (FIPS 204) construction available on arXiv [1] and scheduled for presentation at the NIST Threshold Call Preview Talks Round 2 (TCPT-2, `https://csrc.nist.gov/events/2026/tcpt`).

For *TALUS-MPC*, which is claimed EUF-CMA secure against an adversary corrupting up to $T-1$ parties, we show the claim is false via two independent attacks, both exploiting the same root cause. TALUS-MPC uses Feldman commitments that apply the public matrix A to secret values: key shares of s_1 during key generation, and contributions to the nonce $\hat{y}$ during signing. Since A is left-invertible in every ML-DSA parameter set, these images are invertible by Gaussian elimination, with no lattice problem to solve. A passive observer recovers all key shares of s_1 directly from the key-generation broadcast, and independently recovers the aggregate nonce $\hat{y}$ from the signing broadcast, which then yields $s_1 = c^{-1}(z - \hat{y})$ from a single signature.

For *TALUS-TEE* and *TALUS-MPC*, we identify a persisting flaw: the rejection-sampling check that protects the error term s_2 in standard ML-DSA has been removed. Each signature leaks a noisy linear equation in s_2; applying least-squares recovery over the cyclotomic ring—an instance of LWE without modular reduction—recovers the full secret from a few hundred million signatures. The sample counts we derive are not optimized and we believe exploiting the bounded noise structure and lattice-reduction techniques would reduce them significantly, but we focus on establishing the structural flaw.

1 Preliminaries

We follow the notation of the TALUS paper. Let $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ with $n = 256$ and $q = 8\,380\,417$. In ML-DSA, a public matrix $A \in R_q^{k \times \ell}$ is expanded from a public seed ρ, where the three FIPS 204 parameter sets have $(k, \ell) \in \{(4,4), (6,5), (8,7)\}$ for ML-DSA-44, -65, -87. The long-term signing key is the vector $s_1 \in R_q^\ell$ with $\|s_1\|_\infty \leq \eta$, shamir shared among users with shares $(s_{1,i})_i$, and the error term is $s_2 \in R_q^k$ with $\|s_2\|_\infty \leq \eta$. The public key is (ρ, t_1) where $t_1 = \mathsf{Power2Round}(As_1 + s_2,\, d)$; writing $t := As_1 + s_2$ we have $t = t_1 \cdot 2^d + t_0$ with secret low-order part $t_0 \in R_q^k$, $\|t_0\|_\infty \leq 2^{d-1}$. A signature is $\sigma = (\tilde{c}, z, h)$ with challenge $c = \mathsf{SampleInBall}(\tilde{c})$ and public response $z = \hat{y} + c\, s_1$, where $\hat{y}$ is the signing nonce.

2 The attack on TALUS-MPC

TALUS-MPC uses Feldman commitment (see their Remark 20) in the same way it is used in the classical (discrete-log) setting: to commit to the coefficients of a secret-sharing polynomial, each party broadcasts the images of those coefficients under a public group operation. In the classical setting, the commitment g^x is hiding because computing a discrete logarithm is hard. TALUS-MPC replaces g^x with Ax, where A is the public ML-DSA matrix. This substitution is unsound: unlike

exponentiation, A is a tall matrix ($k \geq \ell$ in every ML-DSA parameter set), making it injective. Recovering x from Ax requires no lattice assumption, simply Gaussian elimination. Concretely, the map is left-invertible with very high probability as shown in the following lemma (failure probability $\leq 2^{-15}$ for ML-DSA-44 and $\leq 2^{-38}$ for ML-DSA-65/87, by a union bound over the 256 NTT slots), so there exists $A^+ \in R_q^{\ell \times k}$ with $A^+A = I_\ell$, and any x is recovered from Ax in closed form as $x = A^+(Ax)$.

Lemma 1 (Left-invertibility of A)**.** *Since $512 \mid q-1$, the ring R_q splits via the NTT isomorphism $R_q \cong \prod_{j=0}^{n-1} \mathbb{F}_q$; we write $f^{(j)}$ for the evaluation of $f \in R_q$ at the j-th NTT point $\zeta^{2j+1} \in \mathbb{F}_q$, where ζ is a fixed primitive $2n$-th root of unity. Under this isomorphism, A decomposes into n independent blocks $A^{(j)} \in \mathbb{F}_q^{k \times \ell}$. For a uniform $k \times \ell$ matrix over $\mathbb{F}_q$ with $k \geq \ell$, the probability that $A^{(j)}$ fails to have full column rank is at most $q^{-(k-\ell+1)}$. A union bound over 256 slots gives a failure probability of at most $256\, q^{-1} \approx 2^{-15}$ for ML-DSA-44 and $256\, q^{-2} \approx 2^{-38}$ for ML-DSA-65/87. Hence with high probability there exists $A^+ \in R_q^{\ell \times k}$ satisfying $A^+A = I_\ell$, and any $x \in R_q^\ell$ is uniquely recoverable from Ax by $x = A^+(Ax)$ using Gaussian elimination.*

This single structural error yields two independent key-recovery attacks, one during key generation and one during signing. Both attacks are passive: no party corruption is needed, and the adversary need only observe the broadcast channel.

2.1 Attacking key generation

In Protocol 11, Round 2, every party i broadcasts $As_{1,i}$ as a Feldman commitment on their Shamir share of the secret. Since A is left-invertible, each key share is immediately recovered by

$$s_{1,i} \;=\; A^+\big(A\, s_{1,i}\big).$$

An observer of the broadcast channel therefore recovers all shares $s_{1,i}$ and reconstructs the long-term signing key $s_1 = \sum_{i \in S} \lambda_i s_{1,i}$ for any qualified set S.

2.2 Attacking the signing protocol

Should the attacker miss the key-generation broadcast, they can still recover s_1 from a single signature.

For identifiable abort, each party $h \in S$ broadcasts in the preprocessing round (Protocol 13, Step 4) the Feldman commitments

$$\Phi_{h,k} \;=\; A\, \tilde{a}_{h,k}, \quad k = 0, \ldots, T-1,$$

where $\tilde{a}_{h,0} = \hat{y}_h$ is party h's contribution to the signing nonce ($\hat{y} = \sum_{h \in S} \hat{y}_h$). In particular, $\Phi_{h,0} = A\hat{y}_h$ is public. Summing over all parties and applying the left inverse recovers the aggregate nonce:

$$\hat{y} \;=\; A^+\Big(\sum_{h \in S} \Phi_{h,0}\Big) \;=\; A^+\Big(A \sum_{h \in S} \hat{y}_h\Big).$$

With the nonce in hand, the signing key is recovered from the identity $z = \hat{y} + c\, s_1$, which holds over $\mathbb{Z}[X]/(X^{256}+1)$ without any reduction mod q (both $\hat{y}$ and $c\, s_1$ have coefficients bounded well below $q/2$). Since the quotient $\mathbb{Q}[X]/(X^{256}+1)$ is a field, any nonzero c is invertible there via the extended Euclidean algorithm. Rearranging gives

$$s_1 \;=\; c^{-1}(z - \hat{y}) \;\in\; \mathbb{Q}[X]/(X^{256}+1),$$

and the result is automatically in $\mathbb{Z}[X]/(X^{256}+1)$ since s_1 is a known integer polynomial. The broadcasts $\Phi_{h,k}$ are public to all participants by design (they serve the blame protocol), so no party corruption is needed—a passive eavesdropper on the broadcast channel suffices.

3 Attack on TALUS-TEE and TALUS-MPC from polynomially many signatures

In TALUS-TEE the nonce is manipulated inside a trusted execution environment, so Ay is not broadcast. The Feldman commitment attacks of Section 2 therefore do not apply directly. However, a different flaw persists: both TALUS-TEE and TALUS-MPC omit the rejection-sampling check that protects the error term s_2 in standard ML-DSA. This allows a passive adversary to recover s_2 (and hence s_1) from polynomially many signatures.

The observable leakage. Recall that the public key encodes $t_1 = \mathsf{Power2Round}(As_1 + s_2)$, so $t := t_1 \cdot 2^d + t_0$ with t_0 the (secret) low-order part. Given a signature $(\tilde{c}, z, h)$ and the public matrix A, an observer can compute

$$v \;:=\; Az - c\,t_1 \cdot 2^d \;=\; w - c(s_2 - t_0),$$

where $w = Ay$ is the ML-DSA commitment derived from the nonce before rounding. Indeed, $Az = A(y + cs_1) = w + cAs_1 = w + c(t - s_2)$, so subtracting $ct_1 \cdot 2^d$ cancels the public part of ct and leaves the term above. Moreover, any valid signature comes with a hint h such that $w_1 := \mathsf{UseHint}(h, v) = \mathsf{HighBits}(w)$ by the ML-DSA verification equation; the high bits of w are therefore directly recoverable from the signature, reducing the unknown part of w to its low bits $\mathsf{LowBits}(w)$, bounded by γ_2 in ℓ_∞.

In standard ML-DSA, rejection sampling aborts the signing procedure whenever $\|\mathsf{LowBits}(w - cs_2)\|_\infty \geq \gamma_2 - \beta$, which ensures that the distribution of accepted signatures is independent of s_2. TALUS-TEE drops this check, so v is always released.

Statistical key recovery. Let $s' := s_2 - t_0$ be the target. Define the observable quantities

$$b_i \;:=\; v_i - w_{1,i} \cdot 2\gamma_2 \;=\; -\,c_i\,s' + e_i, \qquad e_i := \mathsf{LowBits}(w_i).$$

Here, all arithmetic is over $\mathbb{Z}[X]/(X^{256}+1)$ with no modular reduction, as all terms are bounded well below $q/2$. This is precisely an instance of LWE without modular reduction (ILWE) in the negacyclic ring, a problem studied by Bootle et al. [2], who show that this problem is much easier than standard LWE and expose an efficient key recovery attack under mild conditions. We refer to their paper for the complete analysis and only give a rough heuristic estimate of their key recovery attack applied to TALUS below.

They apply a least-squares (LS) regression, followed by nearest-integer rounding, to recover the secret s'. They define the following LS estimator:

$$\hat{s}' \;=\; -\Big(\sum_{i=1}^{N} \bar{c}_i\,c_i\Big)^{-1} \sum_{i=1}^{N} \bar{c}_i\,b_i \;\in\; \mathbb{Q}[X]/(X^{256}+1)^k.$$

where $\bar{c}_i(X) := c_i(X^{-1}) \bmod (X^{256}+1)$ is the ring adjoint of c_i.

Assuming N is large enough, the LS estimator $\hat{s}'$ is close to s' in coefficient-wise ℓ_∞ norm, so nearest-integer rounding recovers s' exactly. Then, since $t_0 = t - t_1 \cdot 2^d$, we have $As_1 = t_1 \cdot 2^d - s'$, and the left inverse of Lemma 1 gives

$$s_1 \;=\; A^{+}(t_1 \cdot 2^d - \hat{s}').$$

Sample complexity. Let us now estimate the number of signatures N required for the LS estimator to succeed. Each coefficient of $e_i = \mathsf{LowBits}(w_i)$ is approximately uniform in $[-\gamma_2, \gamma_2]$, so $\sigma_e^2 := \gamma_2^2/3$.

The LS estimator satisfies $\hat{s}' = s' + \text{noise}$ where the noise term is $(\sum_i \bar{c}_i c_i)^{-1} \sum_i \bar{c}_i e_i$. We analyze its coefficient-wise variance.

- *Denominator.* $\mathbb{E}[\bar{c}c]$ equals the constant polynomial τ: its constant coefficient is $\|\bar{c}\|_2^2 = \|c\|_2^2 = \tau$, and every other coefficient is $\sum_j c_j c_{j+k}$, whose expectation is zero because the τ nonzero positions and signs of c are chosen independently at random. By the law of large numbers, $\sum_i \bar{c}_i c_i \approx N\tau$.
- *Numerator.* The ℓ-th coefficient of $\bar{c}_i e_i$ is a weighted sum of the coefficients of e_i with weights $\pm\bar{c}_{i,j}$:
$$[\bar{c}_i e_i]_\ell \;=\; \sum_j (\pm\bar{c}_{i,j})\, e_{i,\,\ell-j},$$
so $\mathrm{Var}([\bar{c}_i e_i]_\ell) = \|\bar{c}_i\|_2^2 \cdot \sigma_e^2 = \tau\sigma_e^2$ (since coefficients of e_i are i.i.d. and $\|\bar{c}_i\|_2^2 = \tau$). Summing over N independent samples: $\mathrm{Var}([\sum_i \bar{c}_i e_i]_\ell) = N\tau\sigma_e^2$.
- *Combined.* Dividing the numerator by the (approximately deterministic) scalar $N\tau$ scales the variance by $(N\tau)^{-2}$, giving
$$\mathrm{Var}([\hat{s}']_\ell) \;\approx\; \frac{N\tau\sigma_e^2}{(N\tau)^2} \;=\; \frac{\sigma_e^2}{N\tau}.$$

This matches the general ILWE scaling of [2]: in their framework the sample complexity for recovering a secret s from noisy linear measurements $b_i = a_i s + e_i$ scales as σ_e^2/σ_a^2, where σ_a^2 is the variance of the scaling factor in front of s.

Nearest-integer rounding recovers $[\hat{s}']_\ell$ once the standard deviation $\sigma_e/\sqrt{N\tau}$ drops below $1/2$, requiring
$$N \;\gtrsim\; \frac{4\gamma_2^2}{3\tau}.$$
The concrete estimates for each FIPS 204 parameter set are given in Table 1.

Variant	τ	γ_2	$\beta = \tau\eta$	$N \gtrsim$
ML-DSA-44	39	95 232	78	3.1×10^8
ML-DSA-65	49	261 888	196	1.9×10^9
ML-DSA-87	60	261 888	120	1.5×10^9

Table 1: Estimated number of signatures required for statistical key recovery in TALUS-TEE, per ML-DSA parameter set.

The attack is entirely passive, requires only black-box access to signatures on chosen (or even observed) messages, and succeeds in polynomial time.

Remark. The sample counts in Table 1 are conservative heuristic estimates based on a Gaussian noise model and a simple averaging argument. Several directions may reduce them substantially in practice. First, the noise $e_i = \mathsf{LowBits}(w_i)$ is *bounded* rather than Gaussian, so the distribution has sharp edges that carry additional information not captured by variance alone. Second, lattice-reduction methods may further reduce the number of samples required. We do not pursue these improvements here and our goal is solely to expose important structural errors in the construction TALUS.

4 Discussion

The attacks presented here suggest that TALUS is fundamentally broken: the two attacks stem from distinct design errors and each independently suffice to recover the long-term key. Furthermore, both of these attacks exploit well-known facts about lattice-based cryptography, and do not rely on any novel techniques.

Note that merely removing the Feldman commitments and reintroducing the rejection-sampling check would not fix the construction. The corrupted parties have the ability to bias their nonce contributions y_i before the rejection-sampling step, which can be exploited to leak information about s_1. They could set $y_i = 0$ to ensure the rejection test hiding s_1 is always passed, thus again allowing for a statistical key recovery attack.

Acknowledgments. The author thanks Thomas Prest and Thomas Espitau for pointing out the reference [2]. The author used Claude Opus' (Anthropic) LLM to help draft the text of this note, but all attacks are due to the author.

References

[1] Leo Kao and Raymond Chang. TALUS: Threshold ML-DSA with One-Round Online Signing via Boundary Clearance and Carry Elimination. arXiv:2603.22109 [cs.CR], 2026. `https://arxiv.org/abs/2603.22109`.

[2] Jonathan Bootle, Claire Delaplace, Thomas Espitau, Pierre-Alain Fouque, and Mehdi Tibouchi. LWE without modular reduction and improved side-channel attacks against BLISS. In *Advances in Cryptology – ASIACRYPT 2018*, Lecture Notes in Computer Science, pages 494–524. Springer, 2018. `https://doi.org/10.1007/978-3-030-03326-2_17`.