

TALUS

Threshold ML-DSA via Boundary Clearance

Leo Kao · Codebat · joint work with Raymond Chang · NIST MPTC / TCPT2

Slides updated on 2026-09-11 with technical corrections addressing the key-recovery attacks reported by Niot [ia.cr/2026/1386] against an earlier version of TALUS; see the following slide.
Full treatment: [arXiv:2603.22109v5](https://arxiv.org/abs/2603.22109v5).

Update to Prevent Key-Recovery Attack

Niot [ia.cr/2026/1386] reports two key recoveries against an earlier version of TALUS. Their status in the current construction differs:

1 · Non-hiding commitments $A \cdot v$ → removed by design

No per-party $A \cdot s_1$ is ever published: the DKG's only public lattice output is $t = A \cdot s_1 + s_2$, a hiding MLWE sample. Short-share well-formedness is enforced by secret-shared rejection sampling, not a Feldman commitment; Blame never opens a post- z nonce.

2 · Residual s_2 channel → quantified and capped, not claimed away

Each emitted signature leaks a noisy observation of s_2 .

integer-uniqueness wall $q_{\text{uniq}} \approx 2^{15.2} / 2^{16.4} / 2^{16.2}$, below which the channel's observations do not determine s_2

mandatory rotation cap sits $+2.18 / +2.4 / +2.15$ bits below that wall

lattice-shielded: modeled at the core-SVP Module-LWE bound $\approx 2^{160}$, with an unconditional proven floor $\approx 2^{58} / 2^{49}$

The shield is model-supported, not proven. We state it as such.

Scope. TALUS-TEE is withdrawn from the proposal and retained only as a contrasting example. The proposed system is TALUS-MPC — two online rounds, honest majority, no hardware trust. We do not claim a one-round MPC phase under standard assumptions.

IN ONE LINE

**One geometric fact in ML-DSA's design
lets threshold signing skip the
interaction everyone else pays for.**

Standard FIPS 204 output · s_1 never reconstructed.

Rounding is non-linear

Schnorr thresholdizes cleanly — it's linear. ML-DSA's commitment goes through **HighBits**, which doesn't commute with addition.

Parties can't just round their shares and add them up.

Signing rejects and retries

Pick a nonce, form a candidate, check tight bounds. Fail → throw away, retry.

On one signer this loops ~4–5 times, invisibly. Microseconds. Nobody notices.

Split the key — every retry is a round

Now the nonce is shared. Each retry becomes a fresh round of all parties interacting.

The invisible loop becomes **the dominant cost** — this is where prior schemes' online rounds come from.

THE TWO CHECKS

z-check and r_0 -check — different jobs

z-check $\|z\|_\infty < \gamma_1 - \beta$

keeps the released z statistically independent of s_1 . (linear $\rightarrow$ one round)

r_0 -check $\|r_0\|_\infty < \gamma_2 - \beta, \quad r_0 = \text{LowBits}(w - c \cdot s_2)$

ensures the signature verifies (hint reconstructs w_1). Touches $s_2 \rightarrow$ hard to distribute.

r_0 -check is the expensive one

$$r_0 = \text{LowBits}(w - c \cdot s_2) \quad \text{accept iff } \|r_0\|_\infty < \gamma_2 - \beta$$

- touches the secret s_2 — revealing $c \cdot s_2$ leaks it
- LowBits is non-linear (carries)
- re-run on every rejection $\rightarrow$ multi-round MPC

Every prior scheme evaluates $c \cdot s_2$ each attempt. We ask: is it needed at all?

How ML-DSA signs, faithfully

```

repeat:                                     ← rejection loop
  y ← ExpandMask(...)                      sample nonce
  w = A · y
  w1 = HighBits(w)
  c = SampleInBall(H(μ || w1))
  z = y + c · s1
  r0 = LowBits(w - c · s2)
  if ||z||∞ ≥ γ1 - β or ||r0||∞ ≥ γ2 - β: continue ← two checks, retry
σ = (c̃, z, h)                               standard signature

```

Same steps, split across parties

OFFLINE (before μ)

- ① each party samples $\hat{y}_i$ ($y = \sum \hat{y}_i$, no one holds y)
- ② local $A \cdot \hat{y}_i \rightarrow [w]$ (linear; w never revealed)
- ③ CSCP on shares $\rightarrow [w_1]$ (HighBits carry)
- ④ verify BCC on shares $\rightarrow$ drop failing nonces $\leftarrow$ rejection, offline
- ⑤ pool of BCC-passing nonces (w_1 held, not revealed)

ONLINE (μ arrives)

- R1 release $w_1 \rightarrow c = \text{SampleInBall}(H(\mu \| w_1))$
- R2 reveal $z_i = \hat{y}_i + c \cdot s_{1i} \rightarrow$ aggregate, hint, verify $\rightarrow \sigma = (\tilde{c}, z, h)$

A coefficient sits in a grid cell

← one grid cell ($2\gamma_2$) →



SAFE

HighBits rounds $r = A \cdot y$ to a grid. r_0 says how far each coefficient is from a boundary.

The secret can only nudge it by β



$c \cdot s_2$ has size $\leq \beta$. Near an edge (within β) it might cross — danger. In the interior it can't.

Clear by $\beta \rightarrow$ the secret drops out



BCC: every coefficient clears its boundary by β . Then the r_0 -check passes for any secret — $c \cdot s_2$ never needs to be computed.

Note 2026-09-11: $c \cdot s_2$ is indeed never computed, but each signature still leaks a noisy observation of s_2 [ia.cr/2026/1386] — quantified and capped on the “Update” slide.

Why one in three

ML-DSA-65: $\gamma_2 \approx 261,888$ $\beta = 196$

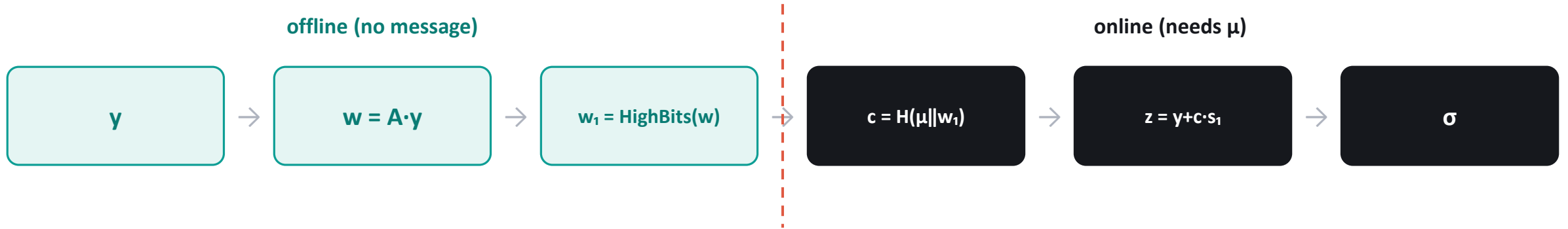
per coefficient in danger band: $196 / 261,888 \approx 0.075\%$

but ALL 1,536 coefficients must clear:

$$(1 - 196/261,888)^{1536} \approx 31.7\%$$

Checkable from $A \cdot y$ alone, without the secret $\rightarrow$ filter offline until one passes.

Compute offline, reveal online



Everything up to w_1 is message-independent → precompute offline.

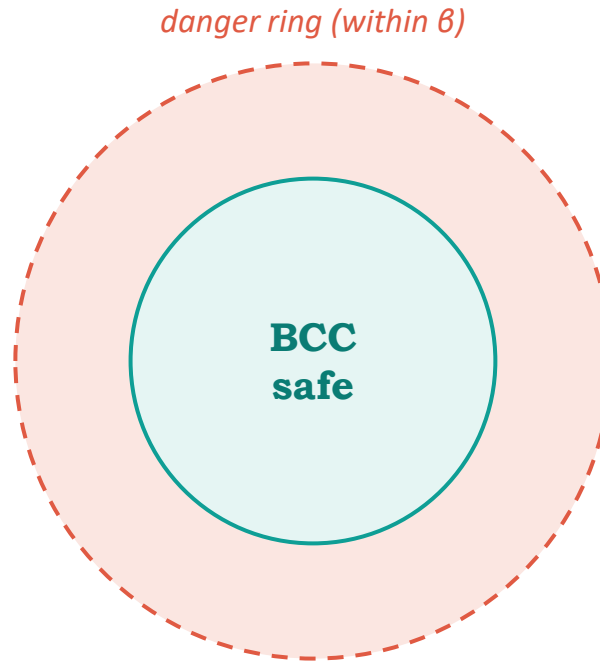
Revealing w_1 is what costs a round

Computing w_1 can be offline. But **revealing** it feeds $c = H(\mu||w_1)$ — so it must wait for the message, in both profiles.

The difference: does releasing w_1 need its own round of interaction? That's what separates the two profiles.

OUR MOVE

Filter the danger away — offline



keep only the core $\rightarrow r_o$ -check can't fail online, no secret-dependent reject

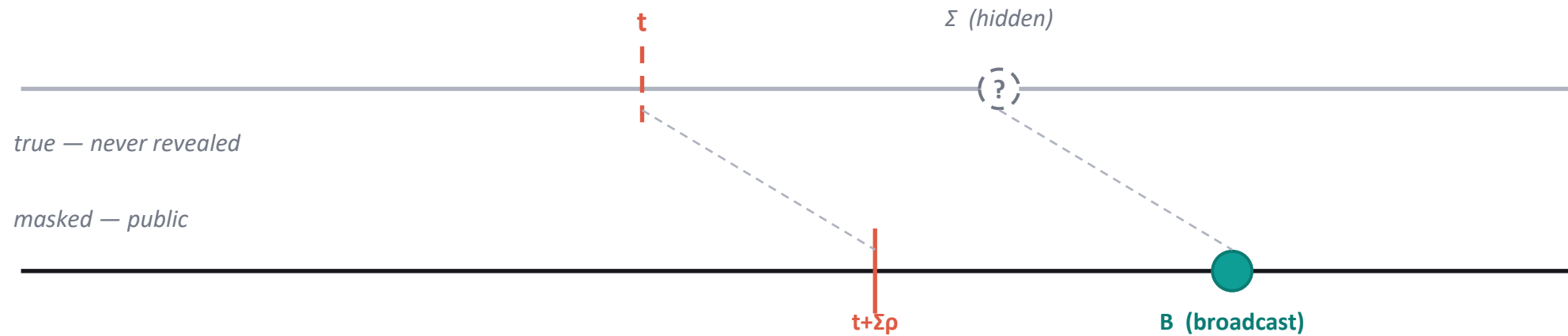
Note 2026-09-11: Moving the secret-dependent rejection offline is what opens that channel [ia.cr/2026/1386] — see the “Update” slide.

On shares, HighBits carries

No party holds the nonce $\rightarrow$ w is shared. To get $w_1 = \text{HighBits}(w)$ you must resolve a carry.

Adding the low parts of shares can overflow the boundary. **$\text{HighBits}(\text{sum}) \neq \text{sum}(\text{HighBits})$.**

Compare without revealing the sum



Everything shifts by $\Sigma\rho$ — the gap is unchanged, so **B's side of $(t + \Sigma\rho) = \Sigma$'s side of t .**

$$B = \Sigma r_{0i} + \Sigma \rho_i \quad \xi = [B \geq t + \Sigma \rho_i]$$

Only the one bit ξ is released — the true sum is never shown (it would be a noise-free lattice sample).

Offline: build a pool of safe nonces

- ① each party samples nonce share $\hat{y}_i$
(no one holds the aggregate y)
- ② each party locally computes $A \cdot \hat{y}_i \rightarrow [w]$ (A public, linear; w never revealed)
- ③ CSCP on shares $\rightarrow [w_1]$, verify BCC $\rightarrow$ keep only passing nonces

Online: reveal w_1 , then reveal z

Round 1 reveal w_1 $\rightarrow$ each party computes $c = \text{SampleInBall}(H(\mu \| w_1))$

Round 2 reveal $z_i = \hat{y}_i + c \cdot s_{1i}$ $\rightarrow$ aggregate z , hint, verify, emit $\sigma=(c,z,h)$

σ is a standard FIPS 204 signature — any unmodified verifier accepts it.

TALUS-TEE — clean one round

The enclave computes w_1 offline and holds it. After μ : it sends w_1 down and collects z back — one round-trip.

The one deployment that cleanly achieves one online round — under the enclave assumption.

Cost = hardware trust. Compromise the enclave $\rightarrow$ nonce $\rightarrow$ key, as with any single-party device.

Note 2026-09-11: TALUS-TEE is retained as a contrasting example only and is not proposed for the Threshold Call.

TALUS-MPC — no trusted hardware

No enclave $\rightarrow$ releasing the held w_1 needs its own round of interaction (no single party can).

Two online rounds (R1 open $w_1 \rightarrow c$; R2 reveal $z \rightarrow \sigma$), standard lattice assumptions, no hardware trust.

The offline shared- w_1 construction and full analysis are being finalized for the package.

Online rounds, per attempt

Scheme	Online rounds	Parties	Model
Mithril	3 / attempt	≤ 6	dishonest-maj
Trilithium	14 (~60 w/ rejection)	2	needs a CRP
Quorus	16 – 29	any N	honest-maj · UC
TALUS	MPC 2	any N ($N \geq 2T-1$)	honest-maj

All output standard FIPS 204 signatures. BCC changes the order of magnitude by moving rejection offline.

Note 2026-09-11: TALUS-TEE is withdrawn from the proposal; this row now reports the proposed MPC profile only.

Honest about what changes

- Standard FIPS 204 output — any unmodified verifier accepts it
- FIPS constrains verification, not the distribution — BCC filtering does change the distribution (public filter)
- s_1 is never reconstructed
- Cost ≈ 3 offline nonce trials / signature — offline only

TAKEAWAY

Established, and in preparation

Established: the BCC observation — a public geometric condition making ML-DSA's secret-dependent check redundant for a third of nonces. It stands on its own.

In preparation: the full security analysis and the fully-distributed protocol → NIST package submission.

What we're building next

- Complete the fully-distributed protocol — the shared- w_1 construction and full security analysis
- Production code package, aligned with the NIST submission requirements
- Real-world deployment testing — performance and integration under WAN conditions
- Finalize the full write-up and complete its peer review

All of the above feed our NIST Threshold Call package submission.

Thank you

Happy to take questions.

Leo Kao · talus@codebat.ai · joint work with Raymond Chang