Pilaf

Tightly and Adaptively Secure Two-Round Threshold Signatures with Preprocessing

Presented by Chen Qian
Shandong University
Joint work with Xingyu Zhao, Hao Cheng, Zengpeng Li, Puwen Wei, and Quan Yuan

2026-10-13 @ NIST TCPT3
Threshold Call Preview Talks #3

Associated preprint: ia.cr/2026/1762
Technical content follows the current revised manuscript

What Pilaf provides

2 **signing rounds**

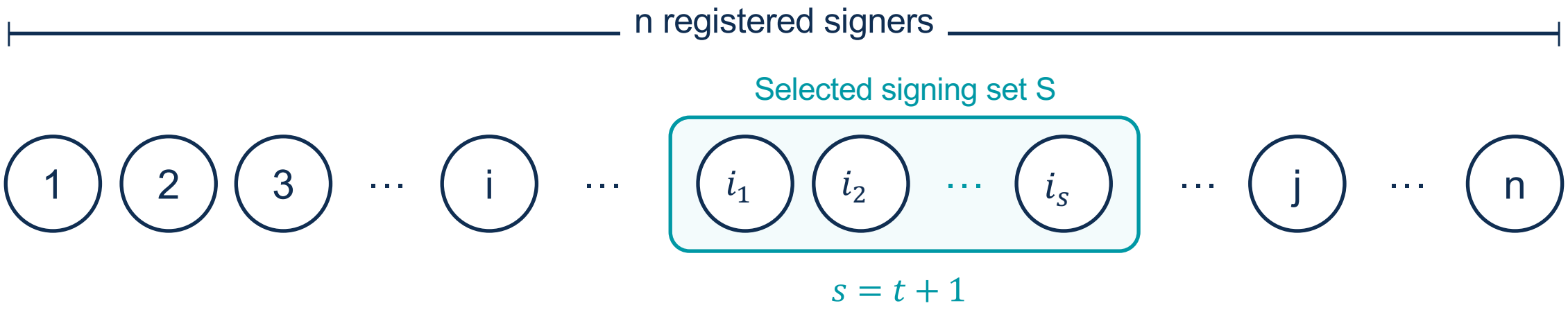


1. Two signing rounds with complete message-independent preprocessing
2. Adaptive security without erasures in the classical random-oracle model (ROM)
3. Decisional Diffie–Hellman (DDH); constant multiplicative reduction loss

4. At $k = 1$: 2 group elements + 15 field elements
5. Custom verifier; security with abort

Threshold and signing model

$$0 \le t < n < q \quad | \quad f \le t \quad | \quad |S| = s = t + 1$$



- n registered signers; at most t corruptions
- Fixed signing quorum: $s = t + 1$
- Centralized key generation; Shamir shares of degree t

- Fix identifier and signer set before preprocessing
- One successful round-two response per signer and identifier
- Completion requires every selected signer to cooperate

The underlying signature

$$a = a_0 + a_1$$

	Branch $b = 0$	**Challenge (shared)**	**Branch $b = 1$**
Share split	a_0		a_1
Share image	$X_0 = F(a_0)$		$X_1 = F(a_1)$
Response	$\rho_0 = z_0 - ca_0$	$c = H(\dots)$	$\rho_1 = z_1 - ca_1$
Reconstruction	$R_0 = F(\rho_0) + cX_0$		$R_1 = F(\rho_1) + cX_1$

Other share image: $X_1 = PK - X_0$

Compact signature: $\sigma = (X_0, \rho_0, \rho_1, r_0^x, r_0^R, r_1^x, r_1^R, c)$

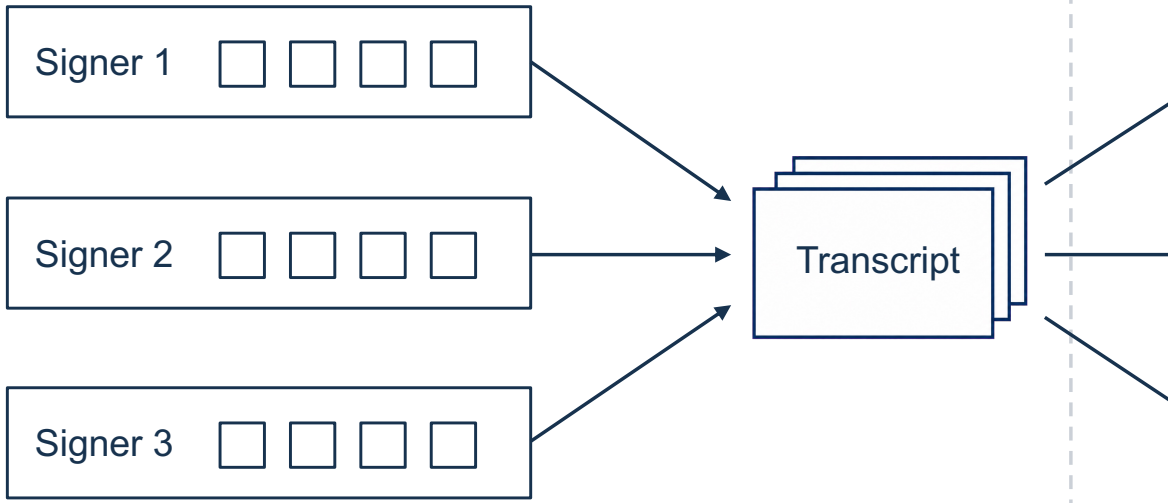
- Split one signing secret into two branches
- Commit to share and nonce images before the message challenge
- Use one challenge $c = H(\dots)$ for both branches
- Reconstruct the other share image, both nonce images, and four commitments
- Check the same challenge hash; a standard Schnorr verifier is incompatible

Two rounds with complete preprocessing

Fixed before Round 1: identifier id and signer set S

Round 1

Offline



Split shares and sample nonces

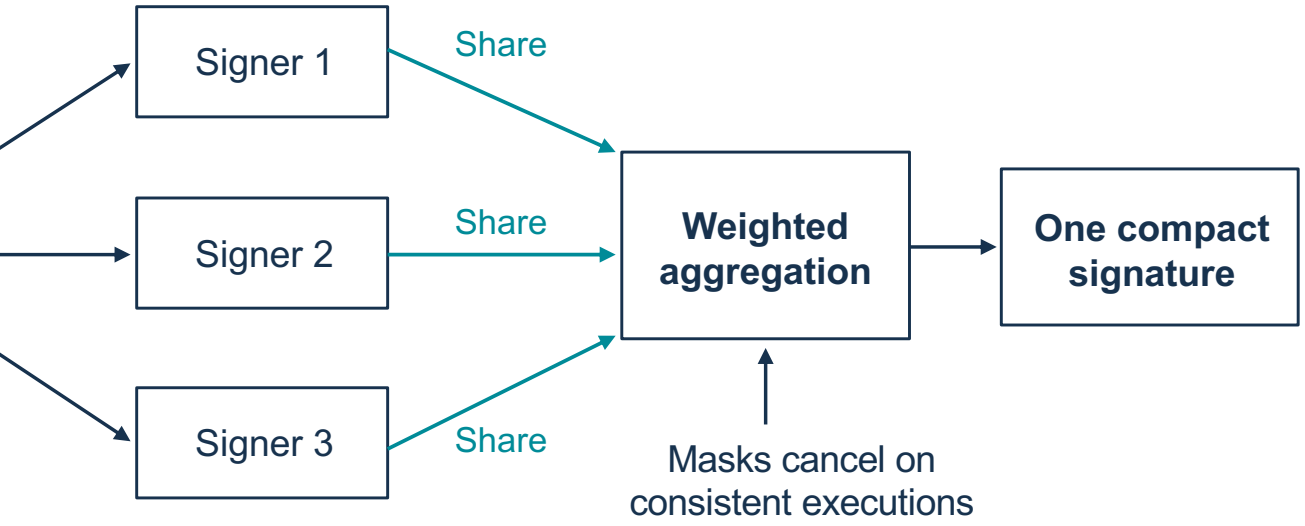
4 masked commitments per signer

Message-independent output, state, randomness

Round 2

Online

Message m

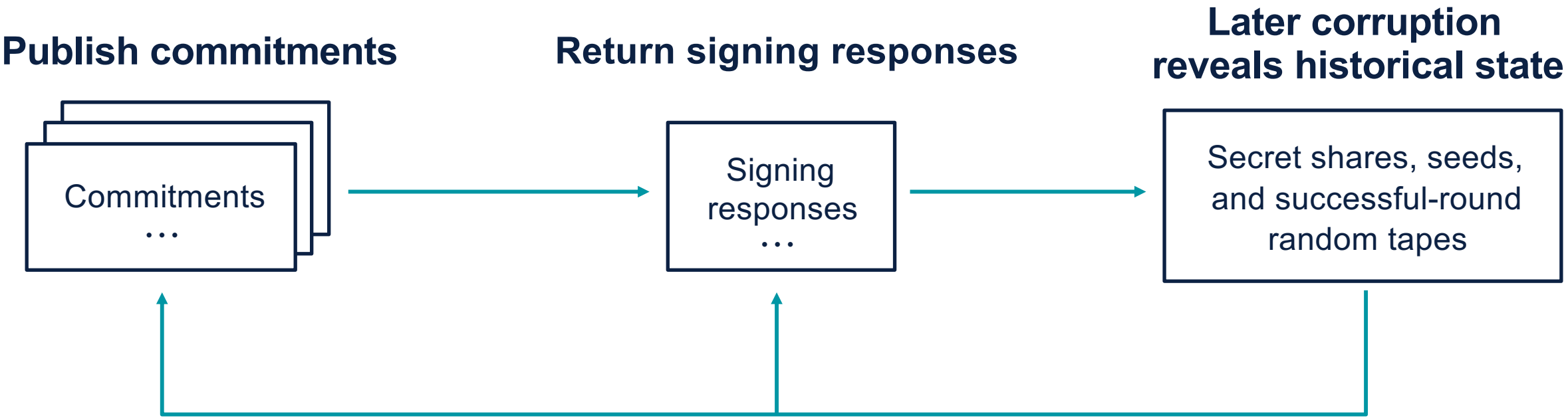


Challenge from aggregate commitments and message

Masks bind the exact transcript and message

Masked images, responses, openings

Adaptive corruption exposes the past



All earlier outputs and later leakage must fit one execution

No erasures; concurrent sessions; message equivocation

Transcript-local freshness

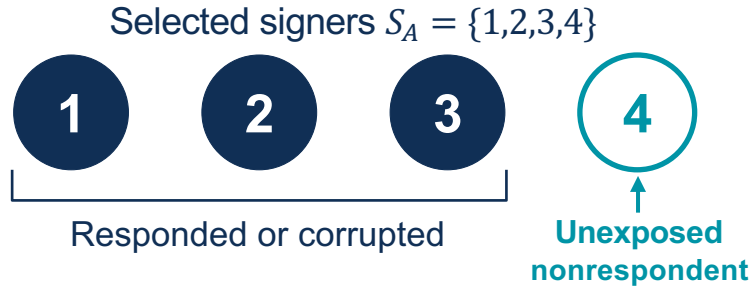
$\lvert S\rvert = t + 1$	For every τ: $\lvert Q_2[\tau] \cup (C \cap S)\rvert \le t$	$\tau = (PK,\ \text{first−round transcript},\ id, S, m)$

Identifier id (same for both transcripts)

$t = 3,\ s = 4 \quad (|S| = t + 1)$

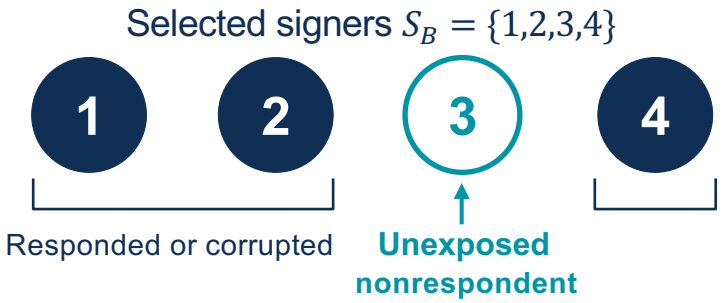
Transcript τ_A

$\tau_A = (PK, \rho_A, id, S_A, m)$



Transcript τ_B

$\tau_B = (PK, \rho_B, id, S_B, m)$



Track successful respondents separately for each exact transcript τ

$Q_2[\tau]$: successful round-two respondents

C: finally corrupted signers in this instance

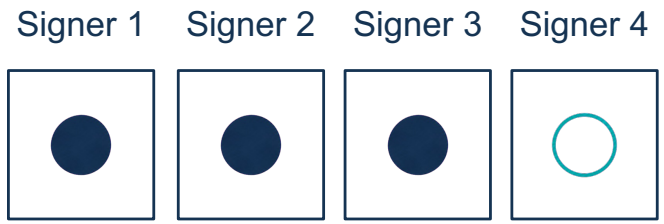
Every transcript retains an unexposed nonrespondent

Target preprocessing and partial responses are allowed

A completed transcript consumes freshness for the entire identifier

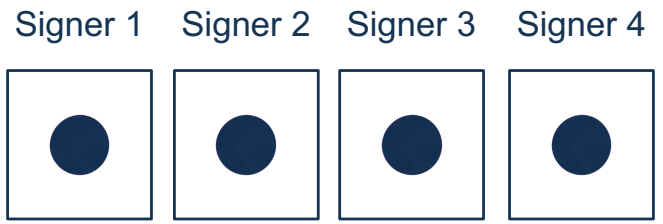
Last touch and the hidden coordinate

Before last touch



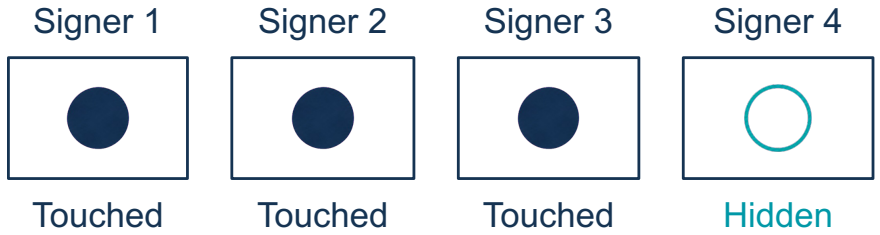
Response or corruption

After last touch



Latent state completes aggregate relation at cached challenge

Fresh target



Keeps a hidden coordinate

- A response or corruption touches one selected signer
- Before last touch, one coordinate remains hidden
- Use latent state to complete the aggregate relation at the cached challenge
- Previously observed commitments and responses stay fixed
- Fresh targets never reach last touch

Concrete homomorphic commitments

$$\mathrm{Com}(K, M; r) = \mathrm{pad}(M) + Kr$$

Simulation keys: $K = QD;\ D$ invertible

Fixed commitment

$$C = Q\zeta$$

Preserve C

Old represented target

$$M = Ud \qquad \zeta = \hat{d} + Dr$$

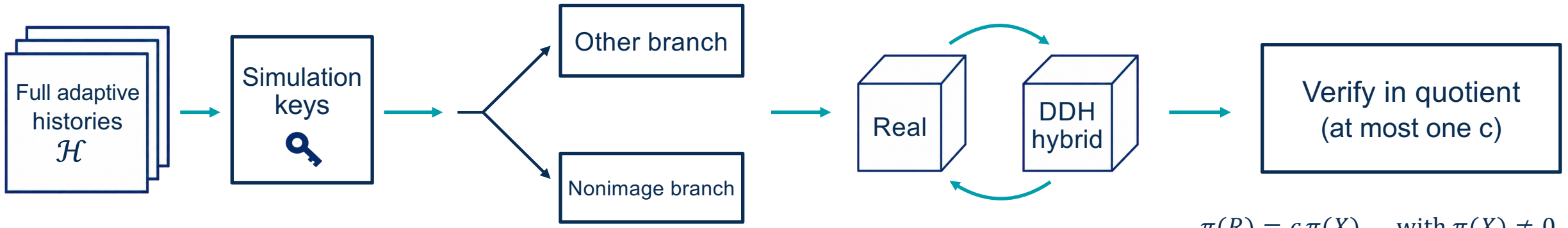
New represented target

$$M' = Ud' \qquad r' = D^{-1}(\zeta - \hat{d}')$$

- Linear commitments aggregate with the signing shares
- Simulation keys: $K = QD;\ D$ invertible
- Retain the full scalar trace to explain later corruption
- Open to a new represented target while preserving C
- Binding is established in a separate shared-factor hybrid
- Padded commitment and factorized trapdoor technique from TZAR

A constant-loss proof

1. **Simulation keys:** reconcile full adaptive histories
2. **Normal extension:** an accepting split has a nonimage branch
3. **Hidden branch and joint DDH switches:** shared-factor binding
4. **Quotient verification:** at most one scalar challenge



Outside the bad-image event

Branch probability $\geq 1/2$

$\pi(R) = c\,\pi(X), \quad \text{with } \pi(X) \neq 0$

π: quotient by the image of F

No guessing loss in users, threshold, instances, or queries

Constant multiplicative loss; additive errors remain

Why the reduction loses only a constant factor

Simulate history → **Choose nonimage branch** → **Bind all source keys jointly**

A hidden fair bit selects a nonimage branch with probability at least $1/2$

For fixed X, R with $X \notin \mathrm{Im}F$, at most one challenge c can work.

$$R = F(\rho) + cX$$

$$\mathrm{Adv}_{\mathrm{forge}} \leq \frac{6q}{q-1}\varepsilon_{\mathrm{DDH}} + \varepsilon_{\mathrm{add}}$$

Query counts and parameters still appear in the additive error

Constant multiplicative loss, independent of k, users, threshold, instances and queries

Signature size and communication

Quantity	Element count
Round 1 / signer	$12k^2G$
Round 2 / signer	$8k^2G + 14k^2F_q$
Final signature	$\mathbf{2k^2G + (14k^2 + 1)F_q}$

At $k = 1$: 2 group elements + 15 field elements

- Round-two count assumes the receiver retains the supplied round-one transcript
- Counts exclude message, identifiers, signer-set encoding, keys, setup, and framing
- Custom signature format and verifier

Comparison with other submissions

PNI: entire round-1 output, state and tape are message-independent.

Scheme	Assumption / model	PNI	Adaptive	Security loss L	Freshness
FROST1	OMDL / ROM	Yes	No	$O(n^2Q^2/\varepsilon)$	TS-UF-3 (static)
ms-FROST	AOMDL / AGM+ROM	Yes	Yes	$O(1)$	TS-UF-4
Li–Liu–Lyu	DDH / ROM (k = 2)	No	Yes	$O(Q_0 + Q_1/\varepsilon)$	TS-UF-1
TZAR	DDH / ROM	No	Yes	$O(1)$	Target R1 excluded
Pilaf	DDH / ROM	Yes	Yes	$O(1)$	TS-UF-2 on matched domain

Advantage loss: $L = \varepsilon/\varepsilon_{\text{base}}$. Additive errors and simulation overhead omitted.

ε: forgery advantage. n: signers. $Q = q_P + q_H + 1$. Q_i: queries to H_i.

Freshness scope

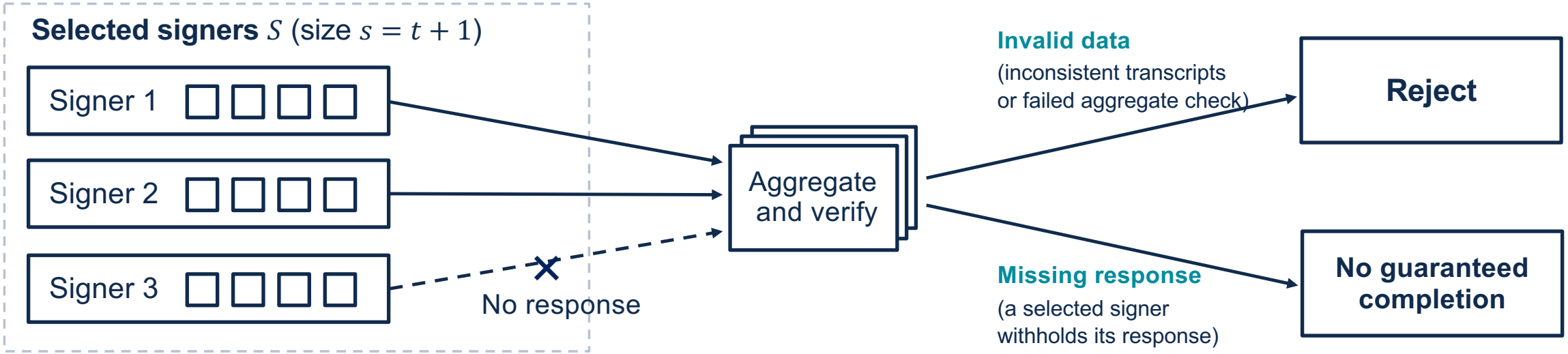
TS-UF-2 containment is restricted to the matched static single-instance domain.

Relation to TZAR

TZAR is independent concurrent work. Pilaf adopts its factorized commitments.

Failure handling and termination

Selected set S with a missing response



Security with abort

The protocol remains secure even if the attempt does not complete.

- Inconsistent transcripts or a failed aggregate check: reject
- A selected signer withholds its response: the attempt can stop
- No proved culprit-identification or recovery protocol
- Extra honest parties outside the selected set do not automatically replace a signer

Takeaways and discussion

1. **Two rounds with complete message-independent preprocessing**

2. **Adaptive security without erasures under DDH in the classical ROM**



Tradeoffs and open questions

3. Constant multiplicative reduction loss: approximately 6

4. At $k = 1: 2G + 15F_q$; custom verifier; fixed quorum; security with abort

5. Deployment evaluation remains open

Thank you

ia.cr/2026/1762

Backup: exact security bound

$$\mathrm{Adv} \le 6\kappa\varepsilon_D + \frac{Q_M}{2^\lambda} + 4Rd_{3k} + 3d_k + 2\eta_k + \beta_{J,k} + \frac{2L}{q}$$

$$d_r = 1 - \prod_{i=4}^{r}(1 - q^{-i})$$

$$\eta_k = 1 - \prod_{i=k+1}^{r}(1 - q^{-i})$$

$$\beta_{J,k} = d_k + (1 - d_k)[1 - (1 - q^{-k^2})^J]$$

- ε_D: maximum advantage of four DDH reductions; $\kappa = q/(q-1)$
- $R = Q_0 + Q_K + Q_H + 1$; $L = Q_H + Q_1 + 1$
- J bounds registered instances; λ is the masking-seed length

- Polynomially bounded parameters and query budgets; $0 \le t < n < q$
- Each reduction runs the adversary once with polynomial overhead

Backup: the matched TS-UF-2 domain

1. Static corruption and a single matched instance
2. Match signed items: identifier, signer set, and message
3. Match leader requests to exact first-round transcripts
4. All successful responses under the target identifier use the target message

Containment holds on this matched domain

Outside it, completing another message can consume identifier freshness

Backup: comparison with TZAR

Both: two rounds and fully tight adaptive DDH security under their respective interfaces

Feature	Pilaf	TZAR
First round	Completely message-independent	Message-dependent
Fresh target interaction	Preprocessing + partial responses	Target round-one queries excluded
Commitment technique	Adopted factorized construction	Source of factorized construction

Pilaf fixes exactly $t + 1$ signers and pays additional communication

Pilaf adopts TZAR's padded commitment and factorized trapdoor technique

Current manuscript: Appendix B; TZAR §4.1

Backup: version differences and references

Earlier preview

- MDDH framework
- Threshold denoted the signing quorum

Current manuscript

- Direct DDH for every polynomially bounded $k \geq 1$
- t bounds corruptions, $s = t + 1$

Acronyms

DDH: Decisional Diffie–Hellman

MDDH: Matrix Decisional Diffie–Hellman

FROST: Flexible Round-Optimized Schnorr Threshold signatures

AOMDL: Algebraic One-More Discrete Logarithm

Selected readings

1. **Pilaf**
   https://ia.cr/2026/1762
2. **FROST**
   RFC 9591; NIST Phase 1 preview
3. **FaFROST: Fully Adaptive FROST with Identifiable Aborts from AOMDL**
   https://ia.cr/2025/1950
4. **TZAR**
   https://ia.cr/2026/1909