Blind Signatures from Proofs of Inequality

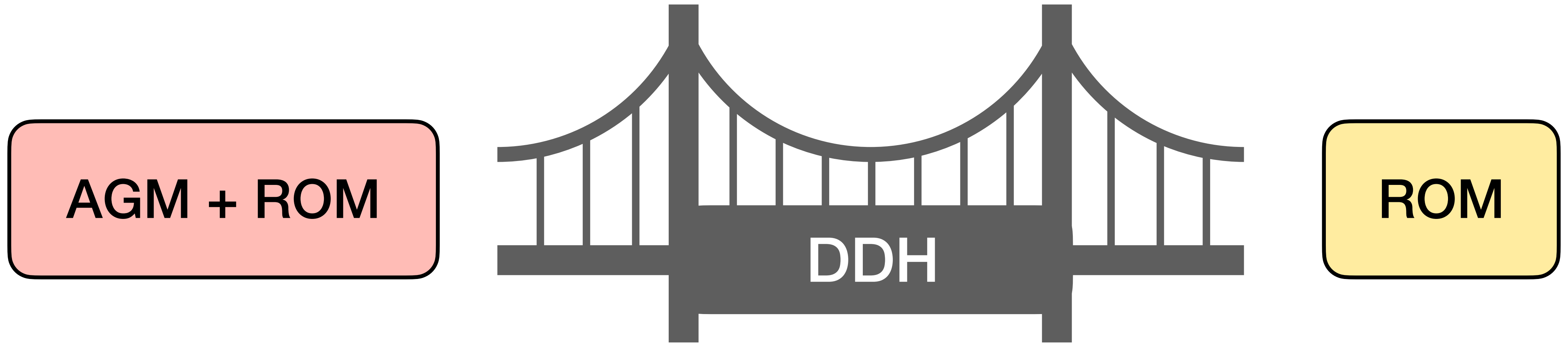
Michael Klooß KIT

Michael Reichle ETH Zurich

Our Contribution

Blind Signatures

- ***Bridge gap in performance between AGM and AGM-free schemes***
  - pairing-free groups
  - standard assumptions in ROM



AGM + ROM

DDH

ROM

Our Contribution

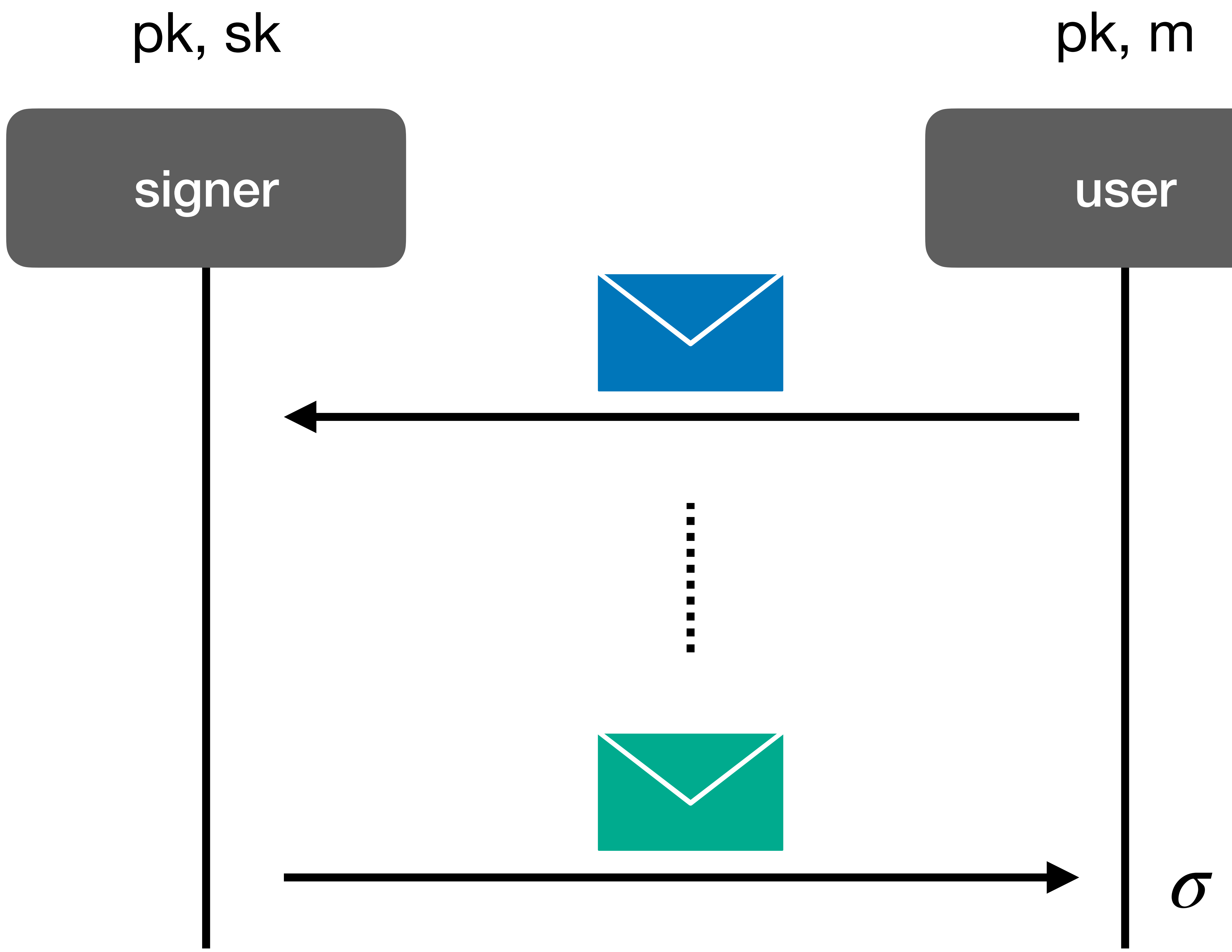
Blind Signatures

- ***Bridge gap in performance between AGM and AGM-free schemes***

Scheme*	Signature Size	Communication Size	Security	Assumption
[CKMTZ23]	$1\mathbb{G} + 2\mathbb{Z}_p$	$2\mathbb{G} + 4\mathbb{Z}_p$	AGM + ROM	DL
[KRW24]	$2\mathbb{G} + 5\mathbb{Z}_p$	poly(λ)	ROM	DDH
Our Work	$1\mathbb{G} + 5\mathbb{Z}_p$	$10\mathbb{G} + 9\mathbb{Z}_p$	ROM	DDH

*representatives for compact AGM and AGM-free blind signatures

Blind Signatures



Correctness:

- honest signatures verify

Blindness:

- signatures are *unlinkable* to signing sessions

One-more Unforgeability:

- user can obtain at most ℓ signatures from ℓ sessions with distinct messages

Our Techniques

Pairing-free blind signature in the ROM

- **Starting Point:** build on recent progress **[CTZ24,KRW24]**
  - remove reliance on NIZK Π for scalars in **[KRW24]**
- **Contributions:**
  - employ tailored Σ-protocol
  - NIZK Π for group elements → less communication
  - **bonus:** $1\mathbb{G}$ smaller signatures

Issuance in [KRW24]

💡 replace pairing-based verification of **[KRS23]** via FS-compiled Σ-protocol

Issuance in [KRW24]

$\mathbf{D} = (D_1, D_2, D_3)$

pk, sk
signer
pk, m
user
C π_m
T, R
c
z
$\mathbb{Z}_p$
⋓
σ

$T = \mathsf{Sign}(\mathsf{sk}, \mathsf{C})$

$R = \Sigma.\mathsf{Init}(\mathbb{X}, \mathbb{W})$

$T = \mathsf{Sign}(\mathsf{sk}, \mathsf{C})$ or $\mathbf{D}$ is DDH tuple

$z = \Sigma.\mathsf{Resp}(\mathsf{c})$

$C = \mathsf{Com}(\mathsf{m}; \mathsf{r})$

$\pi_m = \mathsf{Prove}(''C \text{ is well formed}'')$

$S = T - \mathsf{AdjustSig}(\mathsf{pk}, \mathsf{r})$

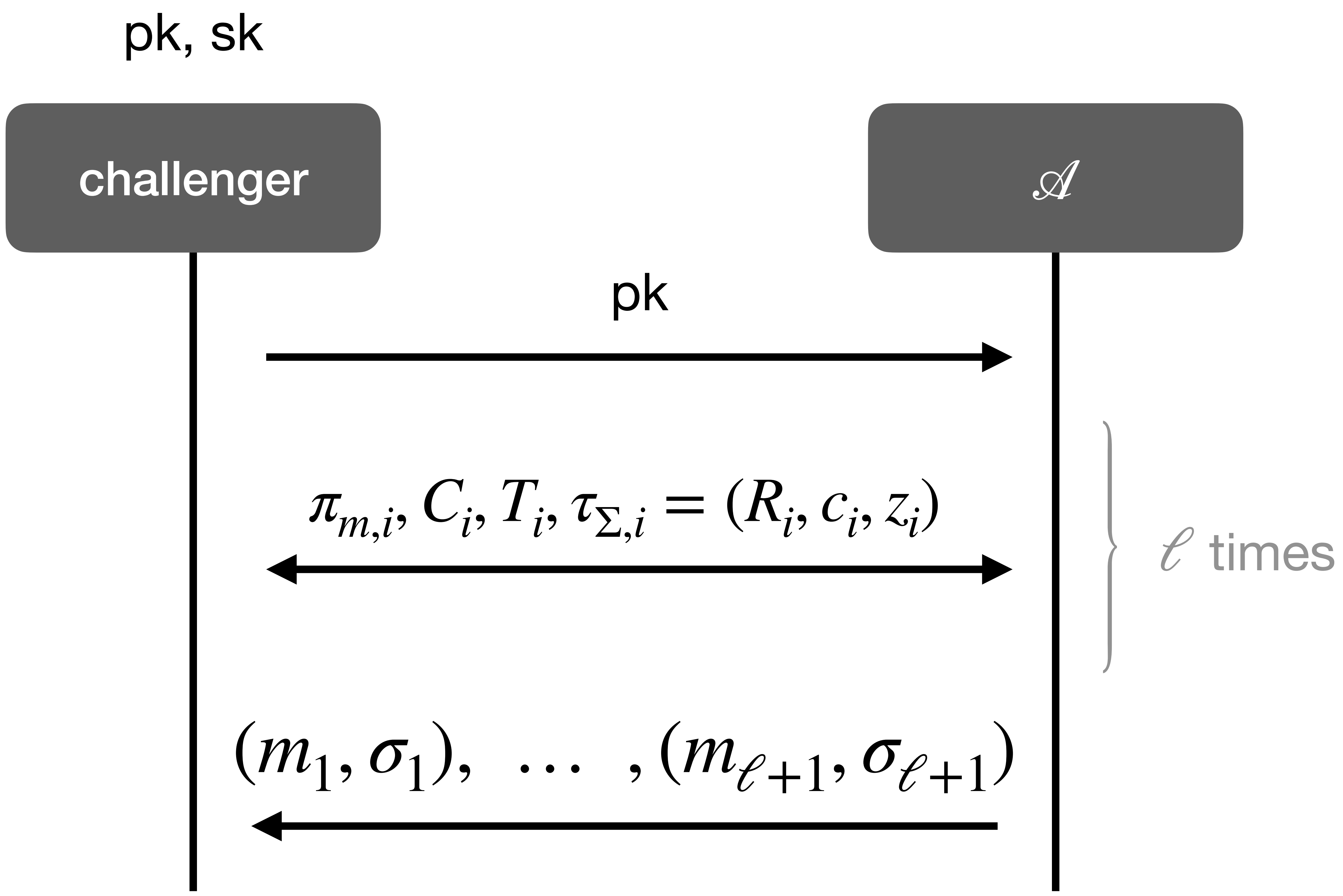
compile proof $\pi = (R, c, z)$ via Fiat-Shamir

$S = \mathsf{Sign}(\mathsf{sk}, \mathsf{m})$ or $\mathbf{D}$ is DDH tuple

One-more Unforgeability

Approach of [KRW24]

$\mathbf{D} = (D_1, D_2, D_3)$


pk, sk
challenger
$\mathscr{A}$
pk
$\pi_{m,i}, C_i, T_i, \tau_{\Sigma,i} = (R_i, c_i, z_i)$
ℓ times
$(m_1, \sigma_1), \ \ldots \ , (m_{\ell+1}, \sigma_{\ell+1})$

$\mathscr{A}$ succeeds if:

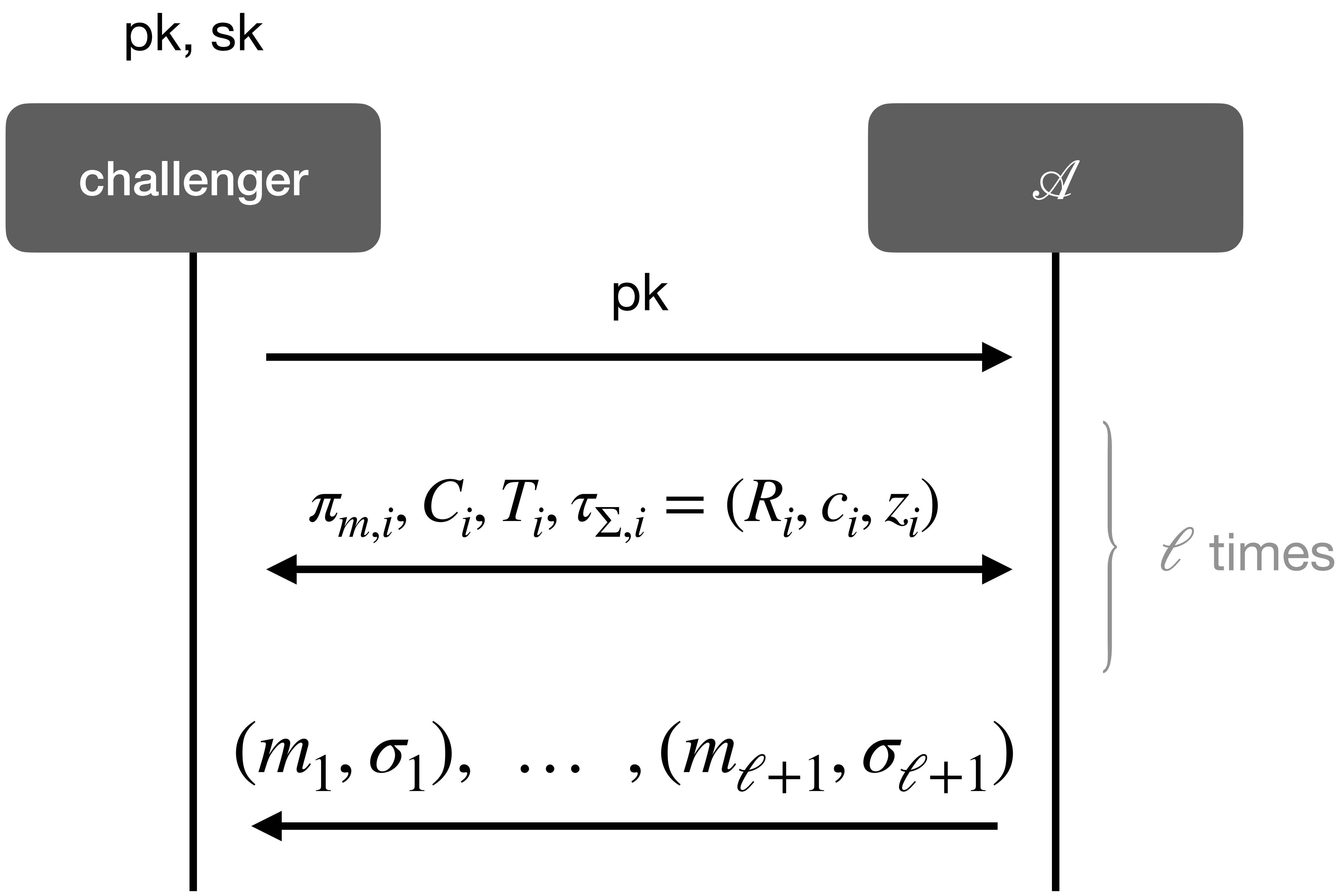
- (m_i, σ_i) verifies,
- m_i pairwise distinct.

One-more Unforgeability

Approach of [KRW24]

- **Step 1**: extract to-be-signed (m_i, r_i) from proof $\pi_{m,i}$
  - requires extracting scalars
  - compute T_i via signature on m_i accounting for r_i

$\mathbf{D} = (D_1, D_2, D_3)$


pk, sk
challenger
$\mathscr{A}$
pk
$\pi_{m,i}, C_i, T_i, \tau_{\Sigma,i} = (R_i, c_i, z_i)$
ℓ times
$(m_1, \sigma_1), \ \dots \ , (m_{\ell+1}, \sigma_{\ell+1})$

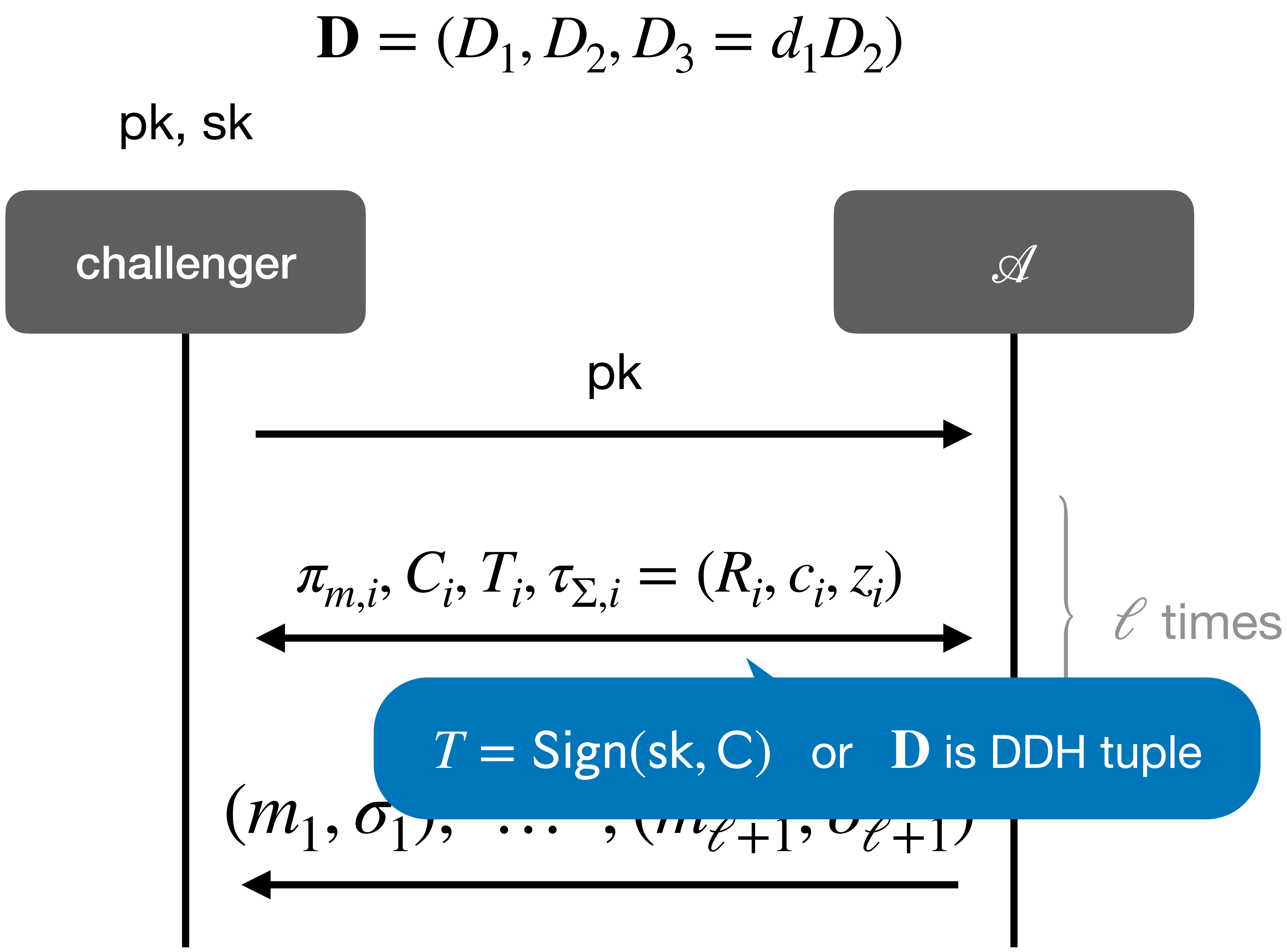
$\mathscr{A}$ succeeds if:

- (m_i, σ_i) verifies,
- m_i pairwise distinct.

One-more Unforgeability

Approach of [KRW24]

- **Step 2:** simulate transcript $\tau_{\Sigma,i}$ via DDH-tuple $\mathbf{D}$
  - simulate Sign-branch
  - compute DDH-branch via d_1


$\mathbf{D} = (D_1, D_2, D_3 = d_1 D_2)$
pk, sk
challenger
$\mathscr{A}$
pk
$\pi_{m,i}, C_i, T_i, \tau_{\Sigma,i} = (R_i, c_i, z_i)$
ℓ times
$T = \mathsf{Sign}(\mathsf{sk}, \mathsf{C})$ or $\mathbf{D}$ is DDH tuple
$(m_1, \sigma_1), \cdots, (m_{\ell+1}, \sigma_{\ell+1})$

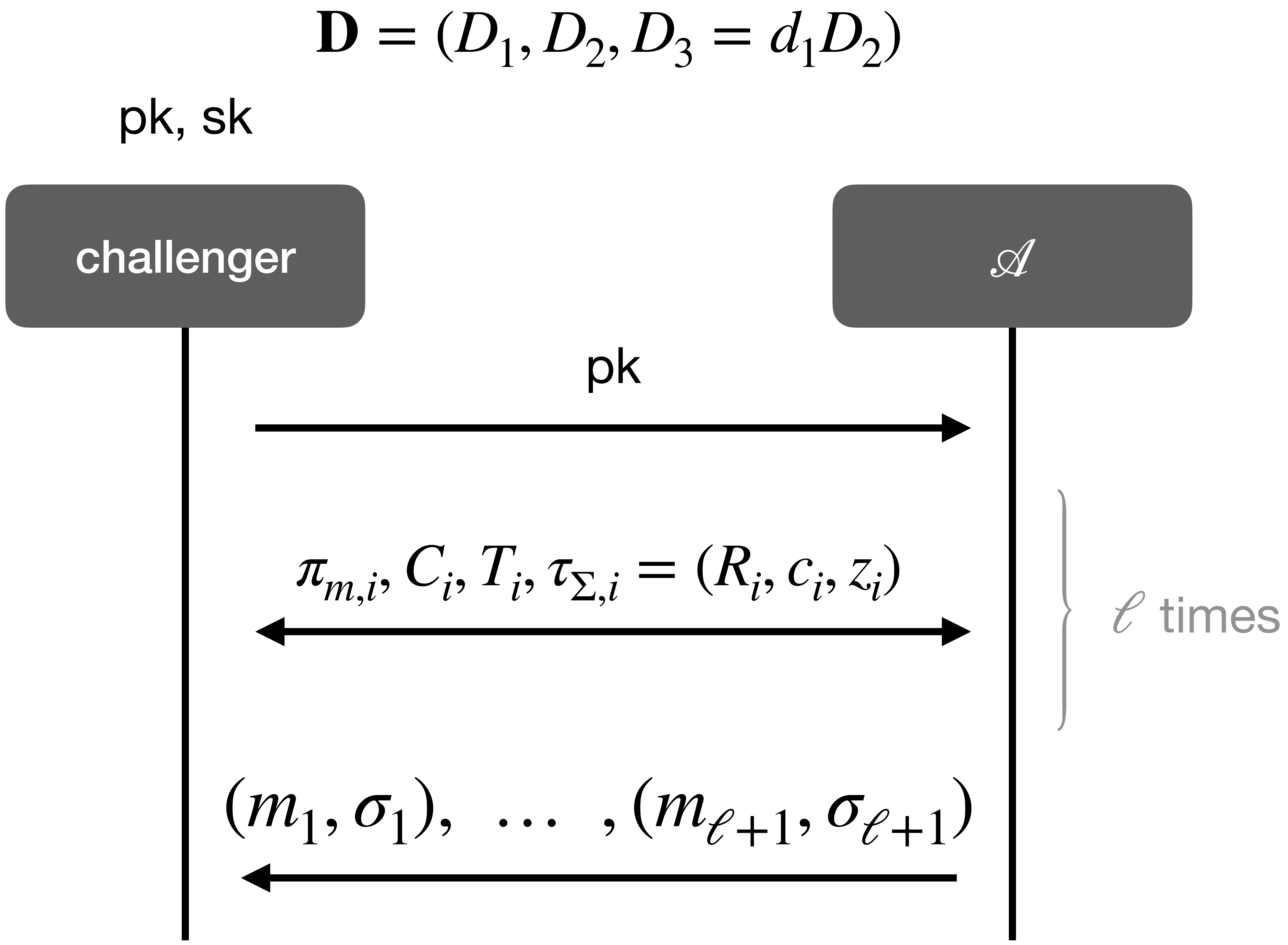
$\mathscr{A}$ succeeds if:

- (m_i, σ_i) verifies,
- m_i pairwise distinct.

One-more Unforgeability

Approach of [KRW24]

- **Step 3:** puncture pk on some message m^*
  - force adversary to provide forgery for m^*
  - never sign m^* in simulation



$\mathbf{D} = (D_1, D_2, D_3 = d_1 D_2)$

pk, sk
challenger
$\mathscr{A}$
pk
$\pi_{m,i}, C_i, T_i, \tau_{\Sigma,i} = (R_i, c_i, z_i)$
ℓ times
$(m_1, \sigma_1), \quad \ldots \quad , (m_{\ell+1}, \sigma_{\ell+1})$

$\mathscr{A}$ succeeds if:

- (m_i, σ_i) verifies,
- m_i pairwise distinct.

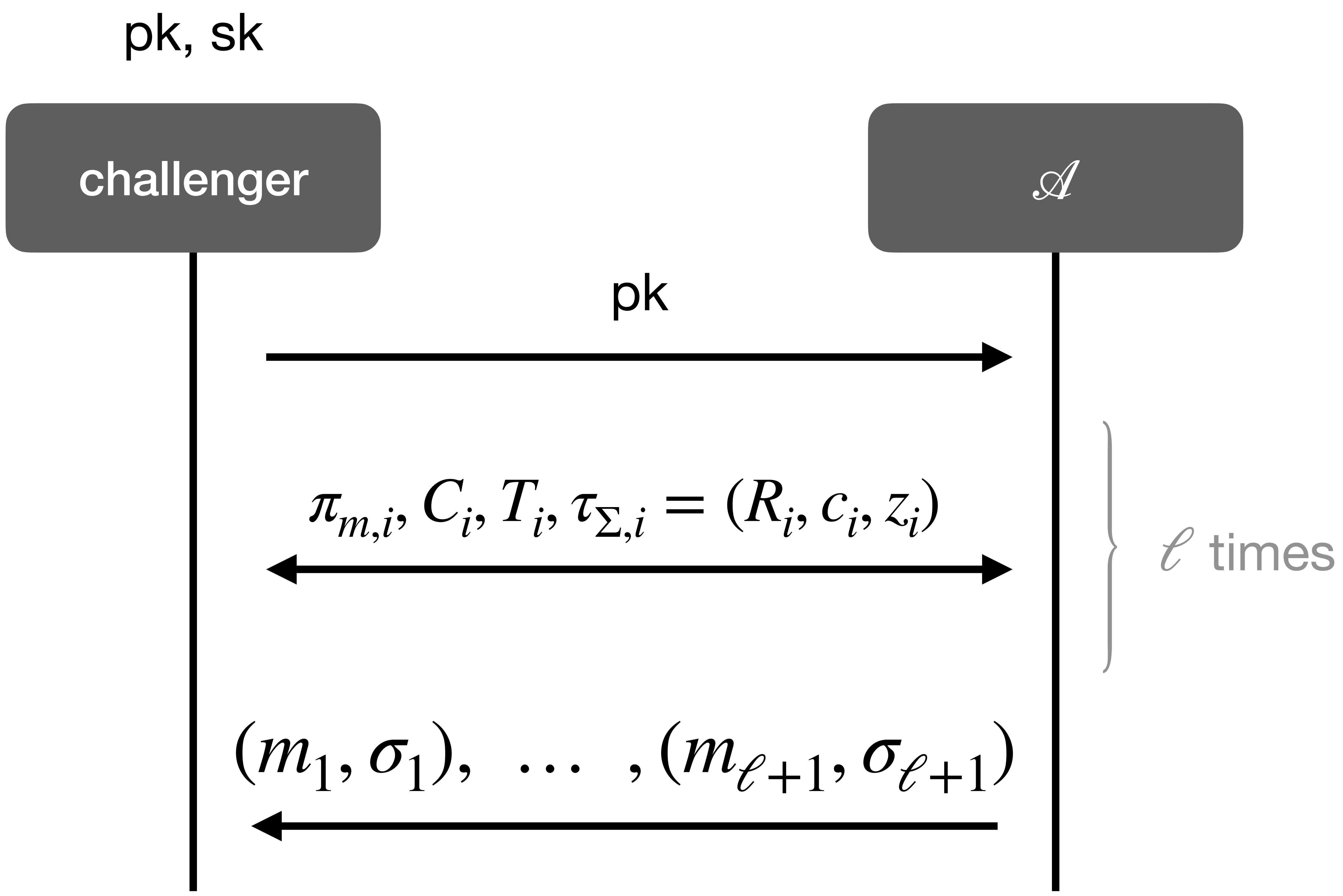
One-more Unforgeability

Approach of [KRW24]

- **Soundness:**
  - signature S^* on m^* valid

    → solves hard problem

$\mathbf{D} = (D_1, D_2, D_3 = d_1 D_2)$


pk, sk
challenger
$\mathscr{A}$
pk
$\pi_{m,i}, C_i, T_i, \tau_{\Sigma,i} = (R_i, c_i, z_i)$
ℓ times
$(m_1, \sigma_1), \quad \ldots \quad, (m_{\ell+1}, \sigma_{\ell+1})$

$\mathscr{A}$ succeeds if:

- (m_i, σ_i) verifies,
- m_i pairwise distinct.

Our Approach

Tailored Trapdoor based on [BS02, CS03]

- **Idea:** craft tailored statement $\mathbb{X}$ for Fiat-Shamir such that
  - $\mathbb{X}$ can be punctured over $\mathbb{G}$ → message extracted from π_m is in $\mathbb{G}$
  - $\mathbb{X}$ is compact and linear → efficient blind issuance
- **Statement $\mathbb{X}$:** inequality of encrypted messages

$C := C^* - \mathrm{Enc}(\mathrm{pk}, M; 0)$ does not encrypt 0

Our Approach

Tailored Trapdoor

$$\Phi(C,(x,y)) = \begin{pmatrix} yH - xG \\ yC_1 - xC_0 \end{pmatrix}^T = \begin{pmatrix} 0 \\ yM \end{pmatrix}^T$$

"x is scaled decryption key"

- **Statement:** $C = (C_0, C_1) = (rG, M + rH)$ does not encrypt 0
- **Idea:** scale decryption by y (*i.e.*, decrypt yC via $x = y \cdot \mathsf{sk}$)

Our Approach

Tailored Trapdoor

$$\Phi(C,(x,y)) = \begin{pmatrix} yH - xG \\ yC_1 - xC_0 \end{pmatrix}^T = \begin{pmatrix} 0 \\ yM \end{pmatrix}^T$$

"yC decrypts to yM"

- **Statement:** $C = (C_0, C_1) = (rG, M + rH)$ does not encrypt 0
- **Idea:** scale decryption by y (*i.e.*, decrypt yC via $x = y \cdot \mathsf{sk}$)
- **Observation:**
  - can reveal $M_\$:= yM \sim U_{\mathbb{G}^\times}$ for $M \neq 0, y \leftarrow \mathbb{Z}_p^\times$
  - if $M_\$ \neq 0$ then $M \neq 0$

Our Approach

Tailored Trapdoor

- **Statement $\mathbb{X}$:** inequality of encrypted messages

$C := C^* - \mathsf{Enc}(\mathsf{pk}, M; 0)$ does not encrypt 0

- **Puncturing:** encrypt M in C^*

Our Blind Signature

$$\mathbf{D} = (D_1, D_2, D_3 = d_1 D_2)$$

$\mathsf{pk} = (C^*, \mathbf{D}), \mathsf{sk} = d_1$

pk, m

signer

user

$C = \mathsf{Enc}(\mathsf{pk}_{\mathsf{rom}}, \mathsf{M}; \mathsf{r})$

$\pi_m = \mathsf{Prove}(''C \text{ is well formed}'')$

$C \ \pi_m$

$\mathbb{X} = C^* - C$

$R = \Sigma.\mathsf{Init}(\mathbb{X}, \mathbb{W})$

R

$C^* - C$ does not encrypt 0
or
$\mathbf{D}$ is DDH tuple

compile proof $\pi = (R, c, z)$ via Fiat-Shamir

c

$C^* - \mathsf{Enc}(\mathsf{pk}, M; 0)$ does not encrypt 0
or
$\mathbf{D}$ is DDH tuple

$z = \Sigma.\mathsf{Resp(c)}$

z

Conclusion

Blind Signatures

- ***Bridge gap in performance between AGM and AGM-free schemes***

Scheme[1]	Signature Size[2]	Communication Size[2]	Security	Assumption
[CKMTZ23]	96 B	192 B	AGM + ROM	DL
[KRW24]	224 B	2.5 KB	ROM	DDH
Our Work	192 B	608 B	ROM	DDH

[1] representatives for compact AGM and AGM-free blind signatures

[2] assuming 256 bit groups