Pointcheval-Sanders Signature-Based Synchronized Aggregate Signature Scheme

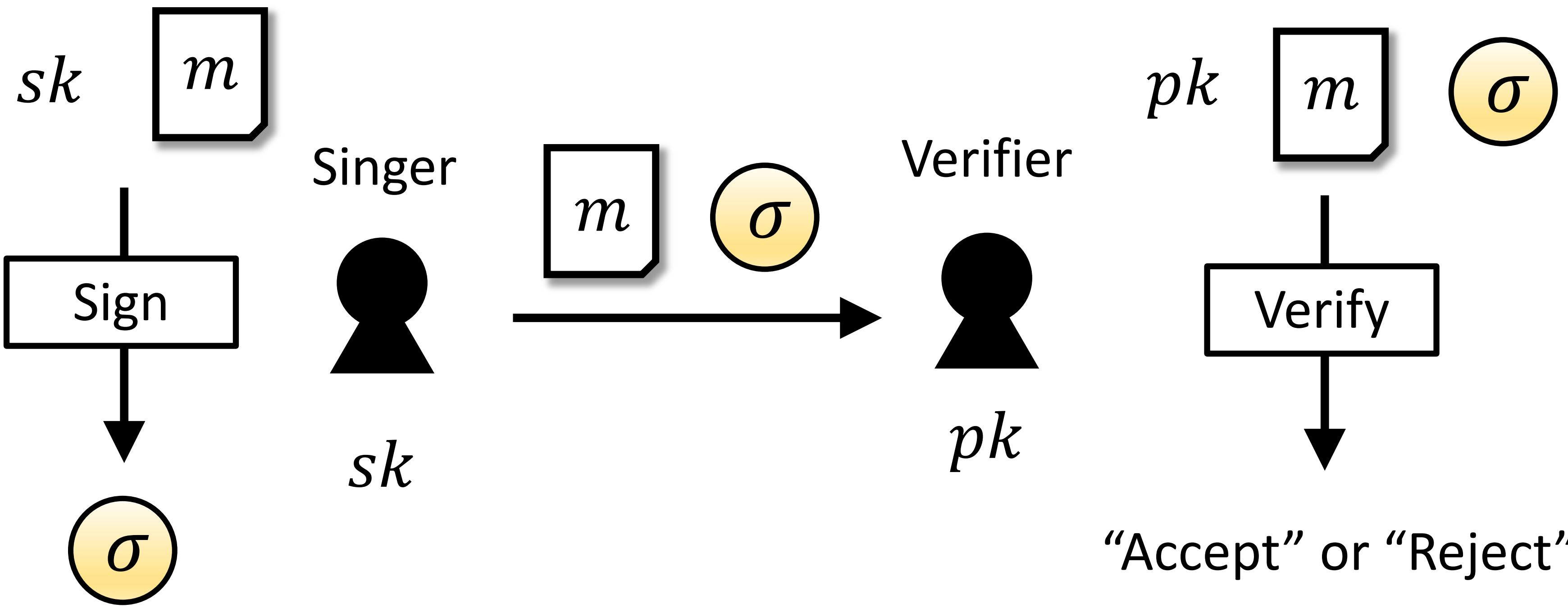
〇Masayuki Tezuka[1]　　Keisuke Tanaka[2]

1 National Institute of Technology, Tsuruoka Collage

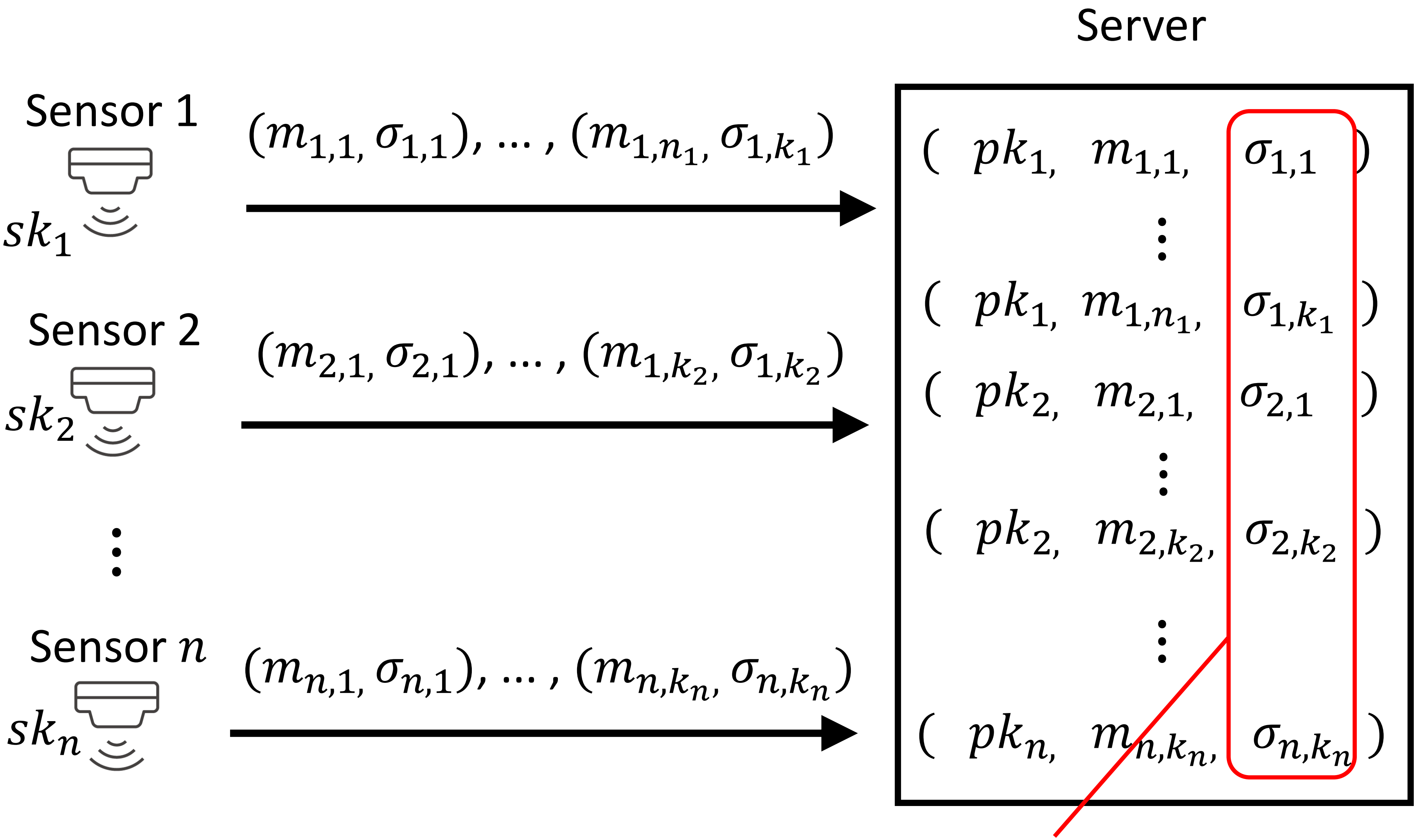
2 Tokyo Institute of Technology

Background

Digital Signature

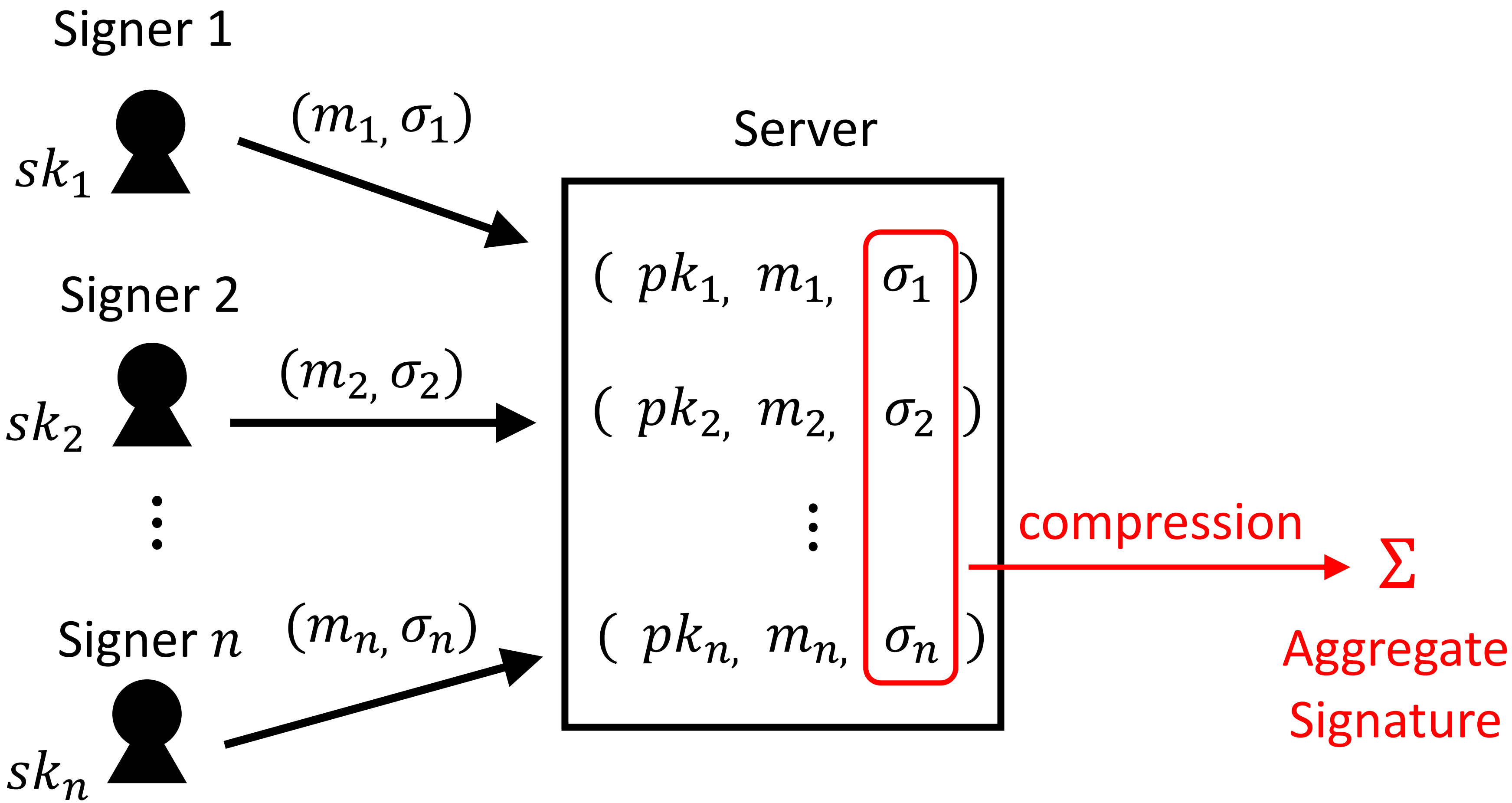


Digital Signature on IoT System



Require large signature storage space

Aggregate Signature [BGLS03]



Existing Aggregate Signature Scheme

Aggregate signature schemes without the random oracle model

- Multilinear map-based scheme [HSW13]
- Indistinguishable obfuscation (iO) based scheme [HKW15]

The aggregate signature scheme in the random oracle model

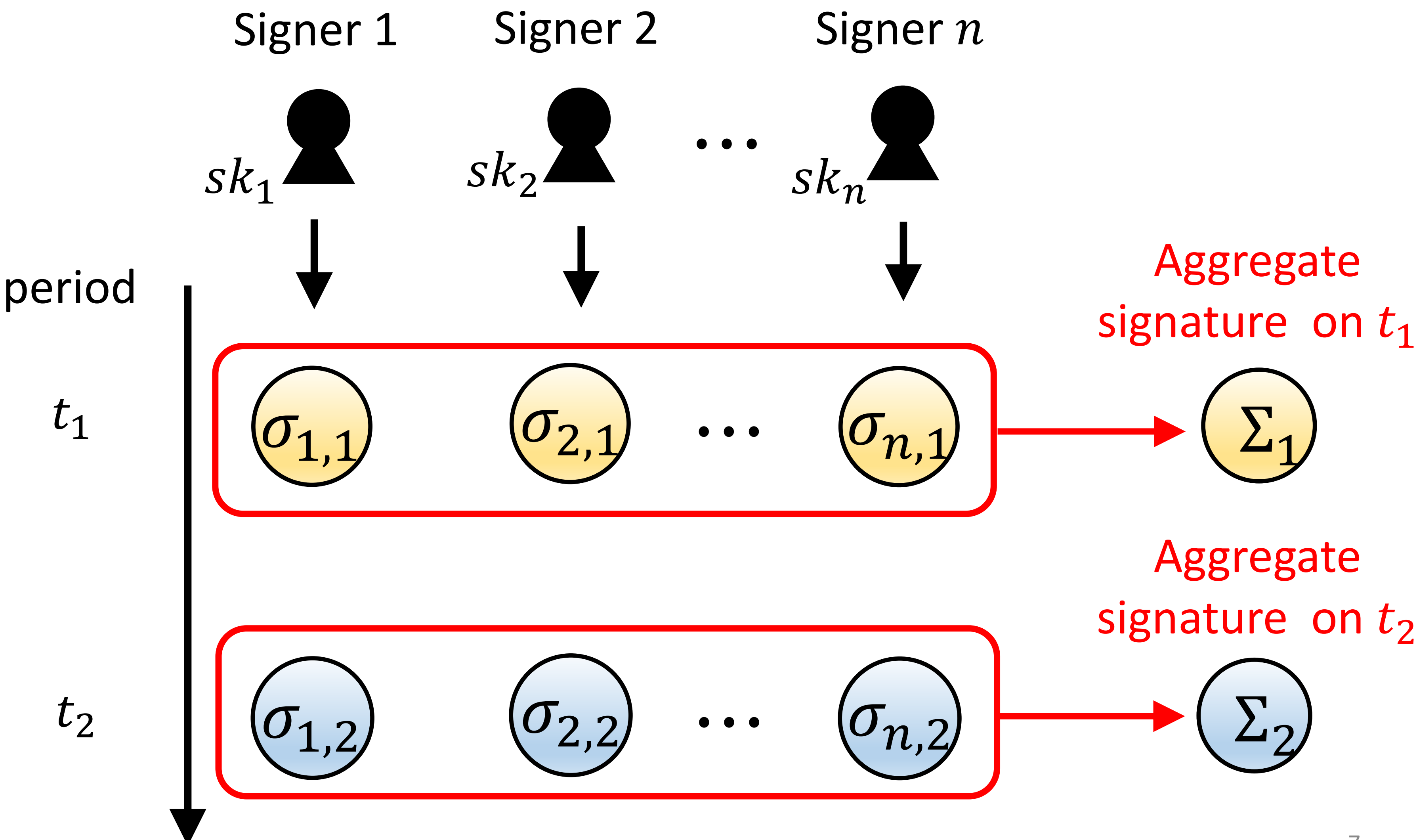
- Pairing based scheme [BGLS03]

Constructing Synchronized AS Scheme is very difficult task !

The pairing-based scheme [BGLS03] needs $(n+1)$ pairing operations to verify an aggregate signature.

(n is the num of signatures which are aggregated)

Synchronized Aggregate Signature [AGH10]



Application of Synchronized AS

A synchronized aggregate scheme can be used systems which has a natural reporting period.

Application

・Sensor data system

・Log data system

・Blockchain protocol

Synchronized AS Scheme in the ROM

Comparison with synchronized aggregate signature schemes in the random oracle model.

Scheme	Assumption	Pk size (elements)	Agg Sig size (elements)	Agg Ver (pairing op)	Pairing Type
[BGLS 03]	co-CDH ROM	1	2	$n+1$	Type-2
[AGH 10]	CDH ROM	1	2	4	Type-3
[LLY 13]	1-MSDH-2 ROM	1	2	3	Type-1

Fewer pairing operations are desirable.

Type-3 pairing based schemes are desirable.

Our Contribution

Comparison with synchronized aggregate signature schemes in the random oracle model.

Scheme	Assumption	Pk size (elements)	Agg Sig size (elements)	Agg Ver (pairing op)	Pairing Type
[BGLS 03]	co-CDH ROM	1	2	$n+1$	Type-2
[AGH 10]	CDH ROM	1	2	4	Type-3
[LLY 13]	1-MSDH-2 ROM	1	2	3	Type-1
Our Scheme	GPS ROM	2	2	2	Type-3

We construct an efficient synchronized aggregate signature scheme based on the Pointcheval-Sanders signature scheme.

Synchronized Aggregate Signature Scheme and Its Security

Syntax of Synchronized AS Scheme

Setup$(1^\lambda, 1^T) \rightarrow pp$

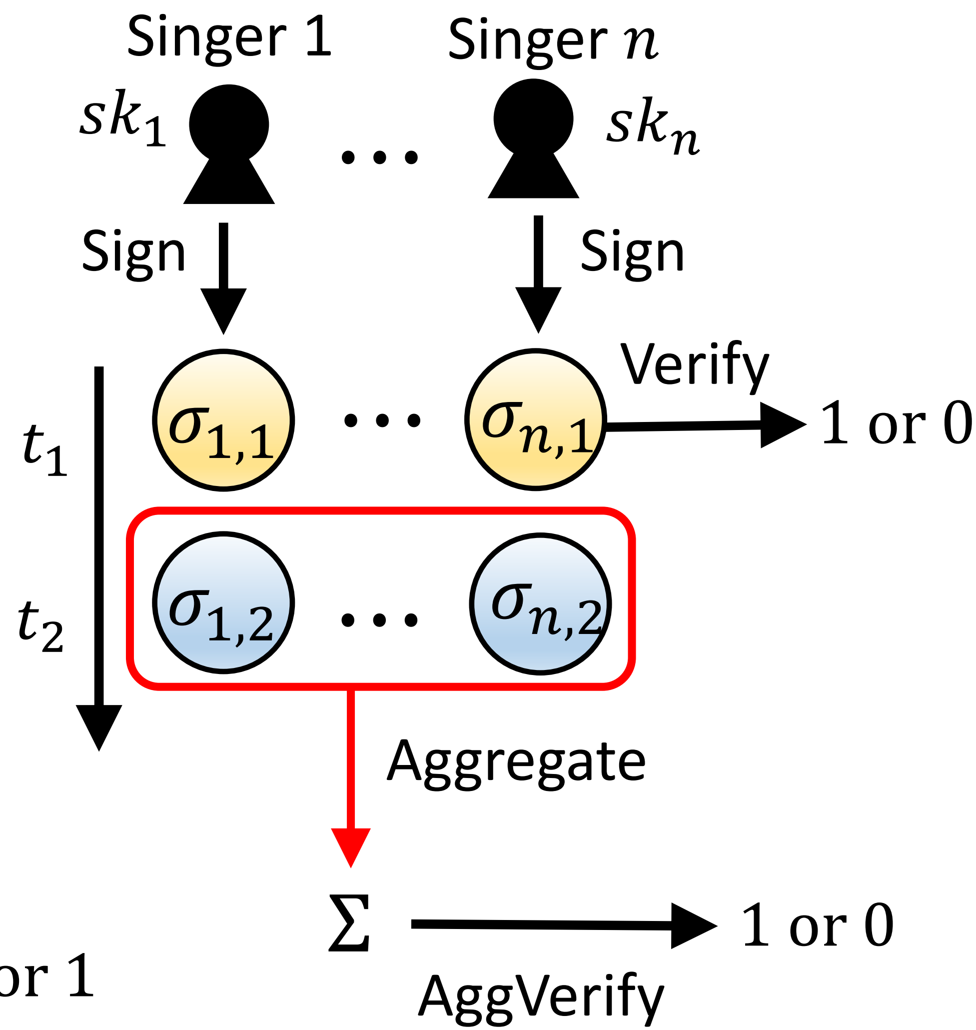
KeyGen$(pp) \rightarrow (pk, sk)$

Sign$(sk, t, m) \rightarrow \sigma$

Verify$(pk, m, \sigma) \rightarrow 0$ or 1

Aggregate$((pk_i, m_i, \sigma_i)_{i=1}^n) \rightarrow \Sigma$

AggVerify$((pk_i, m_i)_{i=1}^n, \Sigma) \rightarrow 0$ or 1



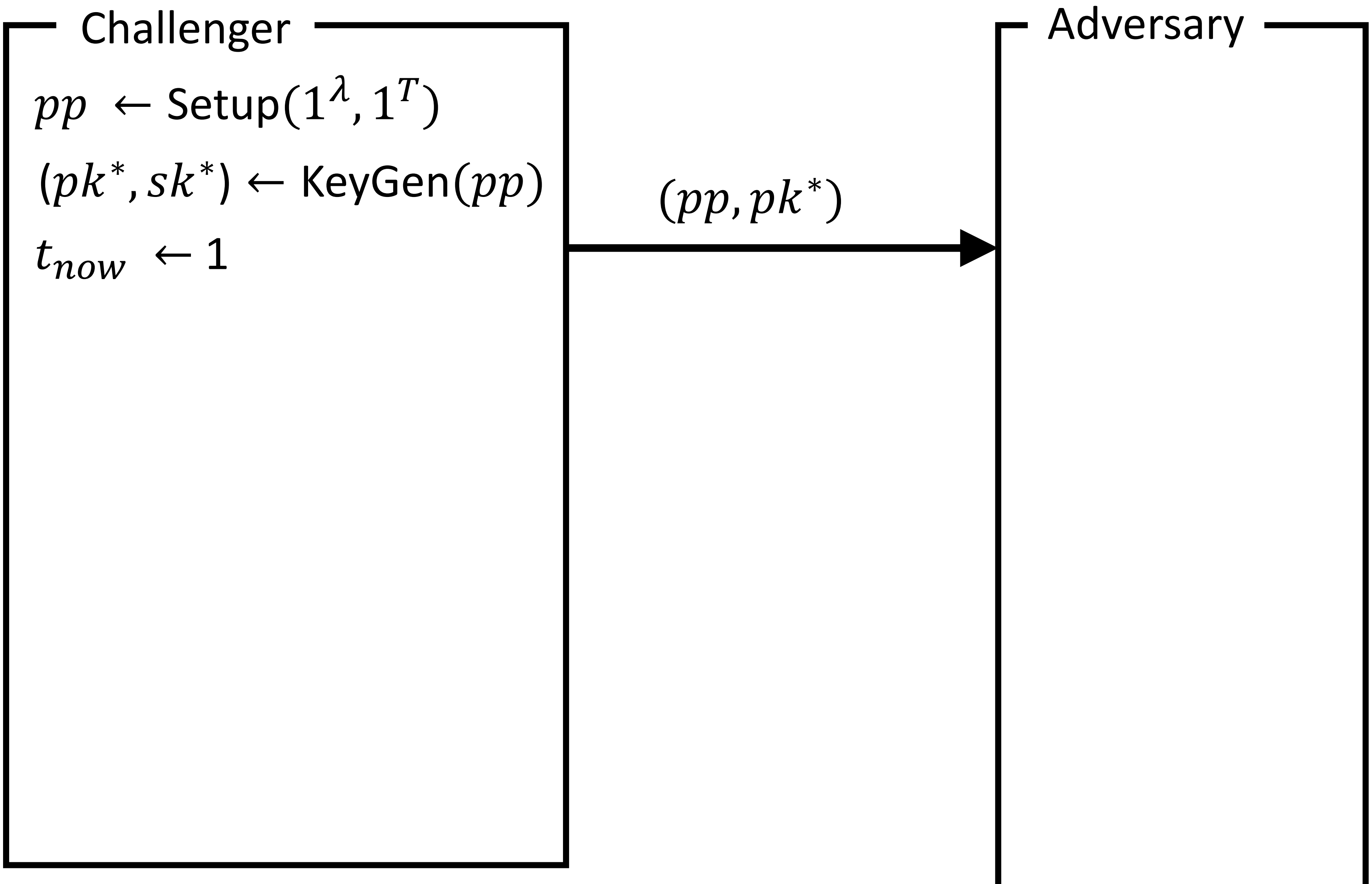
t is implicitly included in σ and Σ.

EUF-CMA Security in the Certified Key Model

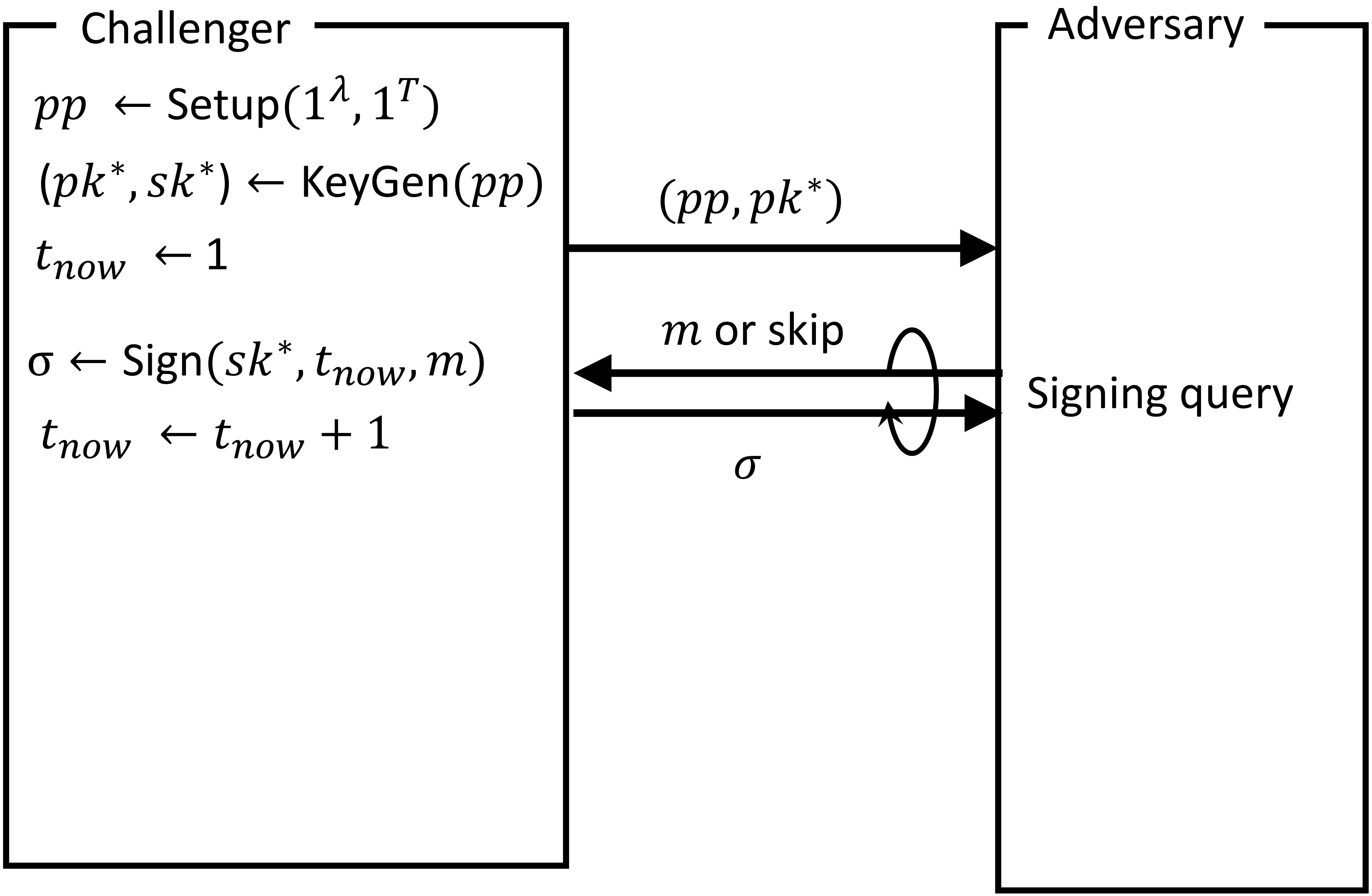
Challenger

Adversary

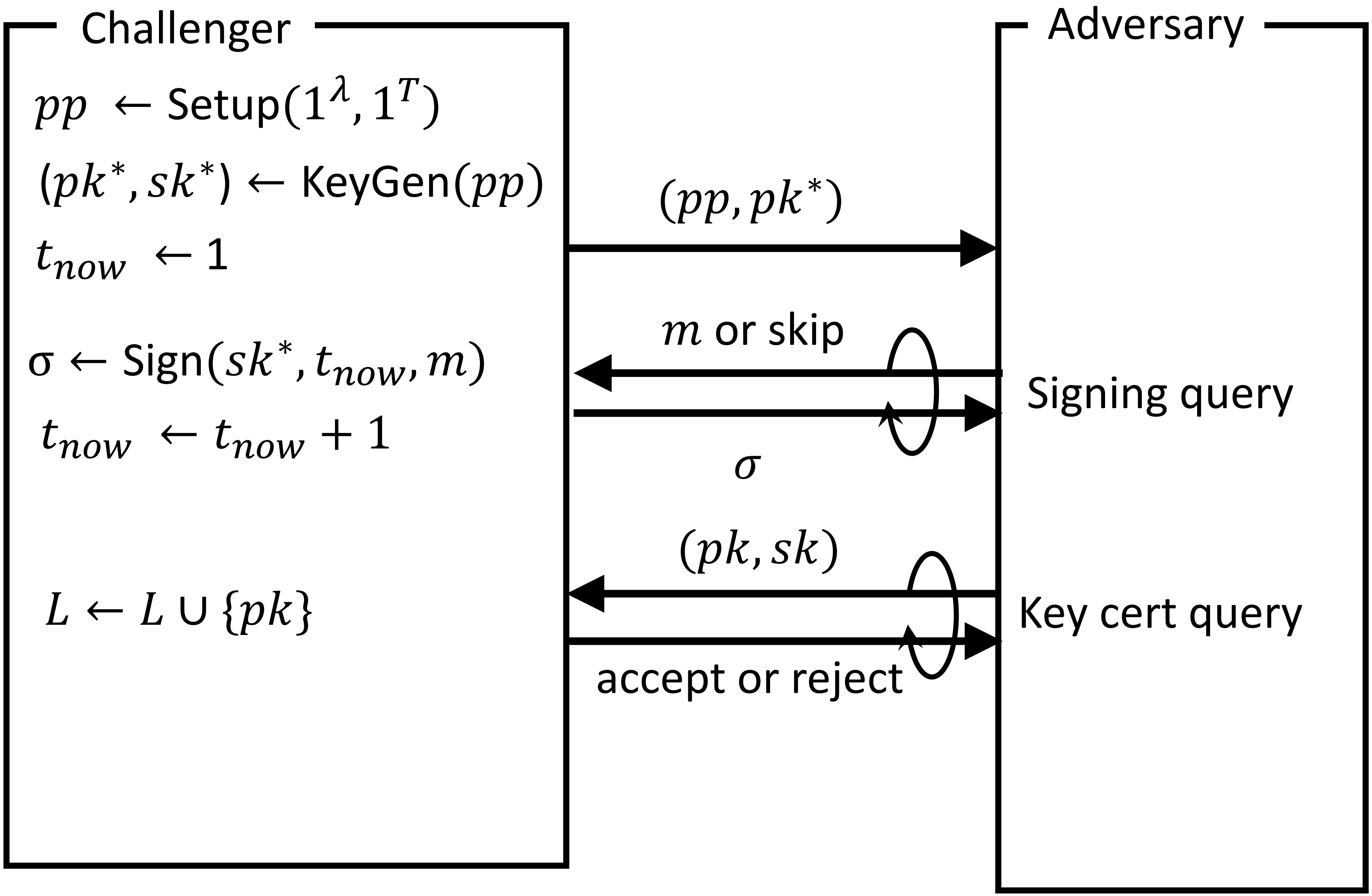
EUF-CMA Security in the Certified Key Model



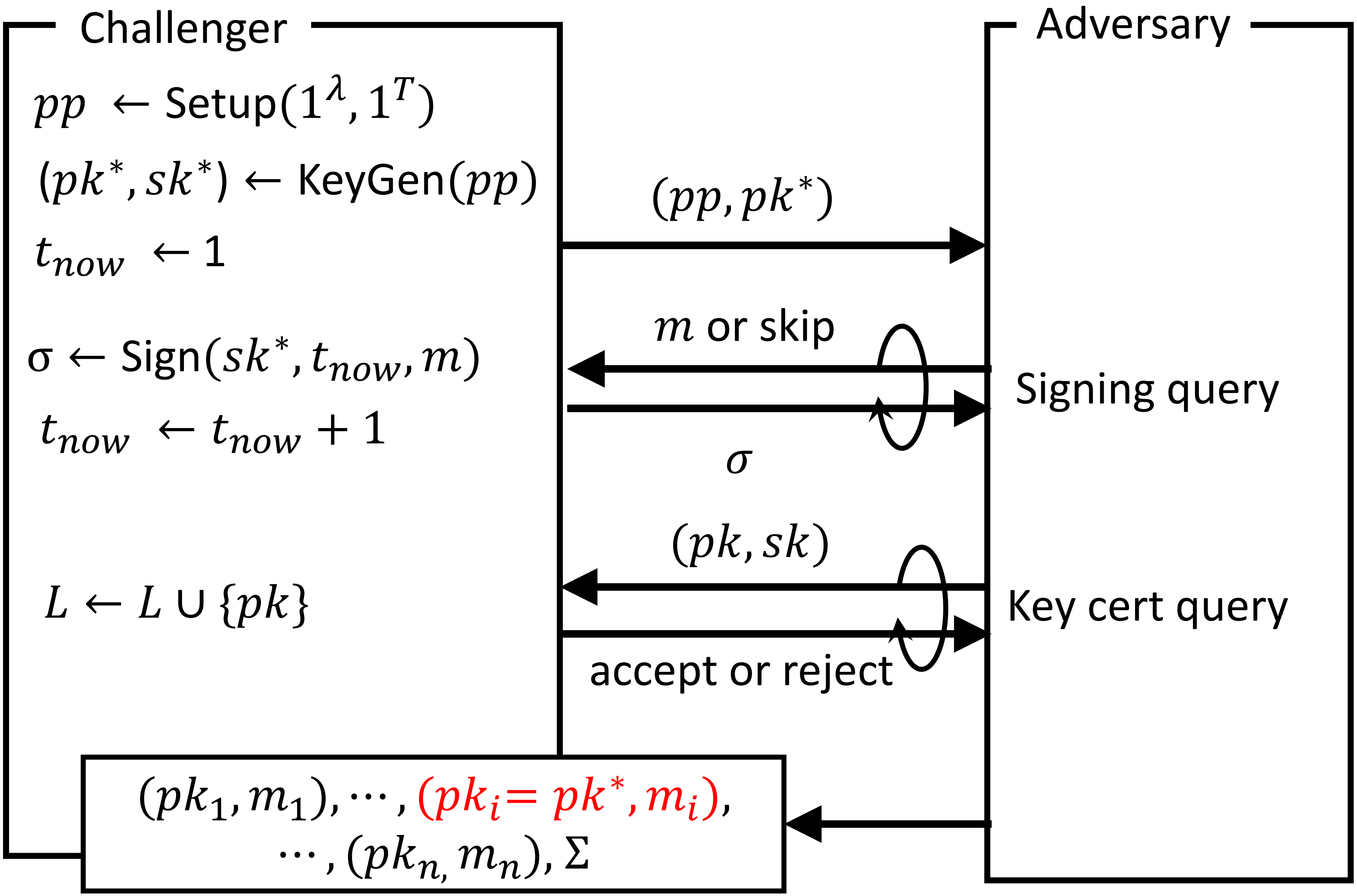
EUF-CMA Security in the Certified Key Model



EUF-CMA Security in the Certified Key Model



EUF-CMA Security in the Certified Key Model



EUF-CMA Security in the Certified Key Model

Final output of the adversary

$$\boxed{(pk_1, m_1), \cdots, (pk_i = pk^*, m_i), \cdots, (pk_n, m_n), \Sigma}$$

The adversary wins if:

1. AggVerify$((pk_i, m_i)_{i=1}^n, \Sigma)$ = 1 holds.
2. All public keys $(pk_1, \ldots, pk_n)$ are certified.
3. m_i is never queried to signing.

Pointcheval-Sanders Signature Scheme and Our Construction

Pointcheval-Sanders Signature Scheme [PS16]

pp:= $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\ e)$

KeyGen(pp)

$\tilde{G} \ \leftarrow_r \mathbb{G}_2^*, \quad x, y \leftarrow_r \mathbb{Z}_p^*, \tilde{X} \leftarrow \tilde{G}^x, \quad \tilde{Y} \leftarrow \tilde{G}^y$

Return $(pk, sk) \leftarrow ((\tilde{G}, \tilde{X}, \tilde{Y}),\ (x, y))$

Sign$(sk = (x, y), m)$

$A \leftarrow_r \mathbb{G}_1^*, \quad B \leftarrow A^{x+m\cdot y}$

Return σ $\leftarrow (A, B)$

Verify$(pk = (\tilde{G}, \tilde{X}, \tilde{Y}), m, \sigma = (A, B))$

If $A \neq 1_{\mathbb{G}_T} \ \wedge e\left(A, \tilde{X}\tilde{Y}^m\right) = e(B, \tilde{G})$, return 1

Otherwise return σ $\leftarrow (A, B)$

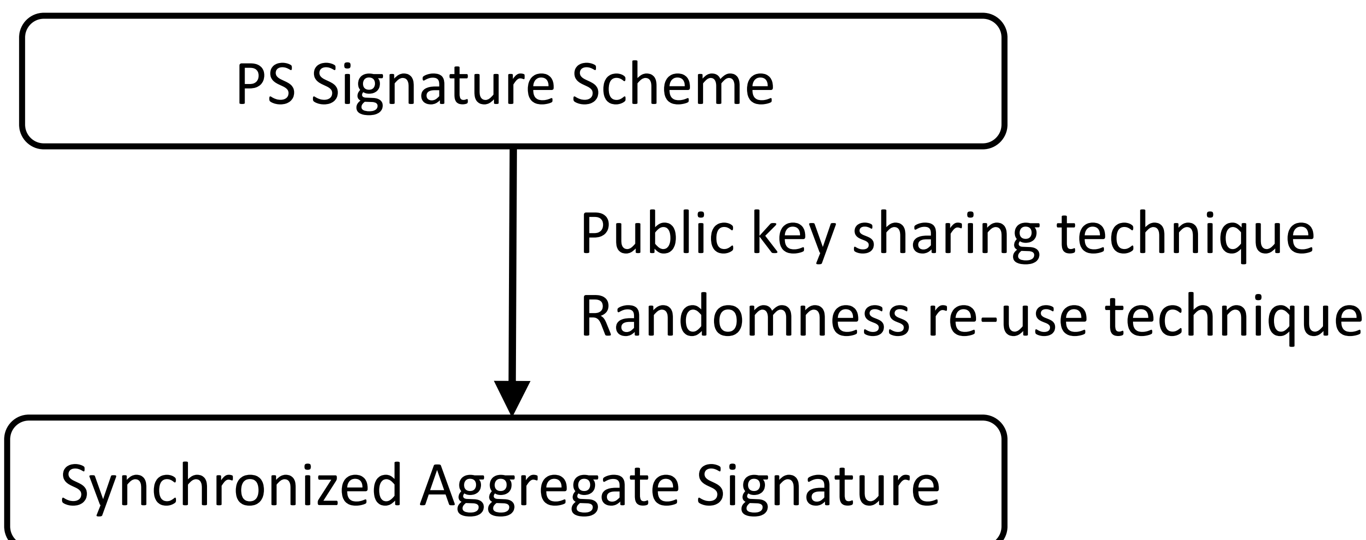
How to Derive Our Scheme

Public key sharing technique

One of element in public key of underlying scheme is replaced by public parameter.

Randomness re-use technique

Force the all signers to use the same randomness to sign a message.



Step 1 (PK Sharing Technique)

$pp := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\ e, \tilde{G})$

Public key sharing technique

KeyGen(pp)

$\cancel{\tilde{G} \leftarrow_r \mathbb{G}_2^*}$, $x, y \leftarrow_r \mathbb{Z}_p^*, \tilde{X} \leftarrow \tilde{G}^x, \quad \tilde{Y} \leftarrow \tilde{G}^y$

Return $(pk, sk) \leftarrow ((\cancel{\tilde{G}}, \tilde{X}, \tilde{Y}),\ (x, y))$

Sign$(sk = (x, y), m)$

$A \leftarrow_r \mathbb{G}_1^*, B \leftarrow A^{x+m \cdot y}$

Return $\sigma \leftarrow (A, B)$

Step 2 (Randomness Re-use Technique)

$pp := (p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\ e, \tilde{G})$

Public key sharing technique

KeyGen(pp)

$\cancel{\tilde{G} \leftarrow_r \mathbb{G}_2^*}$, $x, y \leftarrow_r \mathbb{Z}_p^*, \tilde{X} \leftarrow \tilde{G}^x$, $\tilde{Y} \leftarrow \tilde{G}^y$

Return $(pk, sk) \leftarrow ((\cancel{\tilde{G}}, \tilde{X}, \tilde{Y}),\ (x, y))$

Sign($sk = (x, y), t, m$)

Randomness re-use technique

$A \leftarrow H_1(t), B \leftarrow A^{x+m\cdot y}$

Return $\sigma \leftarrow (\cancel{A}, B, t)$

Aggregate of Our Scheme

pp:= $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T\ e, \tilde{G})$

Public key sharing technique

KeyGen(pp)

$\cancel{\tilde{G} \leftarrow_r \mathbb{G}_2^*}$, $\quad x, y \leftarrow_r \mathbb{Z}_p^*, \tilde{X} \leftarrow \tilde{G}^x, \quad \tilde{Y} \leftarrow \tilde{G}^y$

Return $(pk, sk) \leftarrow ((\cancel{\tilde{G}}, \tilde{X}, \tilde{Y}),\ (x, y))$

Sign$(sk = (x, y), t, m)$

Randomness re-use technique

$A \leftarrow H_1(t), B \leftarrow A^{x+H_2(m,t)\cdot y}$

Return $\sigma \leftarrow (A, B, t)$

To prove the security, change m to $H_2(m, t)$

AggVer of Our Scheme

Aggregate$((pk_i, m_i, \sigma_i = (B_i, t))_{i=1}^{n})$

$$\Sigma = \left(B = \prod_{i=1}^{n} B_i = \prod_{i=1}^{n} H_1(t)^{(x_i + H_2(m,t) \cdot y_i)}, t \right)$$

AggVer$\big((pk_i = (\tilde{X}_i, \tilde{Y}_i), m_i)_{i=1}^{n}, \Sigma = (B, t)\big)$

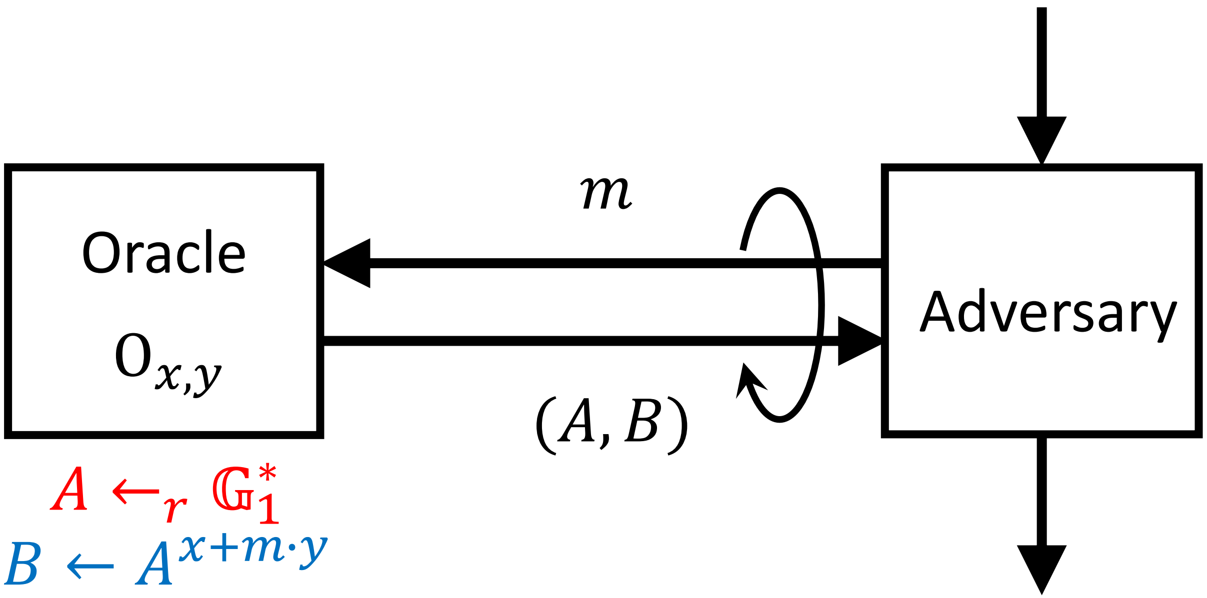
Check $$e\left(H_1(t), \prod_{i=1}^{n} \tilde{X}_i {\tilde{Y}_i}^{H_2(m,t)} \right) = e\big(B, \tilde{G}\big)$$

Only two pairing operations

Security Proof of Our Scheme

PS Assumption [PS16]

Instance $(p, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, \tilde{G}, \tilde{X} = \tilde{G}^x, \tilde{Y} = \tilde{G}^y)$



$A \leftarrow_r \mathbb{G}_1^*$

$B \leftarrow A^{x+m\cdot y}$

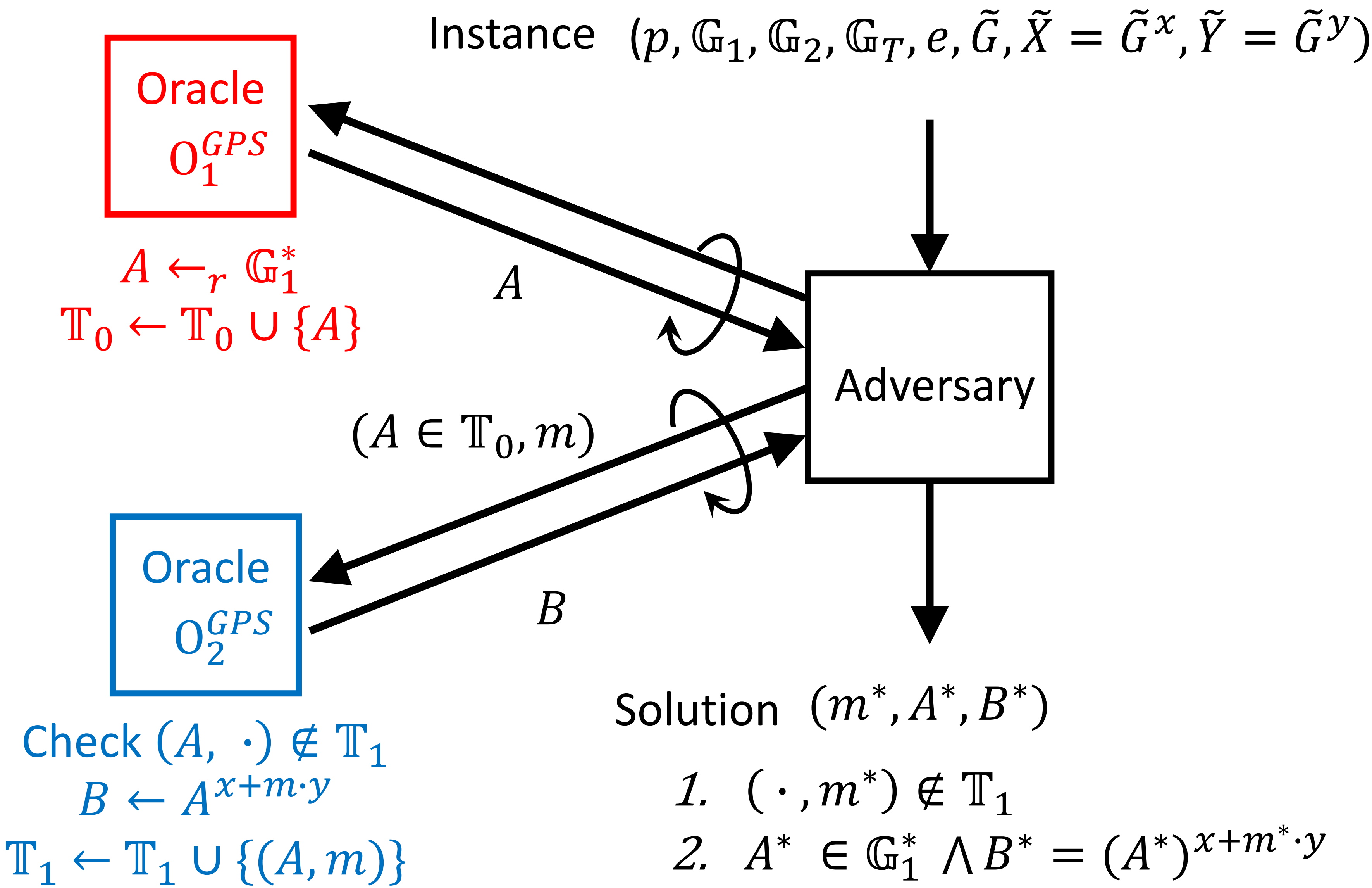
$\mathbb{T} \leftarrow \mathbb{T} \cup \{m\}$

Solution (m^*, A^*, B^*)

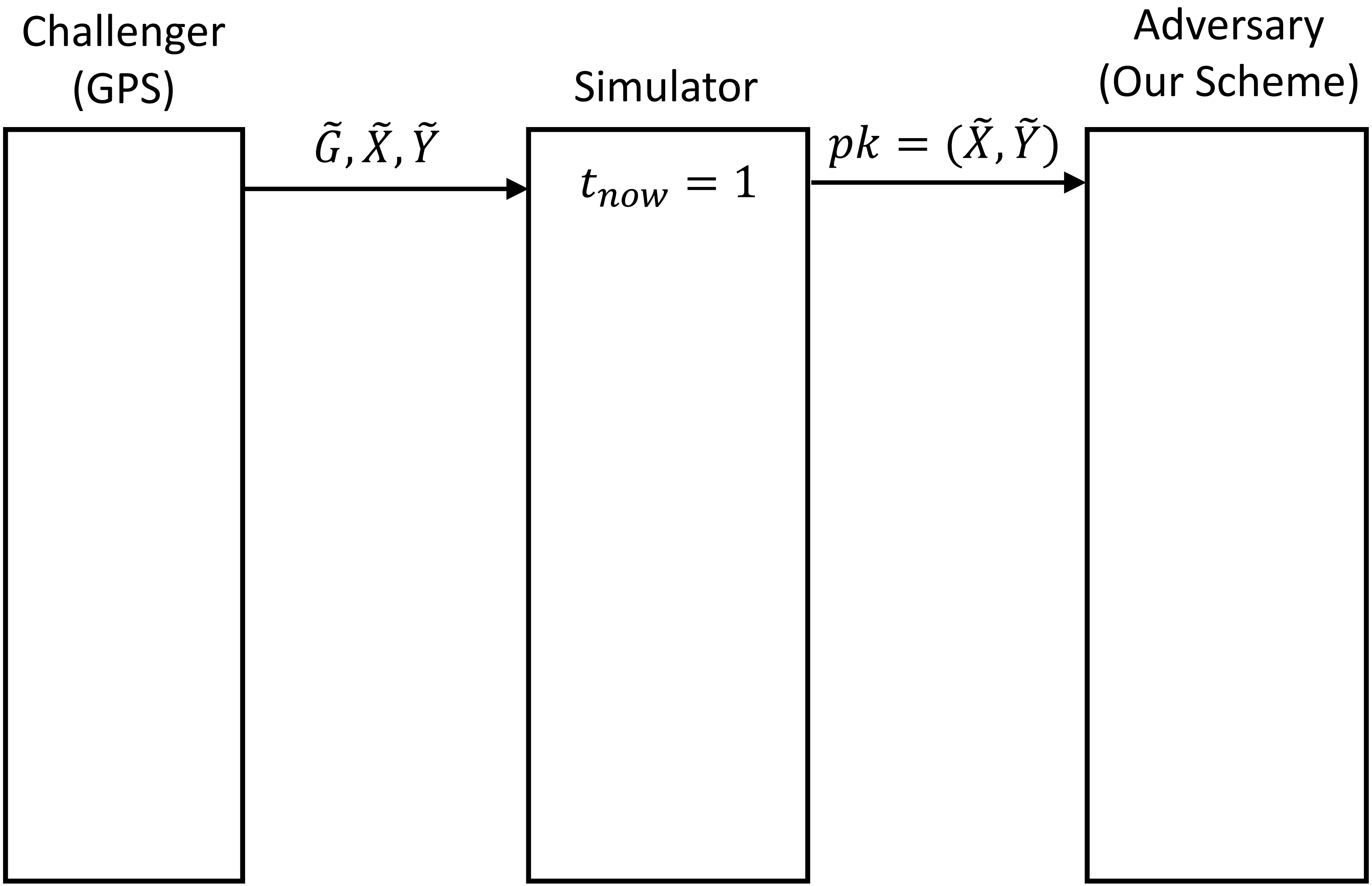
1. $(\cdot, m^*) \notin \mathbb{T}_1$
2. $A^* \in \mathbb{G}_1^* \wedge B^* = (A^*)^{x+m^*\cdot y}$

The PS assumption itself is the EUF-CMA security of the PS Signature Scheme.

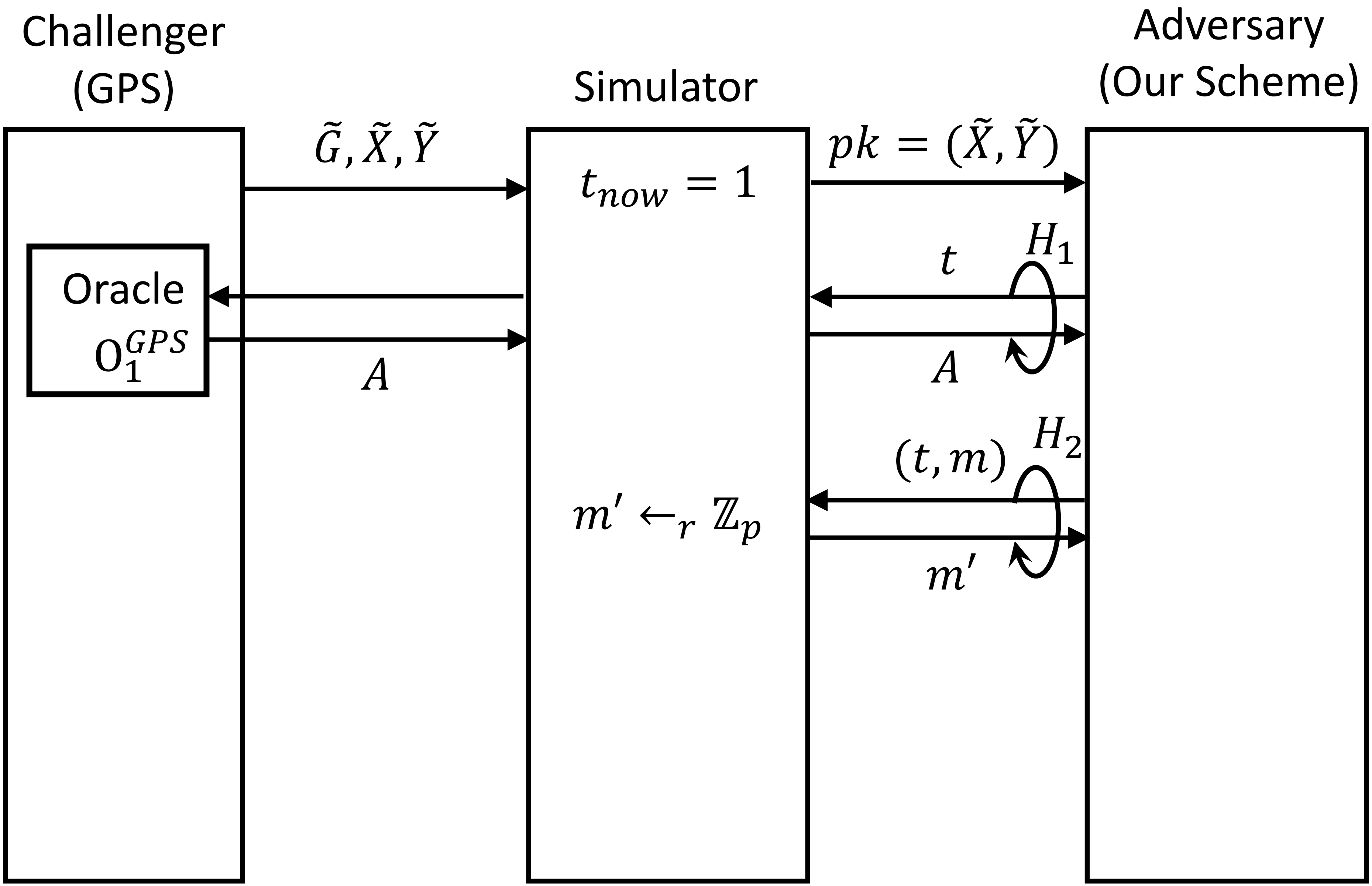
Generalized PS Assumption [KLAP21]



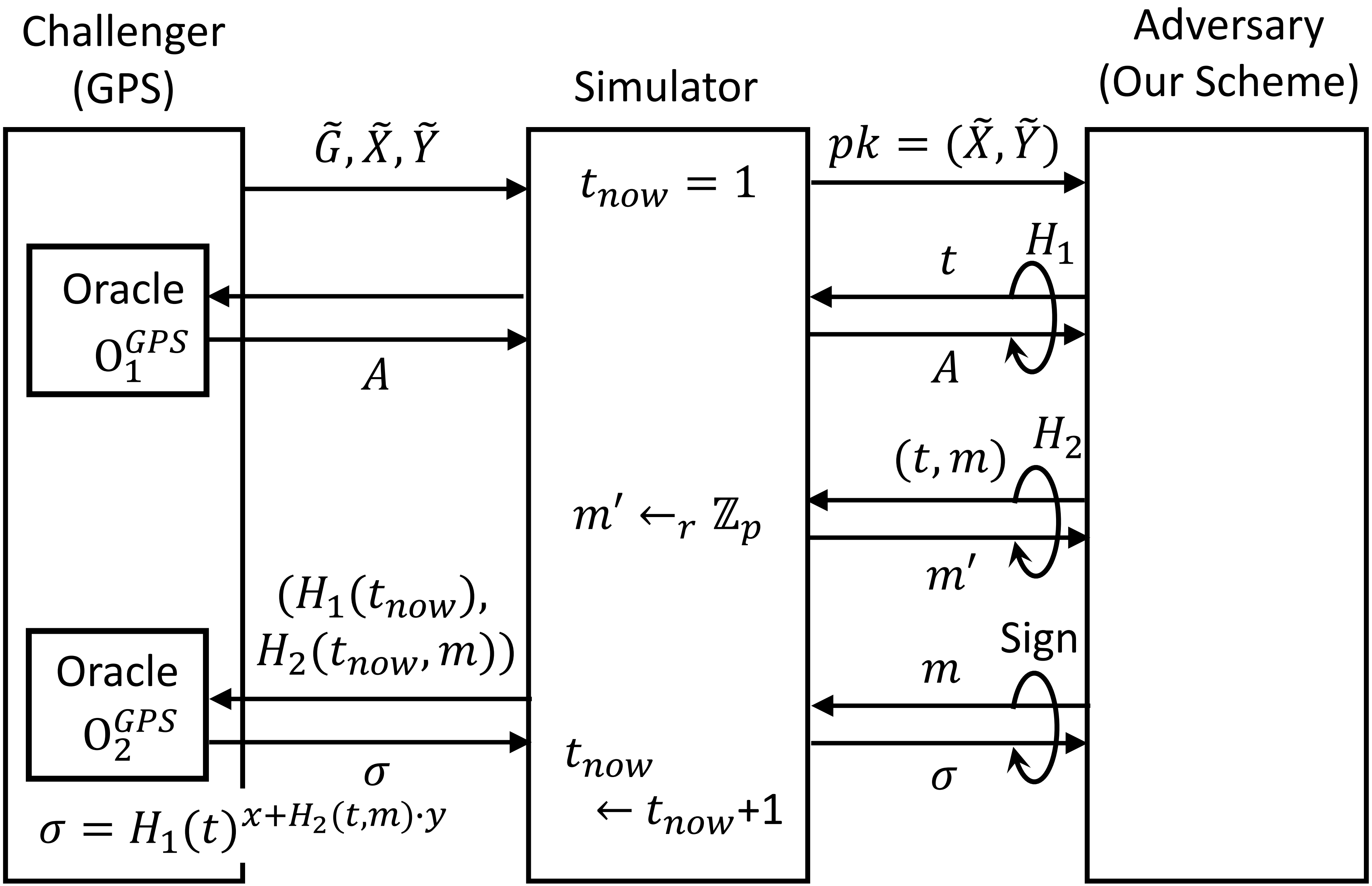
Simulation of EUF-CMA Security Game in ROM



Simulation of EUF-CMA Security Game in ROM



Simulation of EUF-CMA Security Game in ROM



Conclusion

We propose the Pointcheval-Sansers signature based synchronize aggregate signature scheme.

Our scheme is based on type-3 pairing and only needs 2 pairing operations to verify an aggregate signature.

The security of our scheme is proven under
the generalized Pointcheval-Sanders assumption in the ROM.

References

[AGH10] Ahn, Green, and Hohenberger. Synchronized aggregate signatures: new definitions, constructions and applications. (ACM CCS 2010)

[BGLS10] Ahn, Green, and Hohenberger. Synchronized aggregate signatures: new definitions, constructions and applications. (ACM CCS 2010)

[HKW15] Hohenberger, Koppula, Waters. Universal signature aggregators. (EUROCRYPT 2015)

[HSW13] Hohenberger, Sahai, and Waters. Full domain hash from (leveled) multilinear maps and identity-based aggregate signatures . (CRYPTO 2013)

[KLAP21] Kim, Lee, Abdalla, and Park. Practical dynamic group signature with efficient concurrent joins and batch verifications. (J. Inf. Secur. Appl. 63)

[LLY13] Lee, Lee, and Yung. Aggregating CL-signatures revisited: Extended functionality and better efficiency. (FC 2013)

[PS16] Pointcheval and Sanders. Short Randomizable Signatures. (CT-RSA 2016)

Thank you!