

1-out-of-n Oblivious Signatures: Security Revisited and a Generic Construction with an Efficient Communication Cost

○Masayuki Tezuka

Keisuke Tanaka

Tokyo Institute of Technology

Version 2023/12/02

ICISC 2023 Full Presentation Slide

$(1,n)$ -Oblivious Signatures Scheme

(1,n)-Oblivious Signatures [Chen94]

Signer

sk




User

vk



(1,n)-Oblivious Signatures [Chen94]

Make a list of n candidate messages    .

Signer
 sk 

 User
 vk

(1,n)-Oblivious Signatures [Chen94]

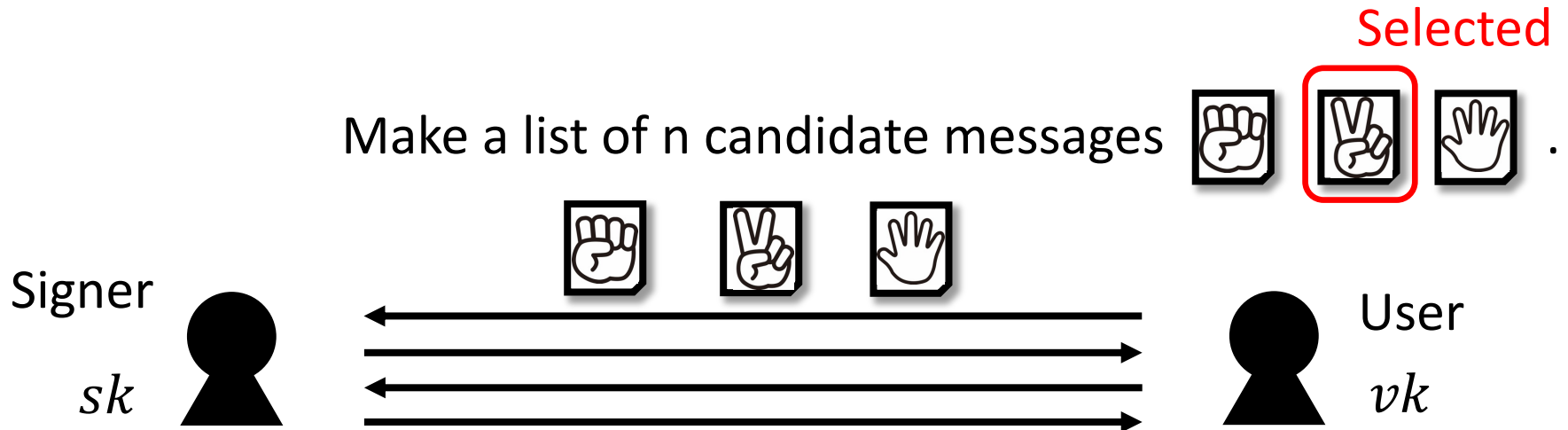
Make a list of n candidate messages    .

Selected

Signer
 sk 

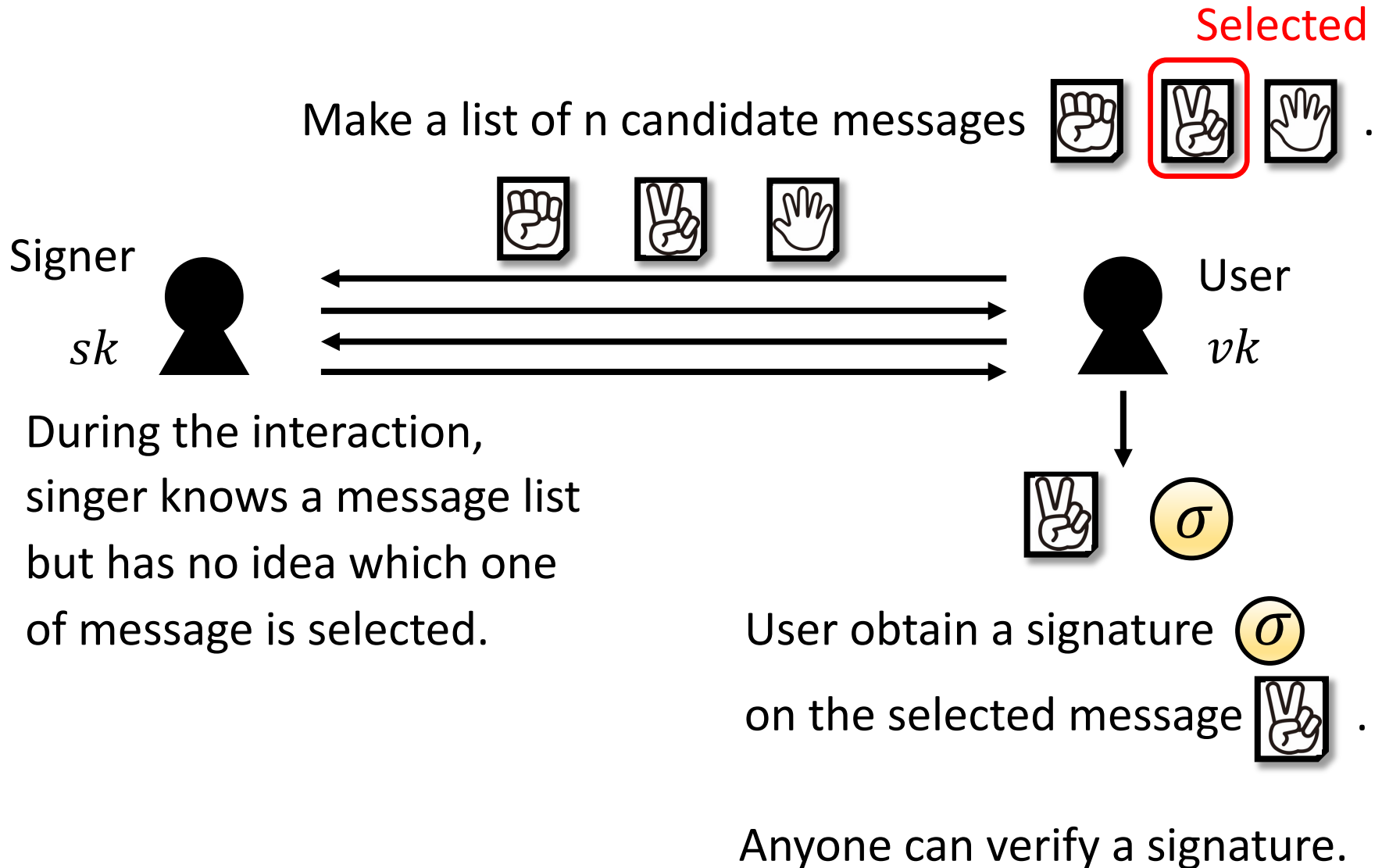
User
 vk 

(1,n)-Oblivious Signatures [Chen94]

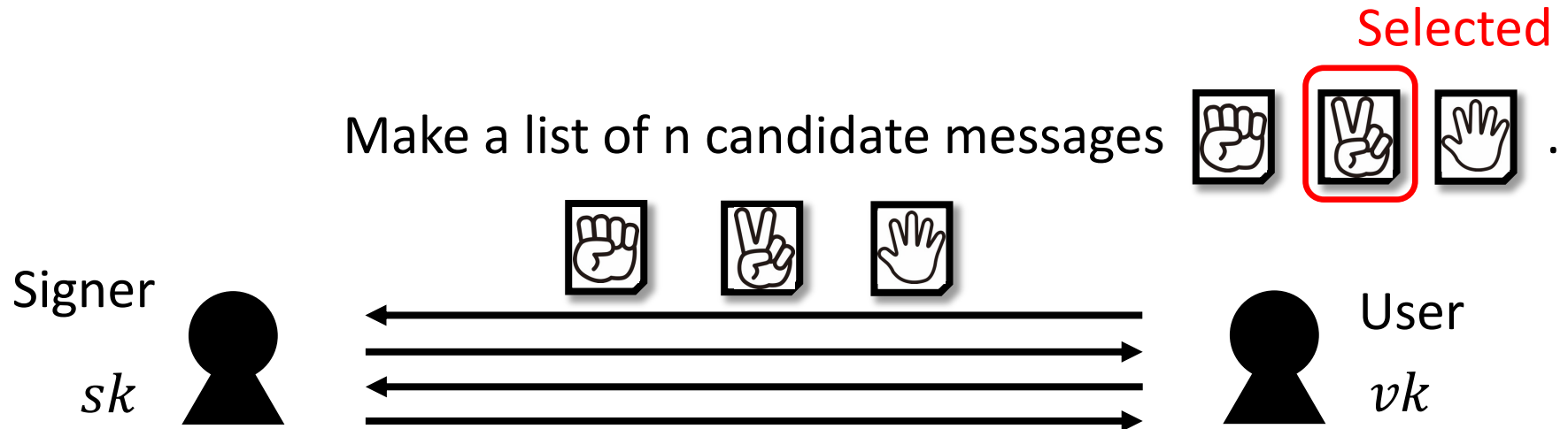


During the interaction,
signer knows a message list
but has no idea which one
of message is selected.

(1,n)-Oblivious Signatures [Chen94]



(1,n)-Oblivious Signatures [Chen94]



During the interaction, signer knows a message list but has no idea which one of message is selected.

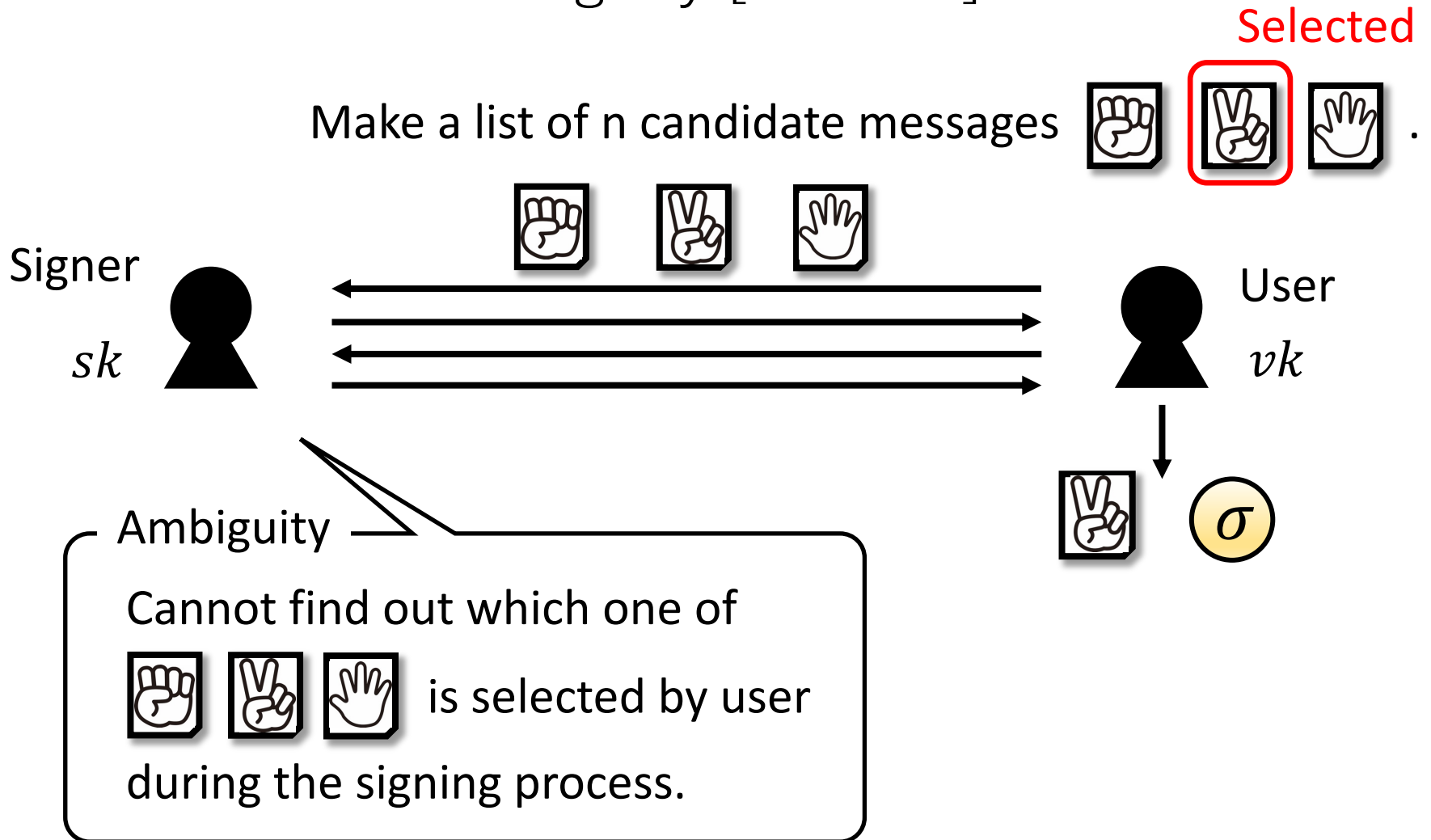
Security Requirements:

- Ambiguity
- Unforgeability

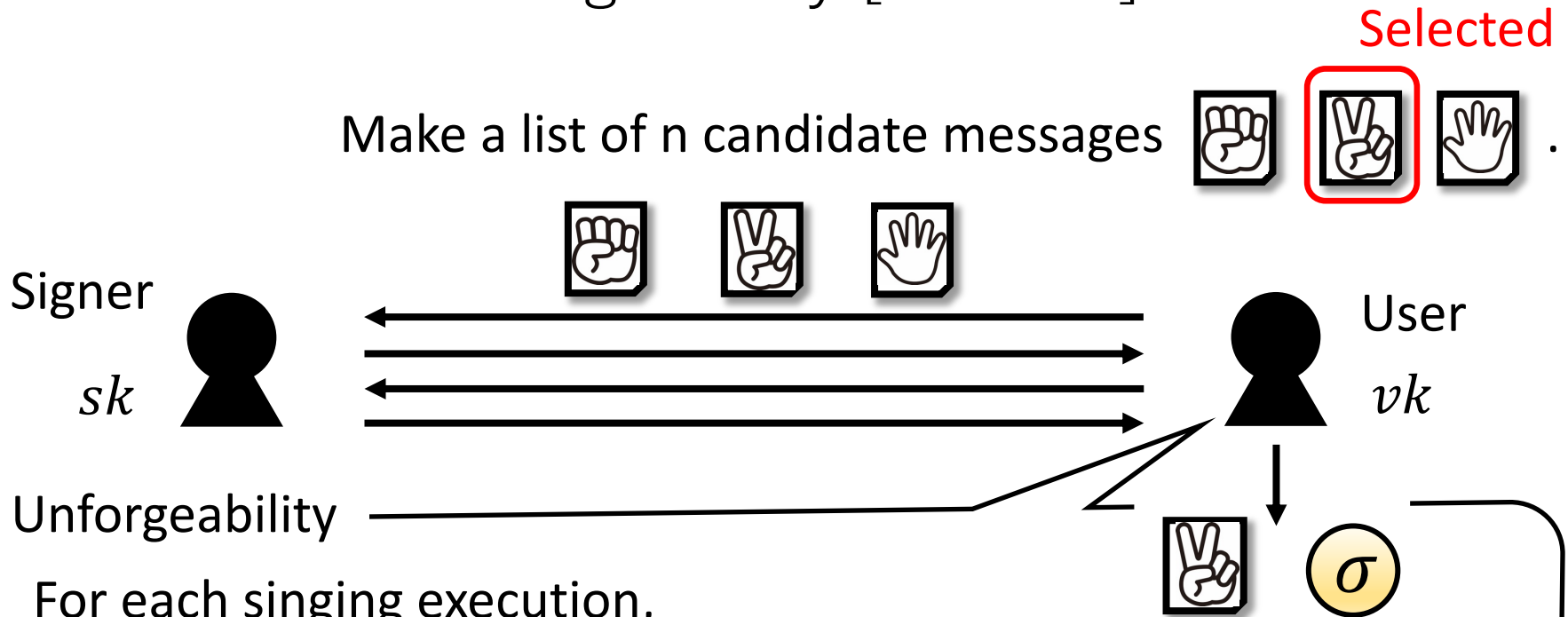
User obtain a signature σ on the selected message.

Anyone can verify a signature.




Ambiguity [Chen94]



Unforgeability [Chen94]



For each signing execution,

- Can derive a signature for only 1 of n messages in the list.
- Cannot obtain signatures on two or more messages in the list.
(e.g.  )
- Cannot obtain a signature on a message which is not in the list.
(e.g. )

Previous works for $(1,n)$ -Oblivious Signatures

Chen [Chen94]

- Notion of $(1, n)$ -oblivious signatures
- The first oblivious scheme

Tso et al. [TOO08]

- Formal definition and security model
- 2-move signing scheme
based on DL assumption in ROM

Zhou et al. [ZLH22]


- Generic construction of 2-move signing scheme
from commitment and a digital signature without ROM

Our Contributions

Chen [Chen94]

- Notion of $(1, n)$ -oblivious signatures
- The first oblivious scheme

1. Revisit the unforgeability security model



Tso et al. [TOO08]

- Formal definition and security model
- 2-move signing scheme
based on DL assumption in ROM

2. Second communication size improvement.



Zhou et al. [ZLH22]

- Generic construction of 2-move signing scheme
from commitment and a digital signature without ROM

Syntax and Unforgeability Security model in the Previous Work

Syntax of $(1,n)$ -Oblivious Signature Scheme

2-move $(1,n)$ -OS (KGen, U_1 , S_2 , Derive, Verify)

Syntax of $(1,n)$ -Oblivious Signature Scheme

2-move $(1,n)$ -OS (KGen , U_1 , S_2 , Derive , Verify)

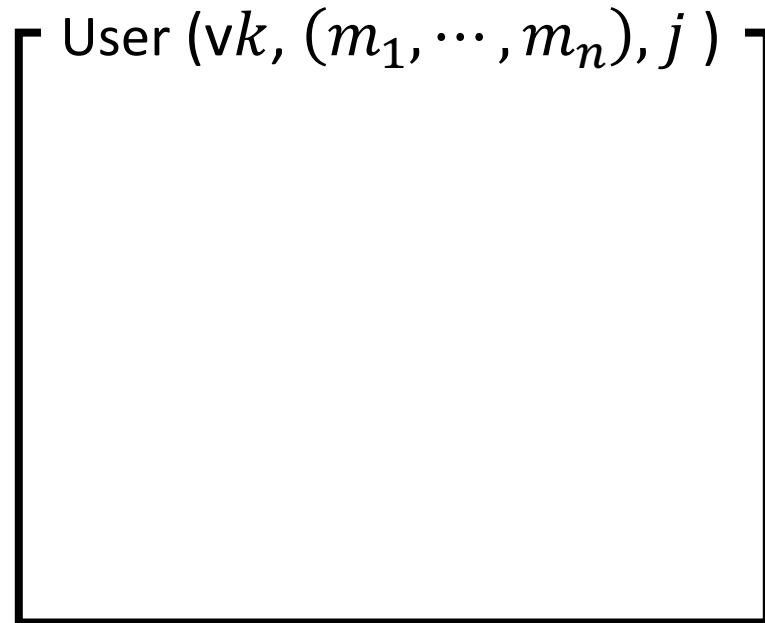
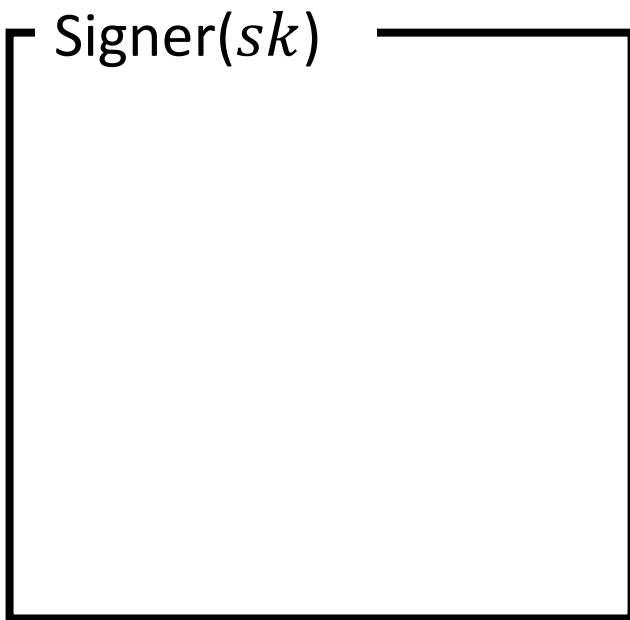
$\text{KGen}(1^\lambda) \rightarrow (vk, sk)$

Syntax of (1,n)-Oblivious Signature Scheme

2-move (1,n)-OS (KGen, U_1 , S_2 , Derive, Verify)

$\text{KGen}(1^\lambda) \rightarrow (vk, sk)$

Signing protocol (U_1 , S_2 , Derive)

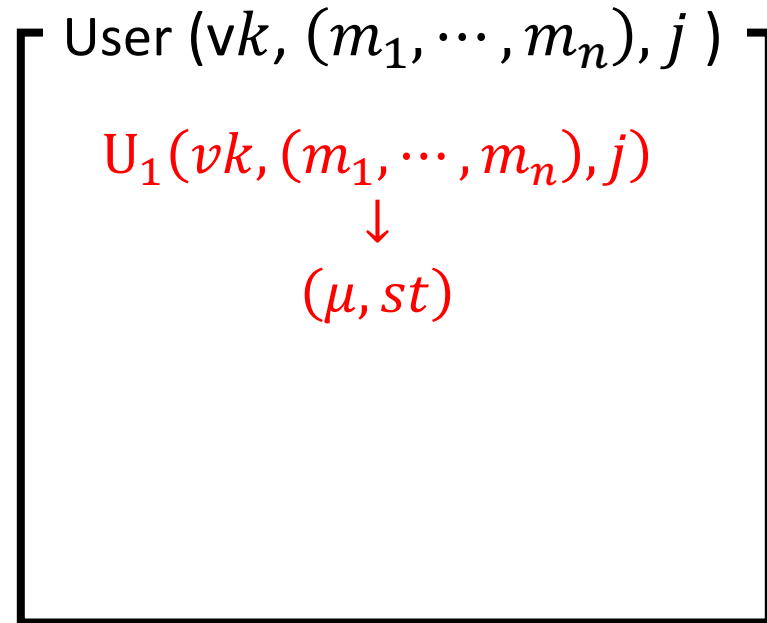
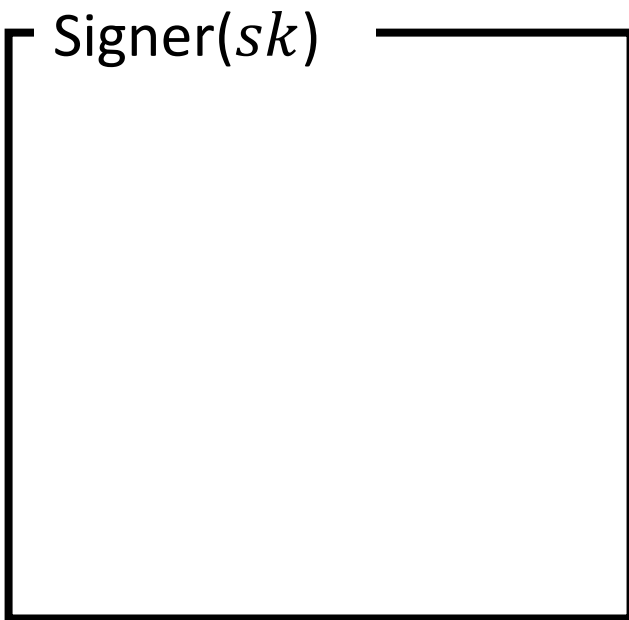


Syntax of (1,n)-Oblivious Signature Scheme

2-move (1,n)-OS $(\text{KGen}, U_1, S_2, \text{Derive}, \text{Verify})$

$\text{KGen}(1^\lambda) \rightarrow (vk, sk)$

Signing protocol $(U_1, S_2, \text{Derive})$

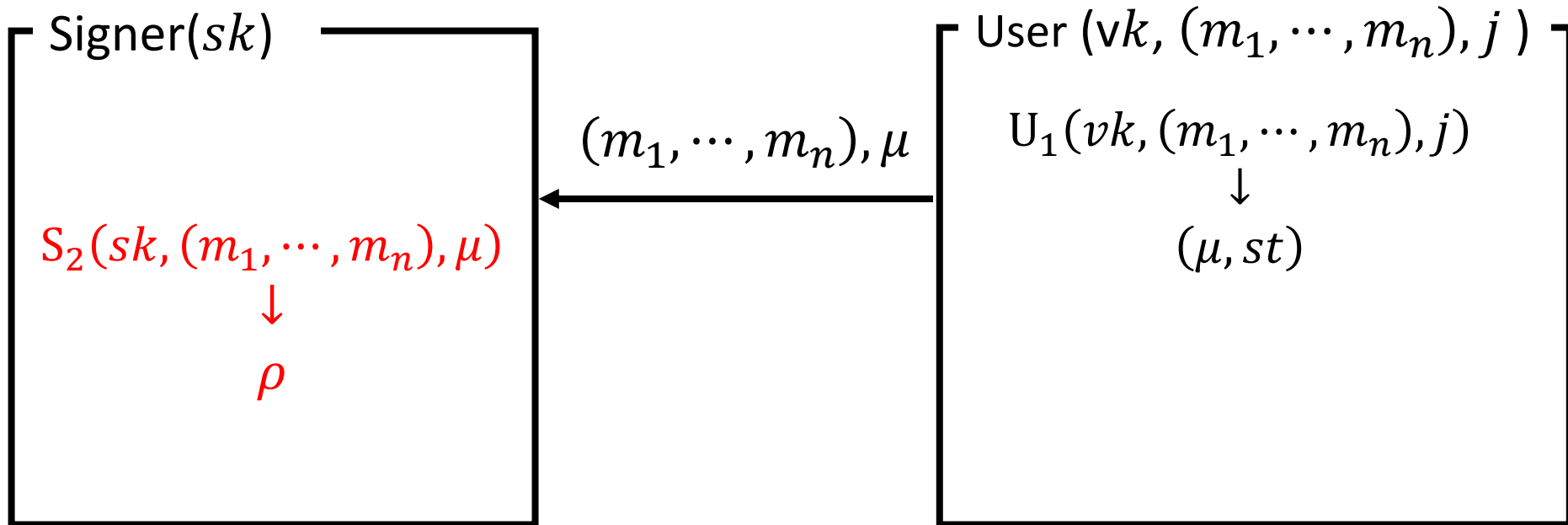


Syntax of (1,n)-Oblivious Signature Scheme

2-move (1,n)-OS $(\text{KGen}, U_1, S_2, \text{Derive}, \text{Verify})$

$\text{KGen}(1^\lambda) \rightarrow (vk, sk)$

Signing protocol $(U_1, S_2, \text{Derive})$

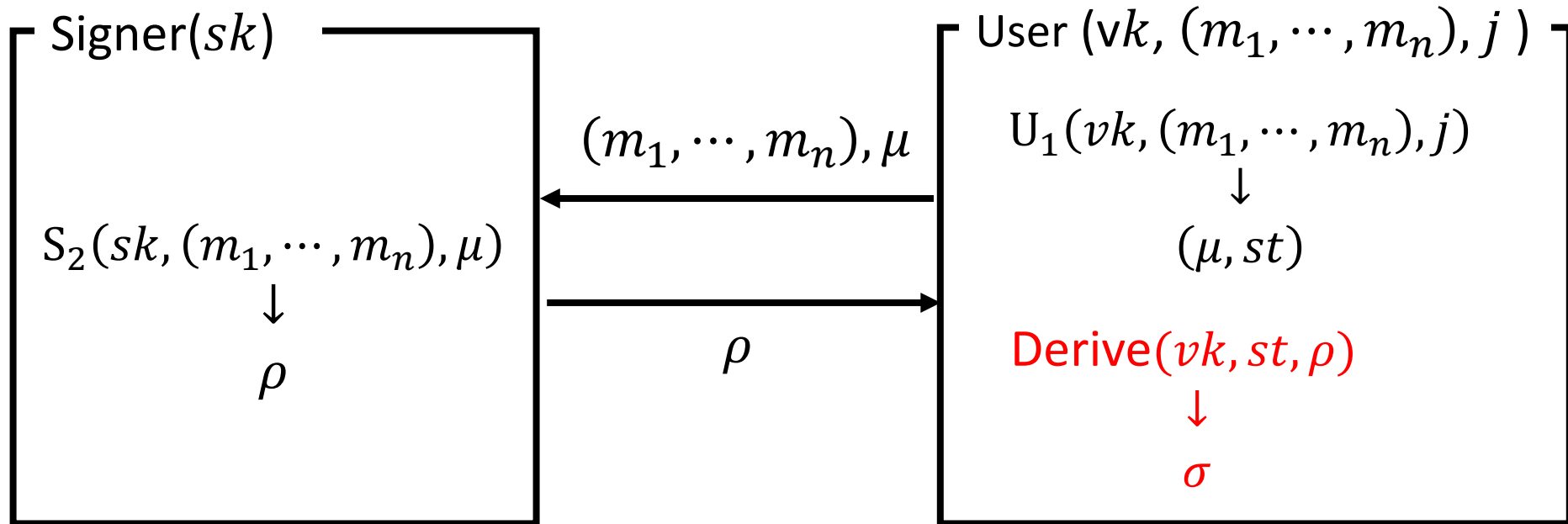


Syntax of (1,n)-Oblivious Signature Scheme

2-move (1,n)-OS (KGen, U_1 , S_2 , Derive, Verify)

$\text{KGen}(1^\lambda) \rightarrow (vk, sk)$

Signing protocol (U_1 , S_2 , Derive)

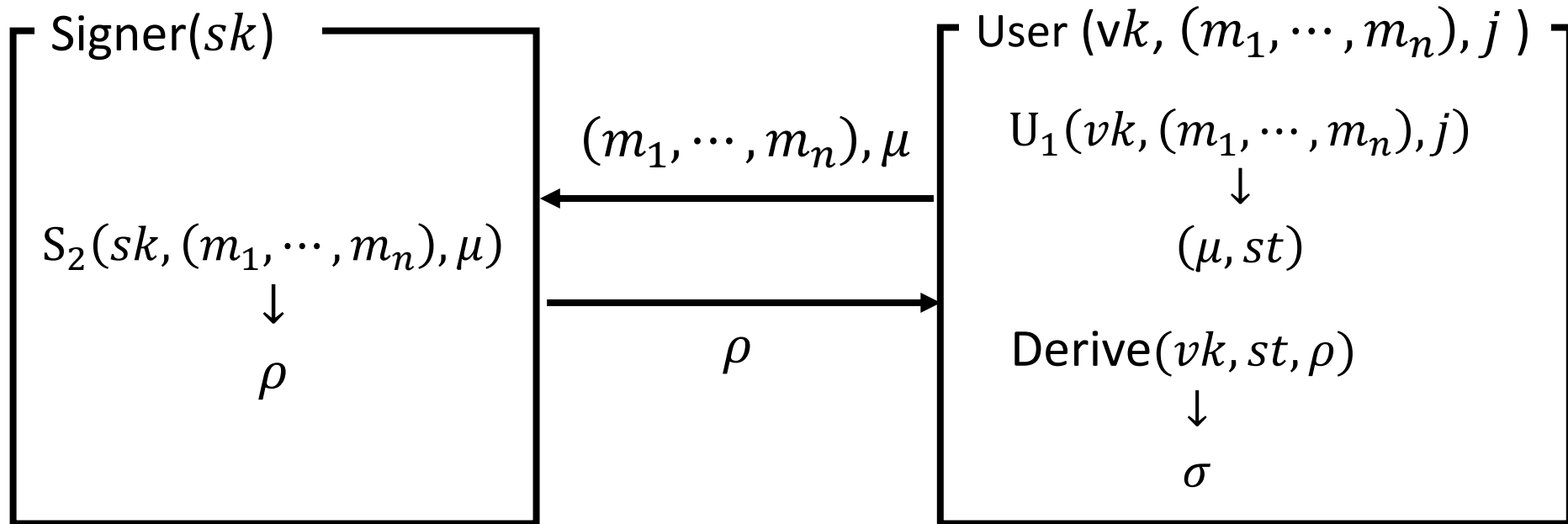


Syntax of (1,n)-Oblivious Signature Scheme

2-move (1,n)-OS (KGen, U_1 , S_2 , Derive, Verify)

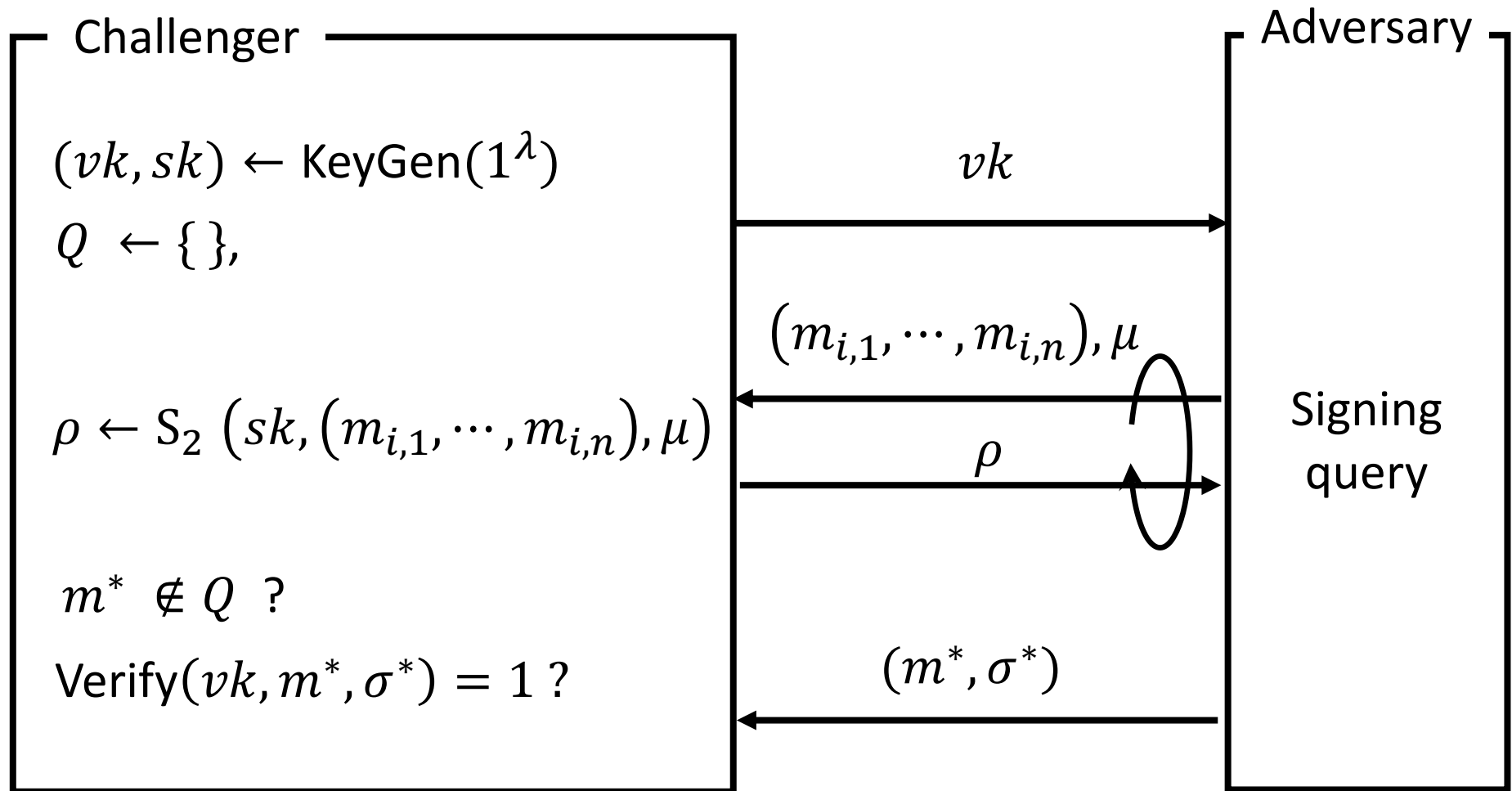
$\text{KGen}(1^\lambda) \rightarrow (vk, sk)$

Signing protocol (U_1 , S_2 , Derive)



$\text{Verify}(vk, m, \sigma) \rightarrow 0 \text{ or } 1$

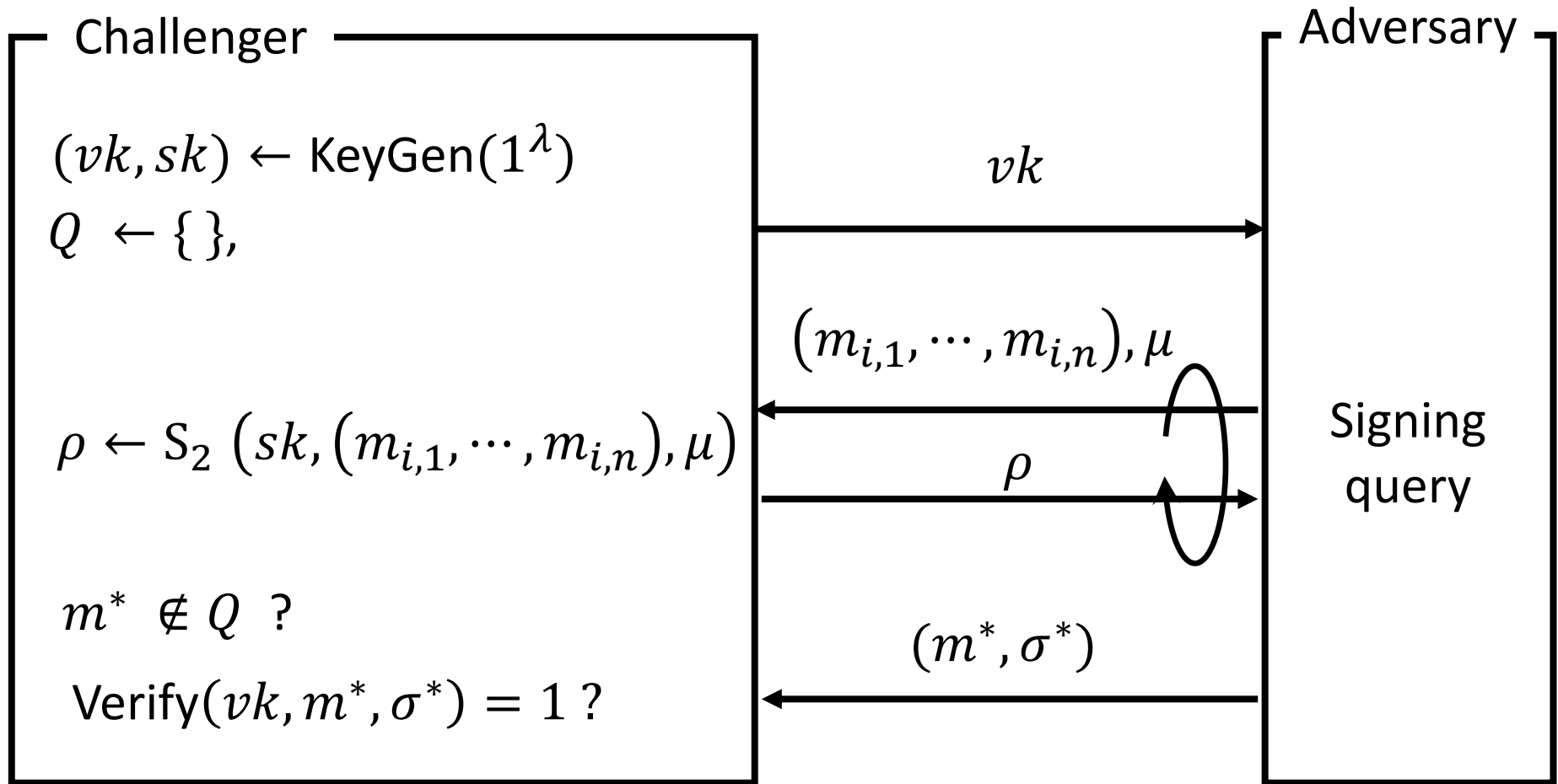
Unforgeability Security Game in [TOO08]



Q records signed messages that the adversary has obtained.

Problems in Unforgeability Security Model and Countermeasures

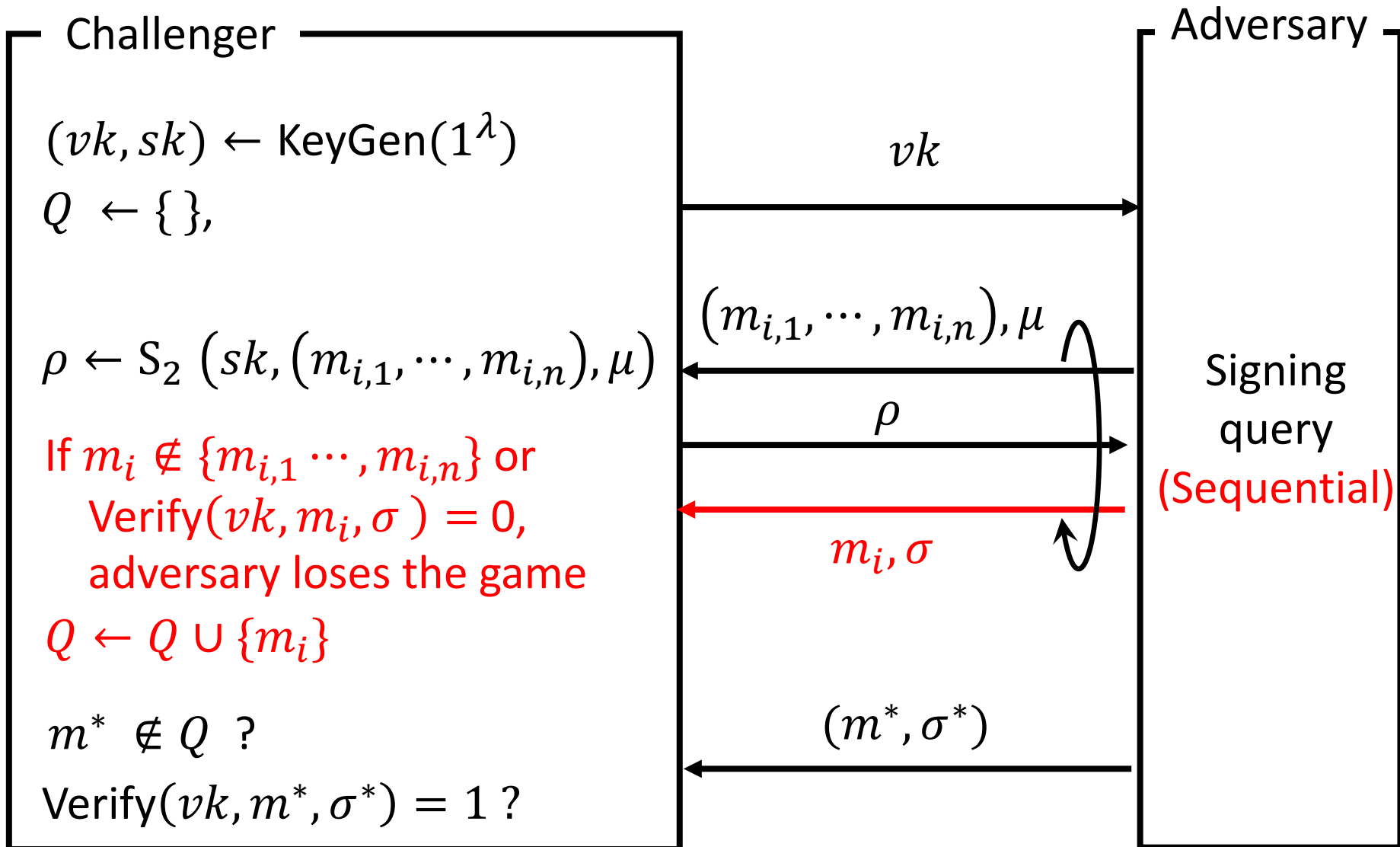
Problem 1 (How to Manage Set Q)



Q is a set of signed messages that the adversary has obtained.

By ambiguity, the challenger cannot know which one of message the adversary gets a signature in each signing query.

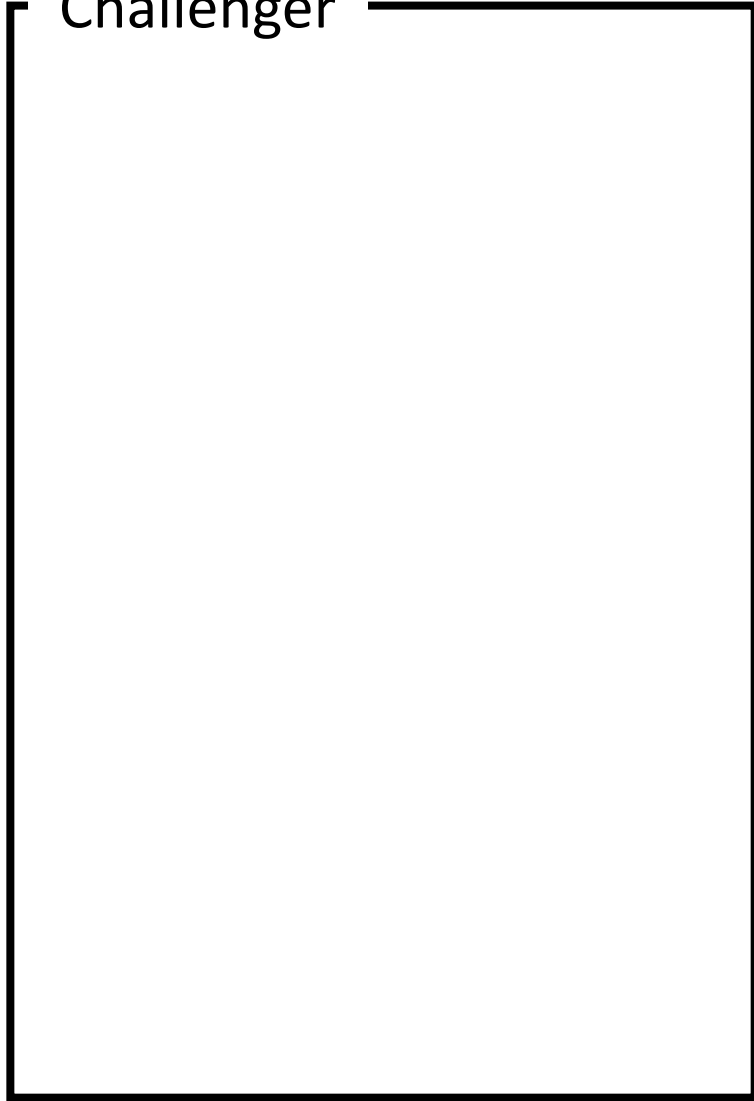
Countermeasure 1



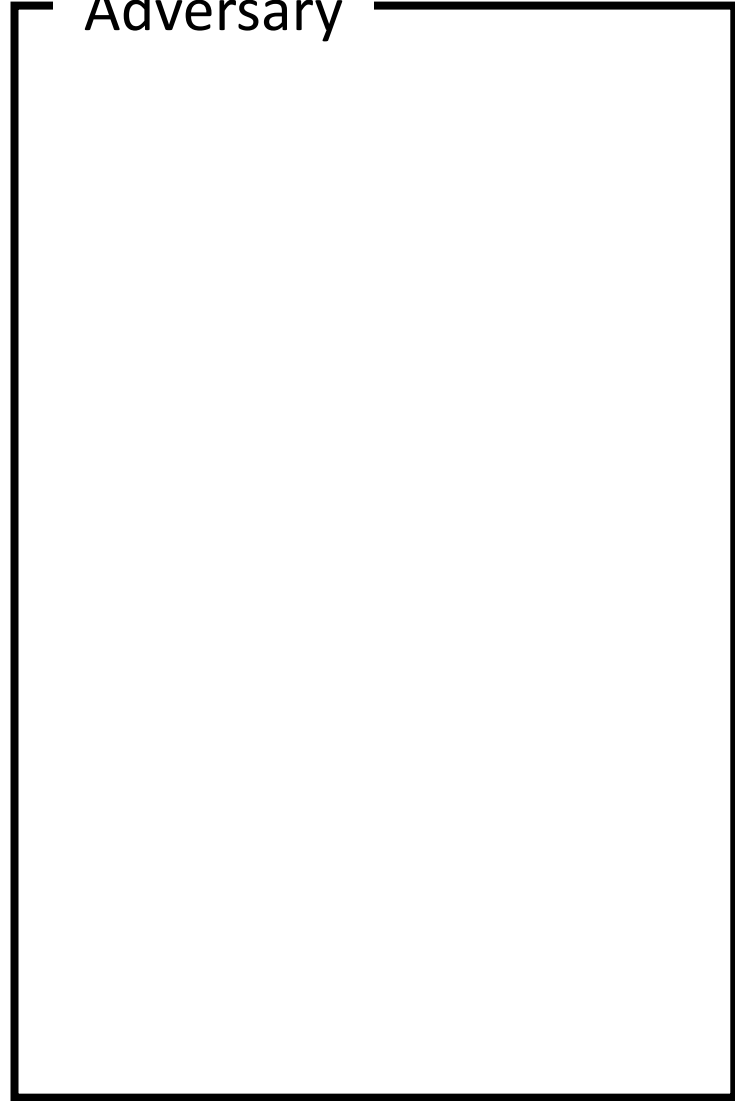
Problem 2 (Trivial Attack)

To simplify the discussion, we assume that n is 2.

Challenger

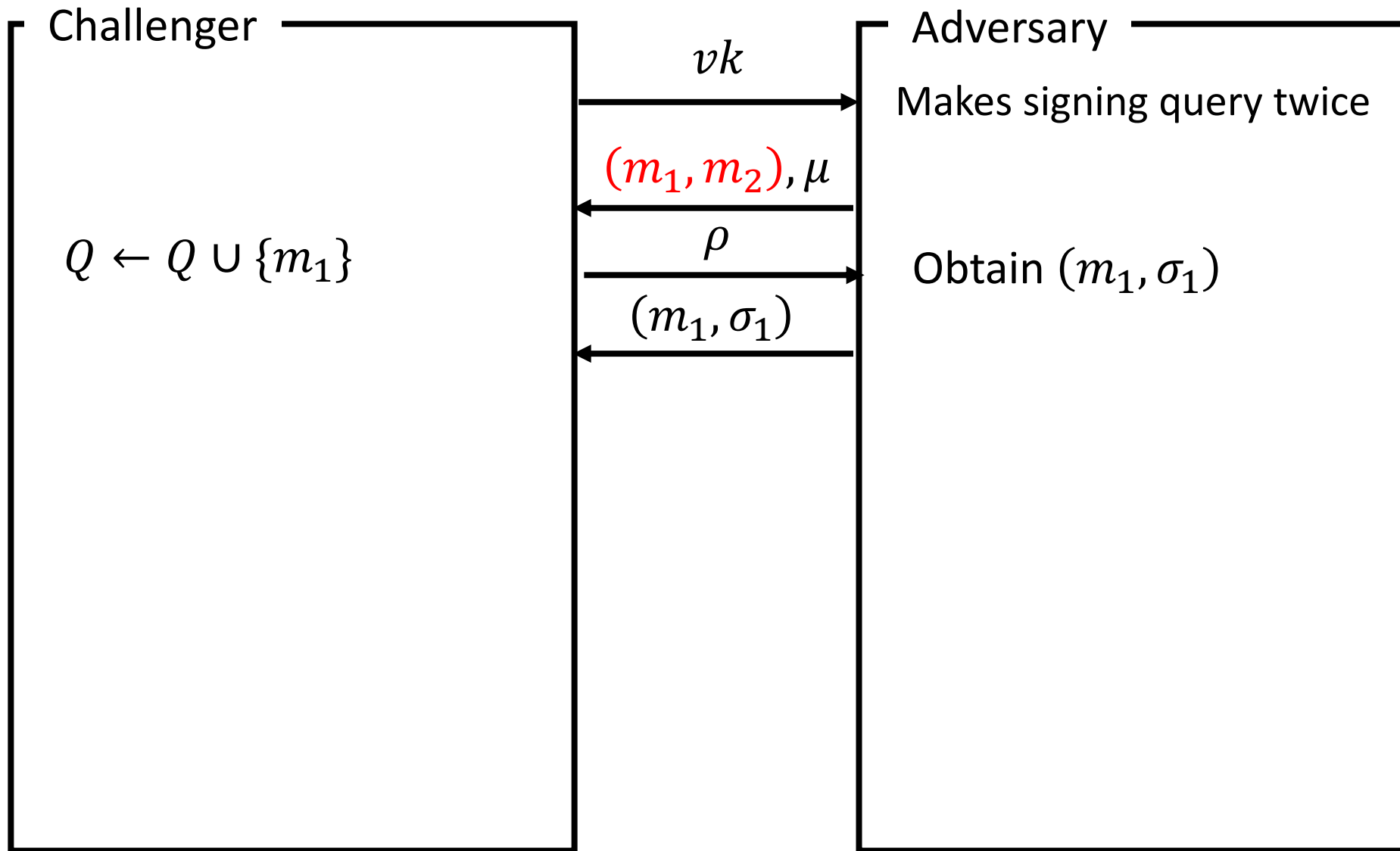


Adversary



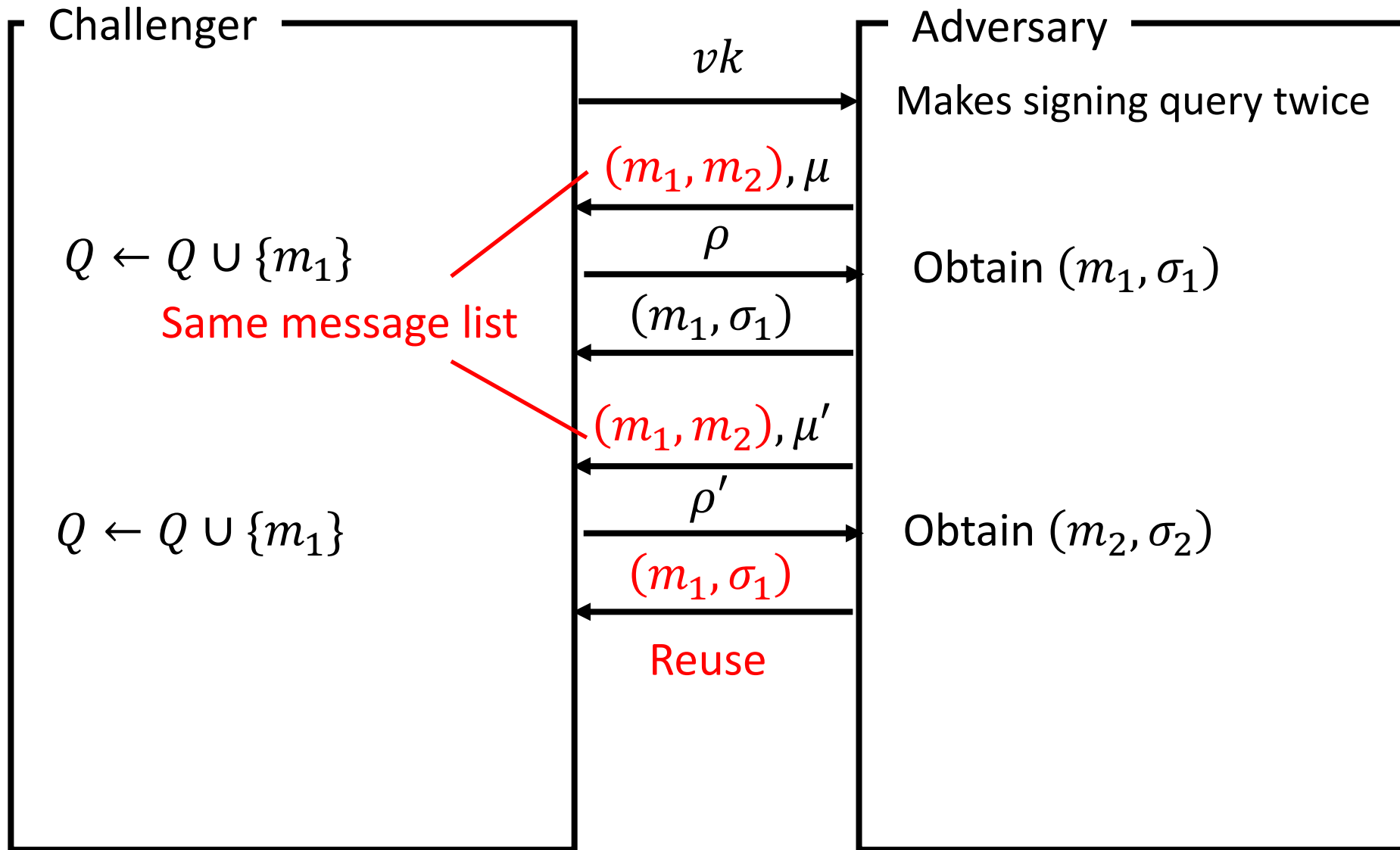
Problem 2 (Trivial Attack)

To simplify the discussion, we assume that n is 2.



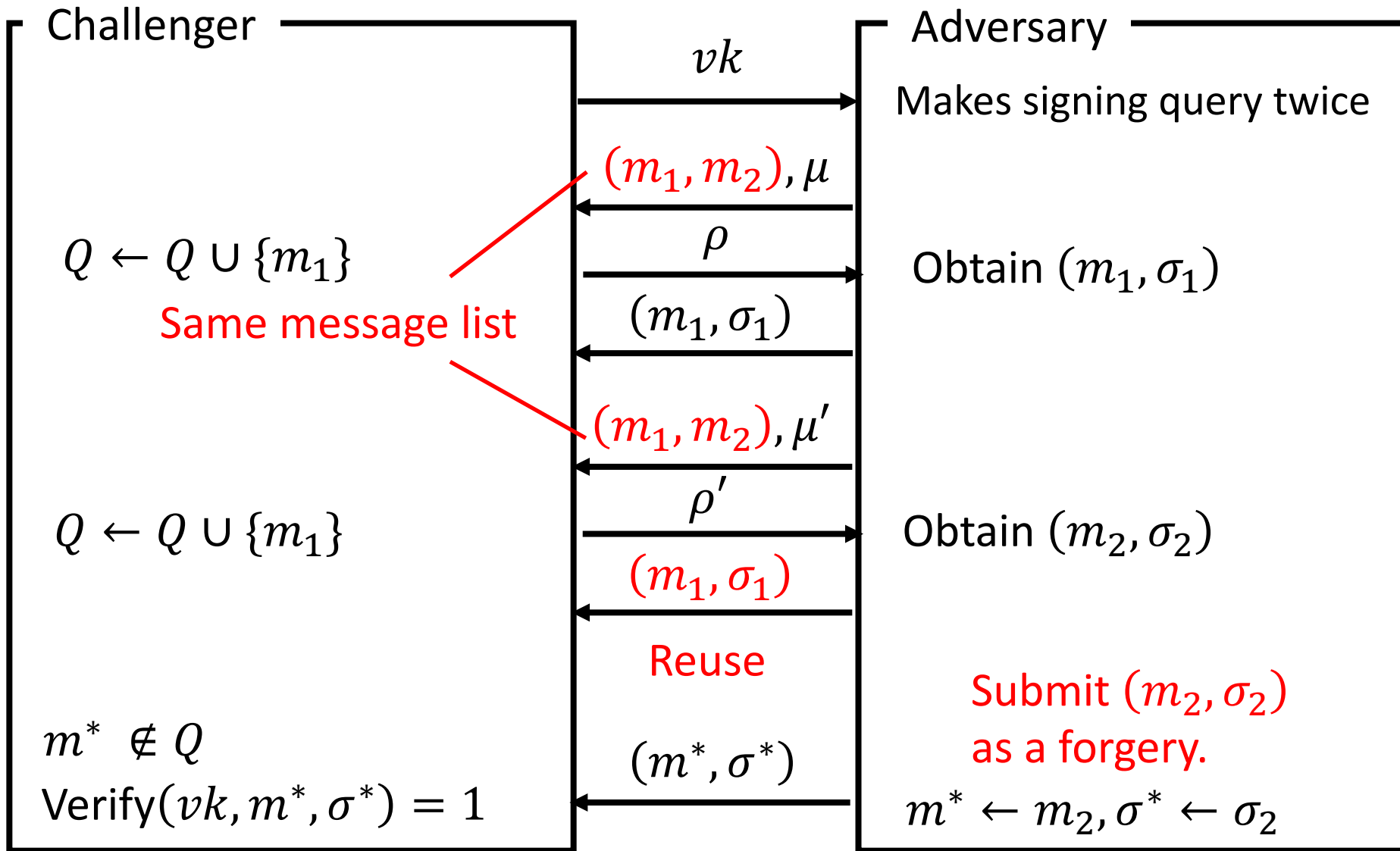
Problem 2 (Trivial Attack)

To simplify the discussion, we assume that n is 2.

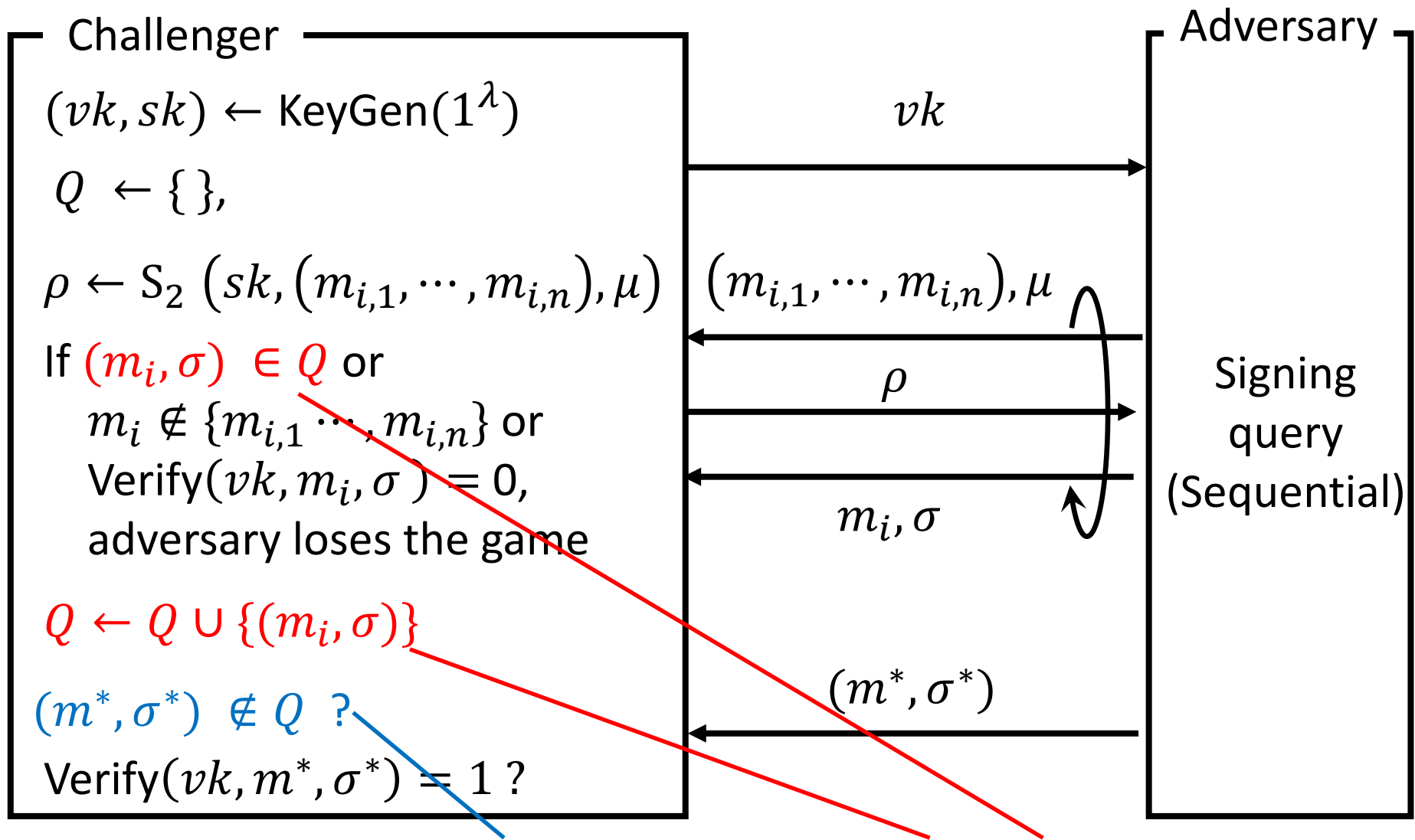


Problem 2 (Trivial Attack)

To simplify the discussion, we assume that n is 2.



Countermeasure 2

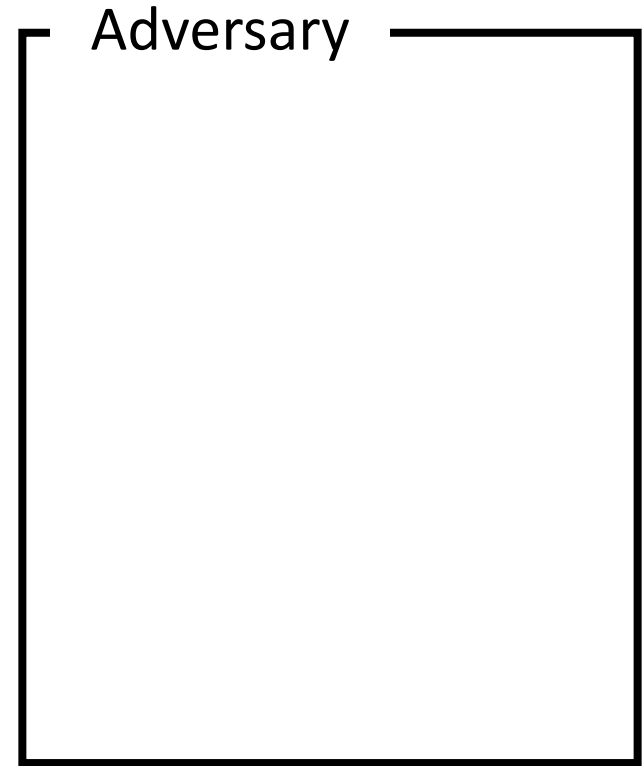


2. Prevent refreshing (reusing) signatures.
We make sUF security as a default!

1. Signature resubmission check!

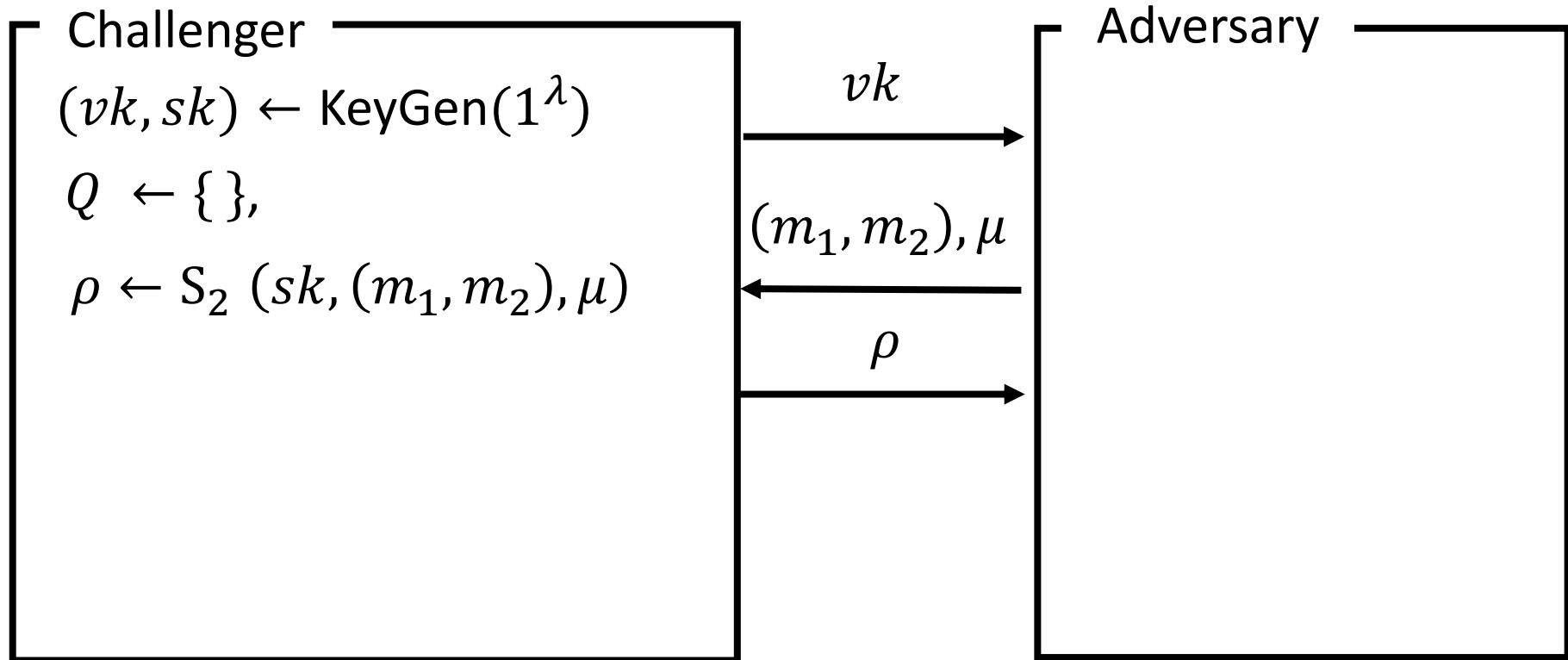
Problem 3 (Missing Adversary Strategy)

To simplify the discussion, we assume that n is 2.



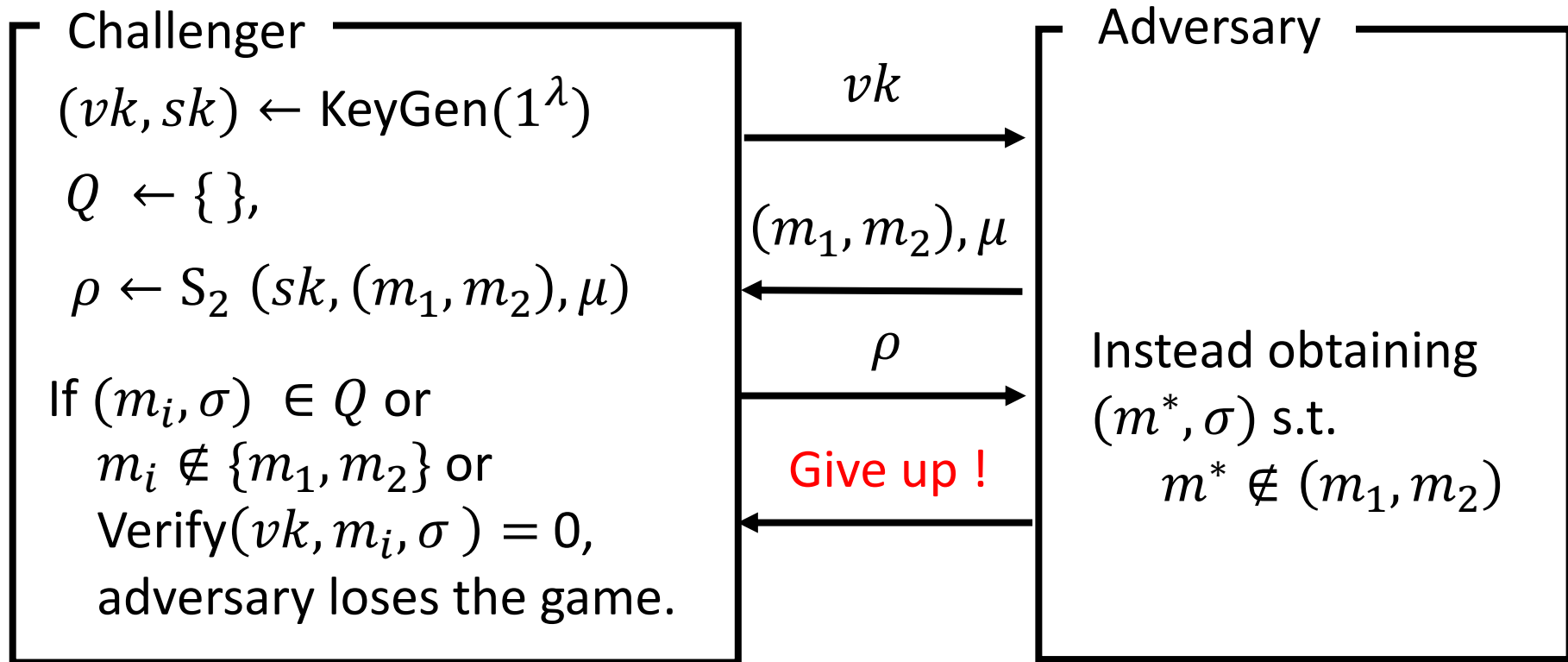
Problem 3 (Missing Adversary Strategy)

To simplify the discussion, we assume that n is 2.



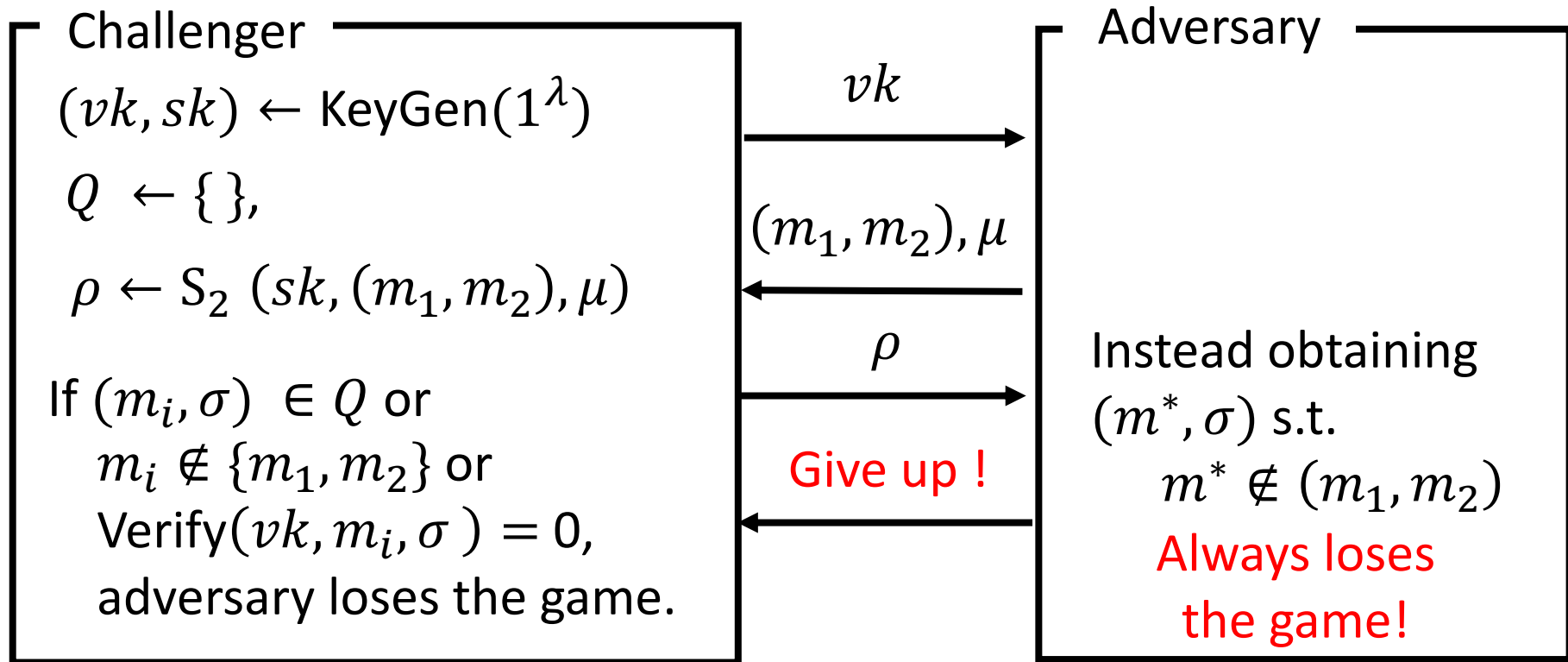
Problem 3 (Missing Adversary Strategy)

To simplify the discussion, we assume that n is 2.



Problem 3 (Missing Adversary Strategy)

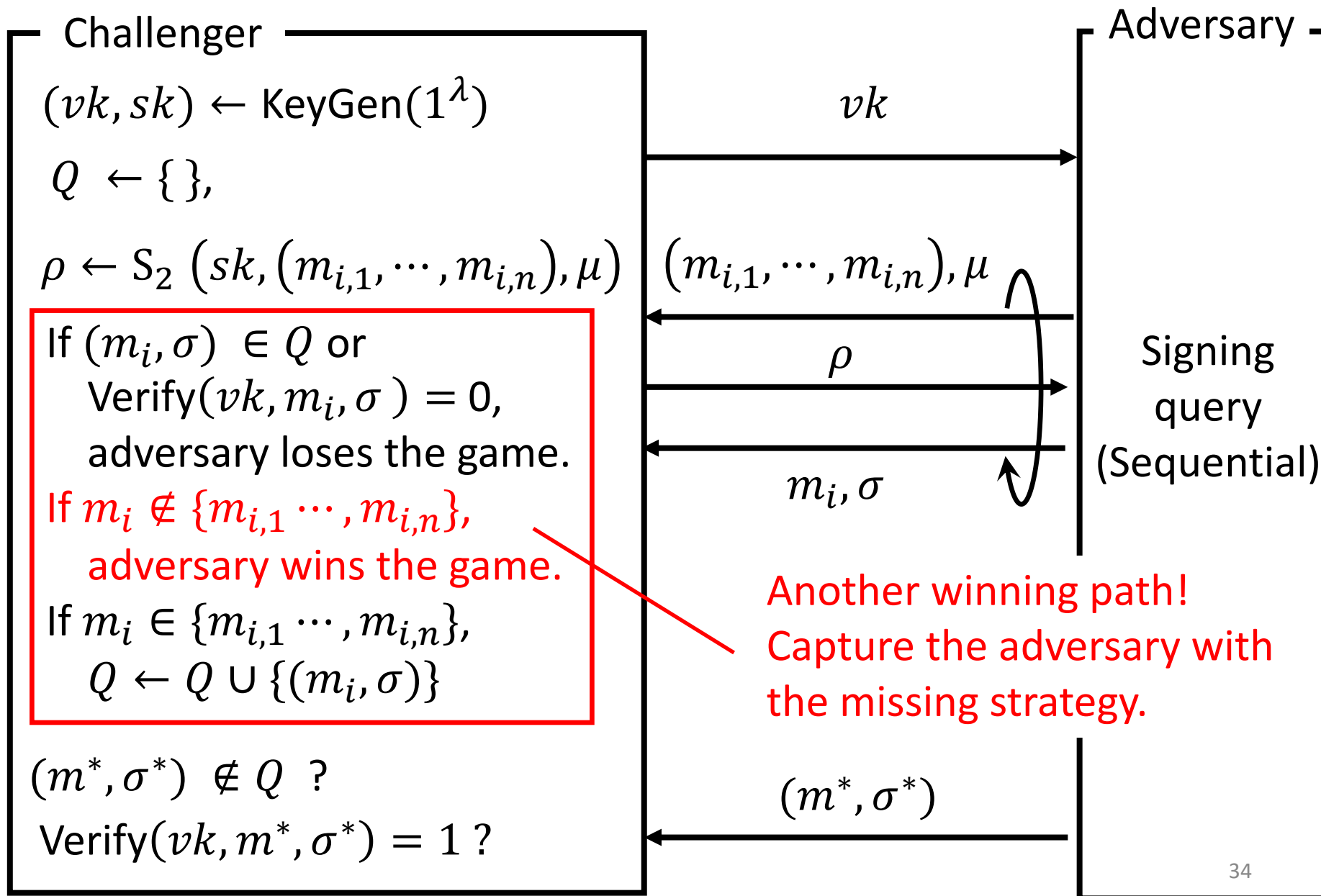
To simplify the discussion, we assume that n is 2.



Unforgeability security must guarantee that the user cannot obtain a signature on a message which is not in the list!

→ This security model does not capture this requirement!

Countermeasure 3



Communication Size Improvement Result in Our Scheme

Communication Message Size

| Scheme | Building Block | First Message μ | Second Message ρ |
|----------|----------------|---------------------|-----------------------------|
| [ZLH 22] | DS COM | 1 com for m_j | n sigs on (m_i, μ) |

We reduce the second message size !

Communication Size Improvement Result

| Scheme | Building Block | First Message μ | Second Message ρ |
|----------|--------------------------|---------------------|----------------------------------|
| [ZLH 22] | DS COM | 1 com for m_j | n sigs on (m_i, μ) |
| Ours | DS COM Merkle Tree | 1 com for m_j | 1 sig on (root, μ) |

root: Assigned root node value of the Merkle Tree on (m_1, \dots, m_n)

Communication Size Improvement Result

| Scheme | Building Block | First Message μ | Second Message ρ |
|----------|--------------------------|---------------------|----------------------------------|
| [ZLH 22] | DS COM | 1 com for m_j | n sigs on (m_i, μ) |
| Ours | DS COM Merkle Tree | 1 com for m_j | 1 sig on (root, μ) |

root: Assigned root node value of the Merkle Tree on (m_1, \dots, m_n)

Security of Our Scheme

Ambiguity Security: Hiding COM

Unforgeability Security: sEUF-CMA DS + Binding COM + Coll resist H

Summary

- We revisited the unforgeability security model by Tso et al.
We identify problems and redefine the security model.
- We improve the generic construction by Zhou et al.
Our scheme offers the smaller second message size.

Thank you!

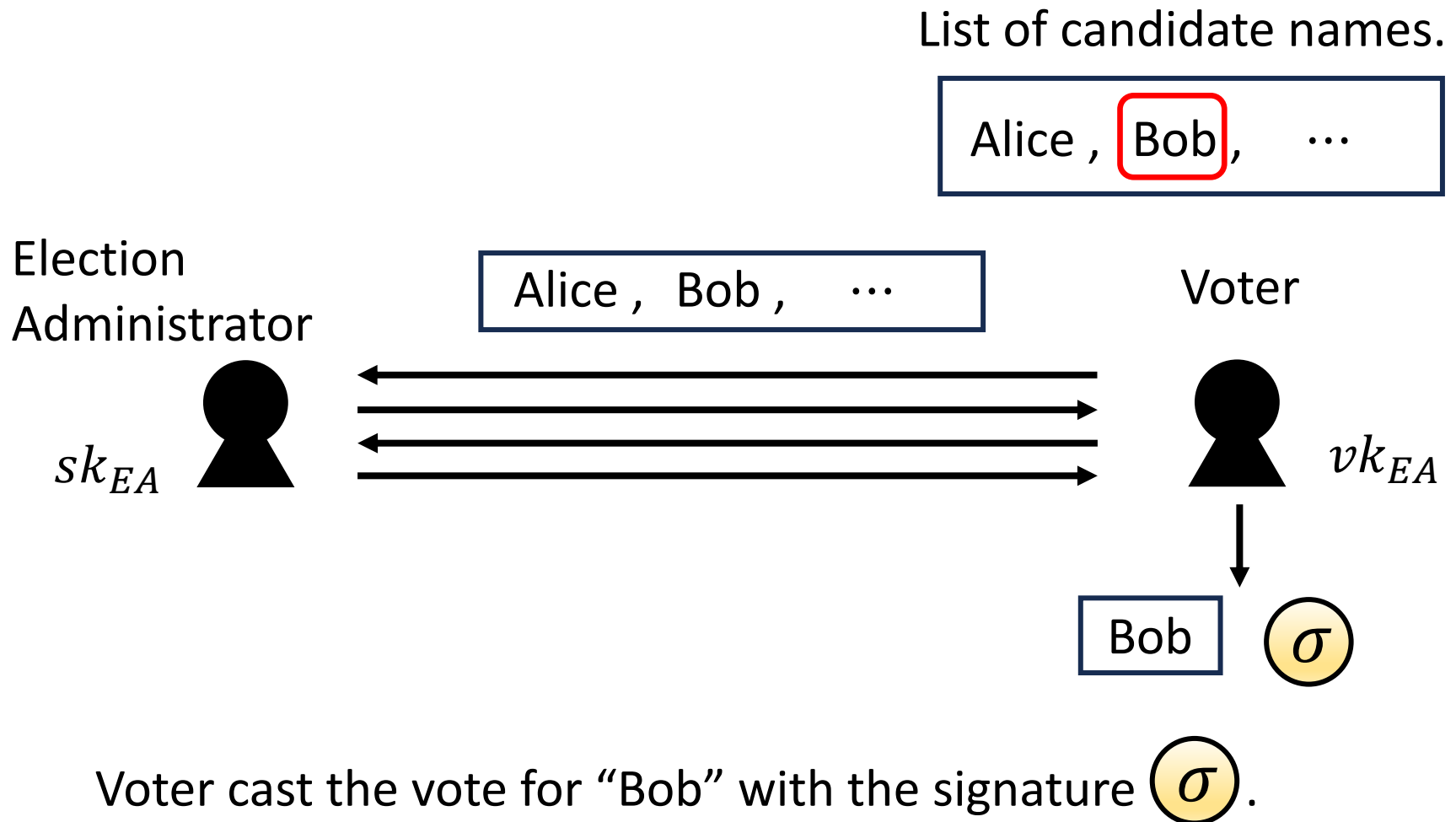
References

- [Chen94] L. Chen. Oblivious signatures. (ESORICS 1994)
- [TOO08] R. Tso, T. Okamoto, and E. Okamoto. 1-out-of-n oblivious signatures. (ISPEC 2008)
- [YLTTM22] J. You, Z. Liu, R. Tso, Y. Tseng, and M. Mambo. Quantum-resistant 1-out-of-n oblivious signatures from lattices. (IWSEC 2022)
- [ZLH22] Y. Zhou, S. Liu, and S. Han. Generic construction of 1-out-of-n oblivious signatures. (IEICE Trans. Inf. Syst. 2022)
- [SYL08] C. Song, X. Yin, and Y. Liu. A practical electronic voting protocol based upon oblivious signature scheme. (CIS 2008)
- [CC18] S. Chiou and J. Chen. Design and implementation of a multiple-choice e-voting scheme on mobile system using novel t-out-of-n oblivious signature. (J. Inf. Sci. Eng. 2018).

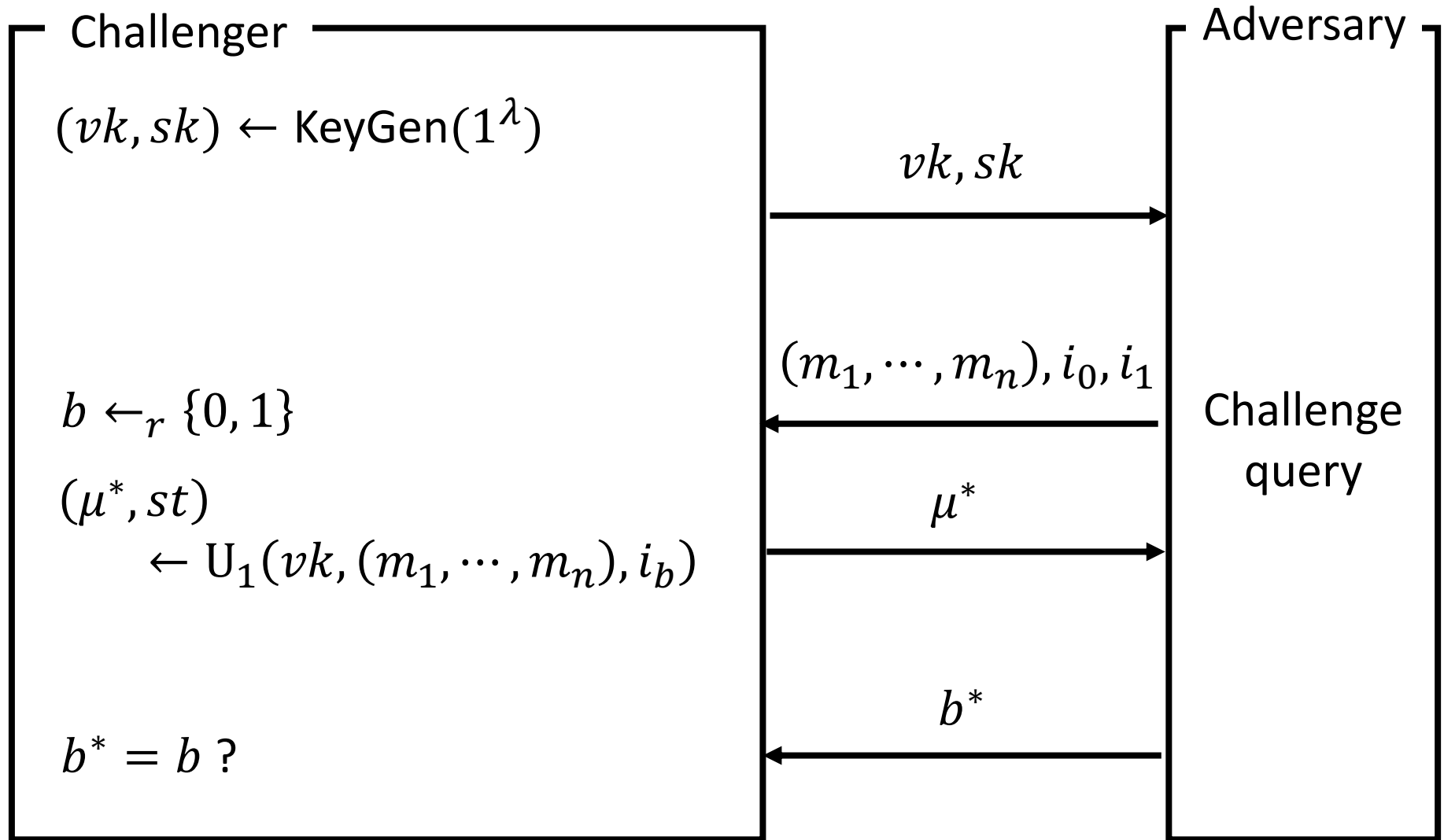
Appendix

Application of (1,n)-Oblivious Signatures

E-voting system based on oblivious signatures [SYL08, CC18]



Ambiguity Security Game



Commitment Scheme

Commitment Scheme

$$ck \leftarrow \text{KGen}(1^\lambda)$$

Sender (ck, m)



(c, m, r)



Receiver(ck)



$$c \leftarrow \text{Commit}(ck, m; r)$$

$$\text{Open}(ck, c, m; r) = 1 ?$$

Security

Hiding: A commitment c hides the committed message m .

Binding: A commitment c can only be opened with the committed message m .

Digital Signature Scheme

Digital Signature Scheme

$$(vk, sk) \leftarrow \text{KGen}(1^\lambda)$$

Signer (sk, m)



$$\sigma \leftarrow \text{Sign}(sk, m)$$

(m, σ)



Verifier (vk)

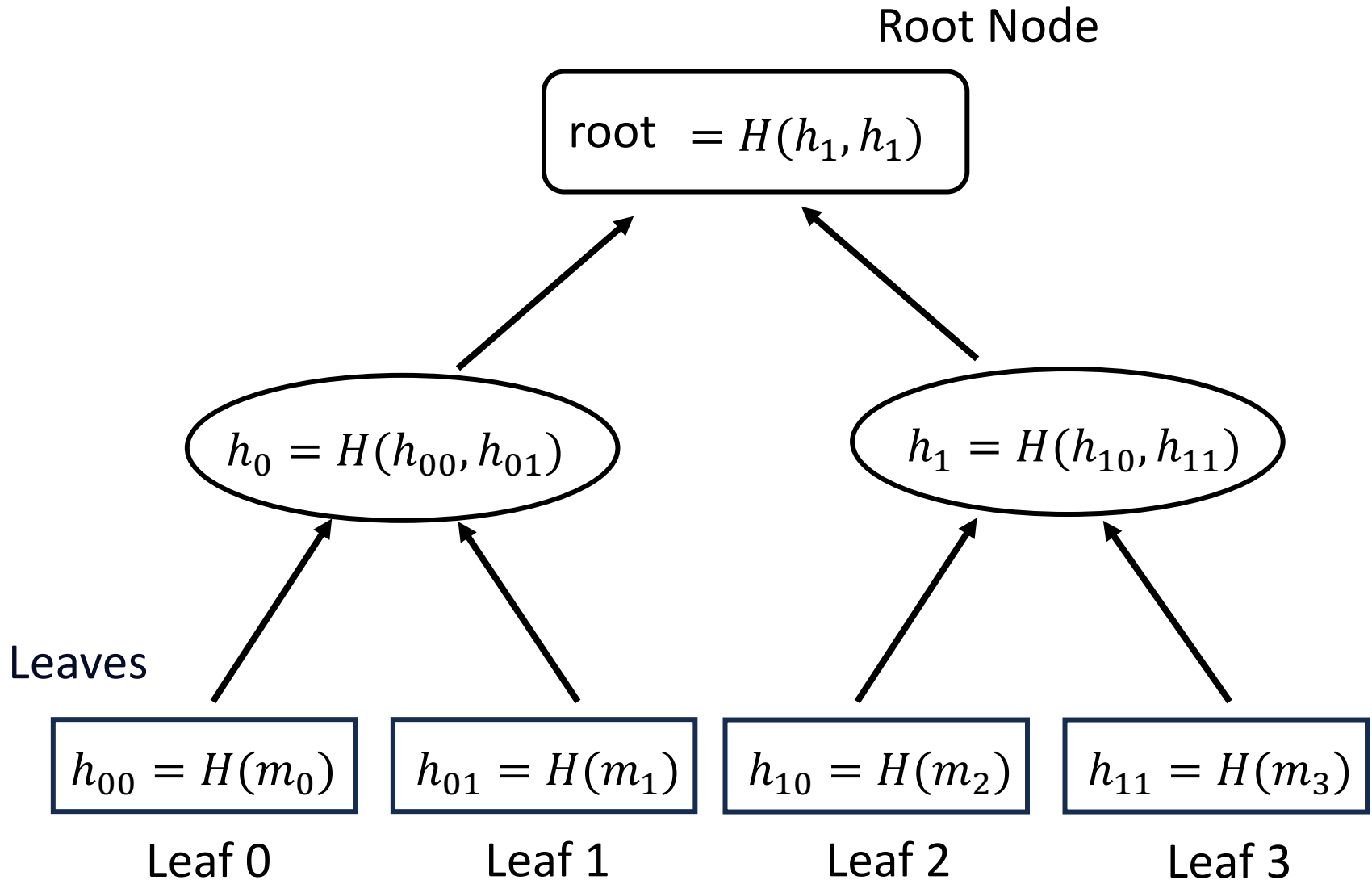


$$\text{Verify}(vk, m, \sigma) = 1 ?$$

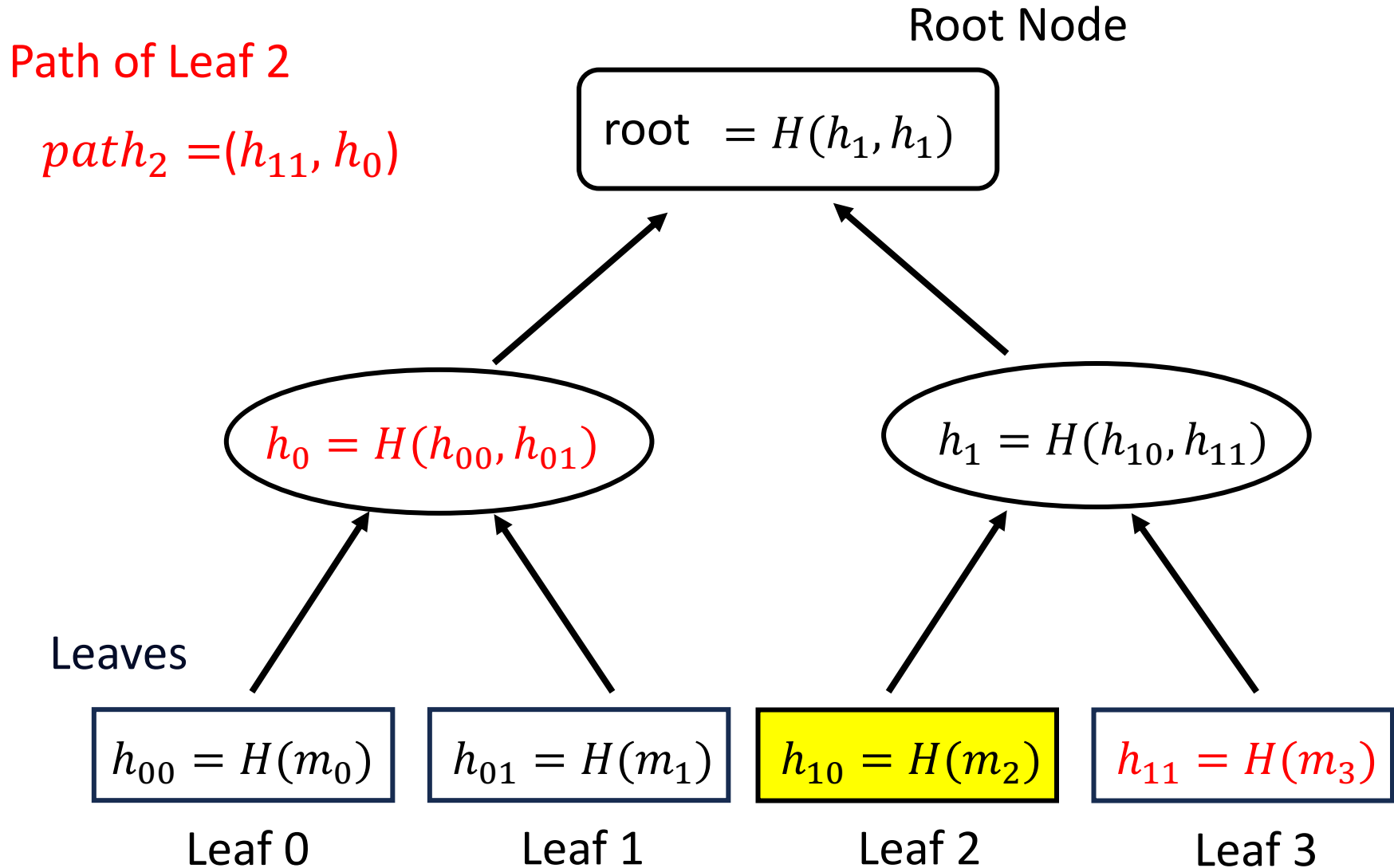
Security (Strong EUF-CMA)

If an adversary obtains message-signature pairs $(m_i, \sigma_i)_i$ on their message choice via signing queries, it is difficult to generate a forgery (m^*, σ^*) which has not been outputted by signing queries.

Merkle Tree



Merkle Tree



Generic Construction by Zhou et al. [ZLH22]

DS: Digital signature scheme

Com: Commitment scheme

Signer ($sk^{OS} = sk^{DS}$)

User ($vk^{OS} = (ck, vk^{DS}),$
 $(m_1, \dots, m_n), j)$

$S_2(sk^{OS}, (m_1, \dots, m_n), \mu)$

For $i \in [n],$

$\sigma_i^{DS} \leftarrow \text{DS.Sign}(sk^{DS}, (m_i, \mu))$

(m_1, \dots, m_n)
 $\mu = c$

$U_1(vk^{OS}, (m_1, \dots, m_n), j)$
 $c \leftarrow \text{Com.Commit}(ck, m_j; r)$

$\rho = (\sigma_i^{DS})_{i \in [n]}$

$\text{Derive}(vk^{OS}, st = (r, j), \rho)$
 $\sigma^{OS} \leftarrow (c, r, \sigma_j^{DS})$

A second communication message ρ needs n signatures !

Signing on (m_i, μ) is seems redundant.

Our Improved Scheme

DS: Digital signature scheme

Com: Commitment scheme

H : Hash function

Signer ($sk^{OS} = (sk^{DS}, H)$)

User ($vk^{OS} = (ck, vk^{DS}, H),$
 $(m_1, \dots, m_n), j)$

$S_2(sk, (m_1, \dots, m_n), \mu)$

Check messages in (m_1, \dots, m_n)
 are all distinct.

Compute the root of
 Merkle Tree from (m_1, \dots, m_n)

$\sigma^{DS} \leftarrow \text{DS.Sign}(sk^{DS}, (\text{root}, \mu))$

(m_1, \dots, m_n)
 $\mu = c$

$U_1(vk, (m_1, \dots, m_n), j)$
 $c \leftarrow \text{Com.Commit}(ck, m_j; r)$


Derive($vk, st = (r, j), \rho$)

Compute root of Merkle
 Tree from (m_1, \dots, m_n) .

Compute $path_j$.

$\sigma^{OS} \leftarrow (c, r, \sigma_j^{DS},$

$\text{root}, path_j)$

$\rho = \sigma^{DS}$


1 signature

Why Our Model Straightforwardly extends to Concurrent Signing Model ?

If we extend our model to concurrent signing setting, there is a problem.

