# M&M'S: Mix and Match Attaks on Schnorr-type Blind Signatures with Repetition

*K. Do, L. Hanzlik, E. Paracucchi*

Eurocrypt 2024 | Zurich | May 27th

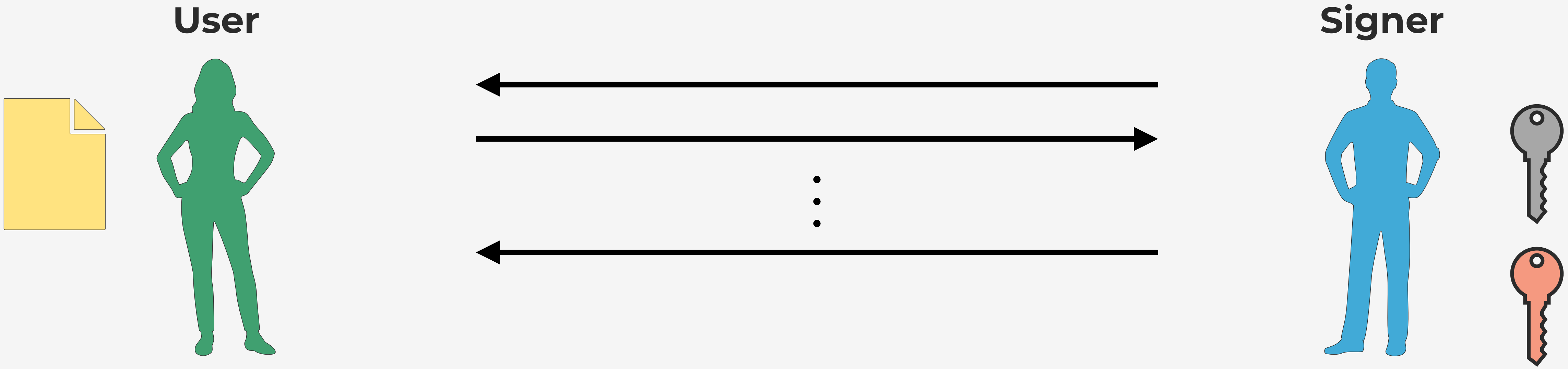
# Goals

I. Introduction: blind signatures and security model

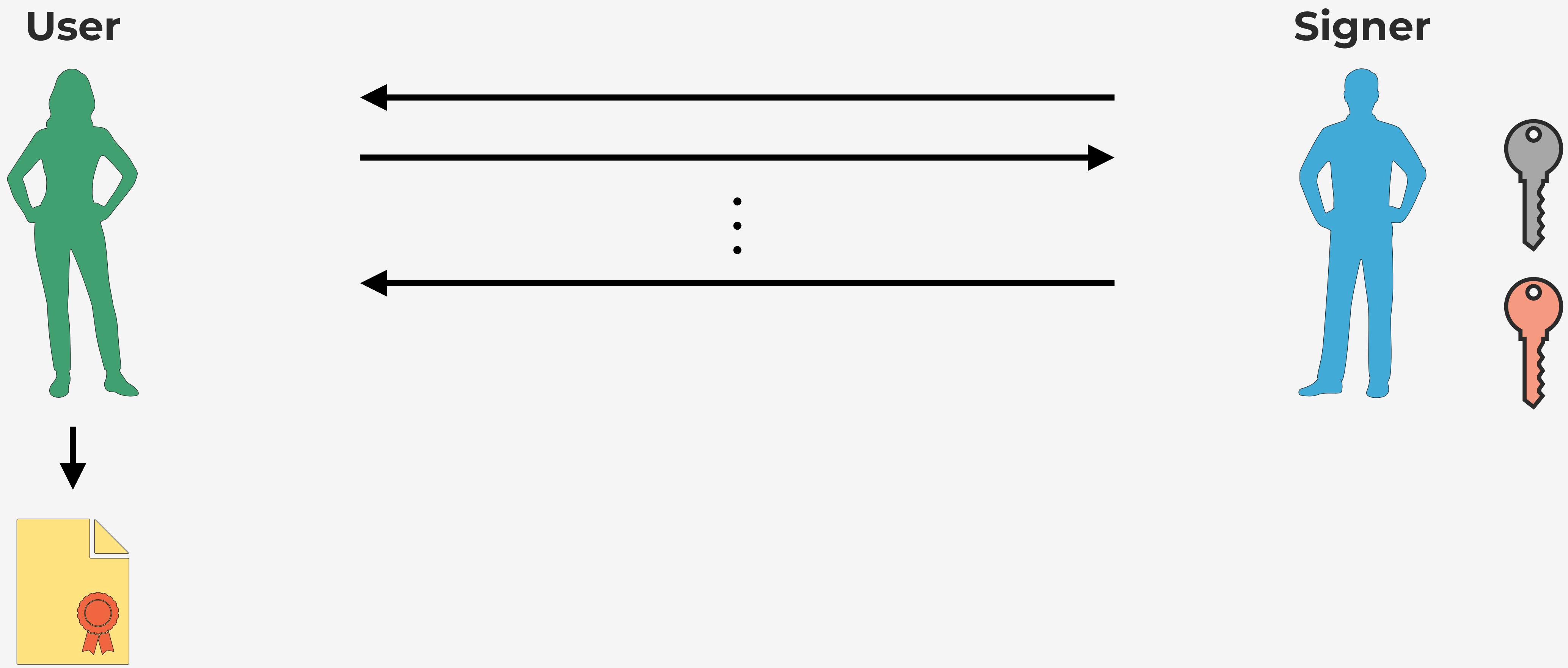
II. Schnorr-type blind signatures

III. Mix and match attacks

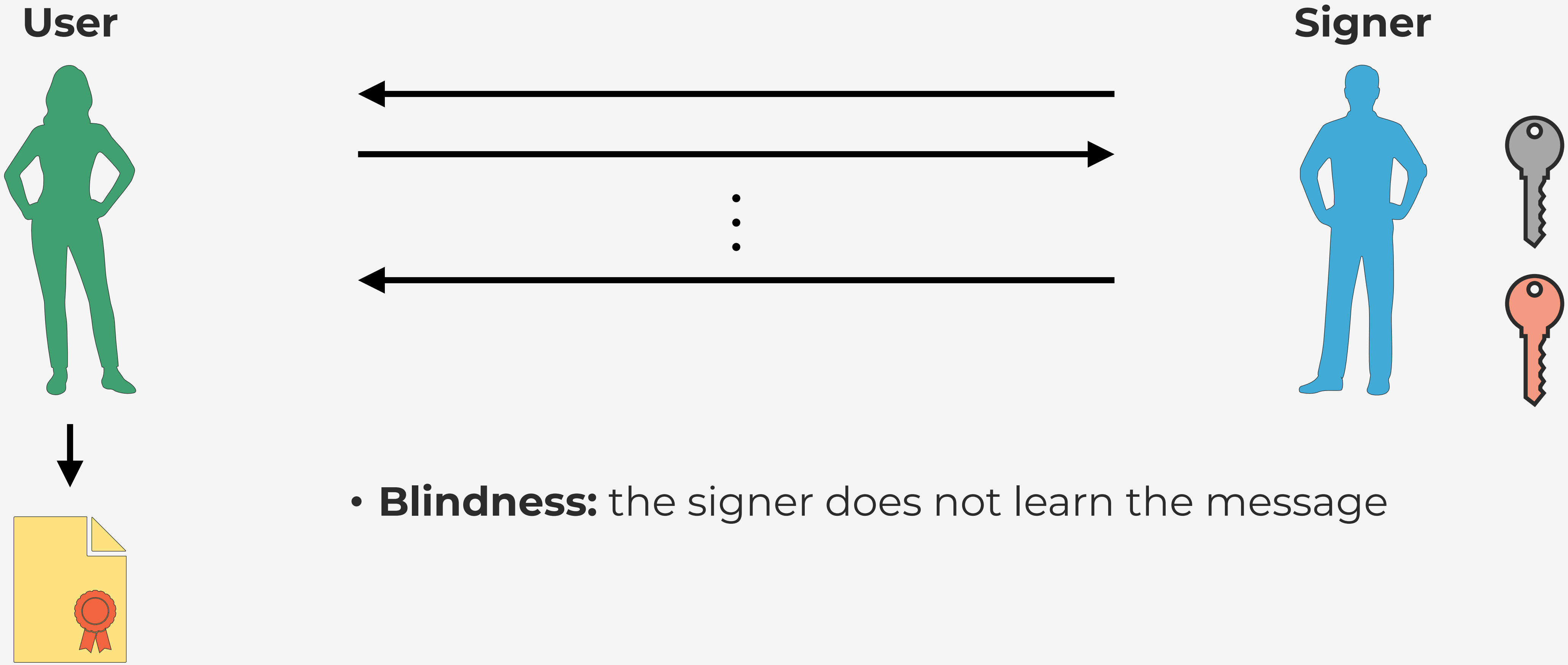
# Blind Signatures



# Blind Signatures

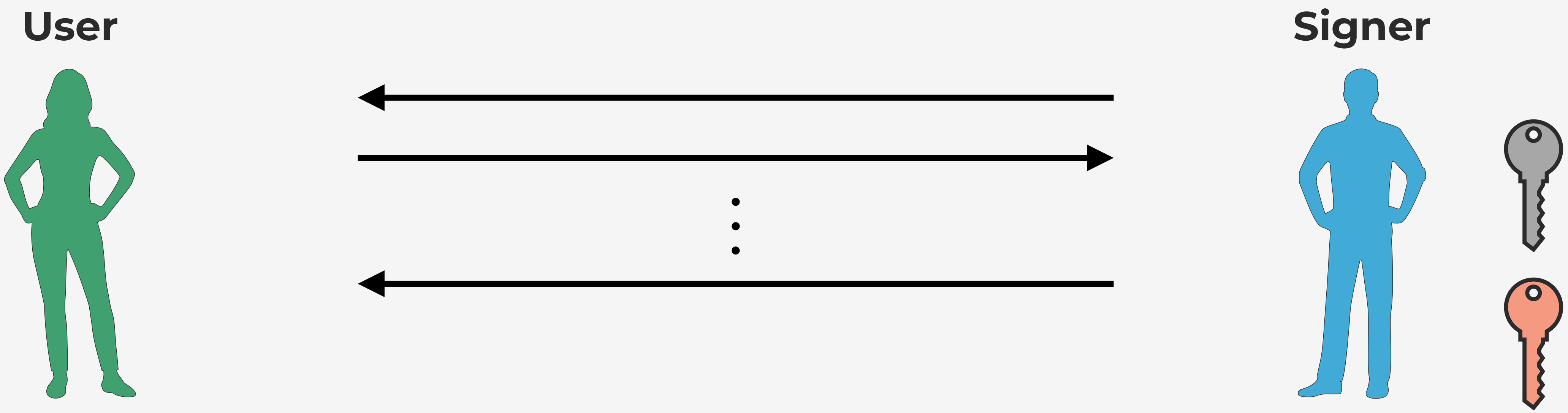


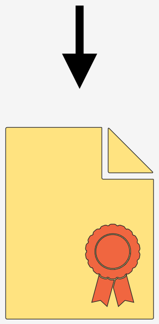
# Blind Signatures




- **Blindness:** the signer does not learn the message

# Blind Signatures



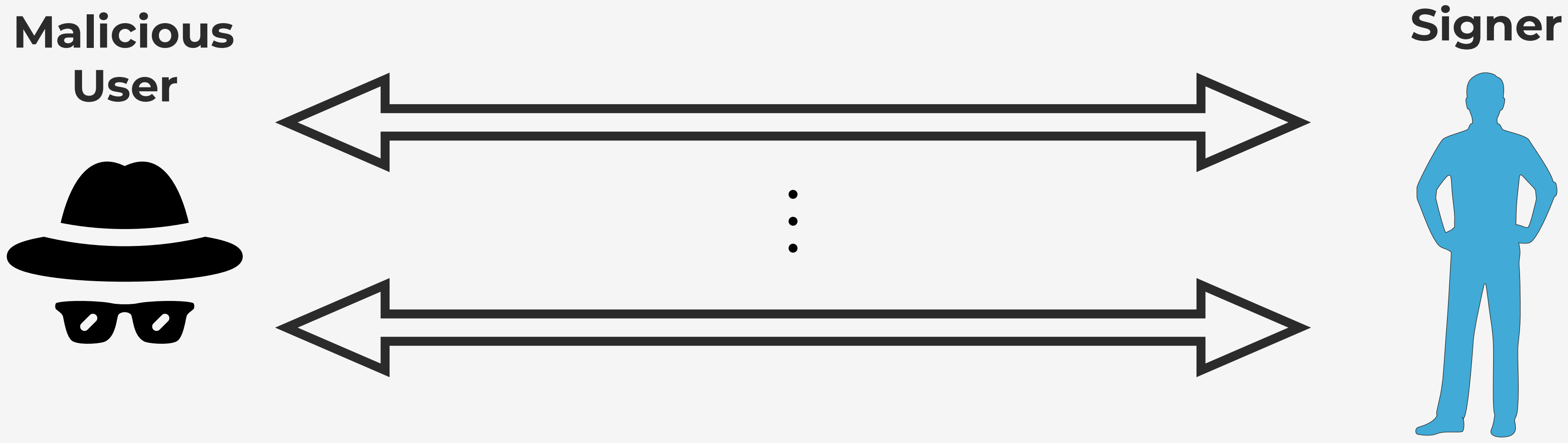



- **Blindness:** the signer does not learn the message
- **Unforgeability*:** the user needs the signer to get a valid signature

# One More Unforgeability

# One More Unforgeability

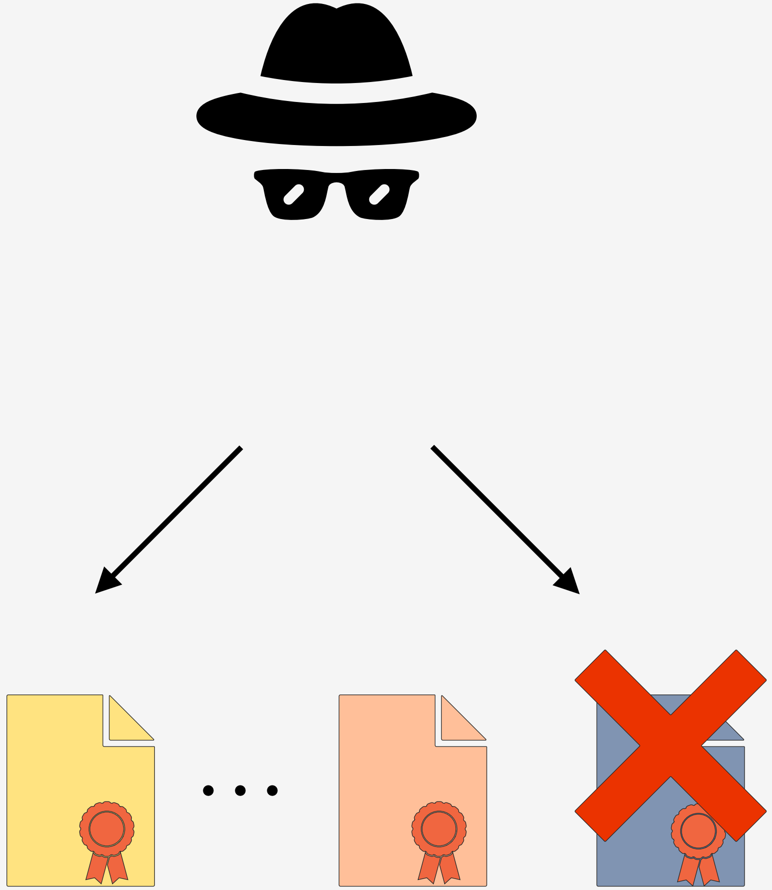
# One More Unforgeability

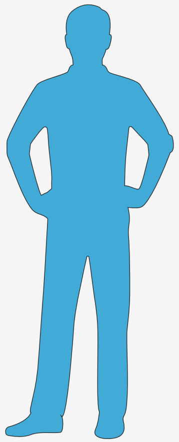


# One More Unforgeability

**Malicious User**

**Signer**





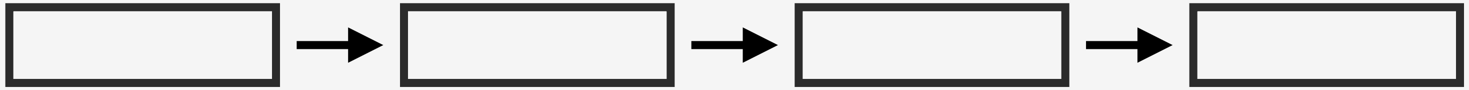
**One more unforgeability:**

The user cannot create $\ell + 1$ valid signatures under different messages while only finishing the signing process $\ell$ times with the signer

# Sequential vs Concurrent Security

The one more unforgeability comes with two flavors:

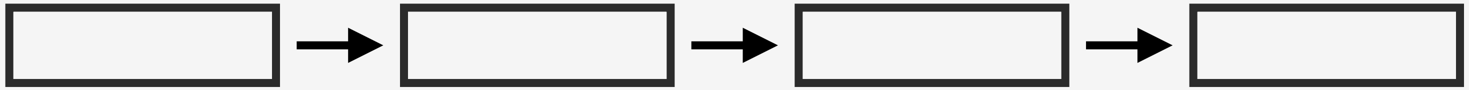
- **Sequential** security: to open a new session one must first close the previous one



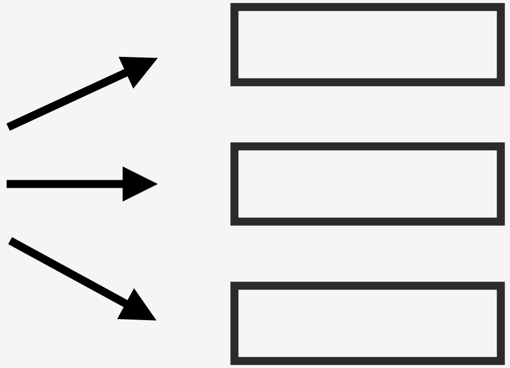
# Sequential vs Concurrent Security

The one more unforgeability comes with two flavors:

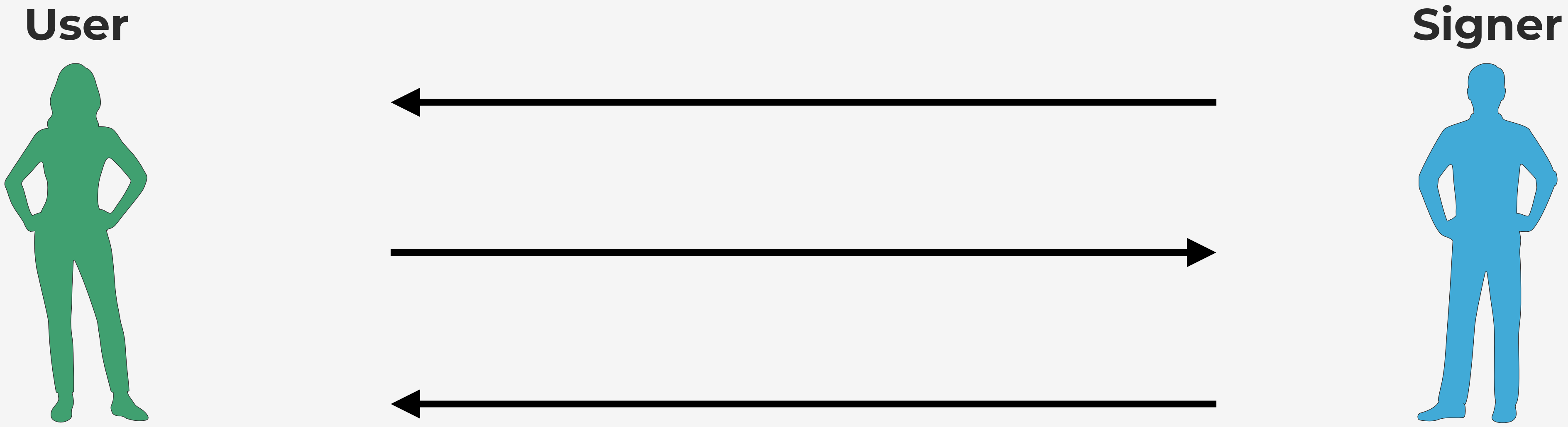
- **Sequential** security: to open a new session one must first close the previous one



- **Concurrent** security: users can execute sessions in parallel



# Schnorr-type Blind Signatures



# Schnorr-type Blind Signatures

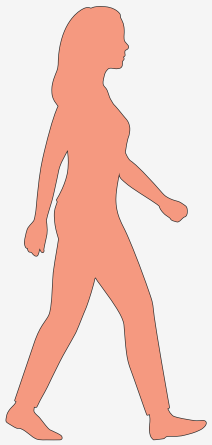

It is a folklore approach to constructing blind signatures on the base of interactive **identification schemes** (sigma protocols)

# Schnorr-type Blind Signatures

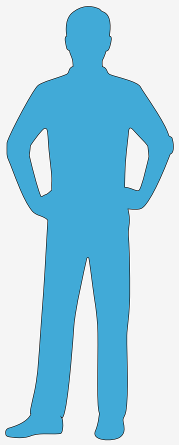

It is a folklore approach to constructing blind signatures on the base of interactive **identification schemes** (sigma protocols)

# Identification Schemes

**Verifier**



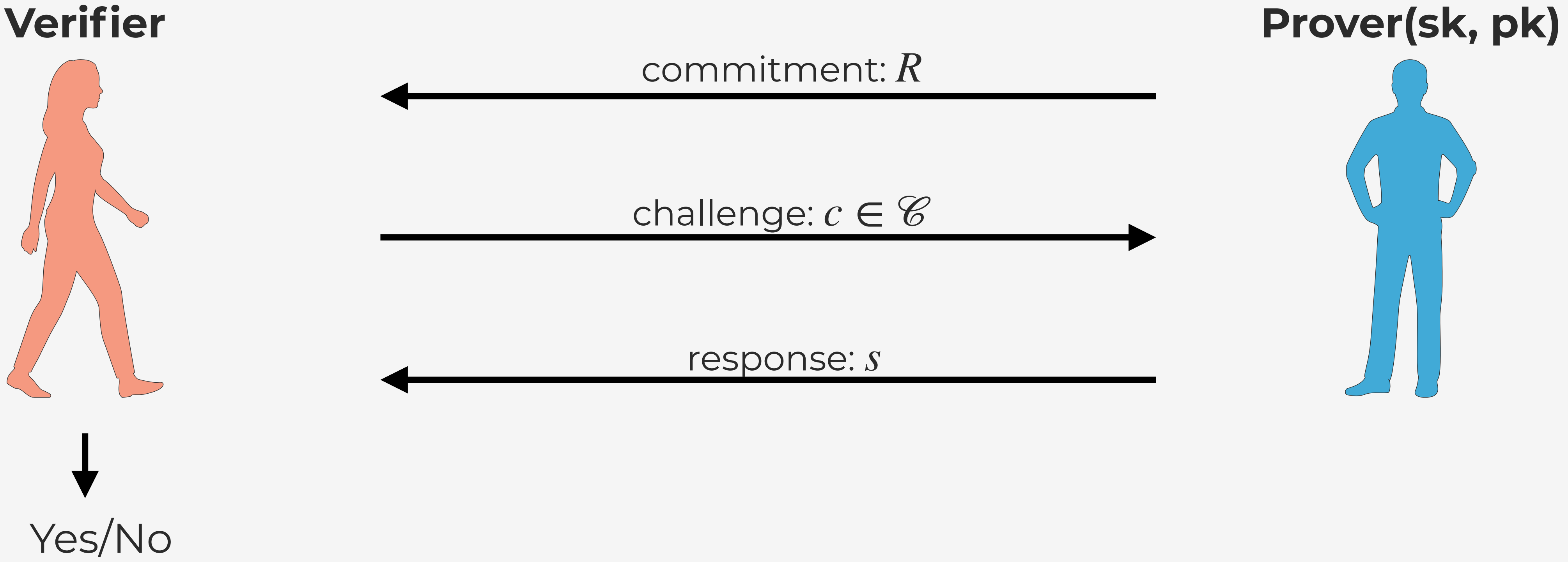
**Prover(sk, pk)**



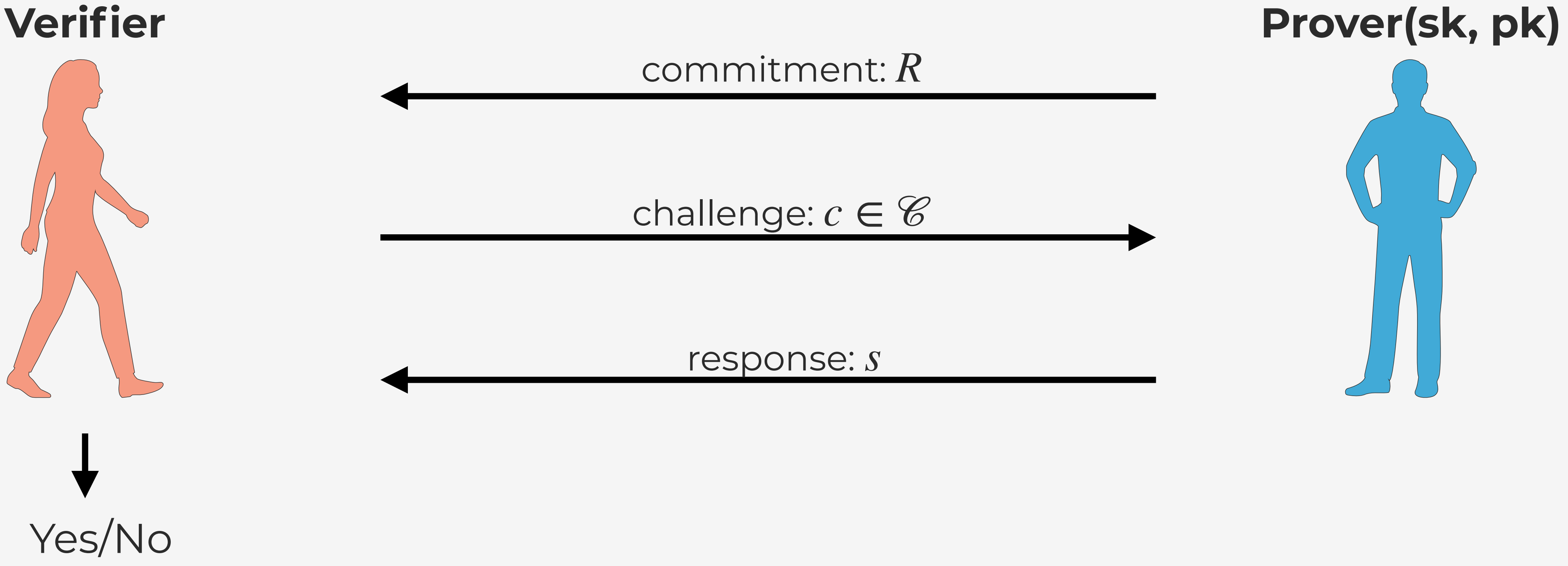
# Identification Schemes

# Identification Schemes

# Identification Schemes

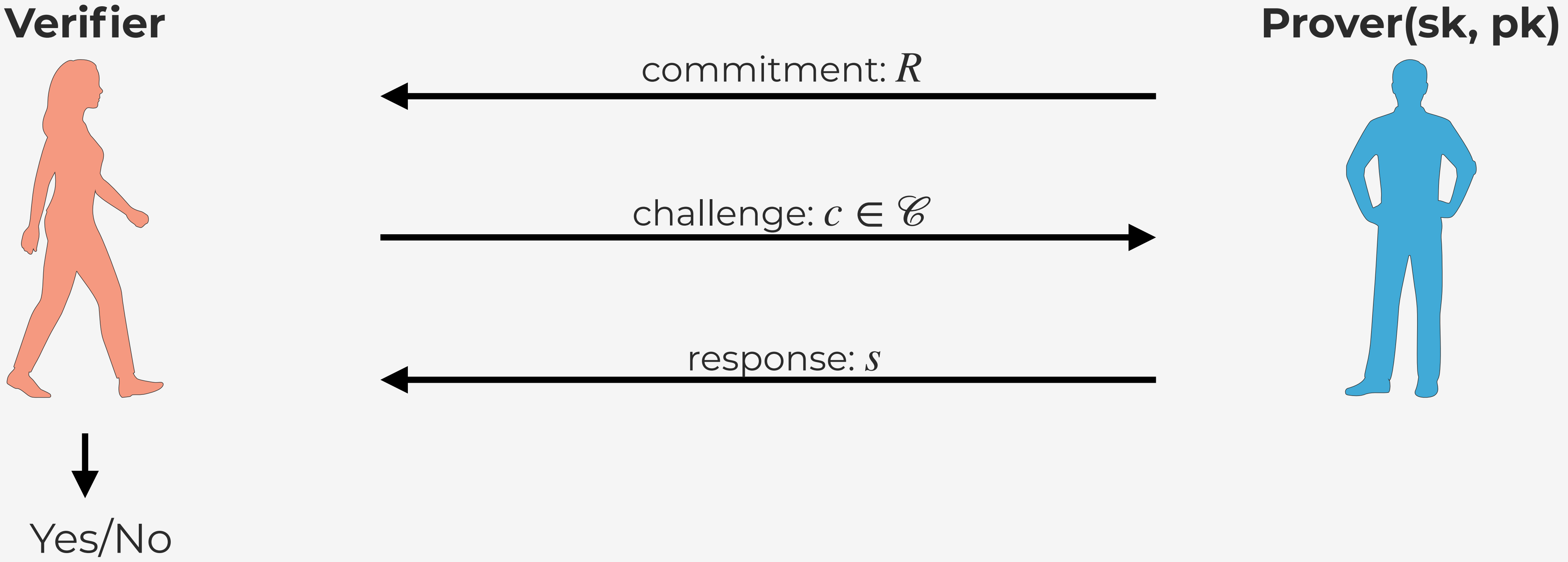


# Identification Schemes



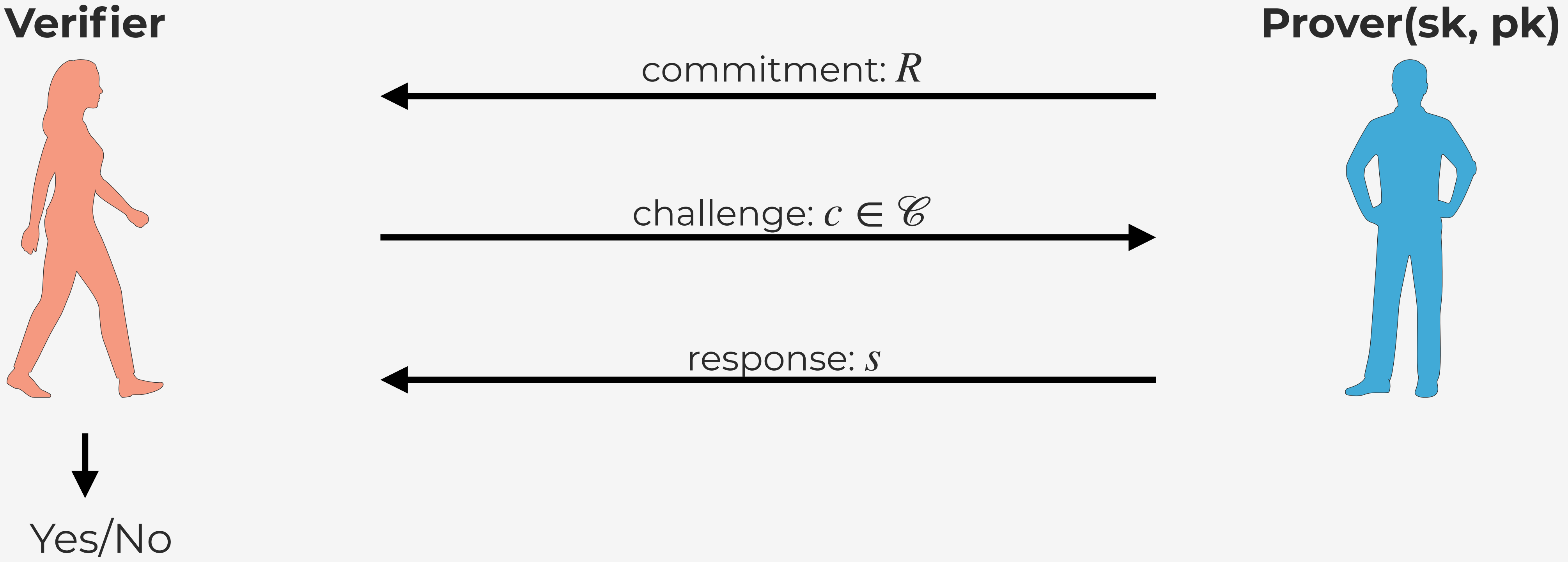

- **Correctness:** an honest prover always succeeds

# Identification Schemes



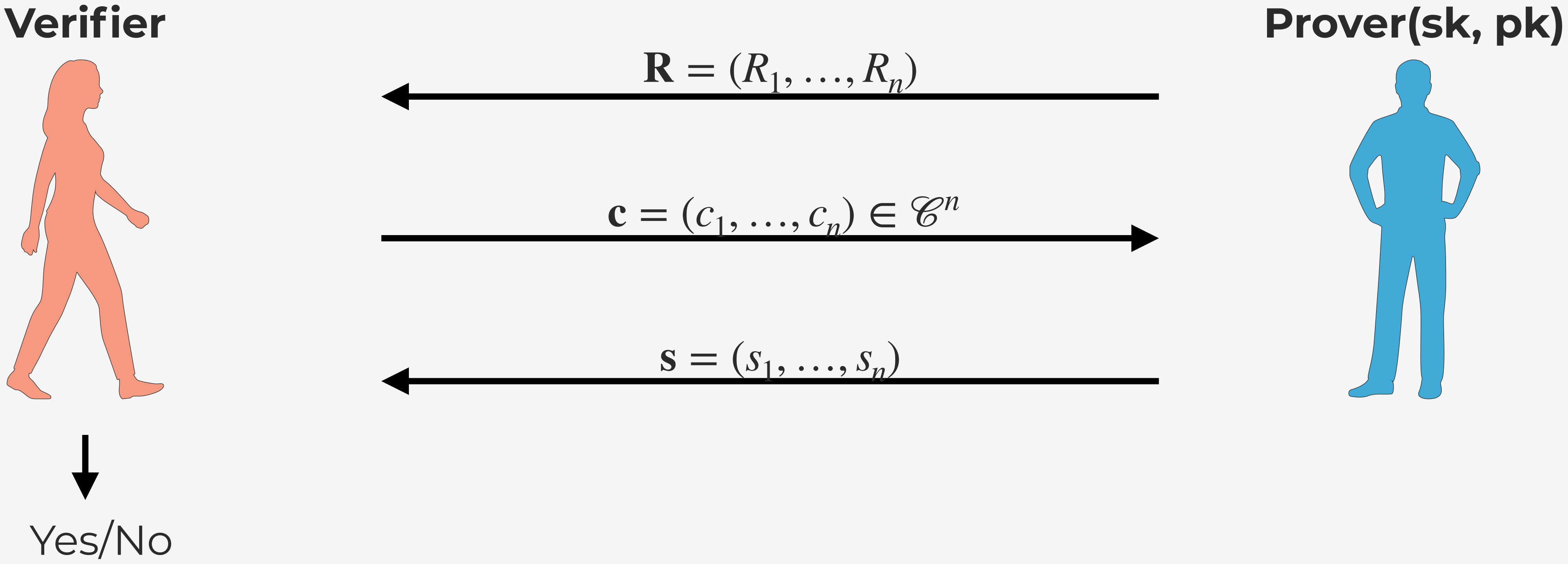

- **Correctness:** an honest prover always succeeds
- **Soundness:** a dishonest prover succeeds with probability $1/|\mathscr{C}|$

# Identification Schemes




- **Correctness:** an honest prover always succeeds
- **Soundness:** a dishonest prover succeeds with probability $1/|\mathscr{C}|$
- **HVZK:** there exists a simulator that, given a challenge $c \in \mathscr{C}$ outputs a valid transcript of the protocol

# Parallel Repetitions




If $\mathscr{C}$ is small then repeat the protocol $n$ times to increase security: now the cheating probability of a dishonest prover is $1/|\mathscr{C}|^n$

# Fiat-Shamir Transform

We replace the interaction with the verifier with a call of a random oracle $\mathscr{H} : \{0,1\}^* \rightarrow \mathscr{C}$

**Sign**

$m \rightarrow$, $sk \rightarrow$

1. $R \leftarrow \mathrm{commit}(sk)$
2. $c \leftarrow \mathscr{H}(R, m)$
3. $s \leftarrow \mathrm{resp}(R, c, sk)$

$\rightarrow (R, s)$

# Parallel Repetitions

If $|\mathscr{C}|$ is small, then repeat the protocol $n$ times to increase security

**Sign**

Inputs: $m$, $sk$

1. $\mathbf{R} \leftarrow \text{commit}(sk)$
2. $\mathbf{c} \leftarrow \mathscr{H}(\mathbf{R}, m)$
3. $\mathbf{s} \leftarrow \text{resp}(\mathbf{R}, \mathbf{c}, sk)$

Output: $(\mathbf{R}, \mathbf{s})$

# Schnorr-type Blind Signatures

**User(pk, m)**

**Signer(pk, sk)**

$R \leftarrow \text{commit}(sk)$

$\xleftarrow{\quad R \quad}$

$c \leftarrow \mathscr{H}(R, m)$

$\xrightarrow{\quad c \quad}$

$s \leftarrow \text{resp}(R, c, sk)$

$\xleftarrow{\quad s \quad}$

$\downarrow$

$(R, s)$

# Schnorr-type Blind Signatures

**User(pk, m)**

**Signer(pk, sk)**

$R \leftarrow \text{commit}(sk)$

$R$ (Signer → User)

$R' \leftarrow \text{blind}(R)$

$c' \leftarrow \mathcal{H}(R', m)$

$c \leftarrow \text{blind}(c')$

$c$ (User → Signer)

$s \leftarrow \text{resp}(R, c, sk)$

$s$ (Signer → User)

$s' \leftarrow \text{unblind}(s)$

↓

$(R', s')$

# Schnorr-type Blind Signatures with Repetitions

**User(pk, m)**

**Signer(pk, sk)**

$\mathbf{R} \leftarrow \mathsf{commit}(sk)$

$\mathbf{R}$

$\mathbf{R}' \leftarrow \mathsf{blind}(\mathbf{R})$

$\mathbf{c}' \leftarrow \mathscr{H}(\mathbf{R}', m)$

$\mathbf{c} \leftarrow \mathsf{blind}(\mathbf{c}')$

$\mathbf{c}$

$\mathbf{s} \leftarrow \mathsf{resp}(\mathbf{R}, \mathbf{c}, sk)$

$\mathbf{s}$

$\mathbf{s}' \leftarrow \mathsf{unblind}(\mathbf{s})$

$(\mathbf{R}', \mathbf{s}')$

# The Attack

We construct an adversary $\mathscr{A}$ against the **one more unforgeability** of a Schnorr-type blind signature:

- small base challenge space $\mathscr{C}$ (polynomial in the security parameter $n$)
- $n$ parallel repetitions

# The Attack

We construct an adversary $\mathscr{A}$ against the **one more unforgeability** of a Schnorr-type blind signature:

- small base challenge space $\mathscr{C}$ (polynomial in the security parameter $n$)
- $n$ parallel repetitions

I. **n-out-of-n**: $n + 1$ signatures after $n$ concurrent sessions

# The Attack

We construct an adversary $\mathscr{A}$ against the **one more unforgeability** of a Schnorr-type blind signature:

- small base challenge space $\mathscr{C}$ (polynomial in the security parameter $n$)
- $n$ parallel repetitions

I. **n-out-of-n**: $n + 1$ signatures after $n$ concurrent sessions

II. **2-out-of-n**: $n + 1$ signatures after $n$ sessions for a scheme allowing at most two concurrent sessions

# The Attack

We construct an adversary $\mathscr{A}$ against the **one more unforgeability** of a Schnorr-type blind signature:

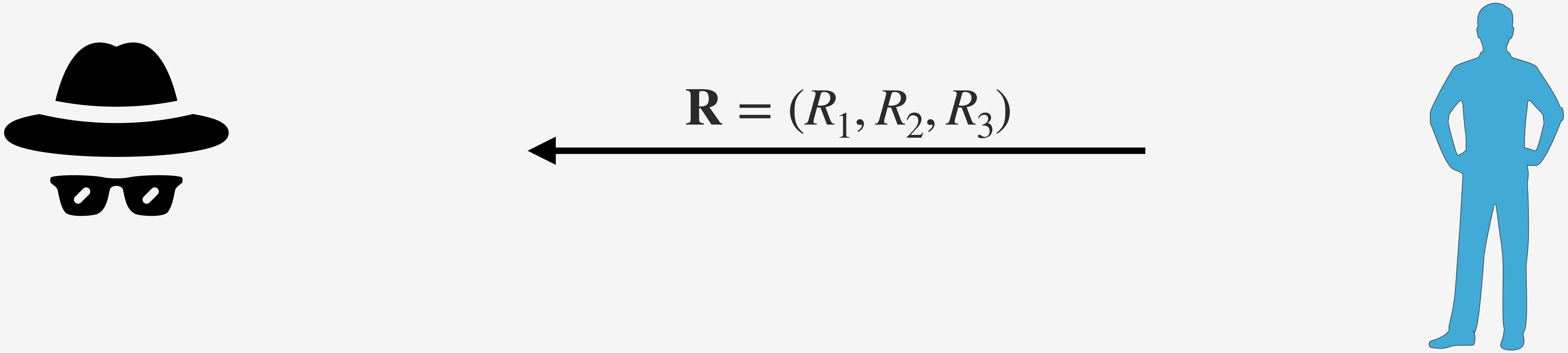
- small base challenge space $\mathscr{C}$ (polynomial in the security parameter $n$)
- $n$ parallel repetitions

I. **n-out-of-n**: $n+1$ signatures after $n$ concurrent sessions

II. **2-out-of-n**: $n+1$ signatures after $n$ sessions for a scheme allowing at most two concurrent sessions

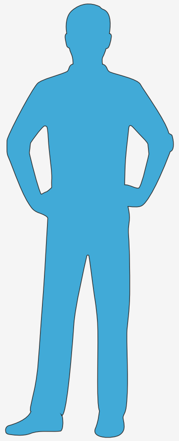
**Runtime**: $\mathcal{O}(n \cdot |\mathscr{C}|)$

# N-out-of-n (High Level, Unblind, n=3)



# N-out-of-n (High Level, Unblind, n=3)

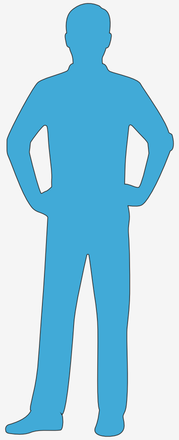
$\mathbf{R} = (R_1, R_2, R_3)$



- Simulate a valid transcript $(e, d, f)$ and replace $\mathbf{R}$ with $(e, R_2, R_3)$
- Find $m$ such that $\mathbf{c} = \mathcal{H}(m, (e, R_2, R_3)) = (d, c_2, c_3)$
- Requires $\mathcal{O}(|\mathcal{C}|)$ queries

# N-out-of-n (High Level, Unblind, n=3)

$\mathbf{R} = (R_1, R_2, R_3)$ ←



- Simulate a valid transcript $(e, d, f)$ and replace $\mathbf{R}$ with $(e, R_2, R_3)$
- Find $m$ such that $\mathbf{c} = \mathscr{H}(m, (e, R_2, R_3)) = (d, c_2, c_3)$
- Requires $\mathcal{O}(|\mathscr{C}|)$ queries

$(*, c_2, c_3)$ →

$\mathbf{s}$ ←

**Advantage**: gets one additional response for any challenge involving $R_1$

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | $R_{1,1}$ | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | $R_{2,2}$ | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | $R_{3,3}$ |

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | $R_{1,1}$ | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | $R_{2,2}$ | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | $R_{3,3}$ |

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | |

| | | | |
|---|---|---|---|
| Forgery | $R_{1,1}$ | $R_{2,2}$ | $R_{3,3}$ |

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | |

| | | | |
|---|---|---|---|
| Forgery | $R_{1,1}$ | $R_{2,2}$ | $R_{3,3}$ |

$$\mathscr{H}(m^*, (R_{1,1}, R_{2,2}, R_{3,3})) = (c_{4,1}, c_{4,2}, c_{4,3})$$

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | $e_1$ | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | $e_2$ | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | $e_3$ |

| | | | |
|---|---|---|---|
| Forgery | $R_{1,1}$ | $R_{2,2}$ | $R_{3,3}$ |

$$\mathscr{H}(m^*, (R_{1,1}, R_{2,2}, R_{3,3})) = (c_{4,1}, c_{4,2}, c_{4,3})$$

- Generate $(e_i, d_i, f_i)$ transcripts for $i = 1,2,3$

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | $e_1$ | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | $e_2$ | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | $e_3$ |

| | | | |
|---|---|---|---|
| Forgery | $R_{1,1}$ | $R_{2,2}$ | $R_{3,3}$ |

$$\mathbf{c}_1 = \mathscr{H}(m_1, (e_1, R_{1,2}, R_{1,3})) = (d_1, *, *)$$

$$\mathbf{c}_2 = \mathscr{H}(m_2, (R_{2,1}, e_2, R_{2,3})) = (*, d_2, *)$$

$$\mathbf{c}_3 = \mathscr{H}(m_3, (R_{3,1}, R_{3,2}, e_3)) = (*, *, d_3)$$

$$\mathbf{c}_4 = \mathscr{H}(m^*, (R_{1,1}, R_{2,2}, R_{3,3})) = (c_{4,1}, c_{4,2}, c_{4,3})$$

- Generate $(e_i, d_i, f_i)$ transcripts for $i = 1,2,3$
- Find $m_i$ for $i = 1,2,3$

# N-out-of-n (High Level, Unblind, n=3)

| | | | |
|---|---|---|---|
| 1st | $e_1$ | $R_{1,2}$ | $R_{1,3}$ |
| 2nd | $R_{2,1}$ | $e_2$ | $R_{2,3}$ |
| 3rd | $R_{3,1}$ | $R_{3,2}$ | $e_3$ |
| Forgery | $R_{1,1}$ | $R_{2,2}$ | $R_{3,3}$ |

$$\mathbf{c}_1 = \mathscr{H}(m_1, (e_1, R_{1,2}, R_{1,3})) = (d_1, *, *)$$

$$\mathbf{c}_2 = \mathscr{H}(m_2, (R_{2,1}, e_2, R_{2,3})) = (*, d_2, *)$$

$$\mathbf{c}_3 = \mathscr{H}(m_3, (R_{3,1}, R_{3,2}, e_3)) = (*, *, d_3)$$

$$\mathbf{c}_4 = \mathscr{H}(m^*, (R_{1,1}, R_{2,2}, R_{3,3})) = (c_{4,1}, c_{4,2}, c_{4,3})$$

- Generate $(e_i, d_i, f_i)$ transcripts for $i = 1,2,3$
- Find $m_i$ for $i = 1,2,3$
- Send the signer: $(c_{4,1}, c_{1,2}, c_{1,3})$, $(c_{2,1}, c_{4,2}, c_{2,3})$, $(c_{3,1}, c_{3,2}, c_{4,3})$ and receive the responses

# N-out-of-n Generalization

Find a message $m$ such that $\mathscr{H}(m, \mathbf{R}) = (d, c_2, \ldots, c_n)$, requires $\mathcal{O}(|\mathscr{C}|)$ queries and $n$ sessions

# N-out-of-n Generalization

Find a message $m$ such that $\mathscr{H}(m, \mathbf{R}) = (d, c_2, \ldots, c_n)$, requires $\mathcal{O}(|\mathscr{C}|)$ queries and $n$ sessions

Find a message $m$ such that $\mathscr{H}(m, \mathbf{R}) = (d_1, d_2, \ldots, d_s, c_{s+1}, \ldots, c_n)$, requires $\mathcal{O}(|\mathscr{C}|^s)$ queries and $\lceil n/s \rceil$ sessions

# N-out-of-n Generalization

Find a message $m$ such that $\mathscr{H}(m, \mathbf{R}) = (d, c_2, \ldots, c_n)$, requires $\mathcal{O}(|\mathscr{C}|)$ queries and $n$ sessions

Find a message $m$ such that $\mathscr{H}(m, \mathbf{R}) = (d_1, d_2, \ldots, d_s, c_{s+1}, \ldots, c_n)$, requires $\mathcal{O}(|\mathscr{C}|^s)$ queries and $\lceil n/s \rceil$ sessions

**Runtime:** $\mathcal{O}(\lceil n/s \rceil \cdot |\mathscr{C}|^s)$

$\Longrightarrow$ trade-off between number of queries to $\mathscr{H}$ and number of sessions

# Conclusion

- Affected scheme: CSI-Otter [KLLQ23], the first isogeny-based blind signature scheme. Our attack is able to efficiently forge **129 valid signatures** after **128 concurrent sessions** with the signer

# Conclusion

- Affected scheme: CSI-Otter [KLLQ23], the first isogeny-based blind signature scheme. Our attack is able to efficiently forge **129 valid signatures** after **128 concurrent sessions** with the signer

- Impossibility result: Shnorr-type blind signatures with repetitions of a small challenge space **are not concurrently secure**

# Conclusion

- Affected scheme: CSI-Otter [KLLQ23], the first isogeny-based blind signature scheme. Our attack is able to efficiently forge **129 valid signatures** after **128 concurrent sessions** with the signer
- Impossibility result: Shnorr-type blind signatures with repetitions of a small challenge space **are not concurrently secure**
- To construct a potential secure blind signature following this paradigm we need a base identification scheme with (exponentially) **big challenge space**

# Contact Information

**Eugenio Paracucchi**

PhD Student @ CISPA

E-Mail:

eugenio.paracucchi@cispa.de