Sécurité Exacte des Schémas de Signature Forward-Secure

Michel Abdalla, Fabrice Ben Hamouda, David Pointcheval

Équipe Crypto CASCADE — ENS

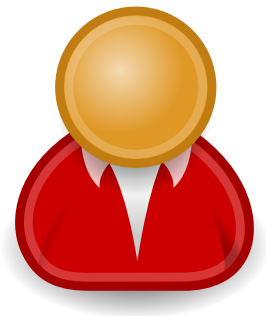
11 octobre 2012

Table of contents

Signature Schemes

$$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$$

M

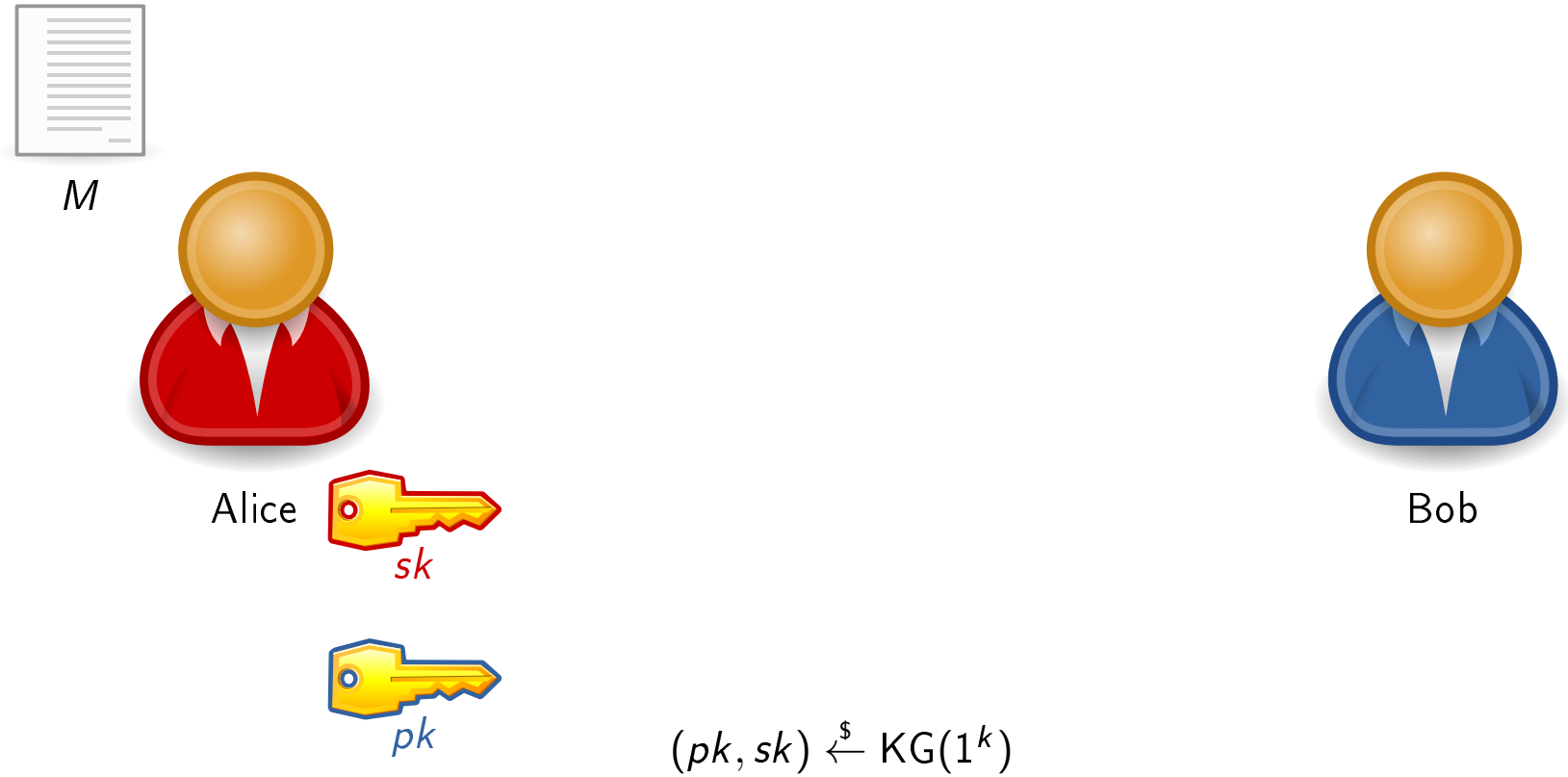


Alice

Bob

Signature Schemes

$$\mathcal{DS} = (\mathbf{KG}, \mathrm{Sign}, \mathrm{Ver})$$



$$(pk, sk) \xleftarrow{\$} \mathrm{KG}(1^k)$$

Signature Schemes

$$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$$

Signature Schemes

$$\mathcal{DS} = (\mathsf{KG}, \mathbf{Sign}, \mathsf{Ver})$$

$$\sigma \xleftarrow{\$} \mathsf{Sign}(sk, M)$$

Signature Schemes

$$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$$

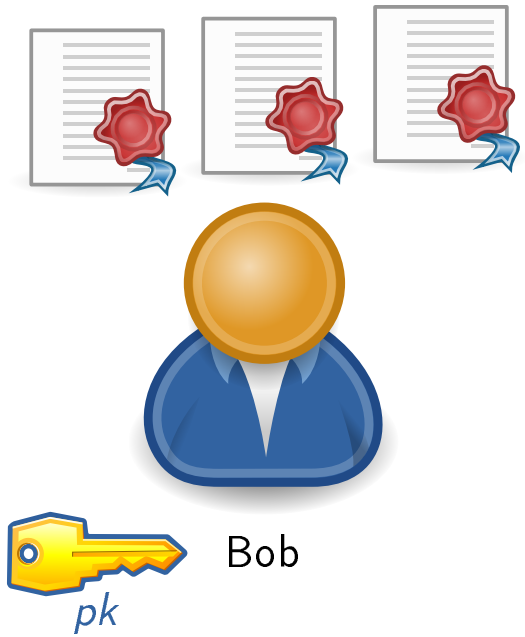
Signature Schemes

$$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathbf{Ver})$$

$$\mathsf{Ver}(pk, \sigma, M)$$

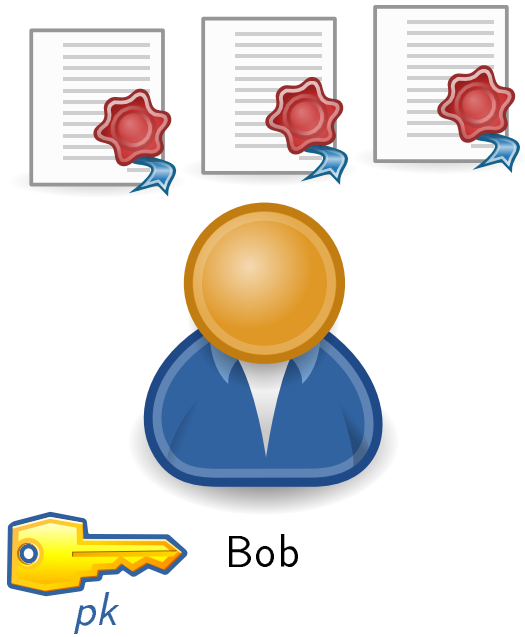
Signature Schemes

$$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$$



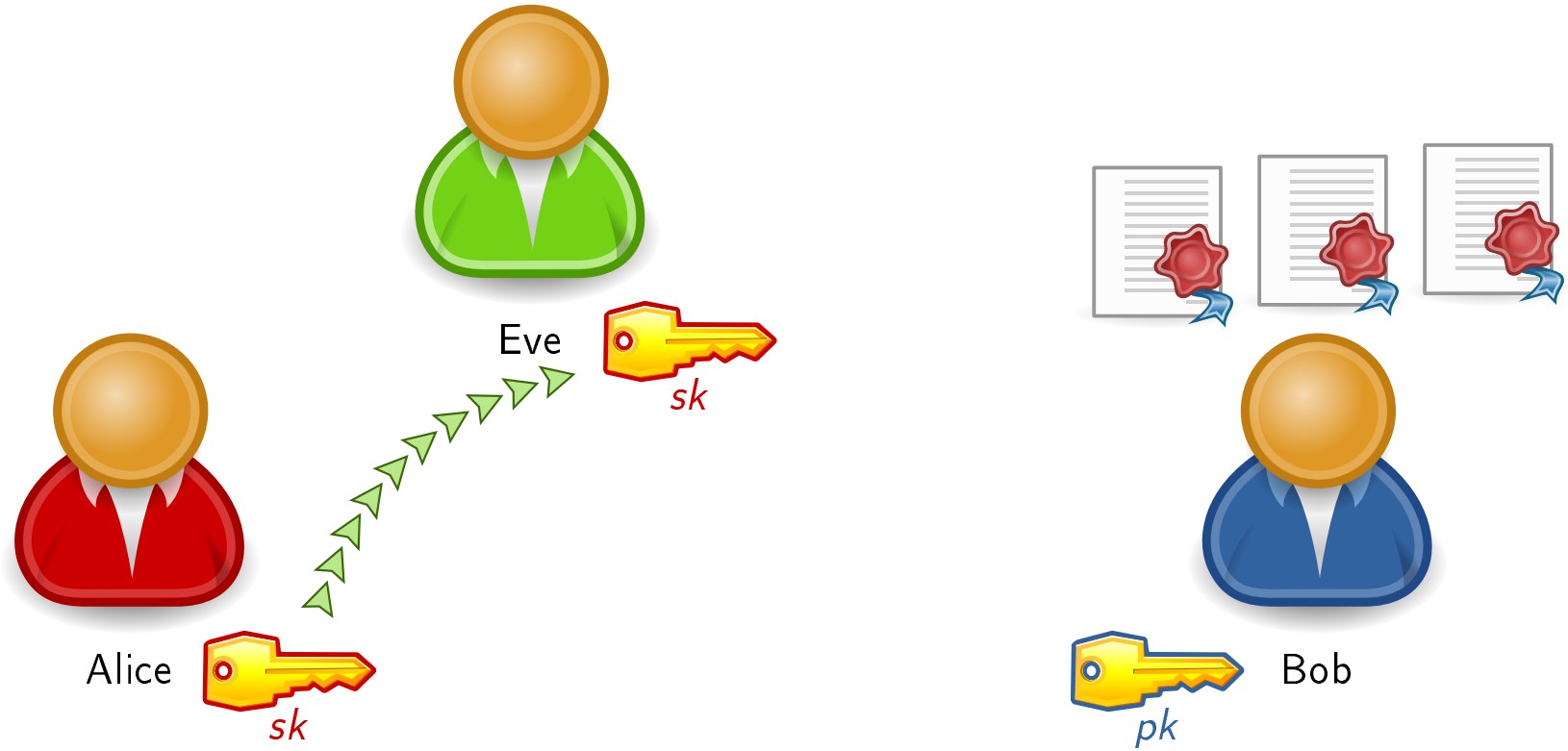
Signature Schemes: Problems with Corruption

$$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$$



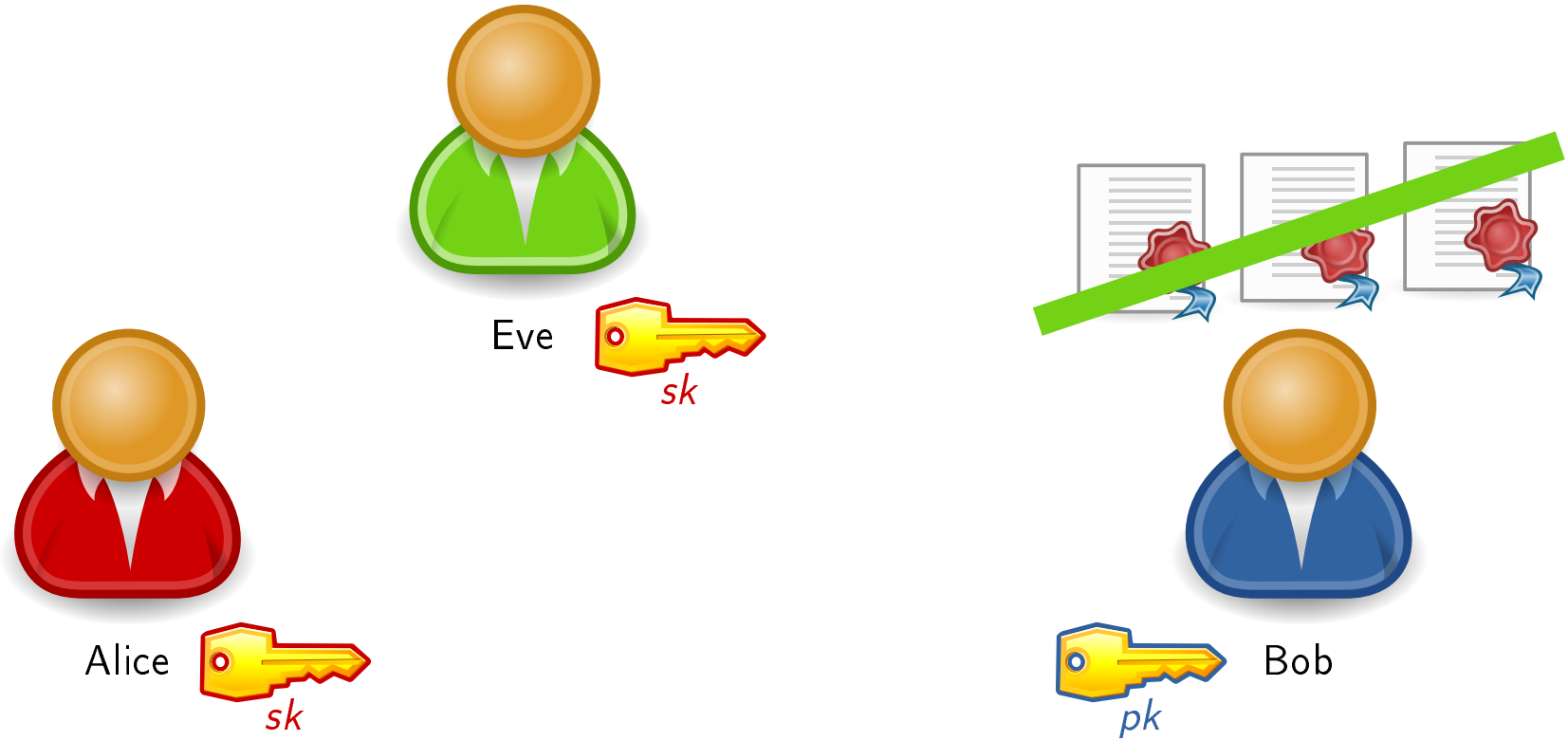
Signature Schemes: Problems with Corruption

$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$



Signature Schemes: Problems with Corruption

$\mathcal{DS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver})$



Forward-Secure Signature Schemes I

- key lifetime divided in T periods,
- a secret key sk_i for each period i,
- the same public key pk for all periods,
- sk_i can sign messages for period $\geq i$.

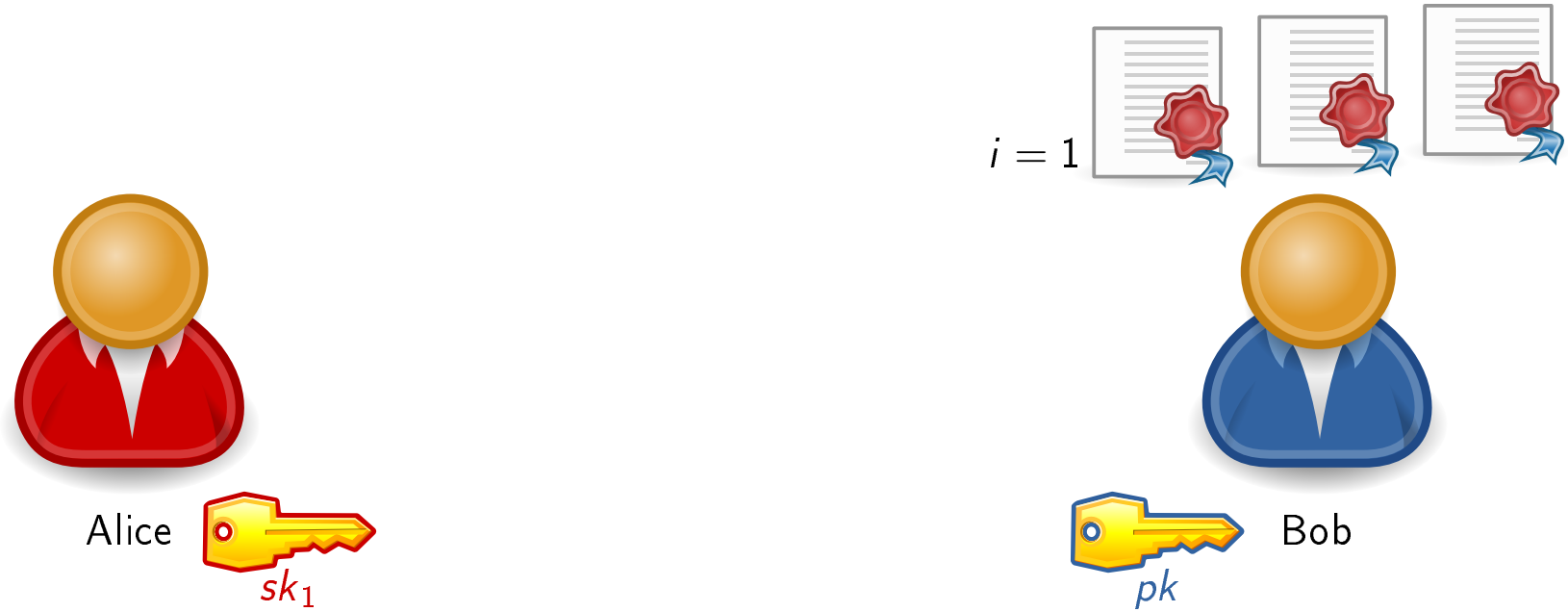
Forward-Secure Signature Schemes II

$$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$$

January ($i = 1$)

Forward-Secure Signature Schemes II

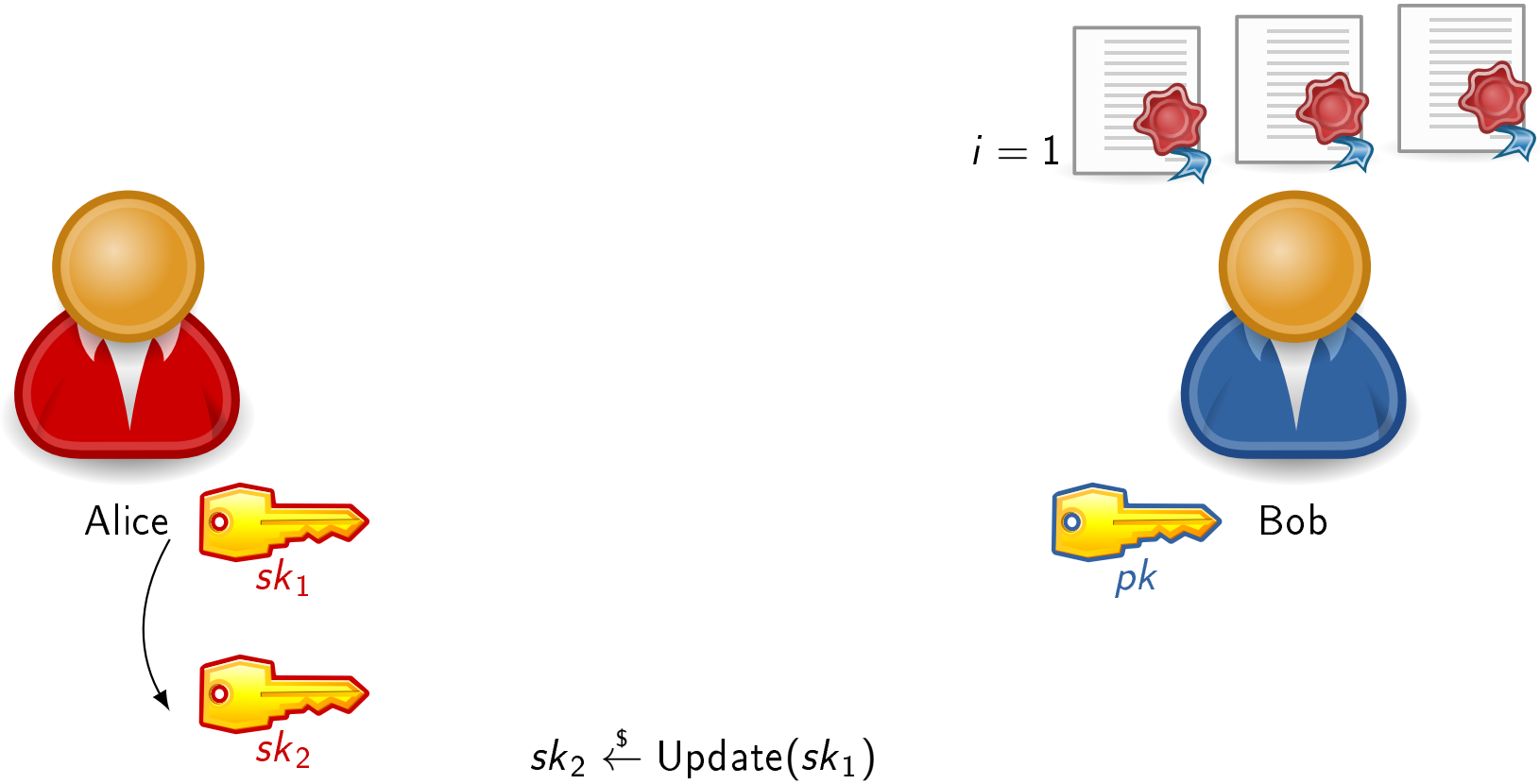
$$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$$



January ($i = 1$)

Forward-Secure Signature Schemes II

$$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \textbf{Update})$$

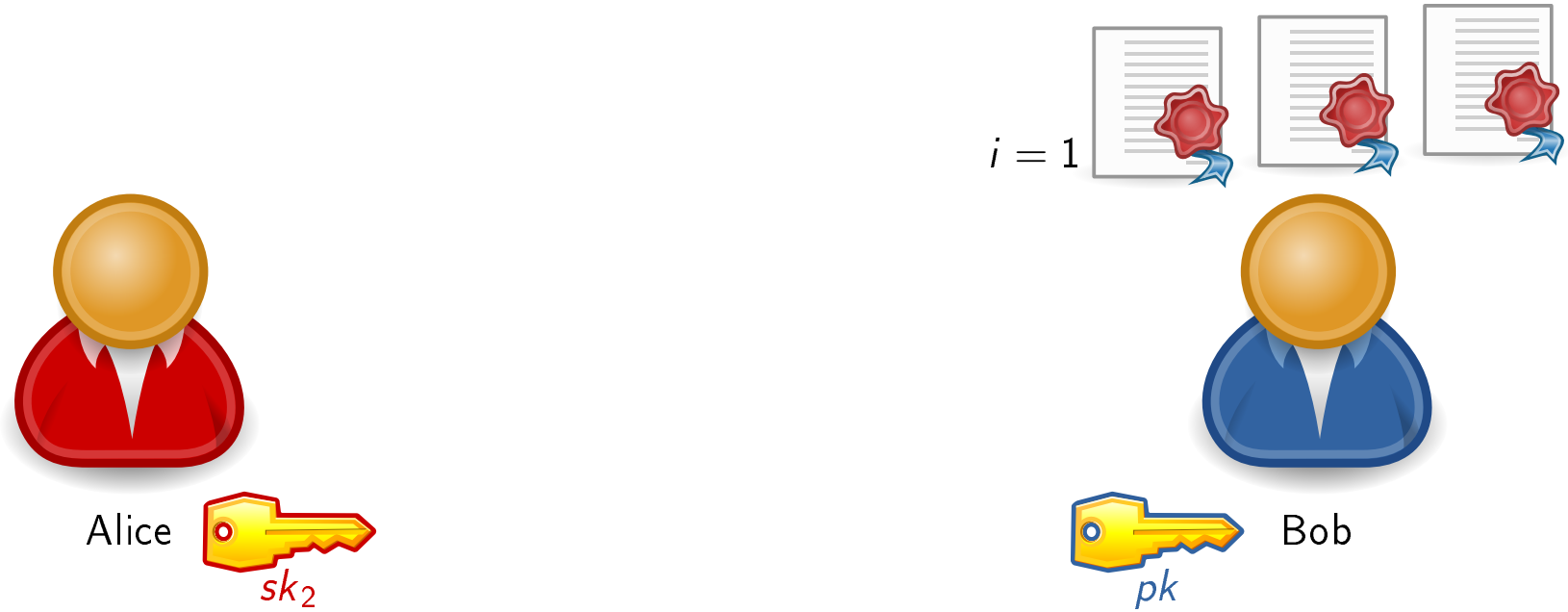


$$sk_2 \xleftarrow{\$} \mathsf{Update}(sk_1)$$

$$\text{January } (i = 1) \longrightarrow \text{February } (i = 2)$$

Forward-Secure Signature Schemes II

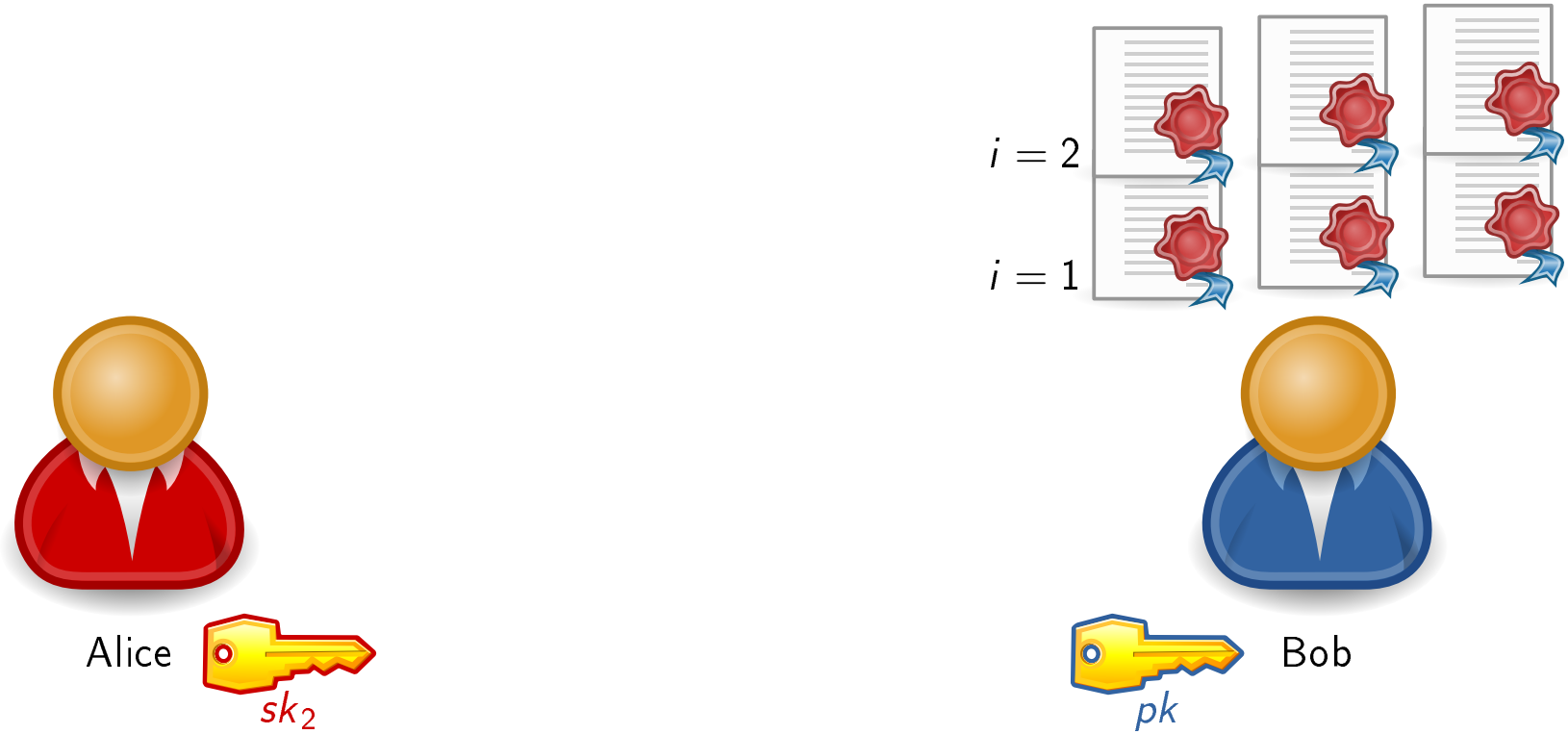
$$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$$



February ($i = 2$)

Forward-Secure Signature Schemes II

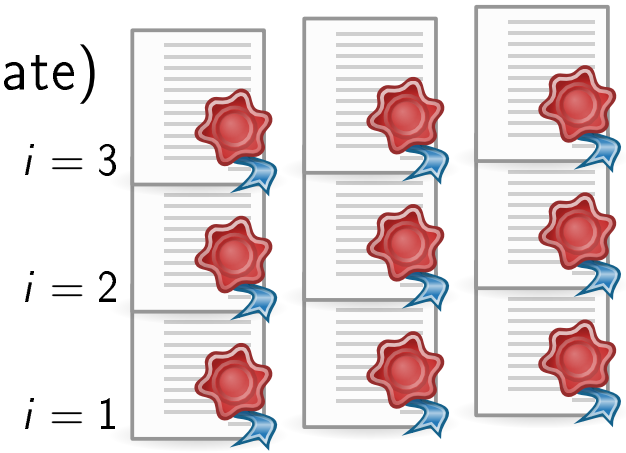
$$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$$

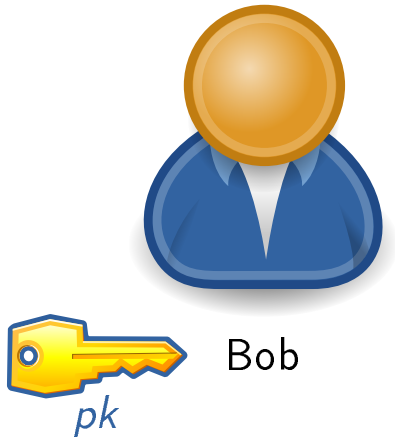


February ($i = 2$)

Forward-Secure Signature Schemes II

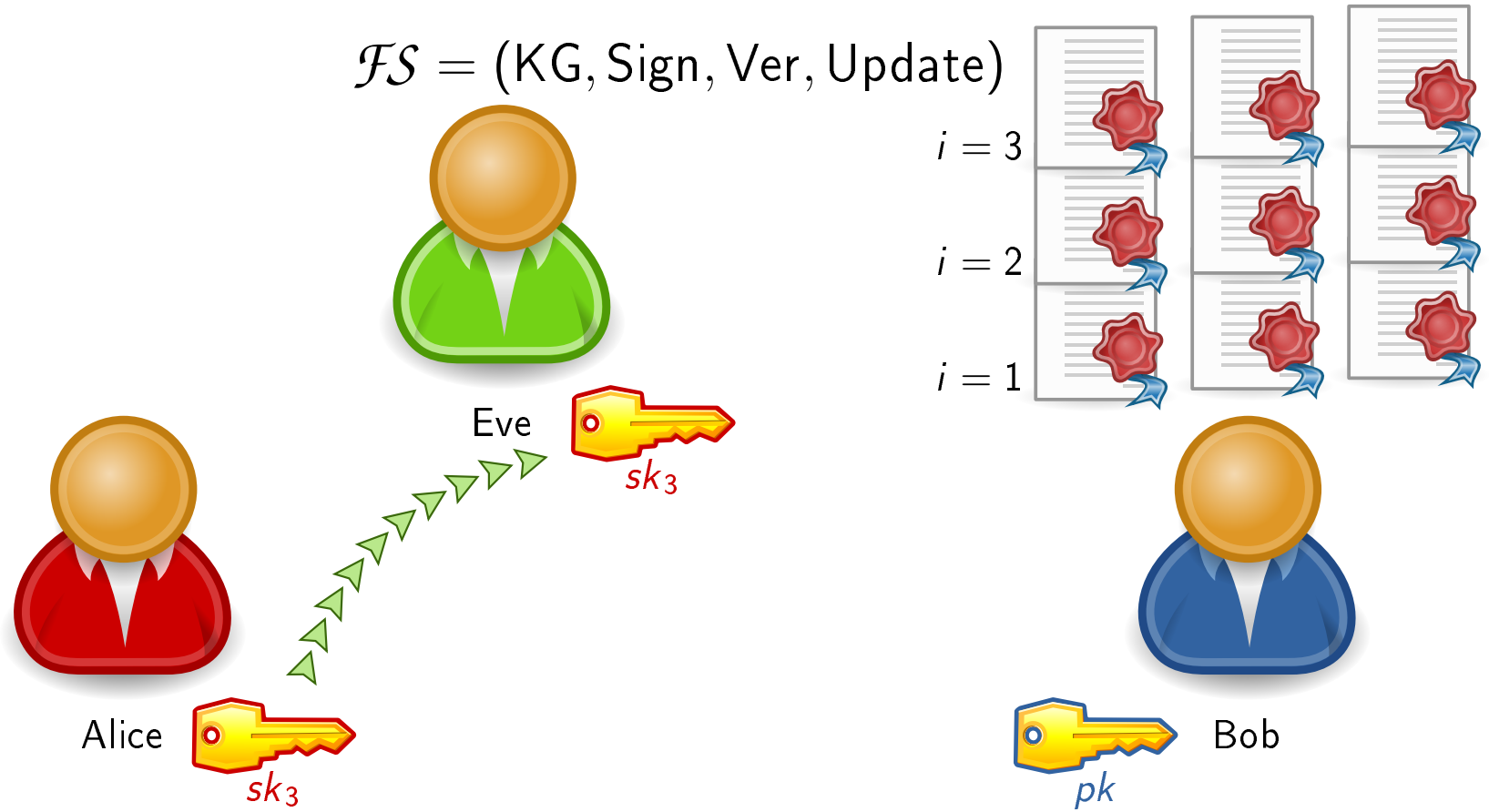
$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$





March ($i = 3$)

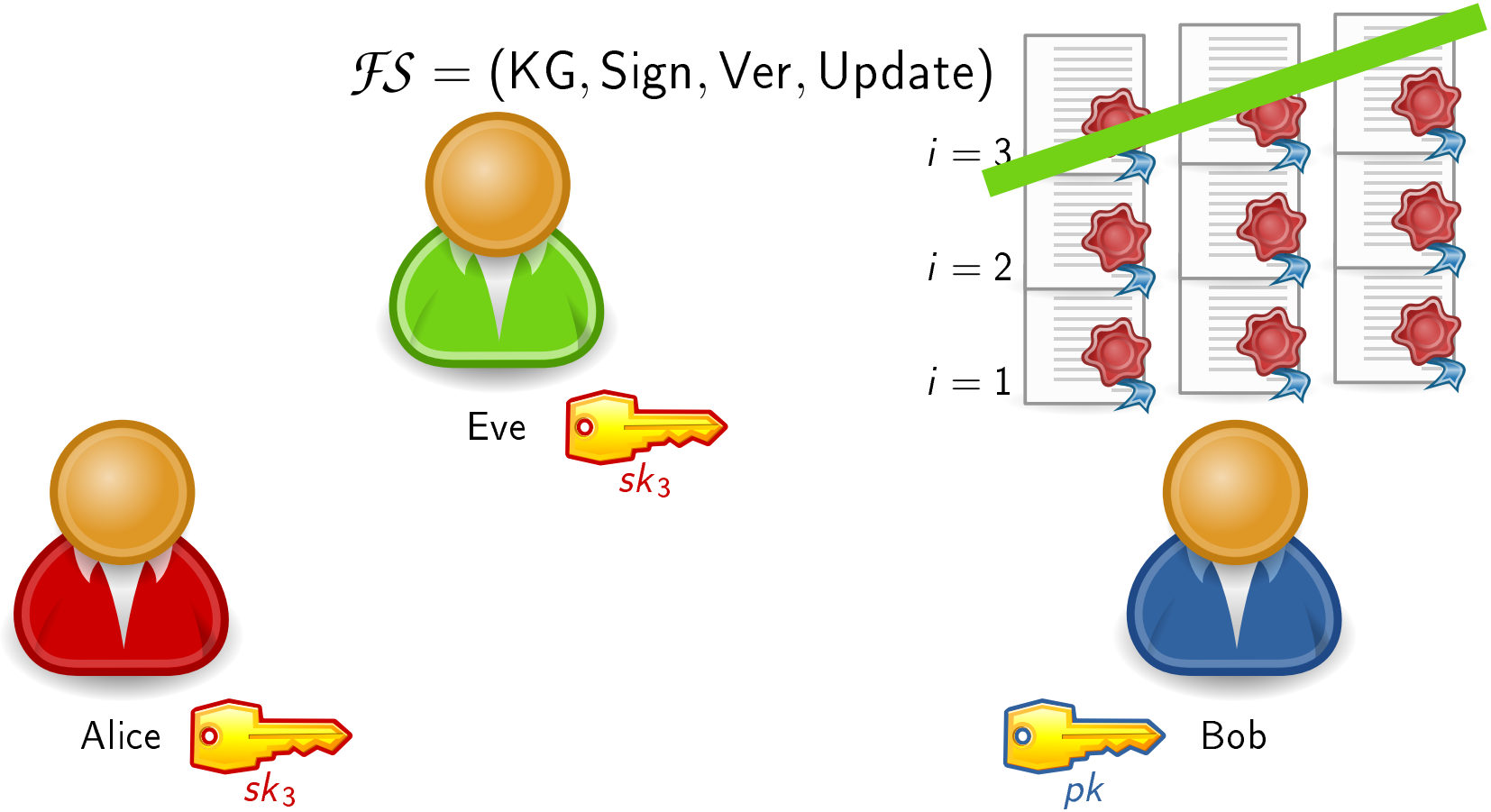
Forward-Secure Signature Schemes II



March ($i = 3$)

Forward-Secure Signature Schemes II

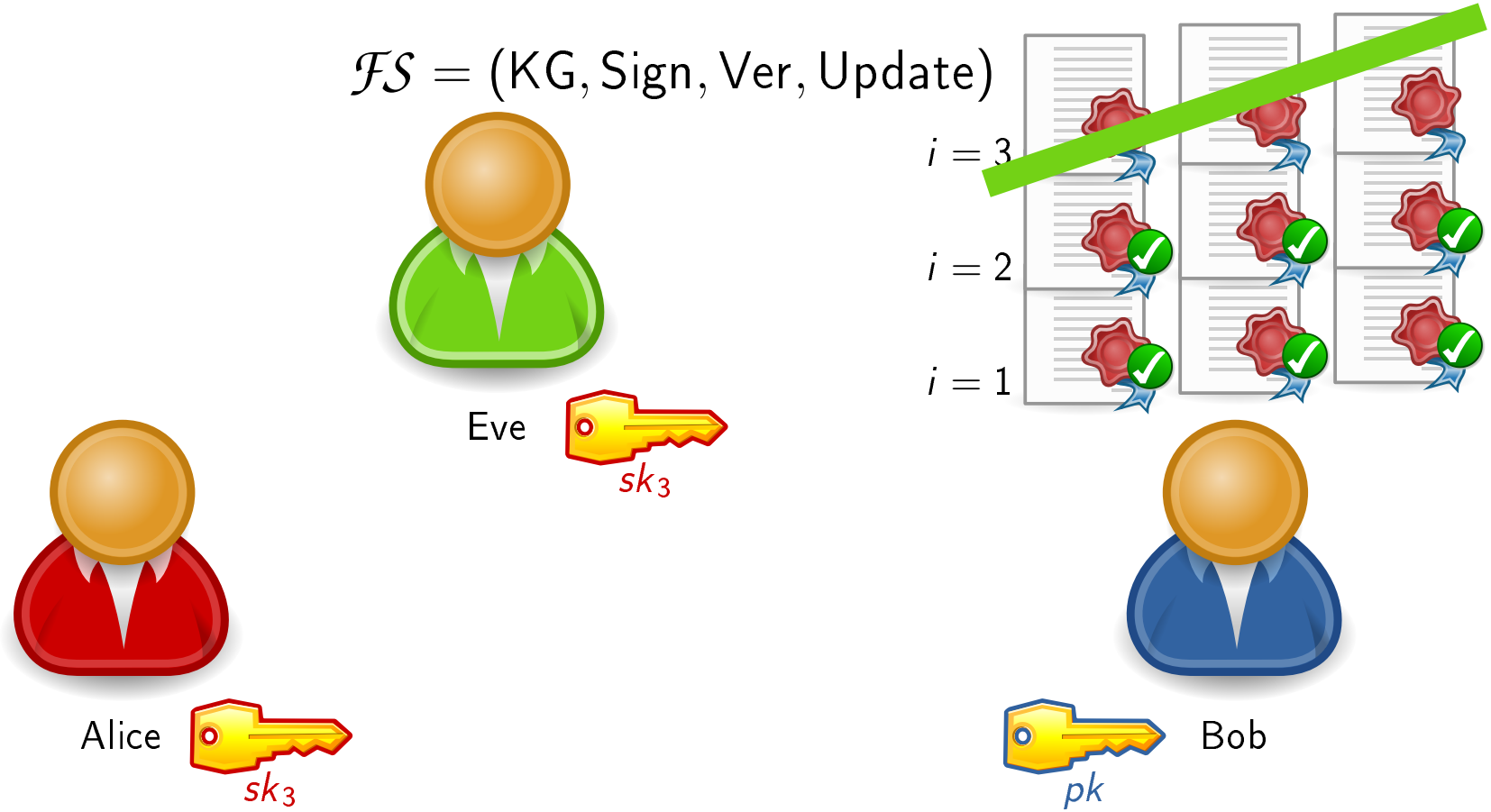
$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$



March ($i = 3$)

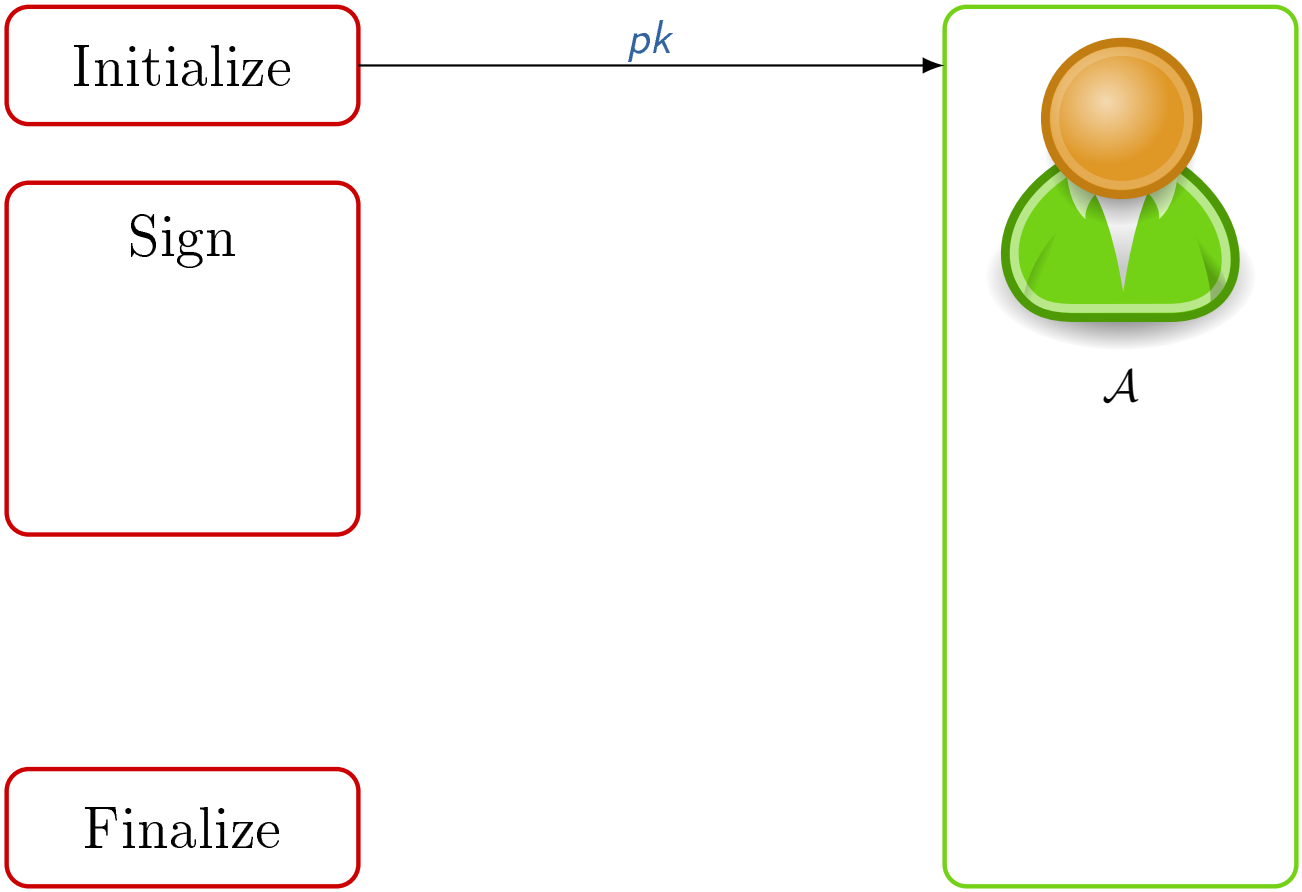
Forward-Secure Signature Schemes II

$\mathcal{FS} = (\mathsf{KG}, \mathsf{Sign}, \mathsf{Ver}, \mathsf{Update})$

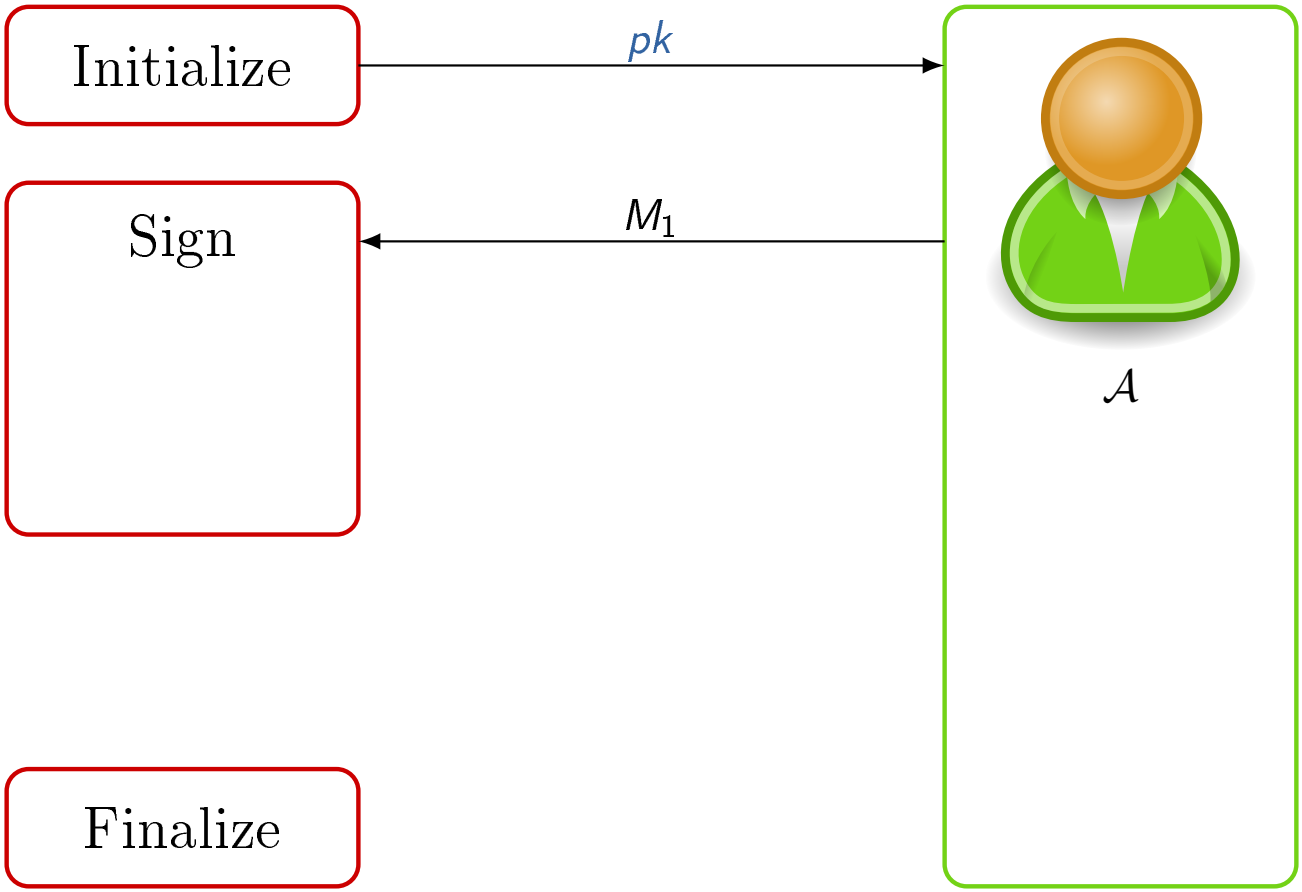


March ($i = 3$)

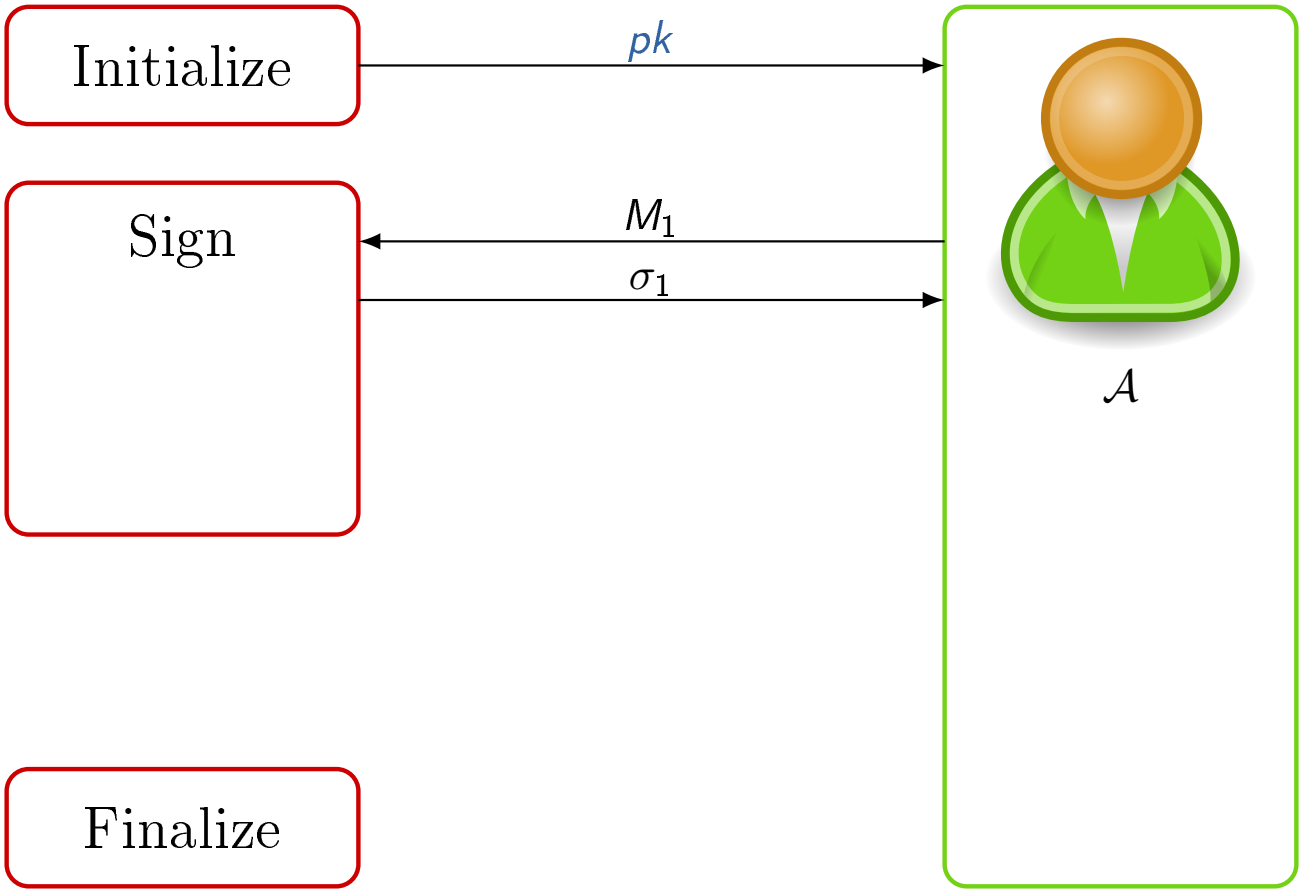
Security Goal: Existential Unforgeability



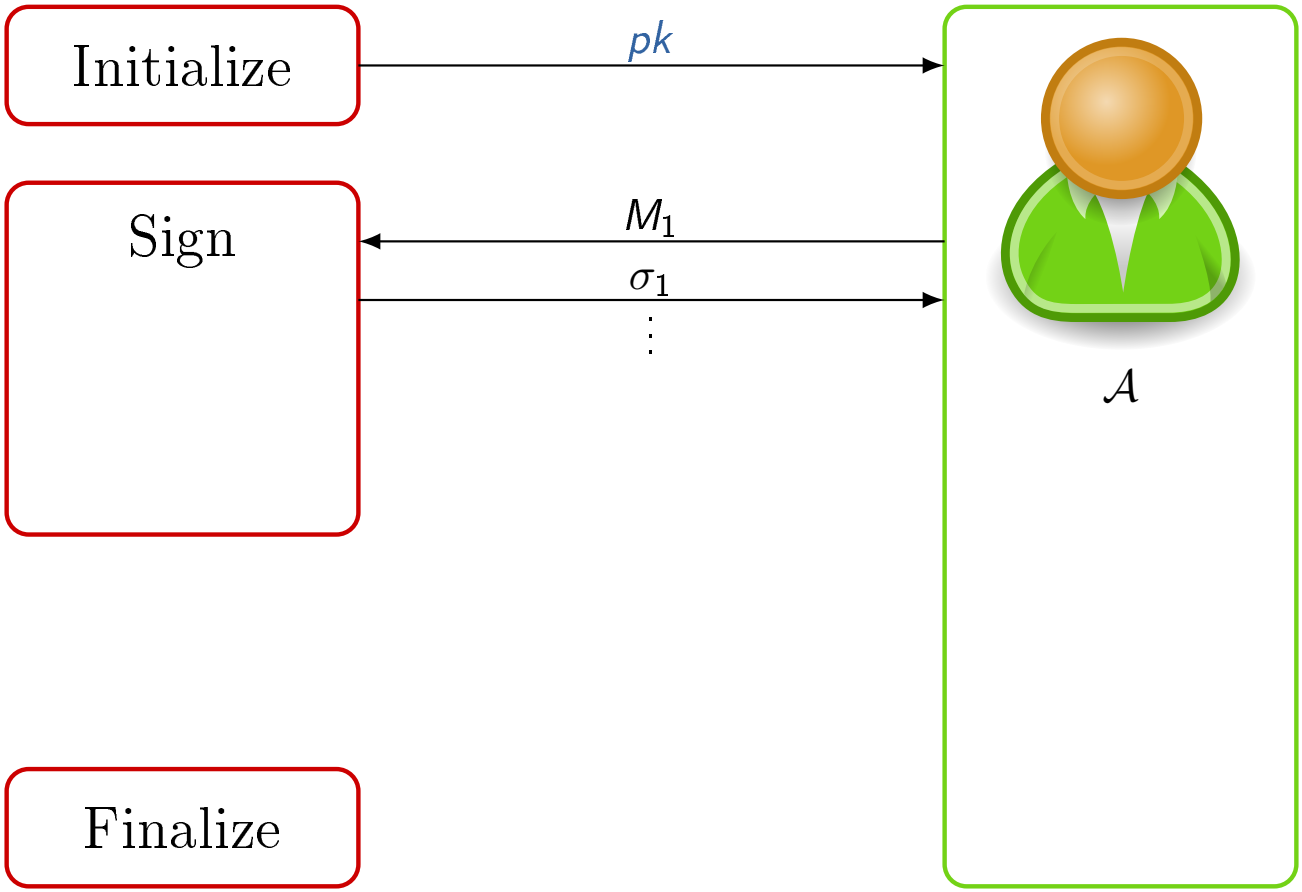
Security Goal: Existential Unforgeability



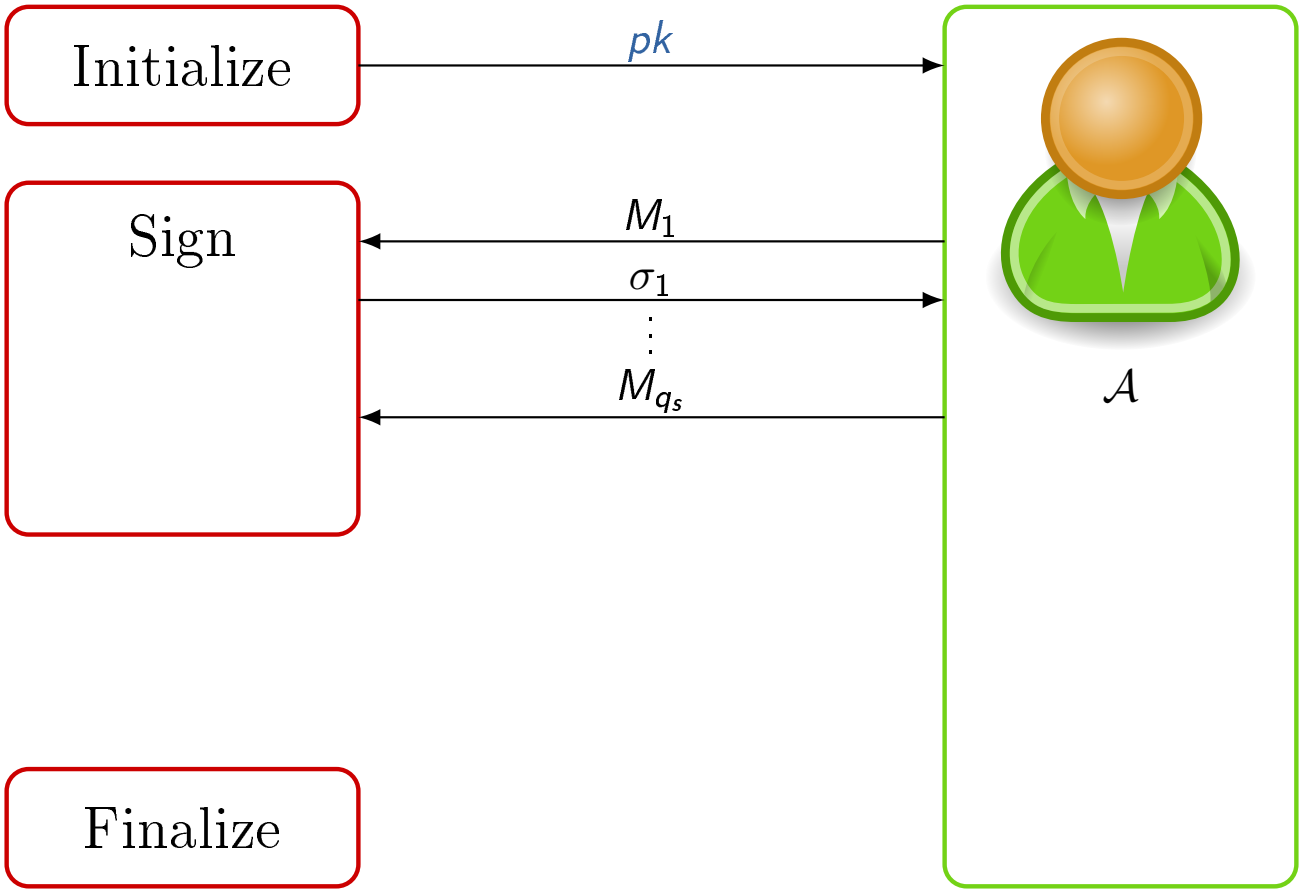
Security Goal: Existential Unforgeability



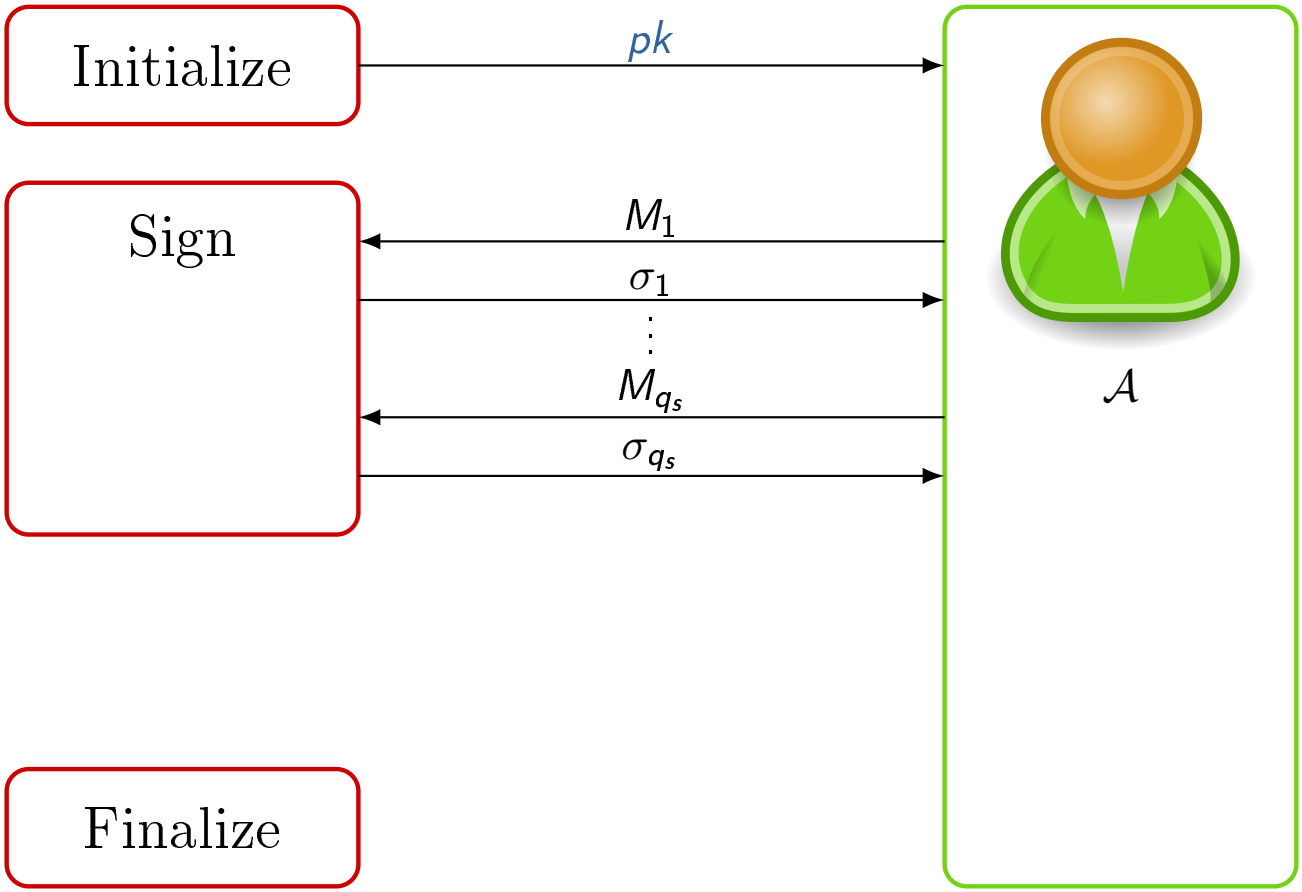
Security Goal: Existential Unforgeability



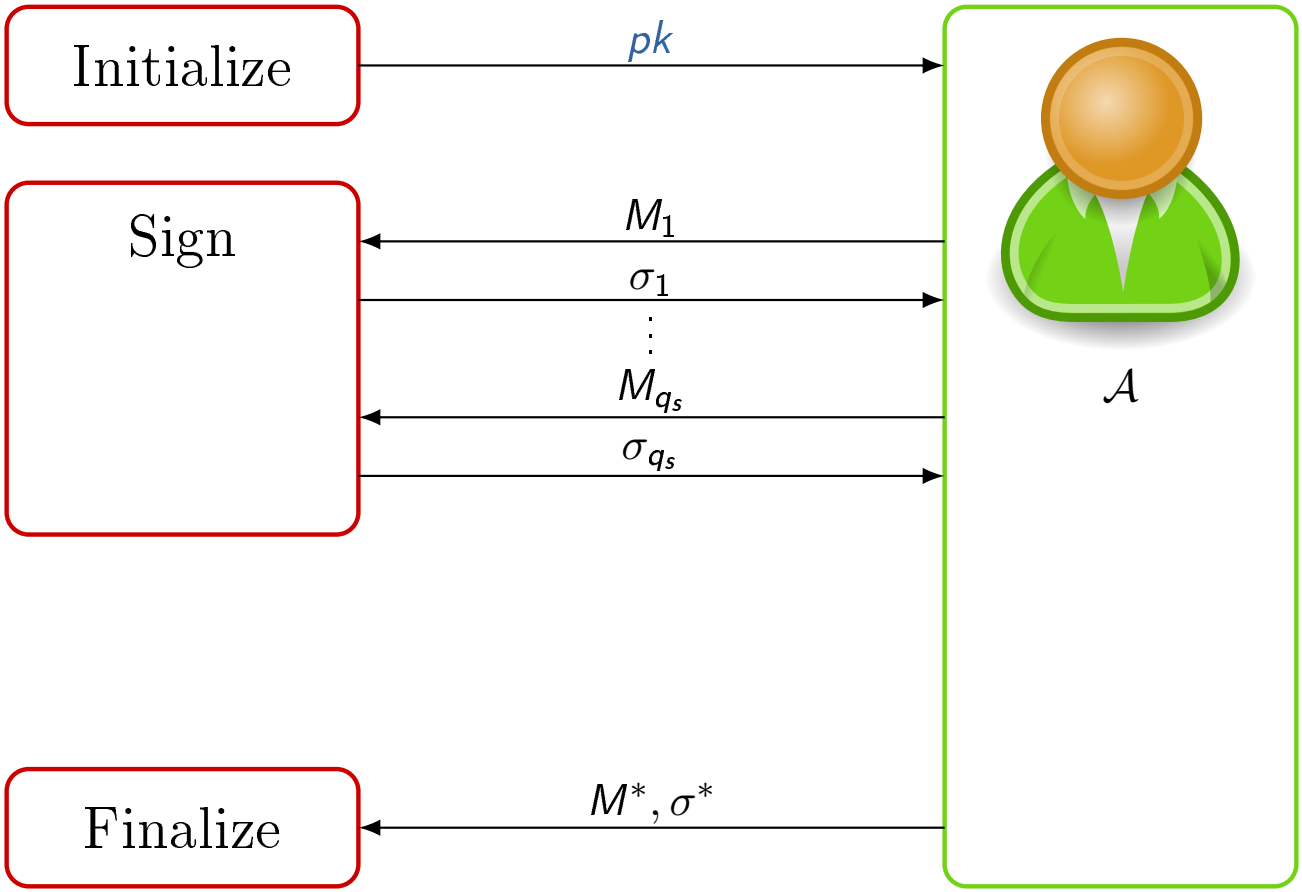
Security Goal: Existential Unforgeability



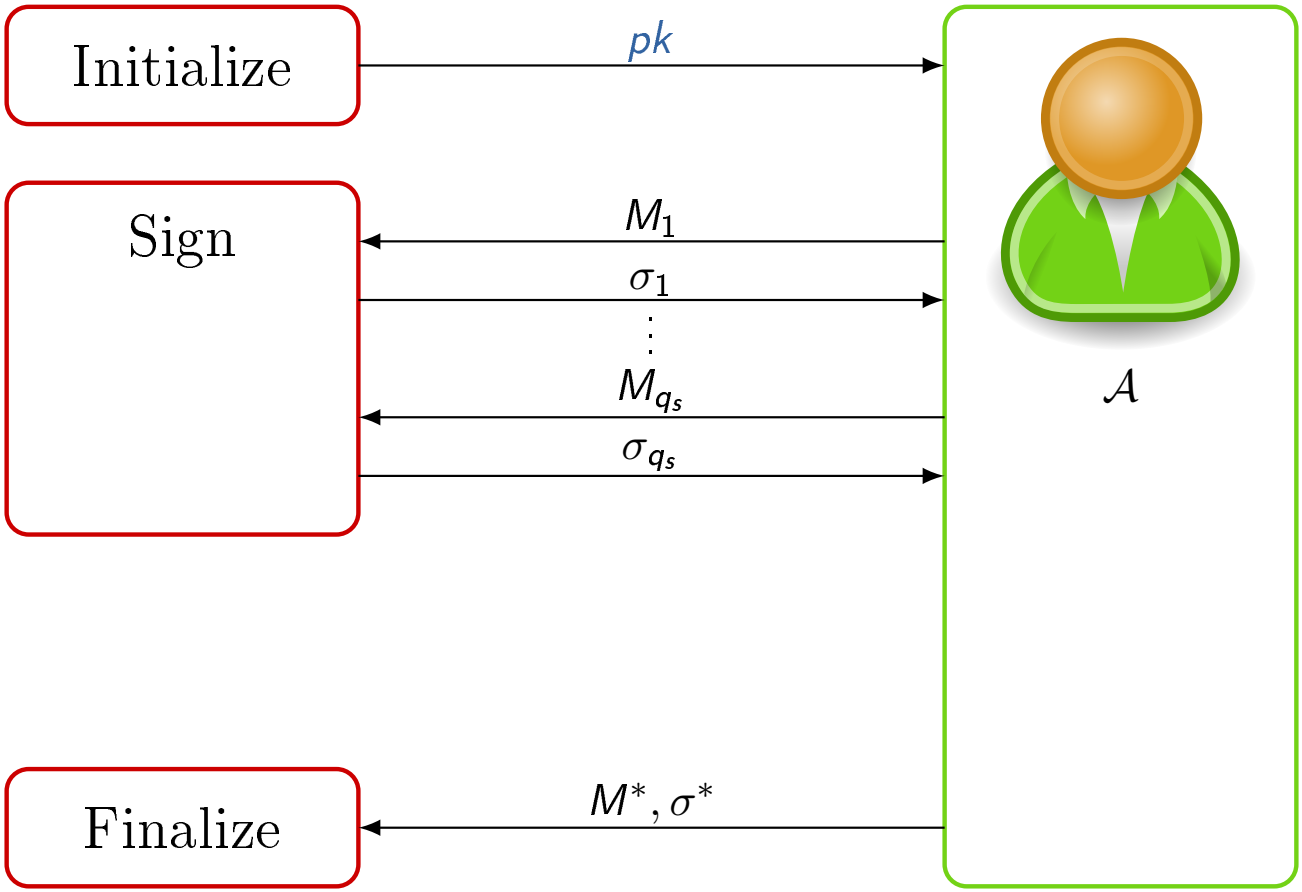
Security Goal: Existential Unforgeability



Security Goal: Existential Unforgeability



Security Goal: Existential Unforgeability

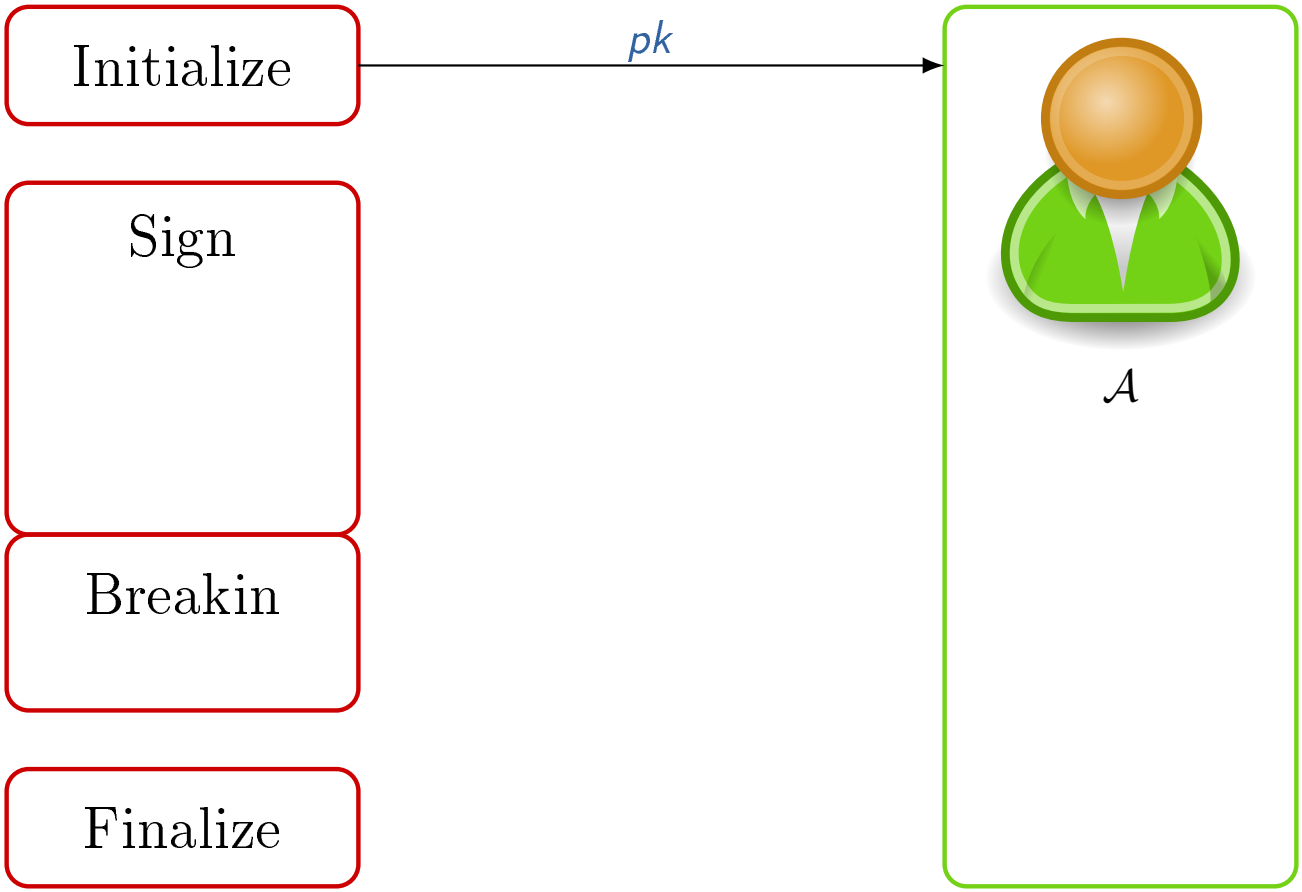


(t, ε)-**existential-unforgeability** (**EUF-CMA**):

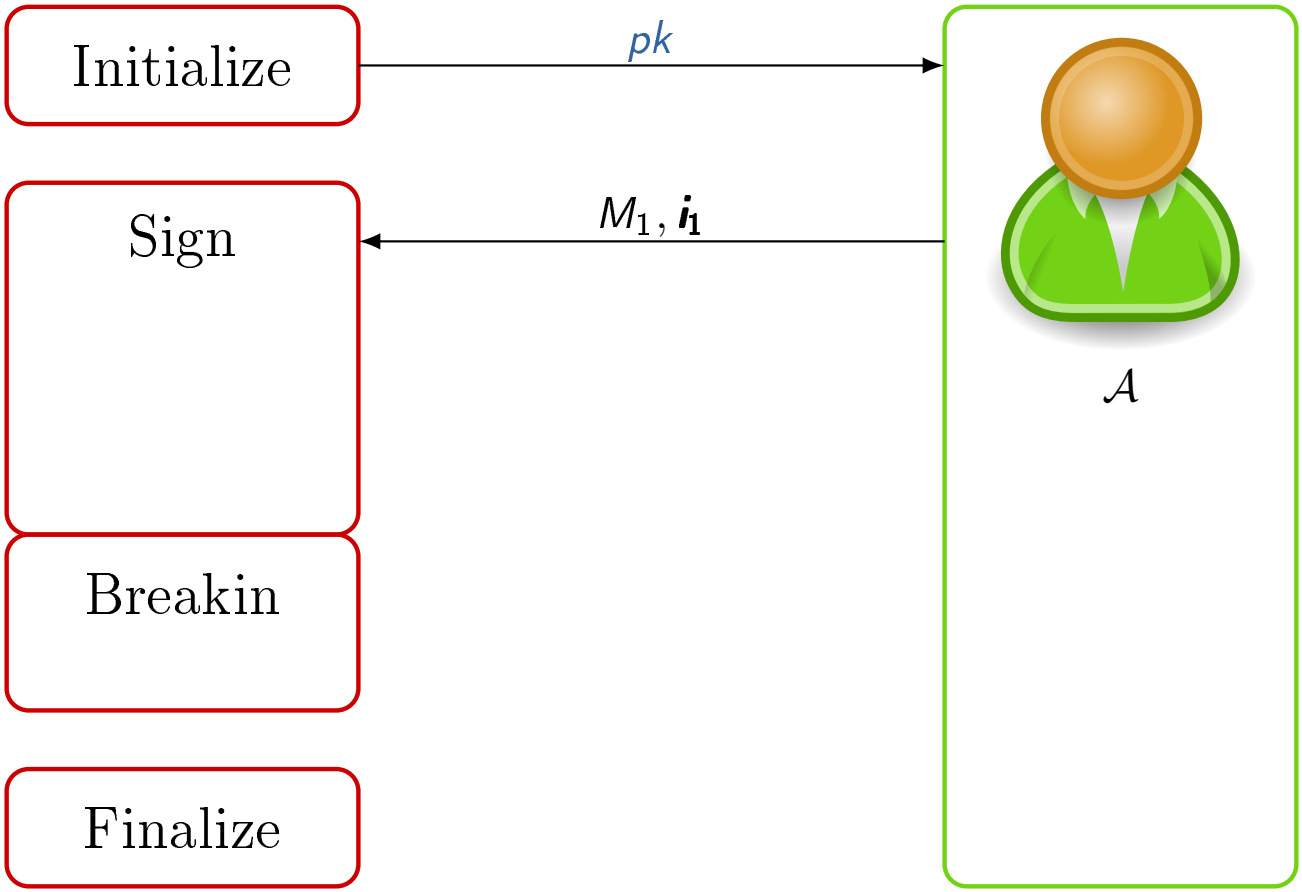
for any adversary $\mathcal{A}$ running in time at most t:

$$\Pr\left[\forall j, M^* \neq M_j \text{ and } \mathrm{Ver}(pk, \sigma^*, M^*) = 1\right] \leq \varepsilon.$$

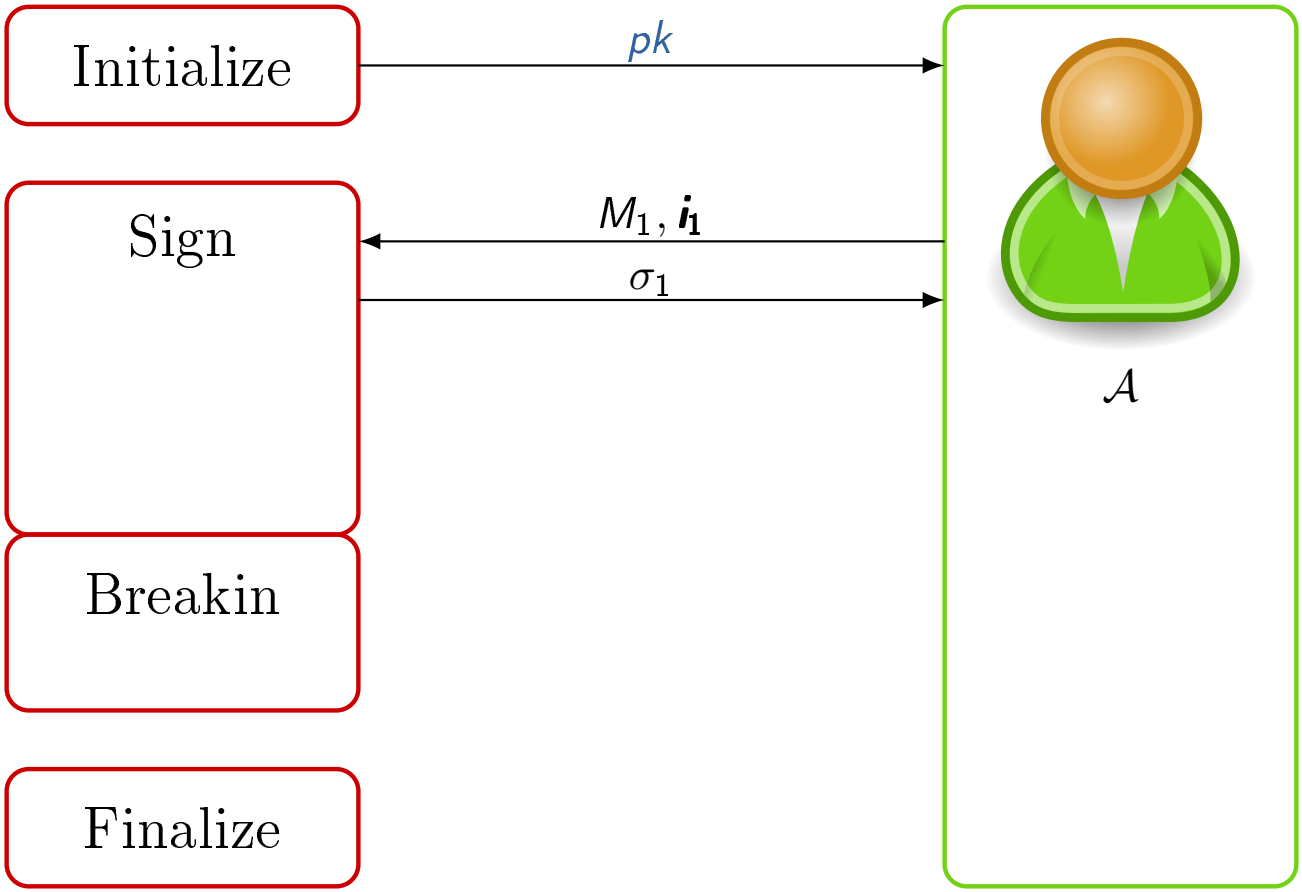
Security Goal: Existential Forward-Security



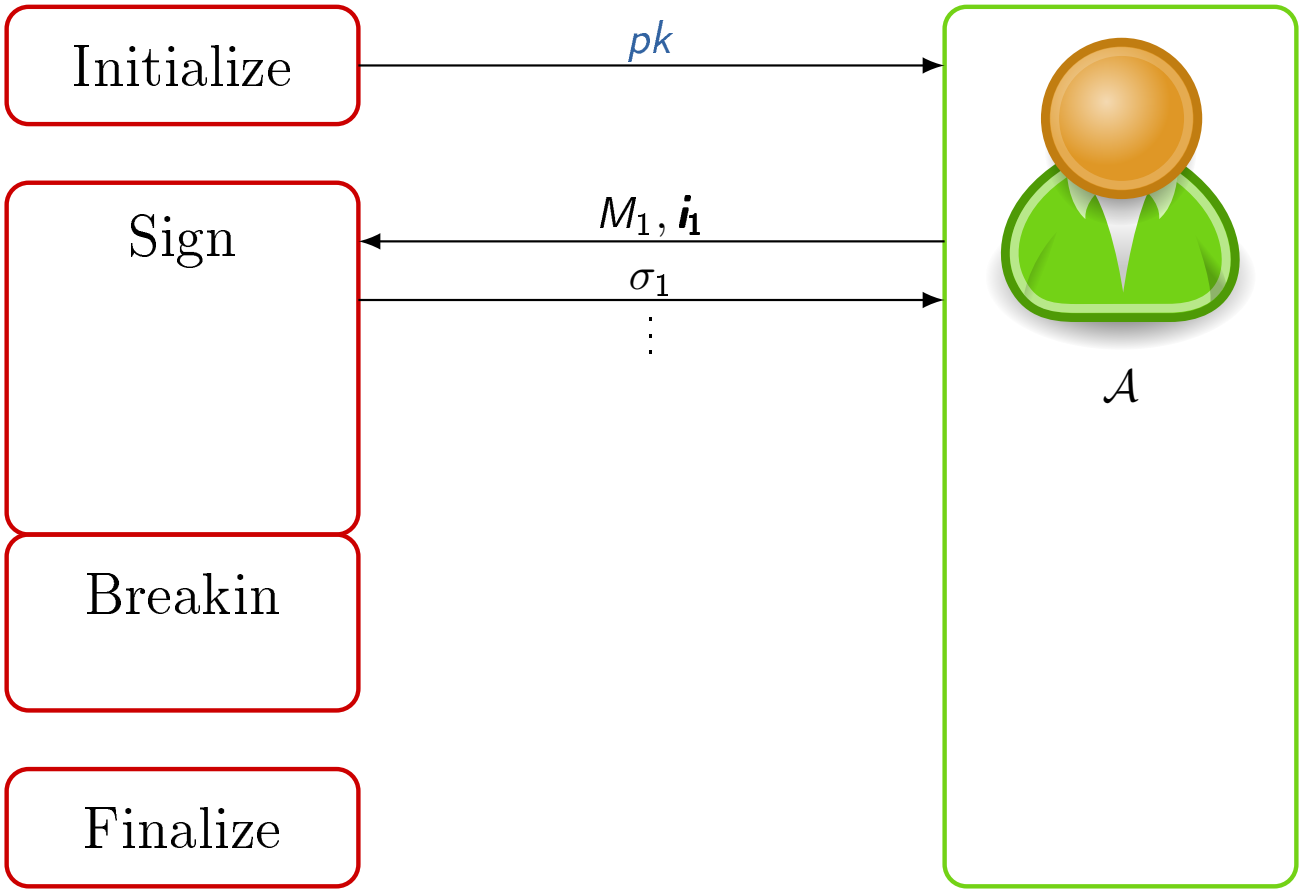
Security Goal: Existential Forward-Security



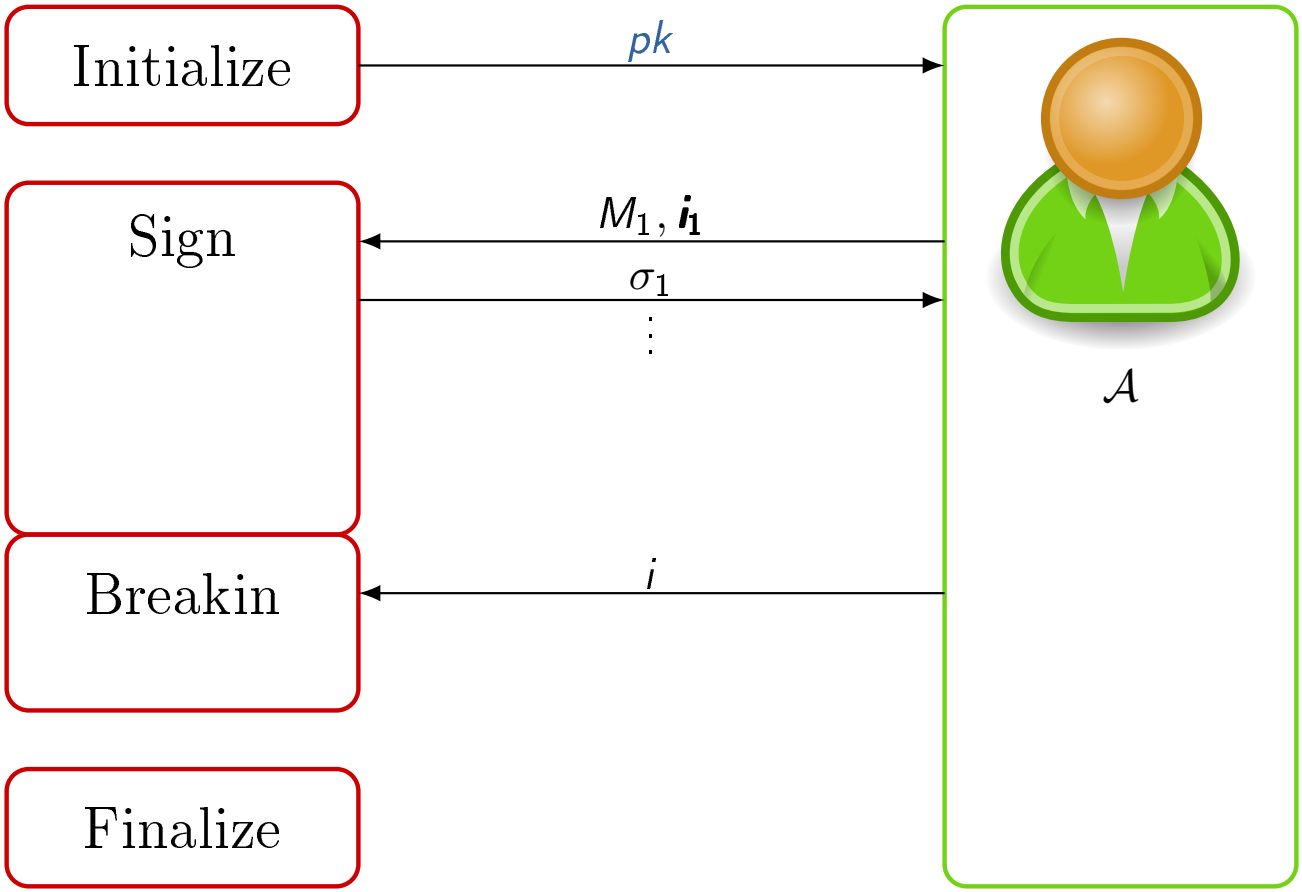
Security Goal: Existential Forward-Security



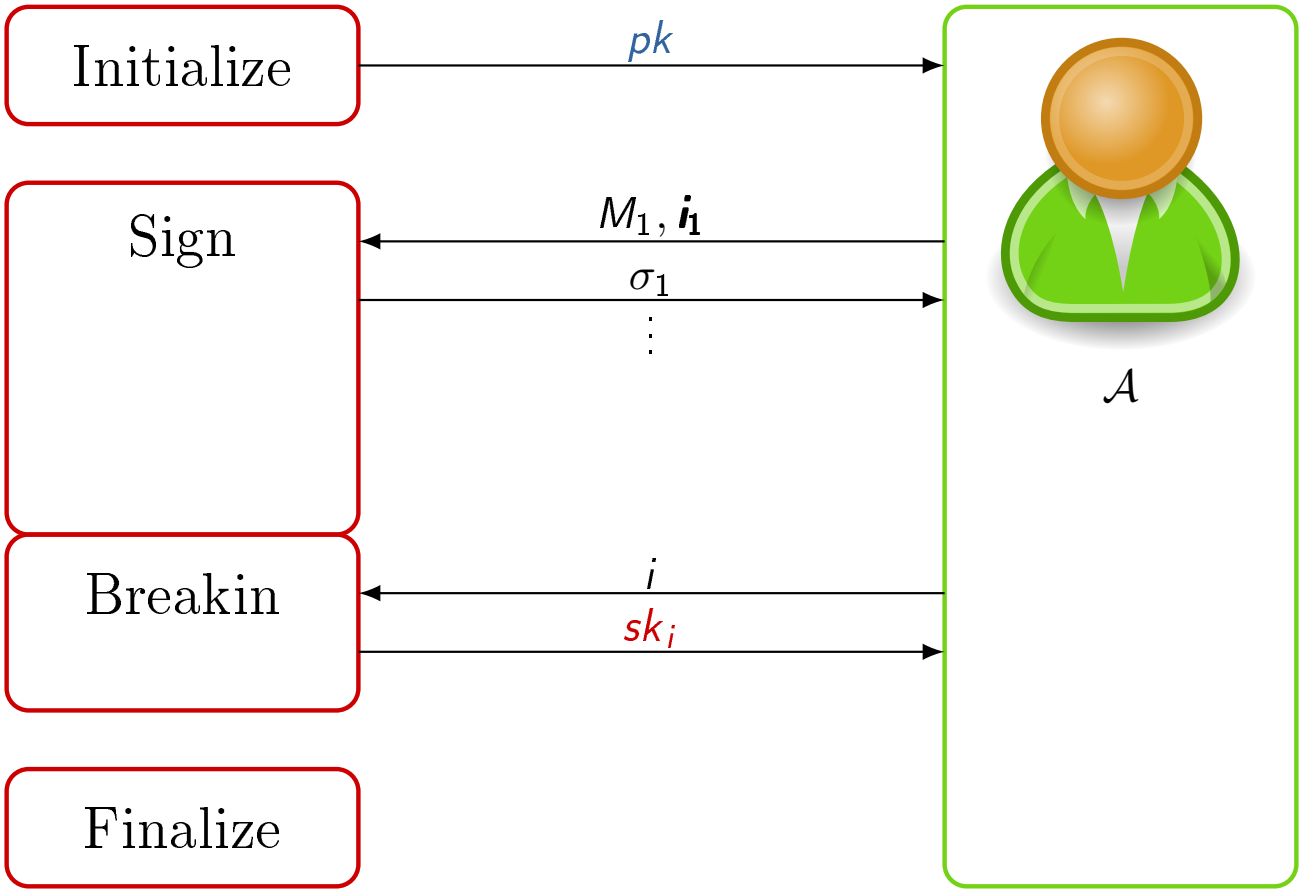
Security Goal: Existential Forward-Security



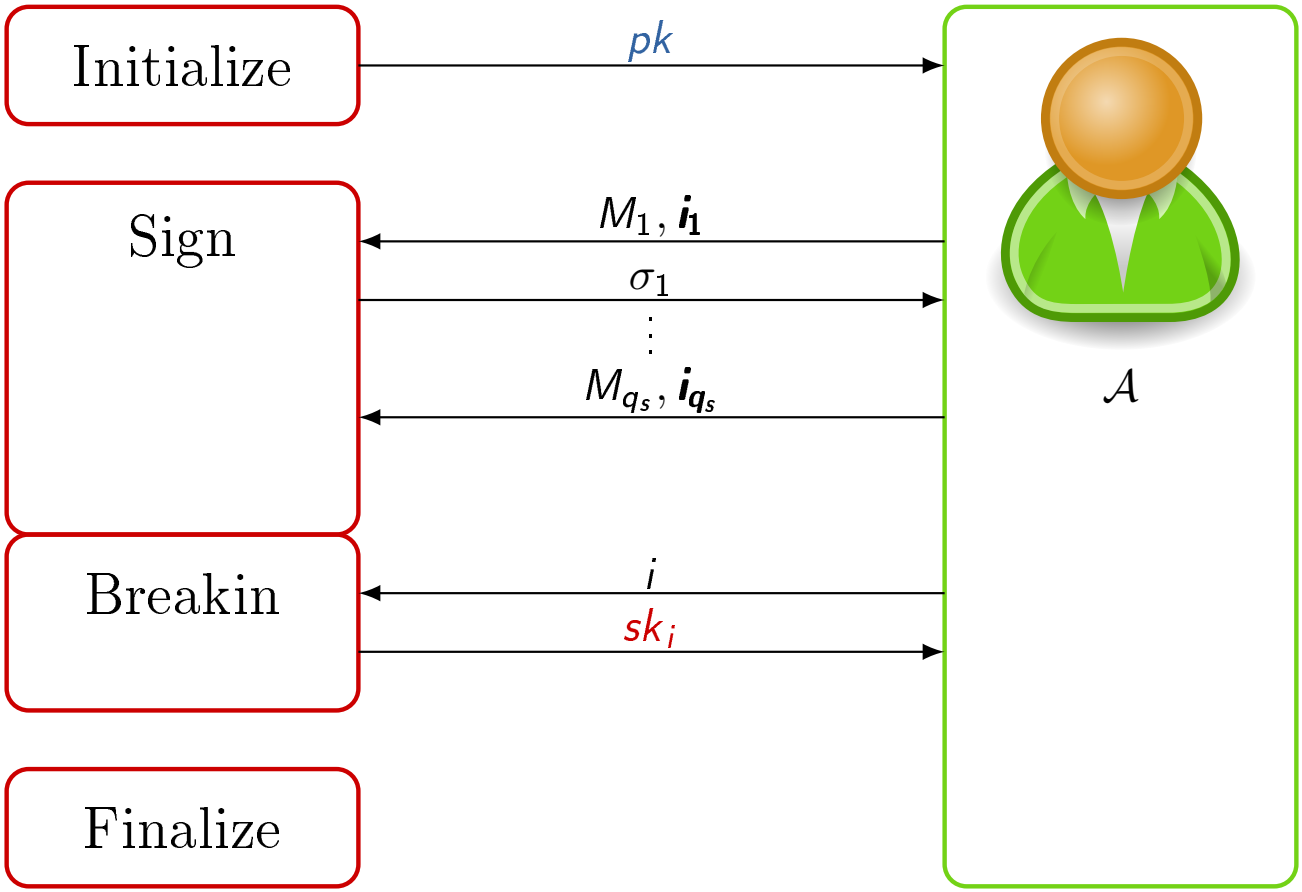
Security Goal: Existential Forward-Security



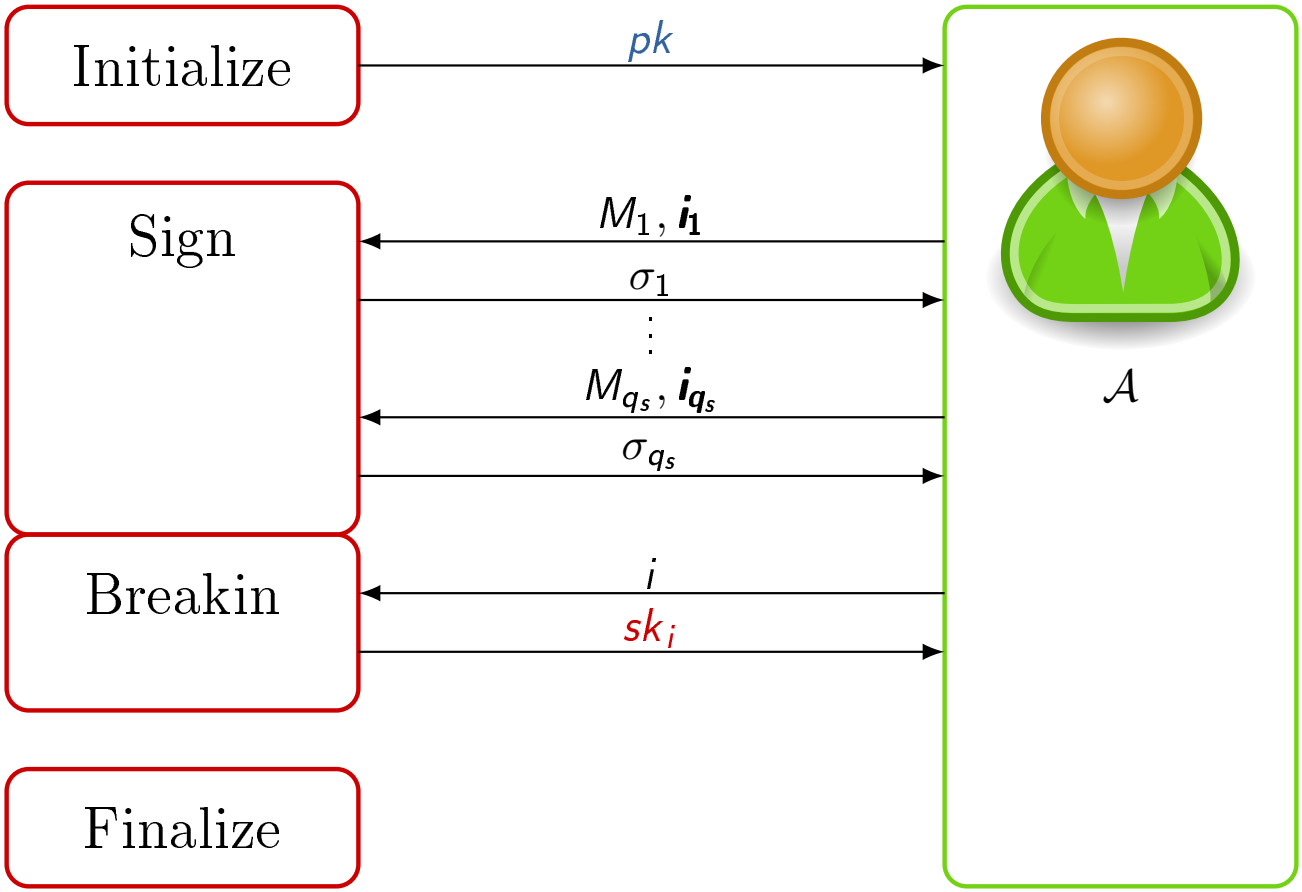
Security Goal: Existential Forward-Security



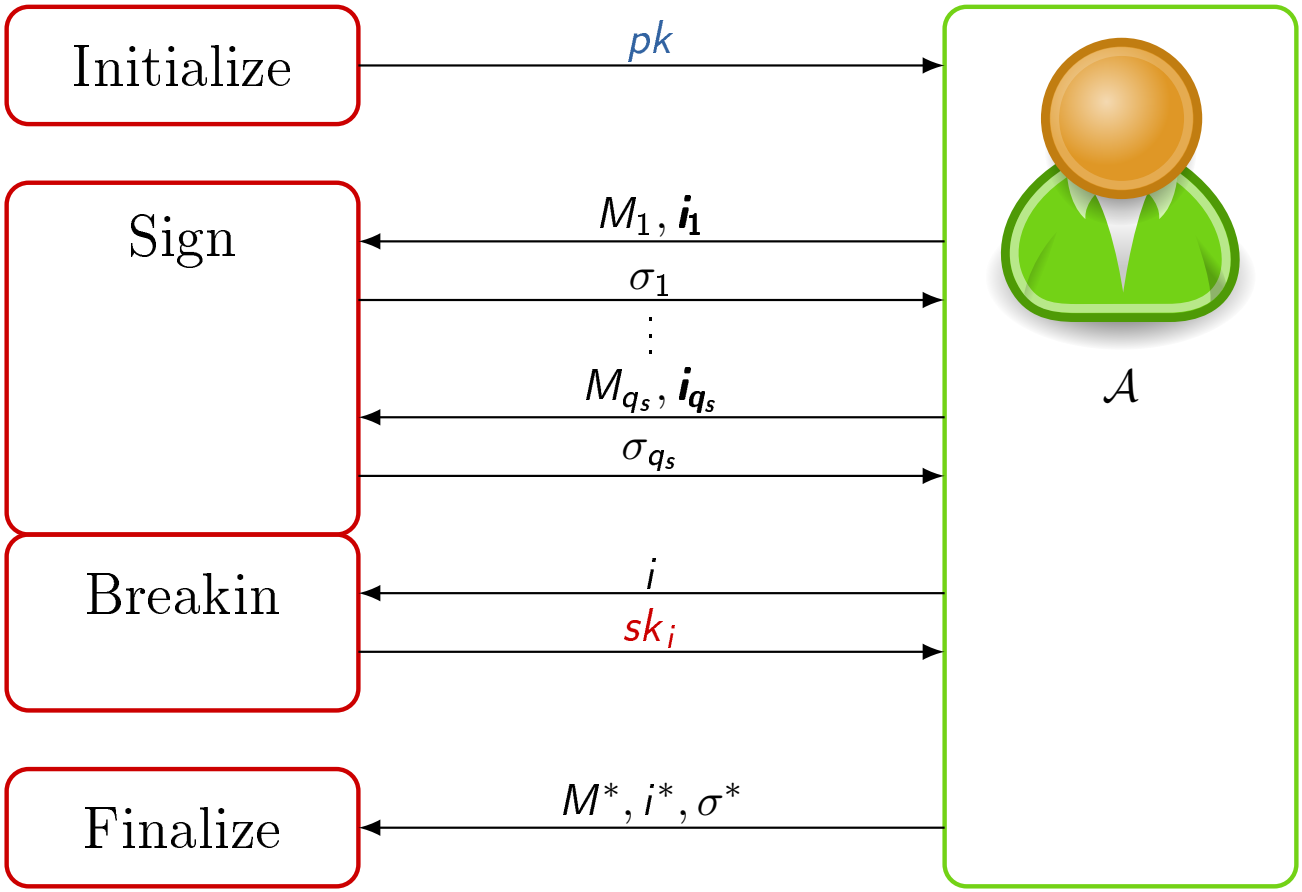
Security Goal: Existential Forward-Security



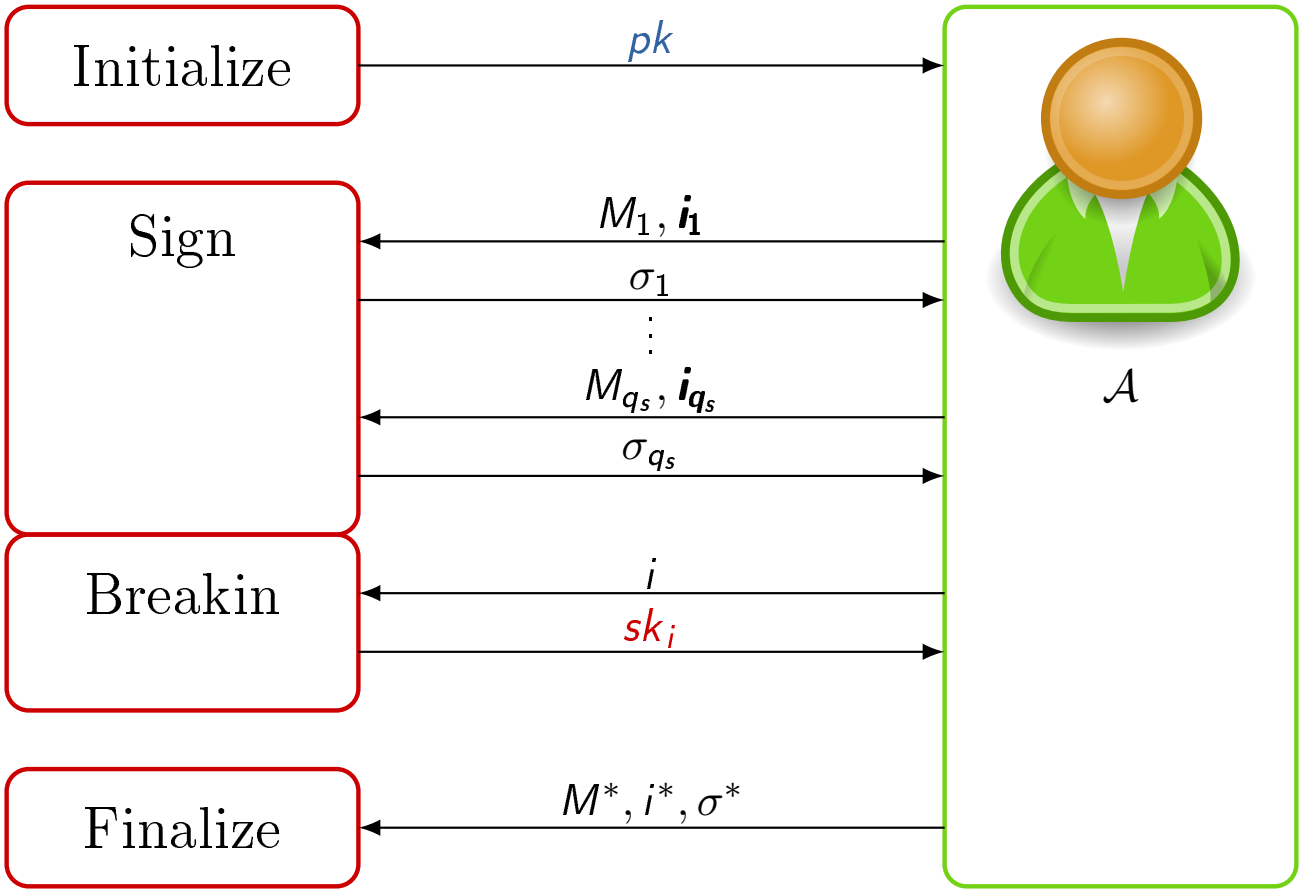
Security Goal: Existential Forward-Security



Security Goal: Existential Forward-Security



Security Goal: Existential Forward-Security

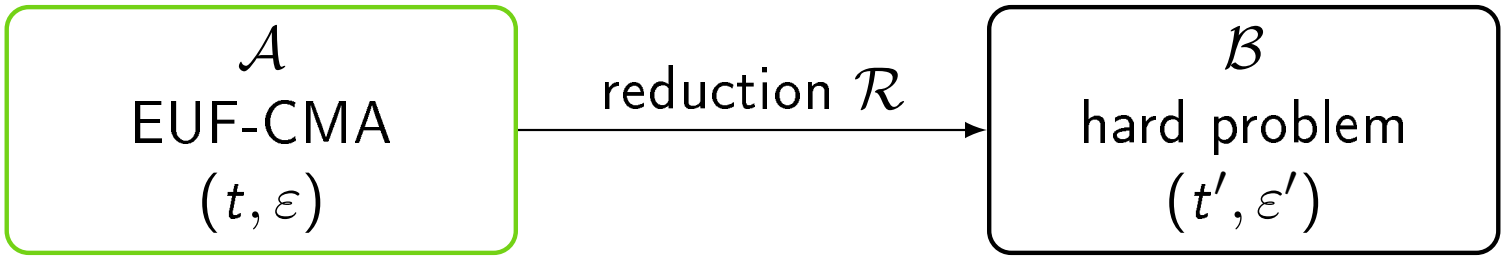


(t, ε)-**existential-forward-security** (**EUF-CMA**):
for any adversary $\mathcal{A}$ running in time at most t:

$$\Pr\left[\, i^* < i,\ \forall j, (M^*, i^*) \neq (M_j, i_j) \text{ and } \mathsf{Ver}(pk, \langle \sigma^*, i^* \rangle, M^*) = 1 \,\right] \leq \varepsilon.$$

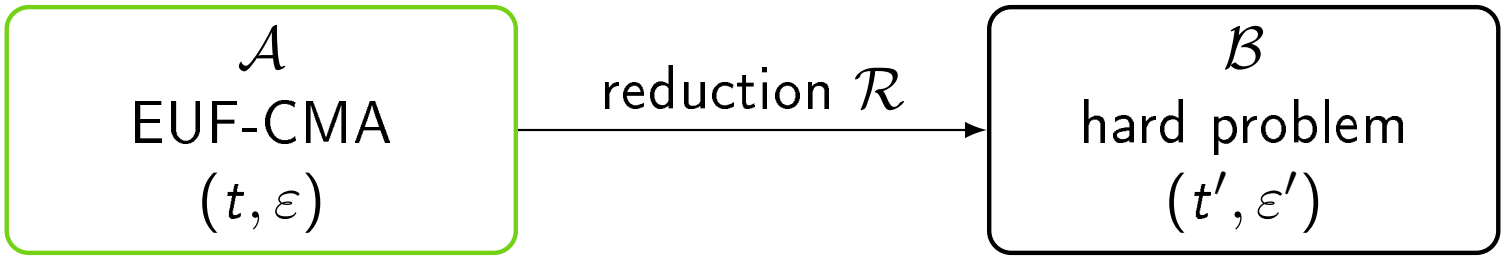
Reduction Tightness and Exact Security

How to prove a scheme is existentially forward-secure ?



Reduction Tightness and Exact Security

How to prove a scheme is existentially forward-secure ?

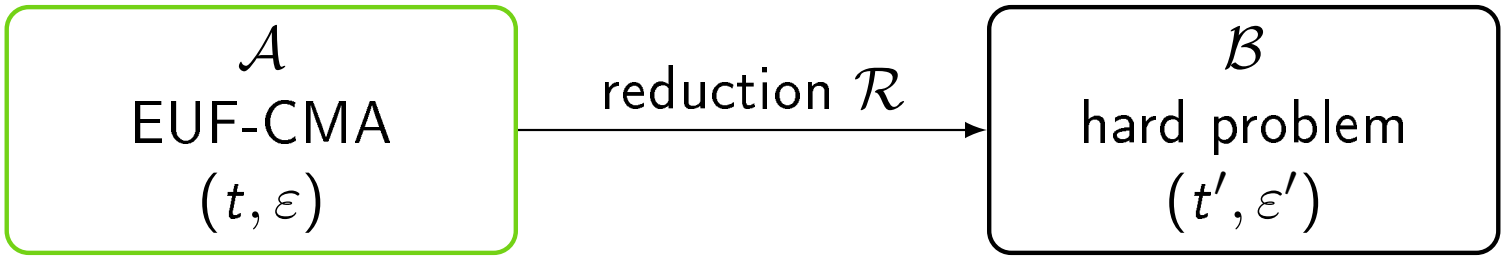


Example of an hard problem: factorization

- instance: N a product of two large primes p and q
- answer: p and q

Reduction Tightness and Exact Security

How to prove a scheme is existentially forward-secure ?



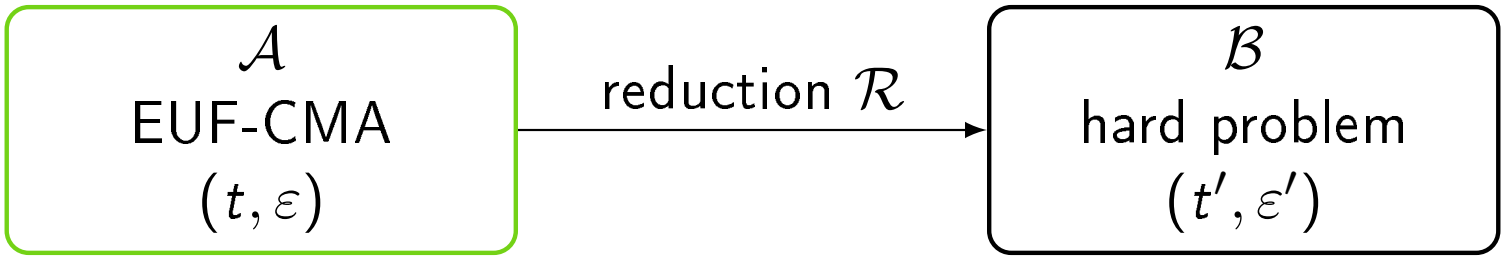
Example of an hard problem: factorization

- instance: N a product of two large primes p and q
- answer: p and q

- **tight** reduction: $t' \approx t$, $\varepsilon' \approx \varepsilon$,
- **loose** reduction otherwise.

Reduction Tightness and Exact Security

How to prove a scheme is existentially forward-secure ?



Example of an hard problem: factorization

- instance: N a product of two large primes p and q
- answer: p and q

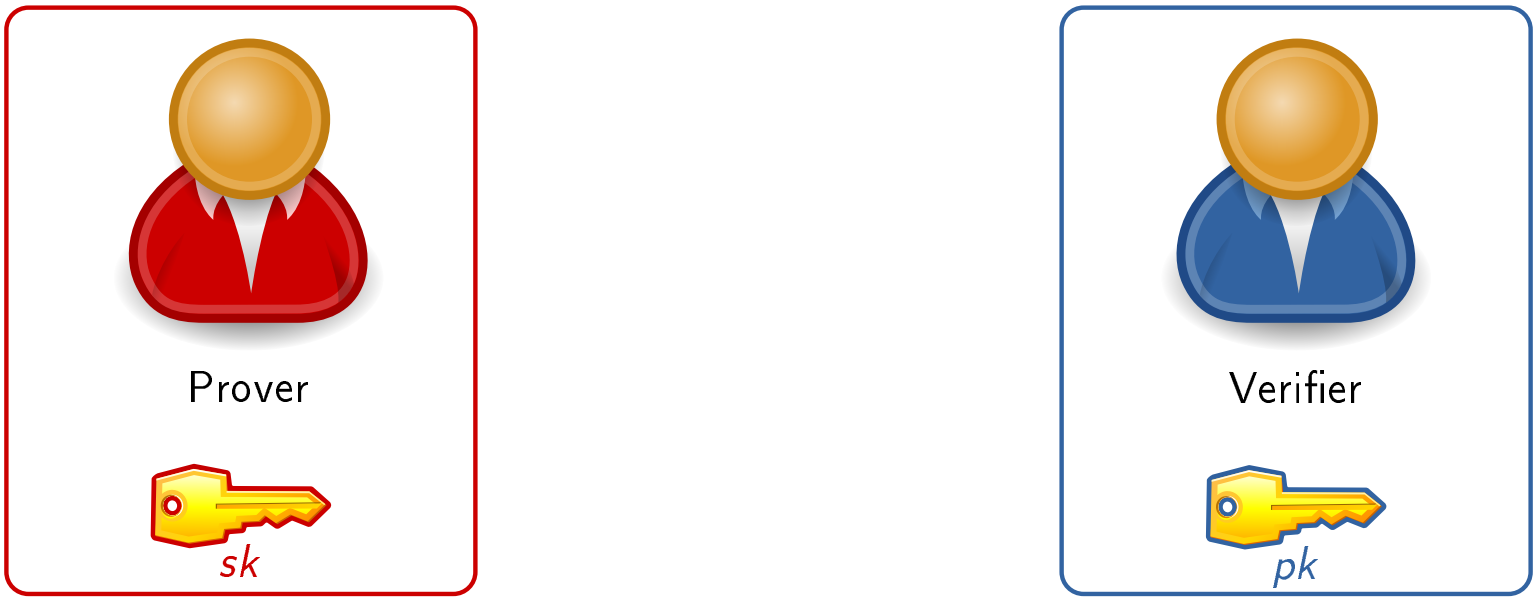
- **tight** reduction: $t' \approx t$, $\varepsilon' \approx \varepsilon$,
  $\rightarrow$ advantage: smaller parameters needed
- **loose** reduction otherwise.

Contributions

- practical forward-secure scheme constructions
  - with tighter reductions than previous "similar" schemes: loss a factor T instead of Tq_h ($q_h \approx 2^{80}$),
  - more efficient than previous schemes (for most aspects),
- optimality of above mentioned reductions and first fully tight forward-secure schemes.

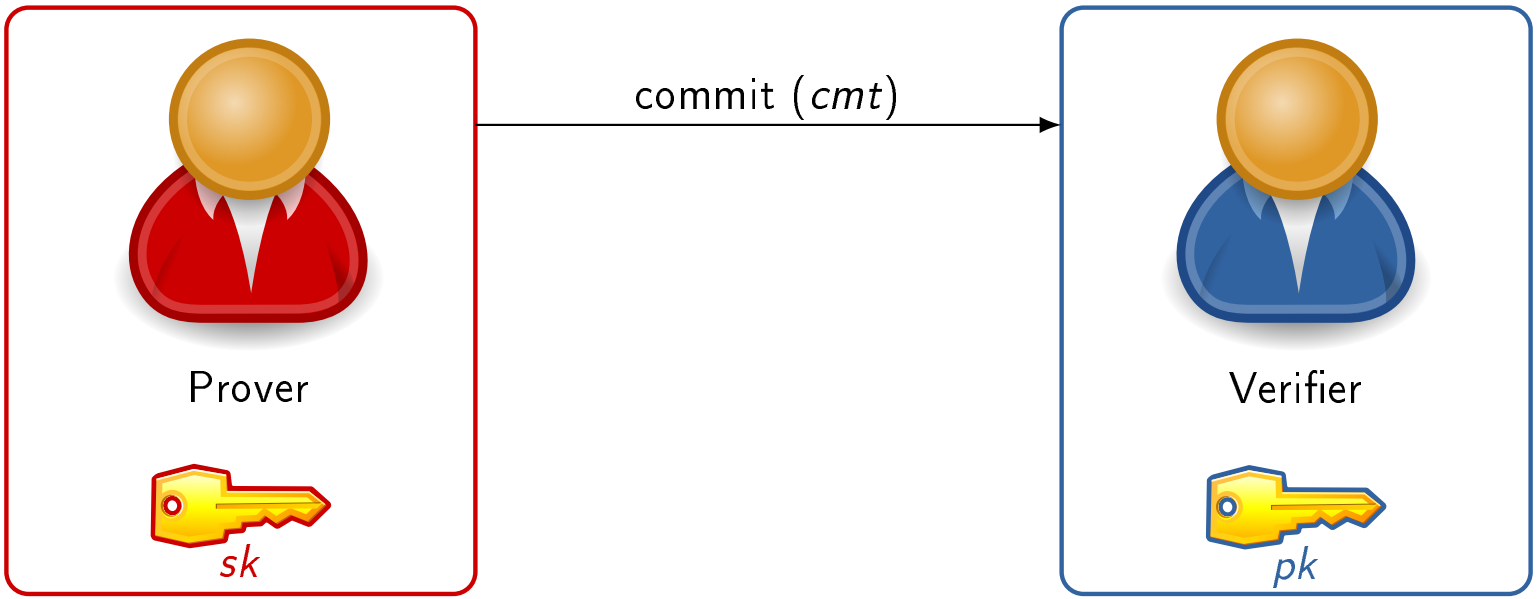
Canonical Identification Scheme

Canonical identification scheme $\mathcal{ID}$
(enables the prover to prove he knows *sk* associated to *pk*)



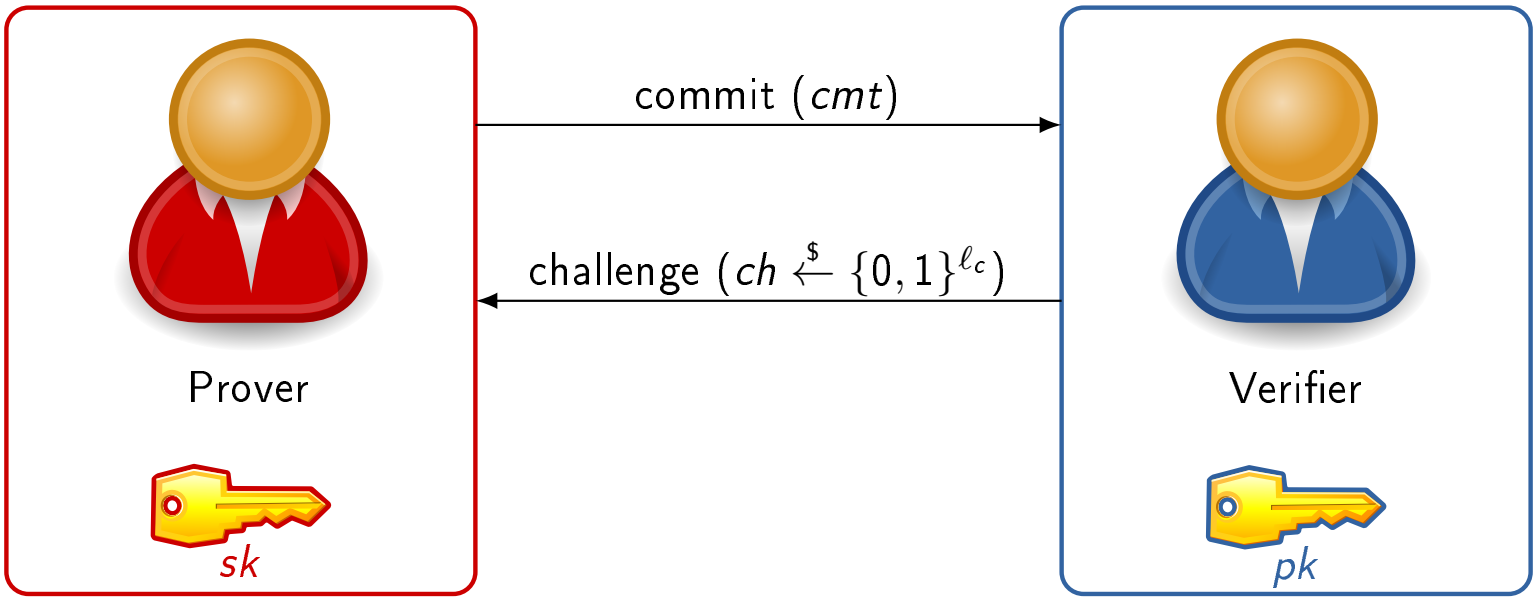
Canonical Identification Scheme

Canonical identification scheme $\mathcal{ID}$
(enables the prover to prove he knows sk associated to pk)



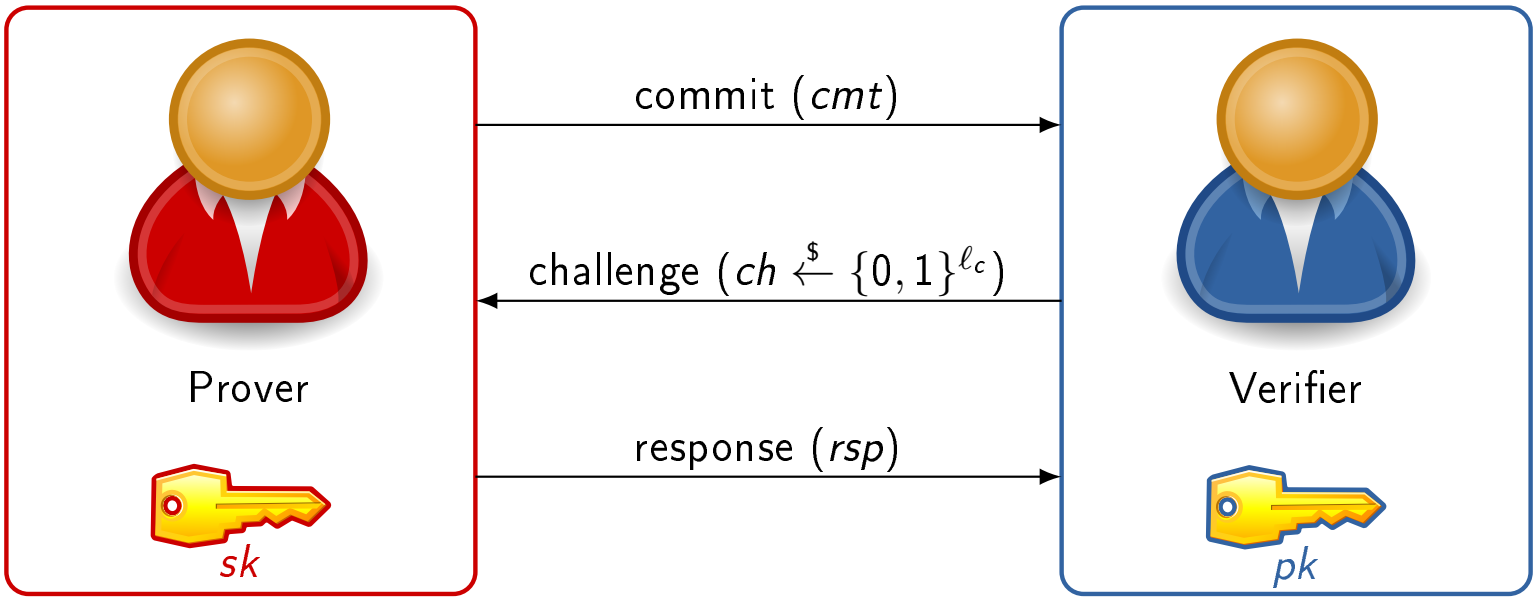
Canonical Identification Scheme

Canonical identification scheme $\mathcal{ID}$
(enables the prover to prove he knows sk associated to pk)



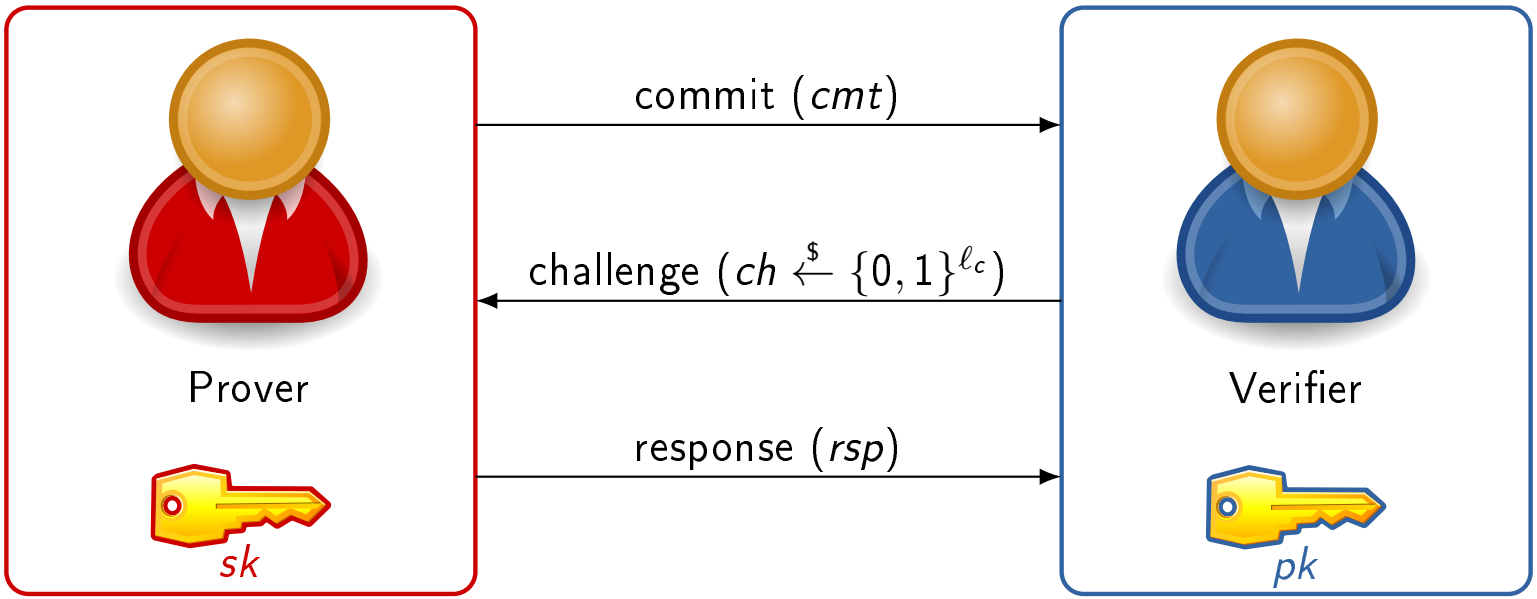
Canonical Identification Scheme

Canonical identification scheme $\mathcal{ID}$
(enables the prover to prove he knows sk associated to pk)



Canonical Identification Scheme

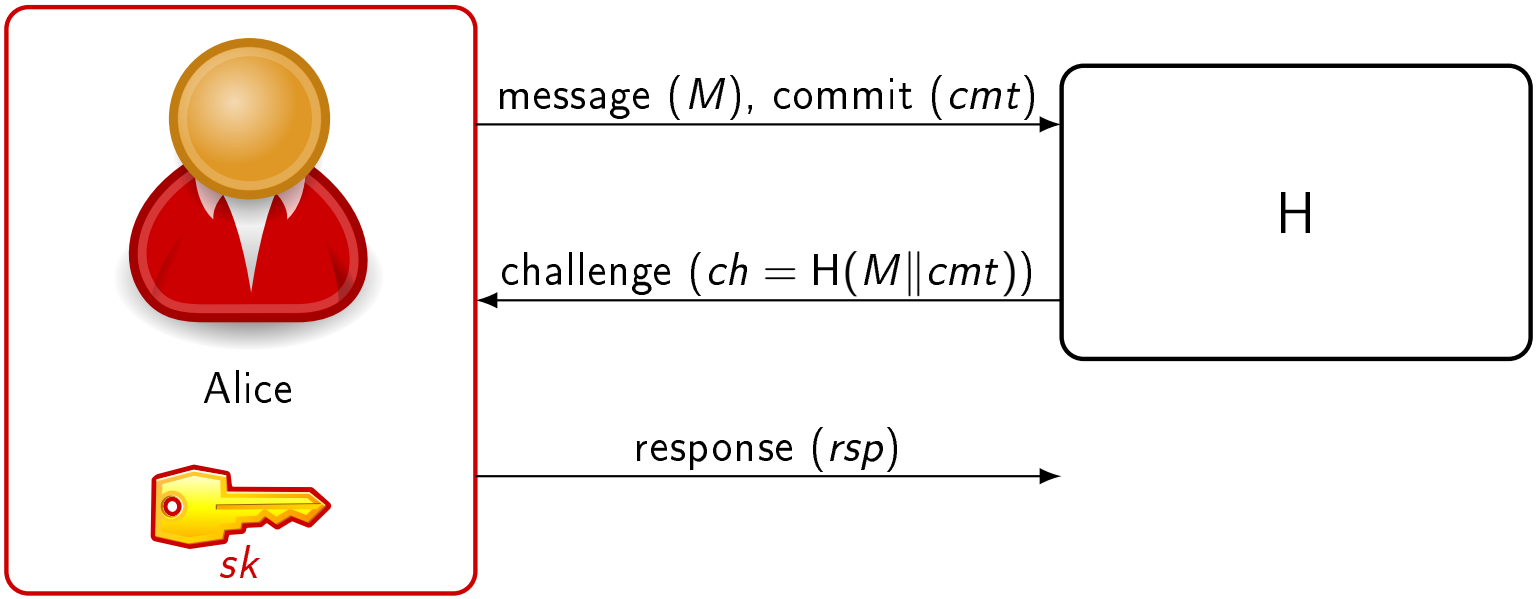
Canonical identification scheme $\mathcal{ID}$
(enables the prover to prove he knows sk associated to pk)



The verifier can accept or reject the transcript (cmt, ch, rsp).

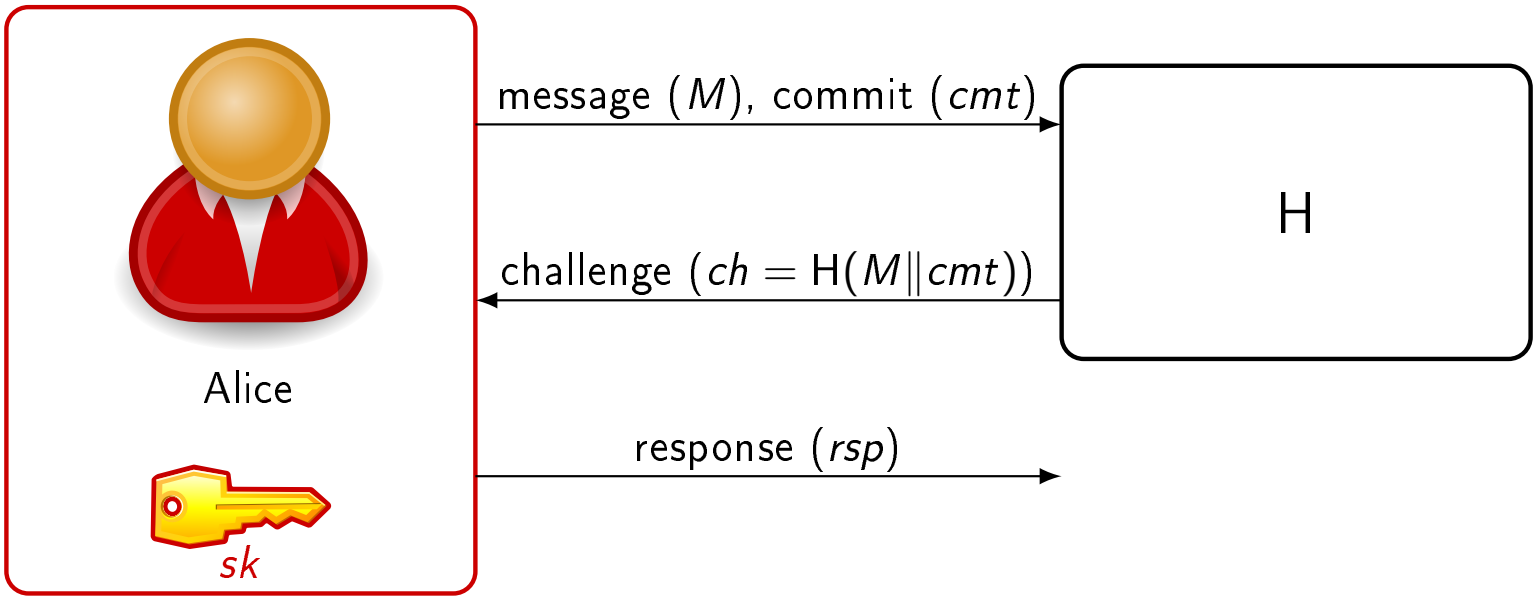
Fiat-Shamir Transform

Identification scheme $\mathcal{ID}$ $\longrightarrow$ Signature scheme $\mathcal{DS}[\mathcal{ID}]$



Fiat-Shamir Transform

Identification scheme $\mathcal{ID}$ $\longrightarrow$ Signature scheme $\mathcal{DS}[\mathcal{ID}]$



Signature: $\sigma = (cmt, rsp)$

Security Reductions

Is $\mathcal{DS}[\mathcal{ID}]$ secure ?

- [AABN02] if $\mathcal{ID}$ is (t, ε)-**secure-against-impersonations**, then $\mathcal{DS}[\mathcal{ID}]$ is about $(t, \frac{\varepsilon}{q_h})$-EUF-CMA.
  - ◇ a scheme is secure against impersonations if no adversary can convince a verifier to accept (even if he can see transcripts),

  → not tight: loss of a factor $q_h \approx 2^{80}$.

Security Reductions

Is $\mathcal{DS}[\mathcal{ID}]$ secure ?

- [AABN02] if $\mathcal{ID}$ is (t, ε)-**secure-against-impersonations**, then $\mathcal{DS}[\mathcal{ID}]$ is about $(t, \frac{\varepsilon}{q_h})$-EUF-CMA.
  - ◇ a scheme is secure against impersonations if no adversary can convince a verifier to accept (even if he can see transcripts),

  → not tight: loss of a factor $q_h \approx 2^{80}$.

- [AFLT12] if $\mathcal{ID}$ is a (t, ε)-"secure" **lossy** identification scheme, then $\mathcal{DS}[\mathcal{ID}]$ is about (t, ε)-EUF-CMA.

  → tight.

Lossy Identification Schemes

A lossy identification scheme is a scheme such that we can generate keys in two ways:

- **normal key** generation: $(pk, sk) \rightarrow$ for the protocol,
- **lossy key** generation: $pk \rightarrow$ for the reduction; no adversary can impersonate a prover with public key pk,

such that lossy public keys and normal public keys are **computationally indistinguishable**.

Lossy Identification Schemes

A lossy identification scheme is a scheme such that we can generate keys in two ways:

- **normal key** generation: $(pk, sk) \rightarrow$ for the protocol,
- **lossy key** generation: $pk \rightarrow$ for the reduction; no adversary can impersonate a prover with public key pk,

such that lossy public keys and normal public keys are **computationally indistinguishable**.

Key indistinguishability $\longrightarrow$ existential unforgeability.

Contributions

- extension of the notion of lossy identification scheme to key-evolving lossy identification scheme, which can be transformed into forward-secure signature schemes,
- generic key-evolving lossy identification scheme using factoring based-problems with tight reductions (for key indistinguishability),
- efficient instantiations of this generic scheme (more efficient than previous schemes for most aspects).

Extension to Forward-Secure Signature Schemes

- **key-evolving** identification schemes:
  - → a secret key sk_i per time period i,
- extension of the Fiat-Shamir transform:
  - → key-evolving identification scheme ⟶ forward-secure signature scheme,
  - → original transform for sk_i.

Extension to Forward-Secure Signature Schemes

- **key-evolving** identification schemes:
  - → a secret key sk_i per time period i,
- extension of the Fiat-Shamir transform:
  - → key-evolving identification scheme ⟶ forward-secure signature scheme,
  - → original transform for sk_i.
- [AABN02] if $\mathcal{ID}$ is (t, ε)-**secure-against-impersonations**, then $\mathcal{FS}[\mathcal{ID}]$ is about $(t, \frac{\varepsilon}{Tq_h})$-EUF-CMA.
  - → not tight: loss of a factor Tq_h.

Extension to Forward-Secure Signature Schemes

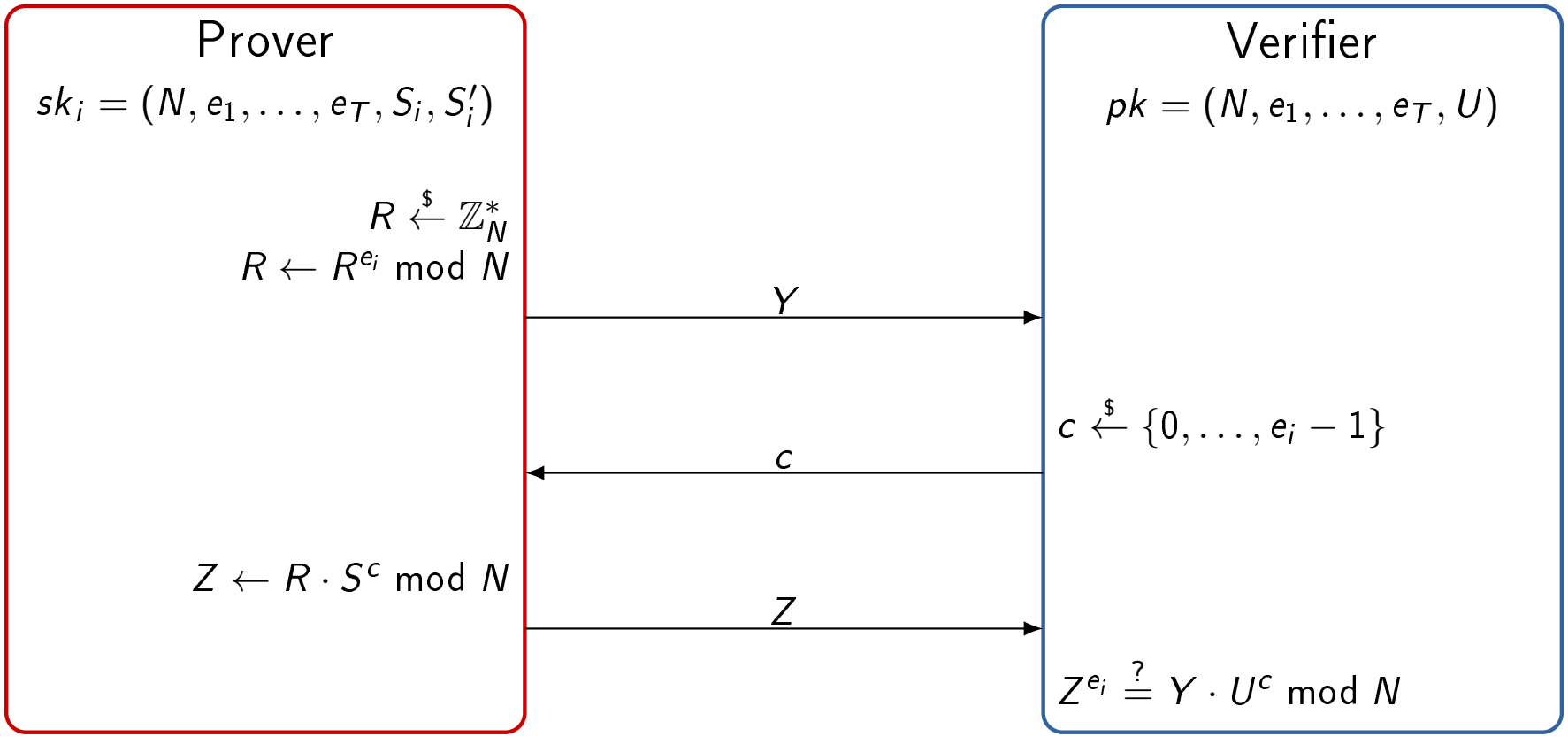
- **key-evolving** identification schemes:
  - → a secret key sk_i per time period i,
- extension of the Fiat-Shamir transform:
  - → key-evolving identification scheme ⟶ forward-secure signature scheme,
  - → original transform for sk_i.
- [AABN02] if $\mathcal{ID}$ is (t, ε)-**secure-against-impersonations**, then $\mathcal{FS}[\mathcal{ID}]$ is about $(t, \frac{\varepsilon}{Tq_h})$-EUF-CMA.
  - → not tight: loss of a factor Tq_h.
- [our work] if $\mathcal{ID}$ is a (t, ε)-key-indistinguishable **lossy** key-evolving identification scheme, then $\mathcal{FS}[\mathcal{ID}]$ is about $(t, \frac{\varepsilon}{T})$-EUF-CMA.
  - → almost tight: loss of a factor T.

Extension to Forward-Secure Signature Schemes II

Lossy **key-evolving** identification schemes:

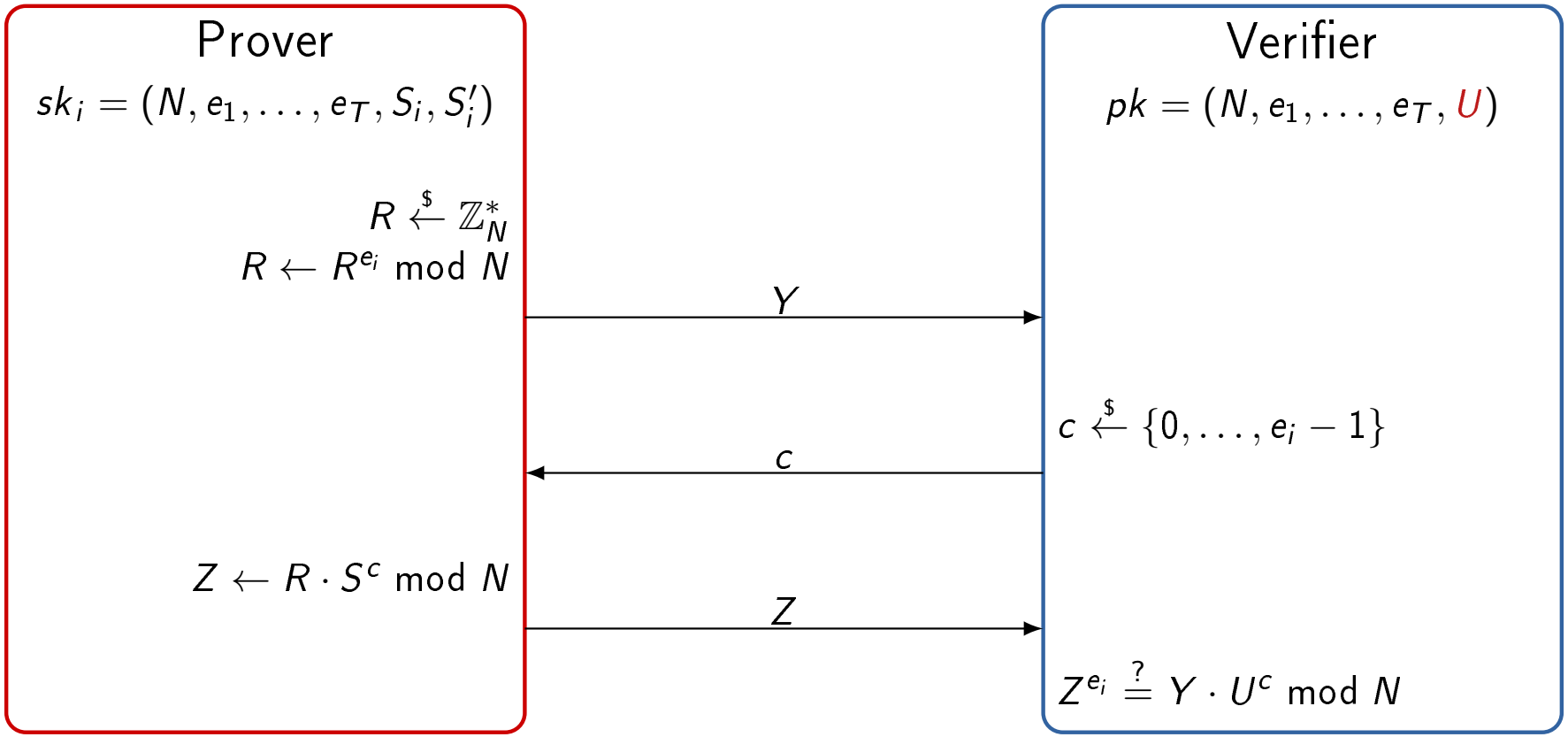
- the secret key changes at each time period,
- the lossy key generation algorithm takes as input a period i and outputs (pk, sk_{i+1}),
- if pk is lossy for period i, no adversary can impersonate a prover for period i with public key pk.

Our Variant of Itkis and Reyzin Scheme



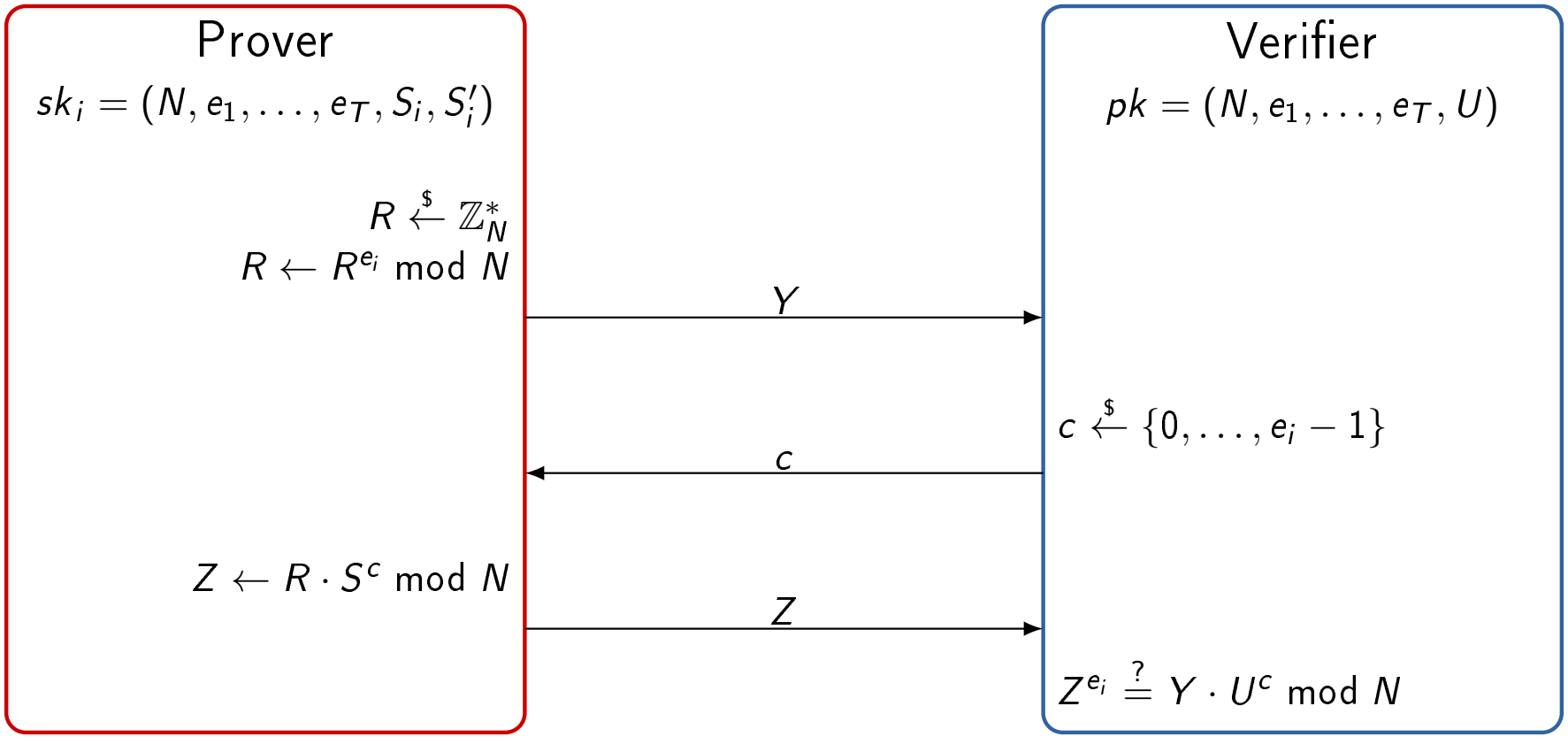
- $N = pq$, with p, q two $l/2$-bit primes,
- $e_1, \ldots, e_T$ are distinct $l/8$-bit primes, coprime with $\phi(N)$,
- $U = S'^{e_i \cdots e_T}_i \bmod N$ and $S_i = S'^{e_{i+1} \cdots e_T}_i \bmod N$ such that $U = S_i^{e_i} \bmod N$

Our Variant of Itkis and Reyzin Scheme



- $N = pq$, with p, q two $l/2$-bit primes,
- $e_1, \ldots, e_T$ are distinct $l/8$-bit primes, coprime with $\phi(N)$,
- $U = S'^{e_i \cdots e_T}_i \bmod N$ and $S_i = S'^{e_{i+1} \cdots e_T}_i \bmod N$ such that $U = S_i^{e_i} \bmod N$

Our Variant of Itkis and Reyzin Scheme



- $sk_{i+1} \xleftarrow{\$} \text{Update}(sk_i)$
- $S'_{i+1} = S'^{e_i}_i \bmod N$
- $S_i = S'^{e_{i+2} \cdots e_T}_{i+1} \bmod N$

Analysis of our Variant of the Itkis-Reyzin Scheme

- Comparison with the Itkis and Reyzin scheme [IR01]:
  - ○ roughly same key length, same key generation time, same or better update time,
  - \+ signing: 9× faster, verifying: 6× faster, signature: 3× shorter.

- Comparison with the MMM scheme [MMM02]:
  - − T× slower key generation and update,
  - ○ roughly same signing time,
  - \+ verifying: 2× faster, signature: 2× shorter.

(for 80 bits of security)

Optimality of Reductions

Theorem (approximate statement)

*For any **key-verifiable** forward-secure scheme, any (black-box) reduction loses at least a factor T.*

A forward-secure scheme is **key-verifiable**, if there exist a set V of bit strings such that:

- all secret keys are in V,
- membership can be tested in polynomial time,
- all bit strings of V can be used as secret keys and yield statistically indistinguishable signatures.

Fully Tight Forward-Secure Schemes

Can we construct a fully tight forward-secure scheme ?

Fully Tight Forward-Secure Schemes

Can we construct a fully tight forward-secure scheme ?

Yes ! We construct the first[1] fully tight forward-secure scheme.

Idea

- consider a fully tight EUF-MUC signature scheme $\mathcal{DS}$,
  - EUF-MUC: extension of EUF-CMA to multiple public keys (with corruptions),
- use one key pair of $\mathcal{DS}$ per period,
- if $\mathcal{DS}$ is (t, ε)-EUF-MUC, the associated forward-secure scheme is about (t, ε)-existentially-forward-secure.

[1] as far as we know

Constructing a Fully Tight EUF-MUC Signature Scheme

Idea

- scheme with multiple secret keys for a given public key,
- all secret keys are indistinguishable (even given access to signatures),
- the reduction knows one secret key for each public key,
- if the adversary creates a signature with another secret key, the reduction can solve a hard problem.

Example of scheme

- public key: perfectly hiding commitment of a random value X,
- signature: non-interactive proof of knowledge of X (labeled by the message M).

Thank you for your attention !

References I

Michel Abdalla, Jee Hea An, Mihir Bellare, and Chanathip Namprempre.

From identification to signatures via the Fiat-Shamir transform: Minimizing assumptions for security and forward-security.

In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 418–433. Springer, April / May 2002.

Michel Abdalla, Pierre-Alain Fouque, Vadim Lyubashevsky, and Mehdi Tibouchi.

Tightly-secure signatures from lossy identification schemes.

In *EUROCRYPT 2012*, LNCS. Springer, 2012.

Gene Itkis and Leonid Reyzin.

Forward-secure signatures with optimal signing and verifying.

In Joe Kilian, editor, *CRYPTO 2001*, volume 2139 of *LNCS*, pages 332–354. Springer, August 2001.

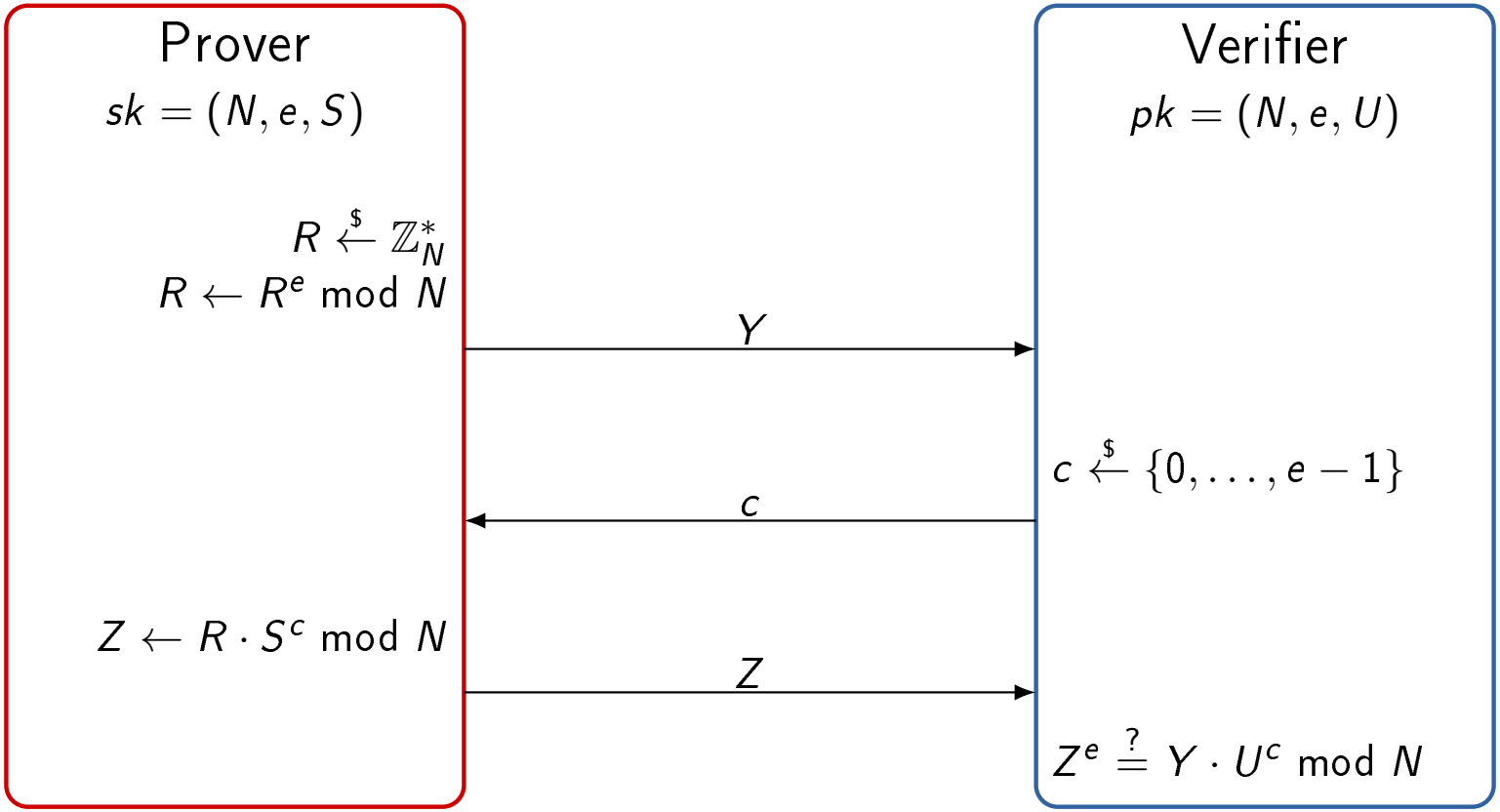
References II

Tal Malkin, Daniele Micciancio, and Sara K. Miner.

Efficient generic forward-secure signatures with an unbounded number of time periods.

In Lars R. Knudsen, editor, *EUROCRYPT 2002*, volume 2332 of *LNCS*, pages 400–417. Springer, April / May 2002.

Guillou-Quisquater Identification Scheme



$N = pq$, with p, q two $l/2$-bit primes,
e is a $l/8$-bit prime, coprime with $\phi(N) = (p-1)(q-1)$,
$U = S^e \bmod N$

Our Variant of Itkis and Reyzin Scheme

Based on the ϕ-hiding assumption.

Comparison with the Itkis and Reyzin scheme [IR01]:

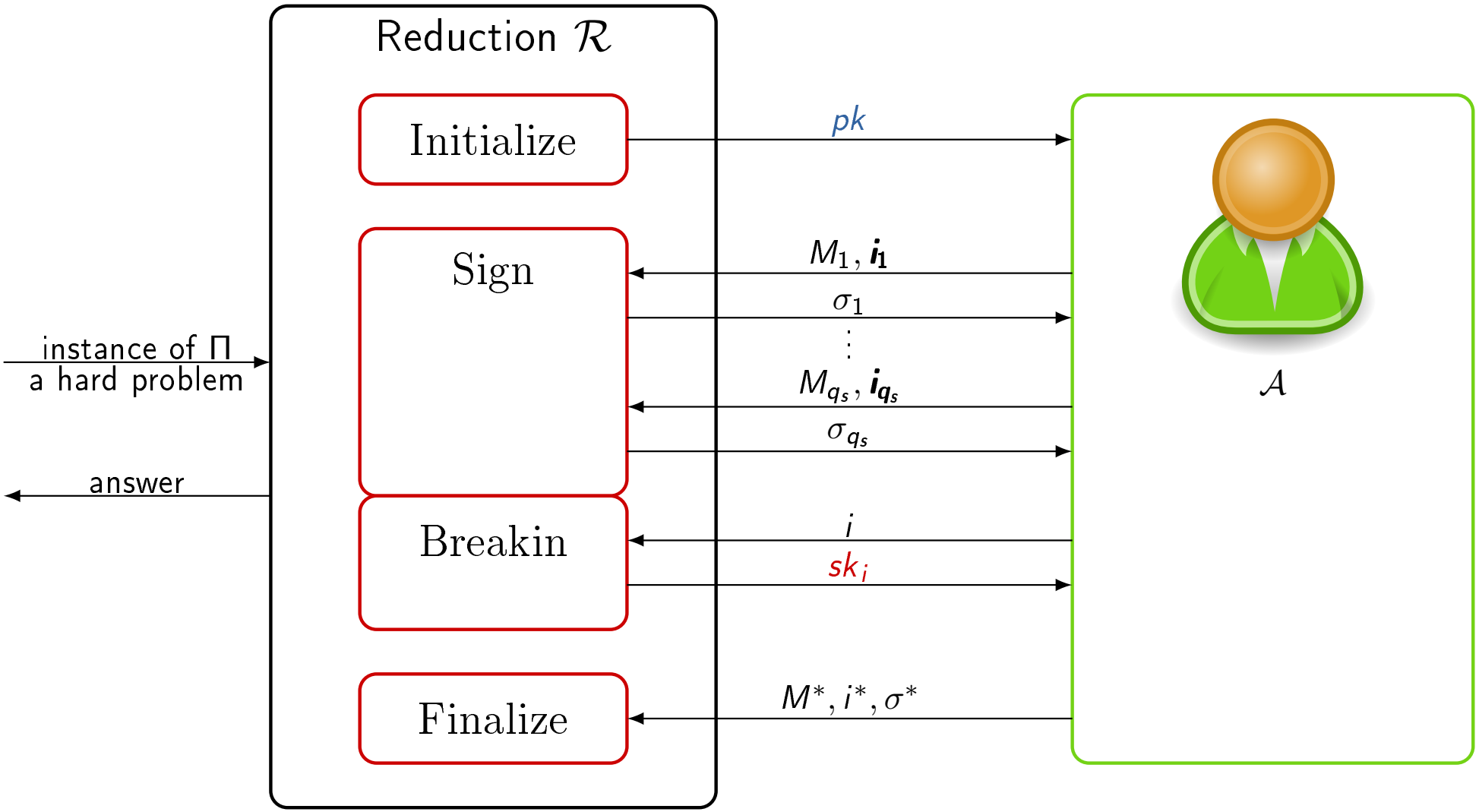
- ◦ roughly same key length, same key generation time, same or better update time,
- \+ signing: $9\times$ faster, verifying: $6\times$ faster, signature: $3\times$ shorter.

Comparison with the MMM scheme [MMM02]:

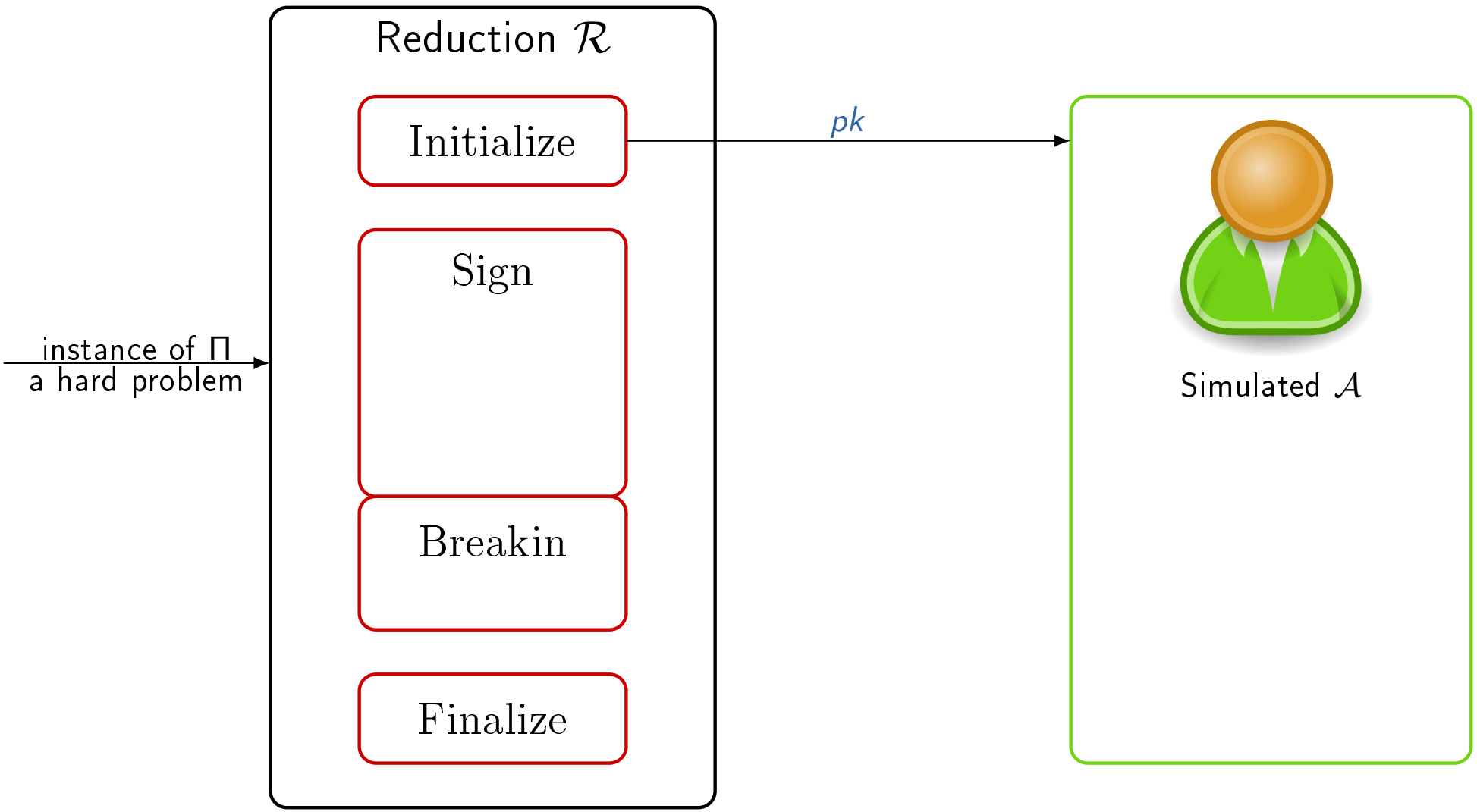
- − $T\times$ slower key generation and update,
- ◦ roughly same signing time,
- \+ verifying: $2\times$ faster, signature: $2\times$ shorter.

(for 80 bits of security)

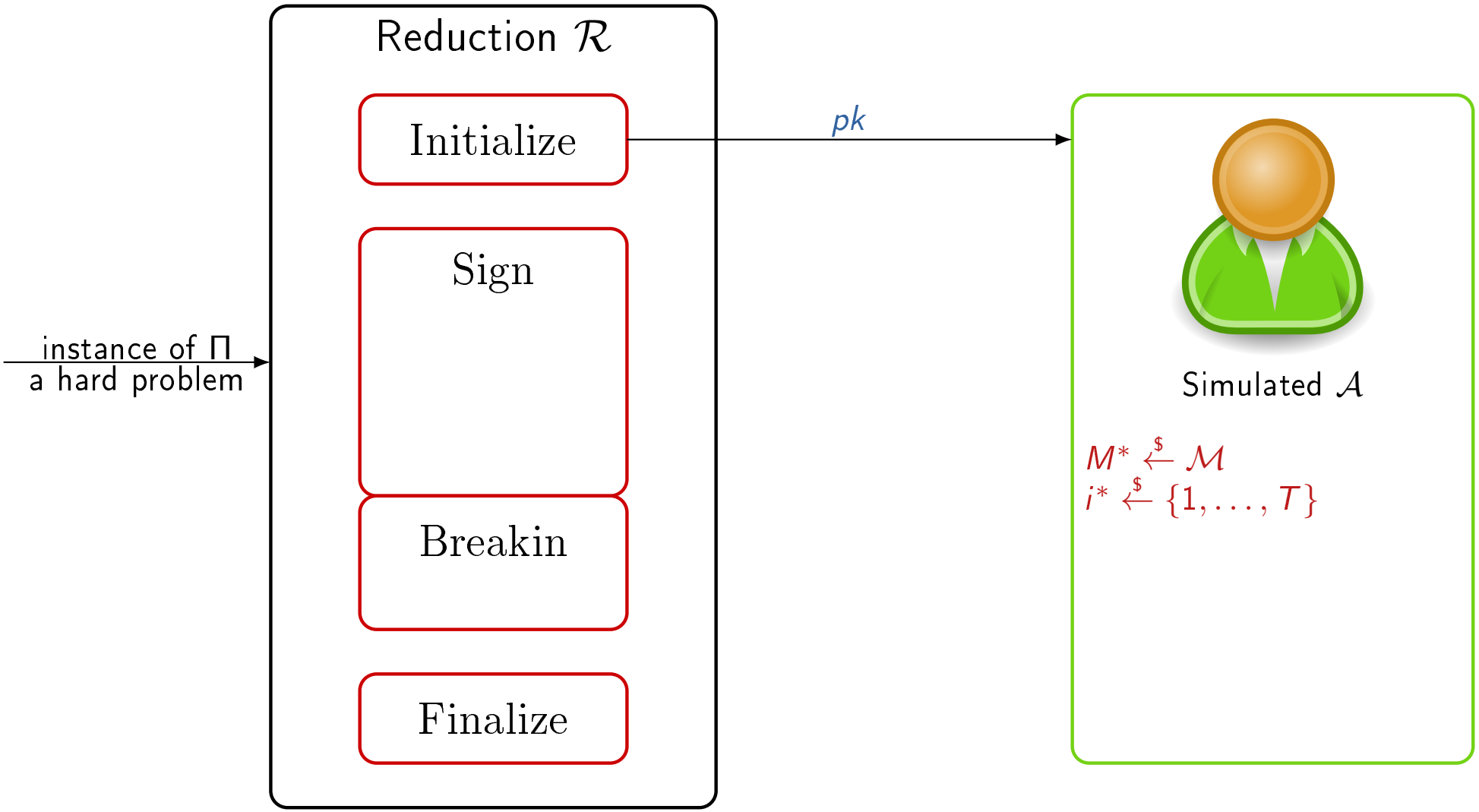
Black-Box Reduction



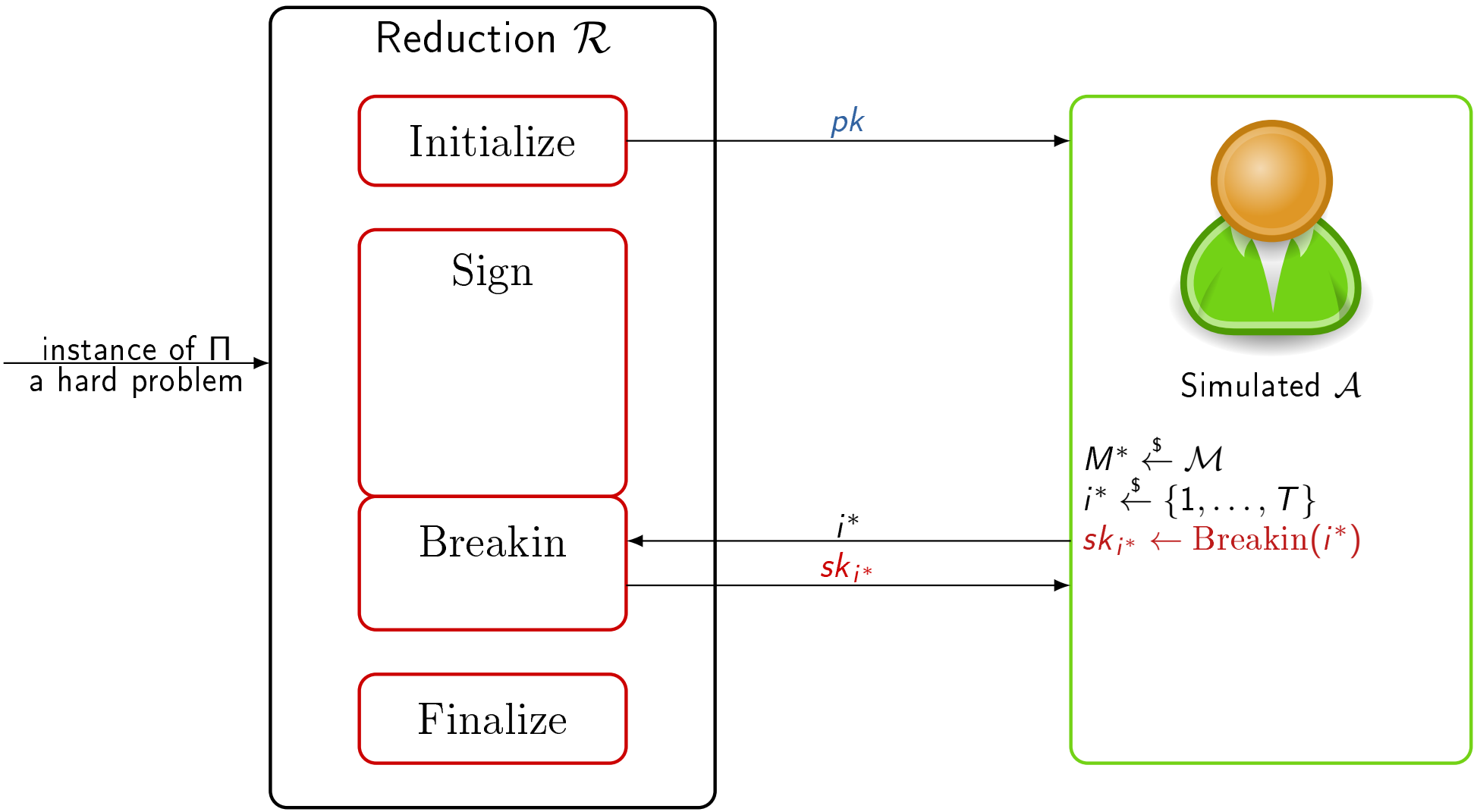
Optimality of Reductions



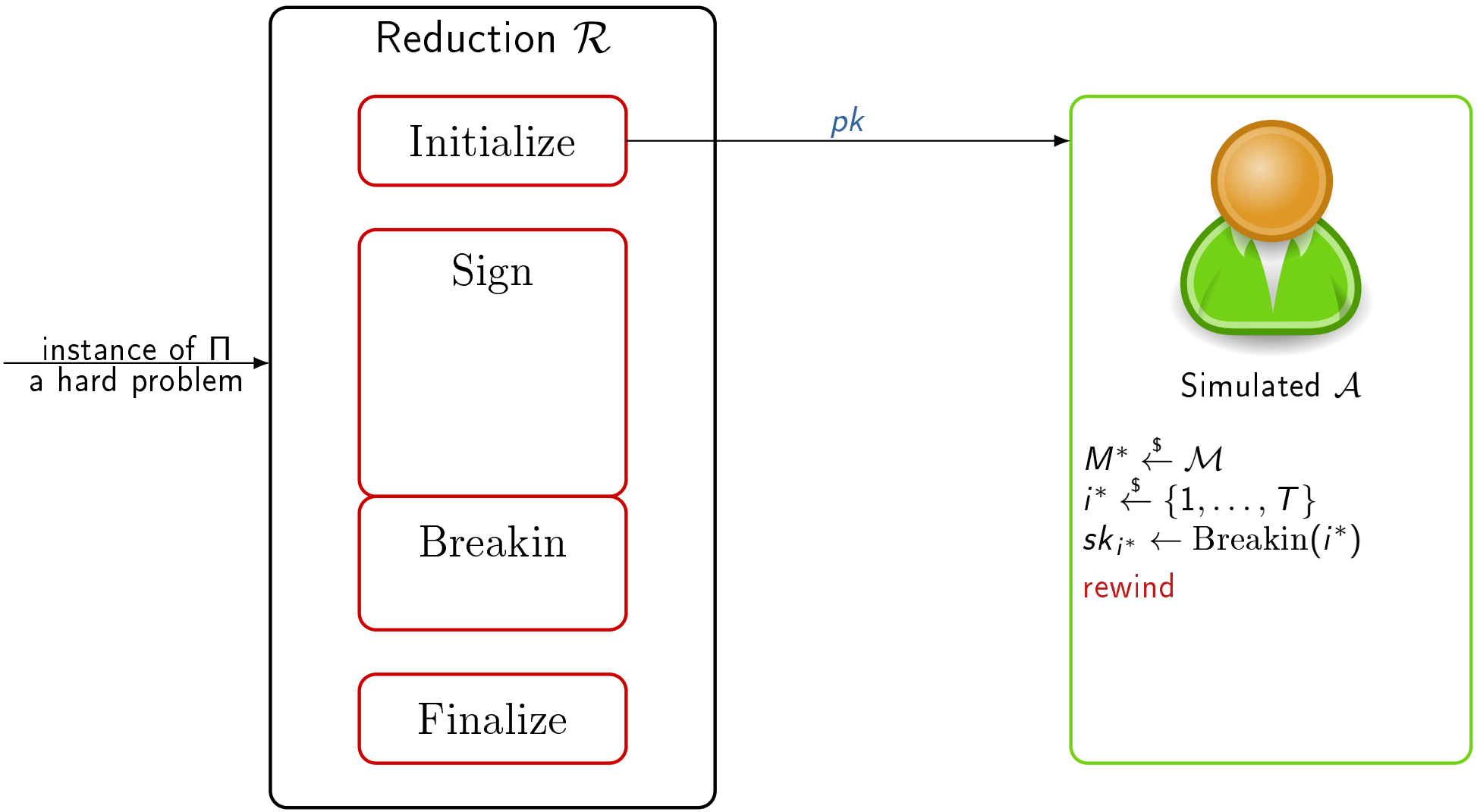
Optimality of Reductions



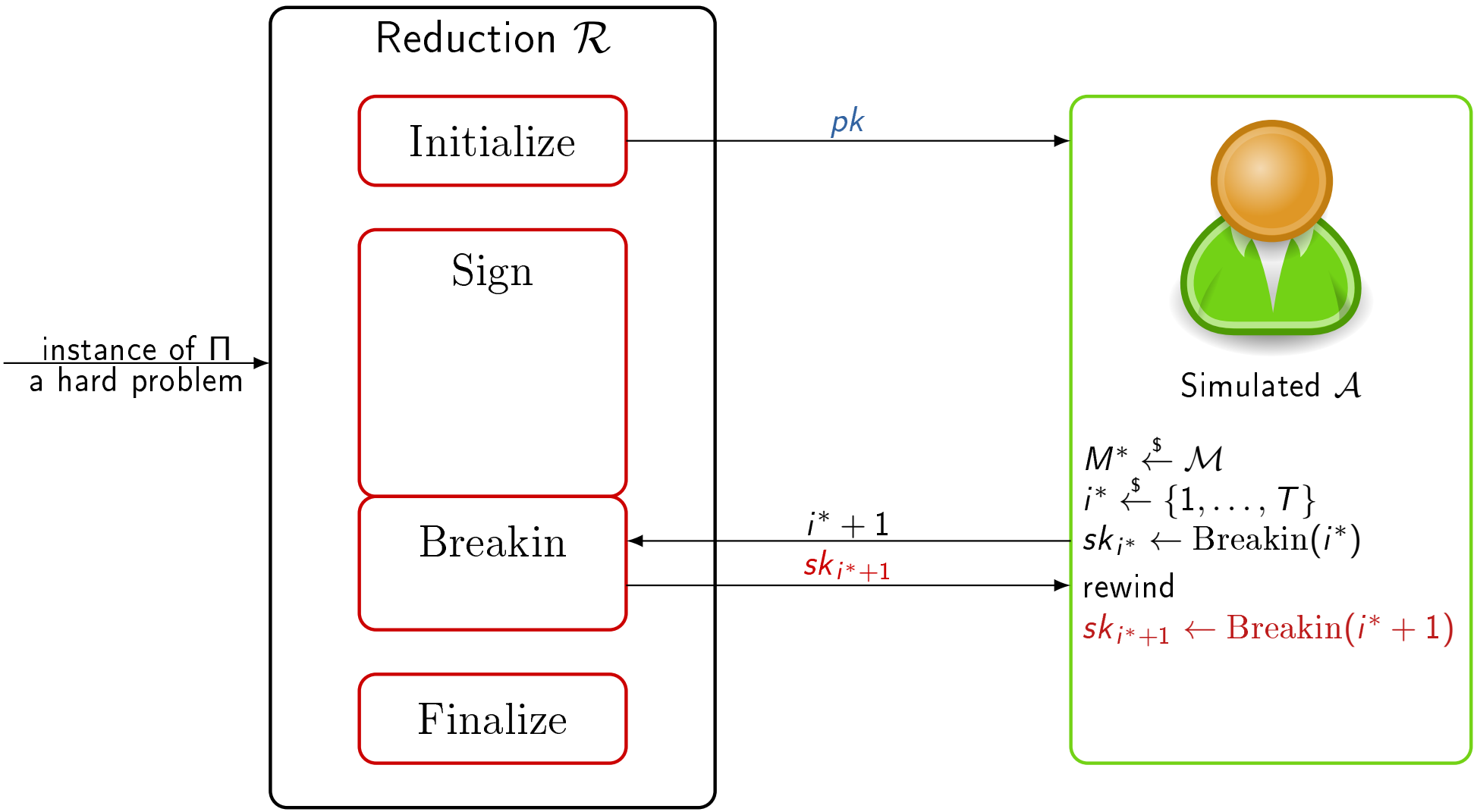
Optimality of Reductions



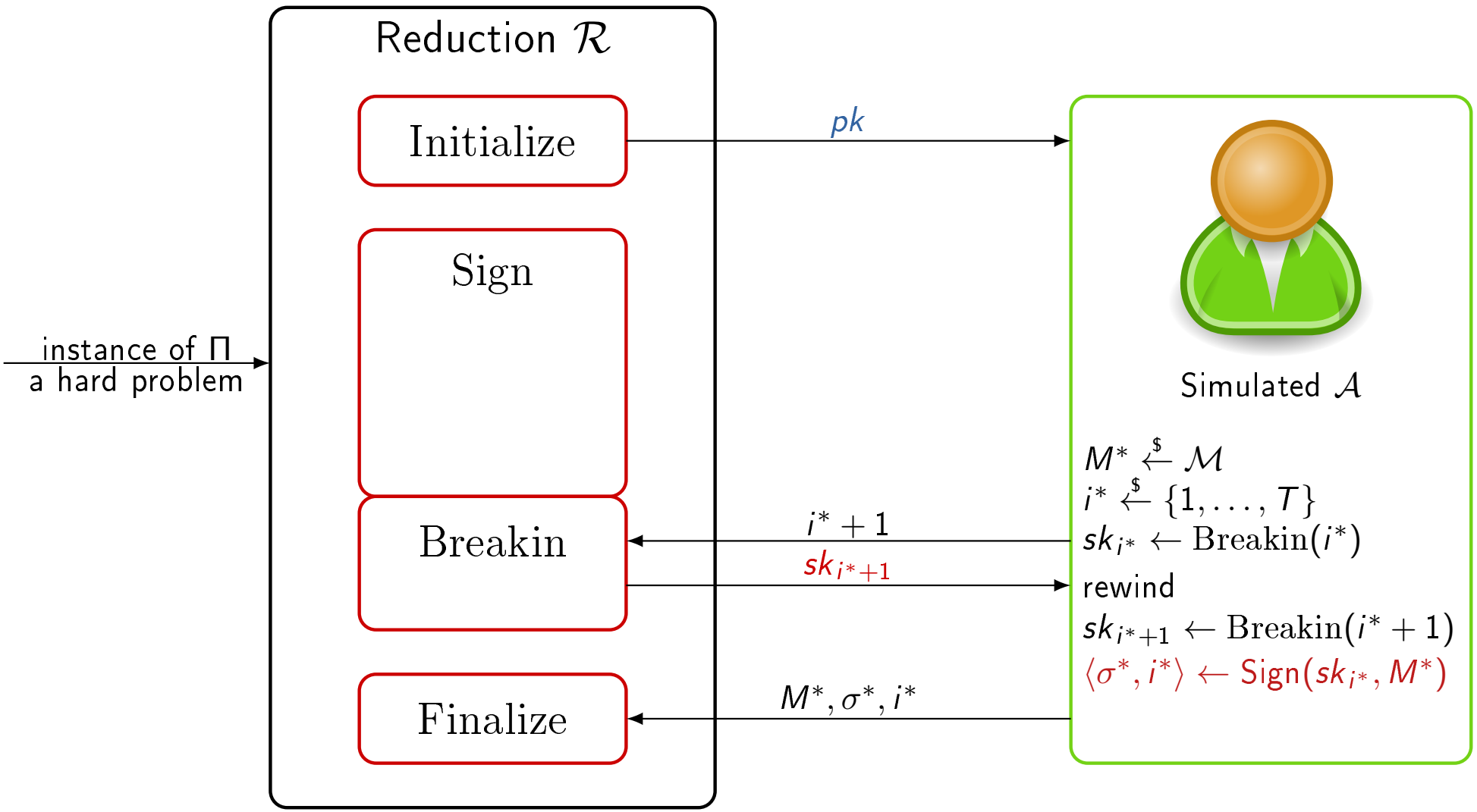
Optimality of Reductions



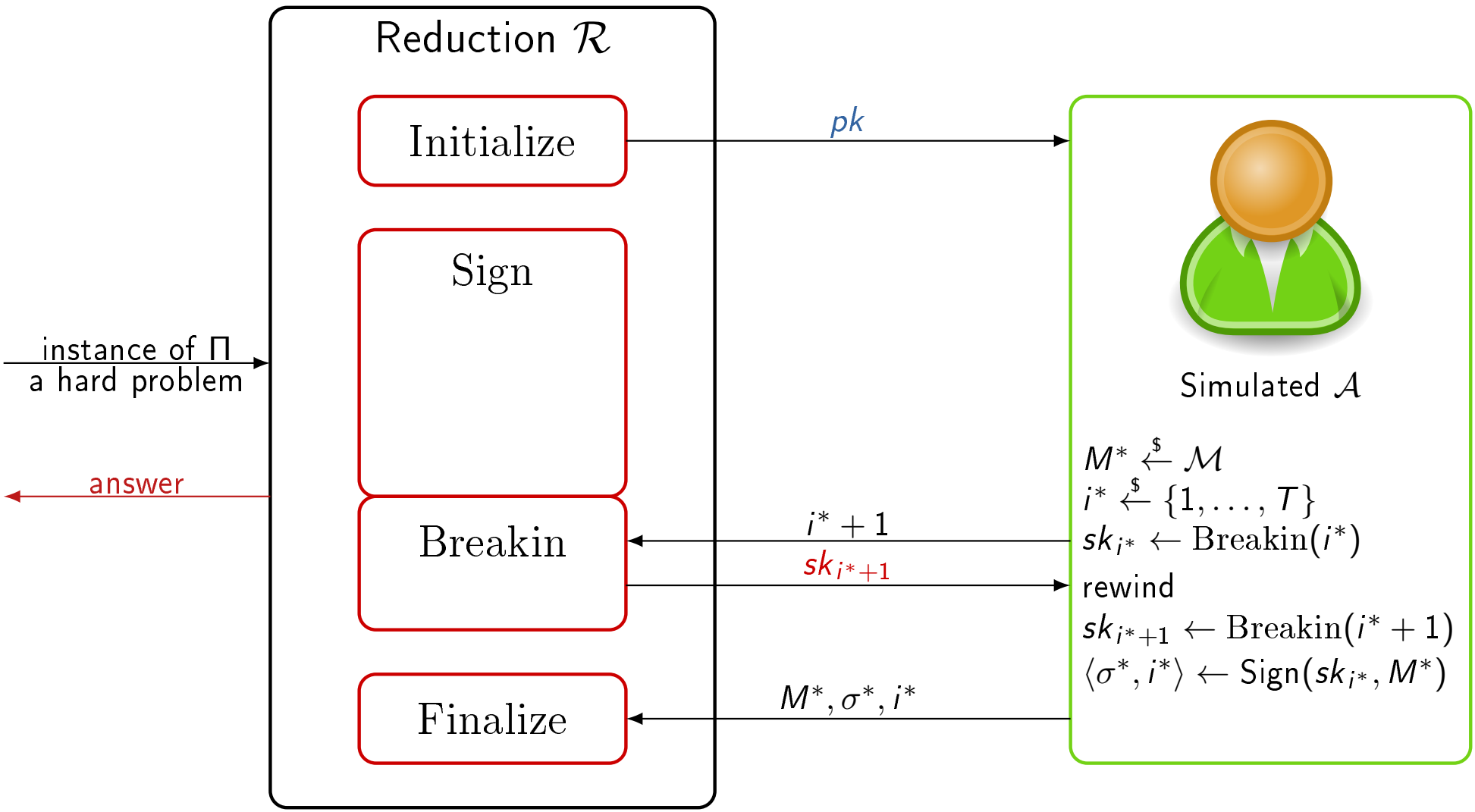
Optimality of Reductions



Optimality of Reductions



Optimality of Reductions



Optimality of Reductions

Theorem (approximate statement)

If $\mathcal{R}$ is such that:

> *if $\mathcal{A}$ $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}})$-breaks the forward-security of $\mathcal{FS}$, then $\mathcal{R}$ $(t_{\mathcal{R}}, \varepsilon_{\mathcal{R}})$-solves the hard problem Π.*

*and if we can check that sk_{i*} is a "valid" key ($\mathcal{FS}$ is said **key-verifiable**), then we can directly (t, ε)-solve Π with:*

$$t \approx 2t_{\mathcal{R}} \text{ and } \varepsilon \approx \varepsilon_{\mathcal{R}} - \frac{\varepsilon_{\mathcal{A}}}{T}.$$

Proof idea.

The reduction can only distinguish our simulated $\mathcal{A}$ from a real adversary if sk_{i*} is invalid and sk_{i*+1} is valid. We can show it happens at most with probability $1/T$. □

Optimality of Reductions

Theorem (approximate statement)

If $\mathcal{R}$ is such that:

> *if $\mathcal{A}$ $(t_{\mathcal{A}}, \varepsilon_{\mathcal{A}})$-breaks the forward-security of $\mathcal{FS}$, then $\mathcal{R}$ $(t_{\mathcal{R}}, \varepsilon_{\mathcal{R}})$-solves the hard problem Π.*

*and if we can check that sk_{i^*} is a "valid" key ($\mathcal{FS}$ is said **key-verifiable**), then we can directly (t, ε)-solve Π with:*

$$t \approx 2t_{\mathcal{R}} \text{ and } \varepsilon \approx \varepsilon_{\mathcal{R}} - \frac{\varepsilon_{\mathcal{A}}}{T}.$$

Corollary

For any key-verifiable forward-secure scheme, there does not exist fully tight black-box reduction.