Foundations of Adaptor Signatures

Paul Gerhart, Dominique Schröder, Pratik Soni, Sri AravindaKrishnan Thyagarajan

Once Upon A Time

Alice
(wants to buy a witness for a statement Y)

Y

Bob
(knows a witness for Y and wants to sell it)

y

wants to rely on minimum trust

does not like the ROM

Once Upon A Time

Alice
(wants to buy a witness for a statement Y)

$$Y$$

Bob
(knows a witness for Y and wants to sell it)

$$y$$

wants to rely on minimum trust

does not like the ROM

Once Upon A Time

Alice
(wants to buy a witness for a statement Y)

$$Y$$

Bob
(knows a witness for Y and wants to sell it)

$$y$$

wants to rely on minimum trust

does not like the ROM

Once Upon A Time

Alice
(wants to buy a witness for a statement Y)

$$Y$$

Bob
(knows a witness for Y and wants to sell it)

$$y$$

wants to rely on minimum trust

does not like the ROM

Adaptor Sigantures

Adaptor Signature Interfaces

Signature Scheme

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

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

$b \leftarrow \mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$

Hard Relation

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$b \leftarrow \mathsf{pVrfy}(\mathsf{pk}, m, \widetilde{\sigma}, Y)$

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

Adaptor Signature Interfaces

Signature Scheme

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

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

$b \leftarrow \mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$

Hard Relation

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$b \leftarrow \mathsf{pVrfy}(\mathsf{pk}, m, \widetilde{\sigma}, Y)$

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

Adaptor Signature Interfaces

Signature Scheme

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

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

$b \leftarrow \mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$

Hard Relation

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$b \leftarrow \mathsf{pVrfy}(\mathsf{pk}, m, \widetilde{\sigma}, Y)$

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

Adaptor Signature Interfaces

Signature Scheme

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

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

$b \leftarrow \mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$

Hard Relation

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$b \leftarrow \mathsf{pVrfy}(\mathsf{pk}, m, \widetilde{\sigma}, Y)$

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

Adaptor Signature Interfaces

Signature Scheme

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

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

$b \leftarrow \mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$

Hard Relation

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$b \leftarrow \mathsf{pVrfy}(\mathsf{pk}, m, \widetilde{\sigma}, Y)$

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

Adaptor Signature Interfaces

Signature Scheme

$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(\lambda)$

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

$b \leftarrow \mathsf{Vrfy}(\mathsf{pk}, m, \sigma)$

Hard Relation

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

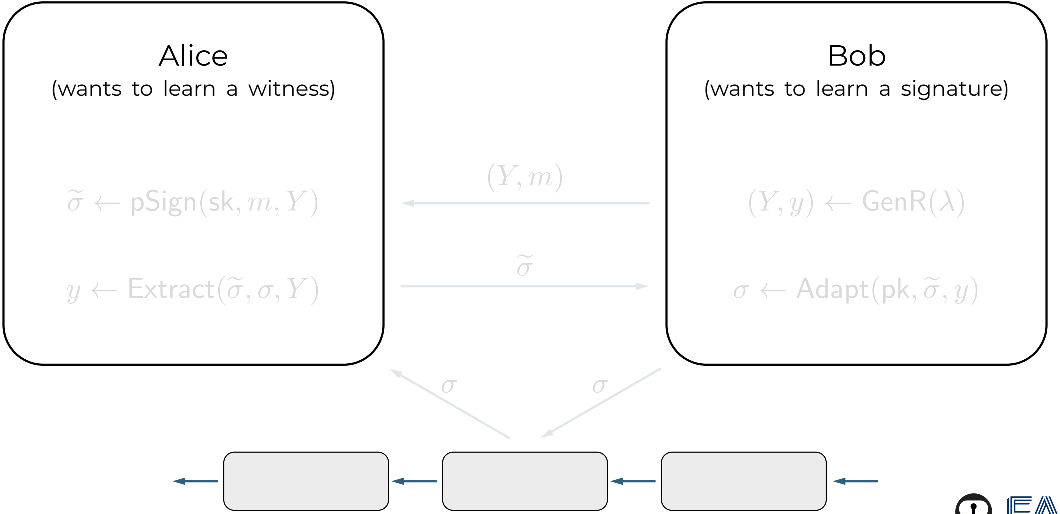
$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$b \leftarrow \mathsf{pVrfy}(\mathsf{pk}, m, \widetilde{\sigma}, Y)$

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

Fair Exchange using Adaptor Signatures



Fair Exchange using Adaptor Signatures

Alice
(wants to learn a witness)

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

(Y, m)

$\widetilde{\sigma}$

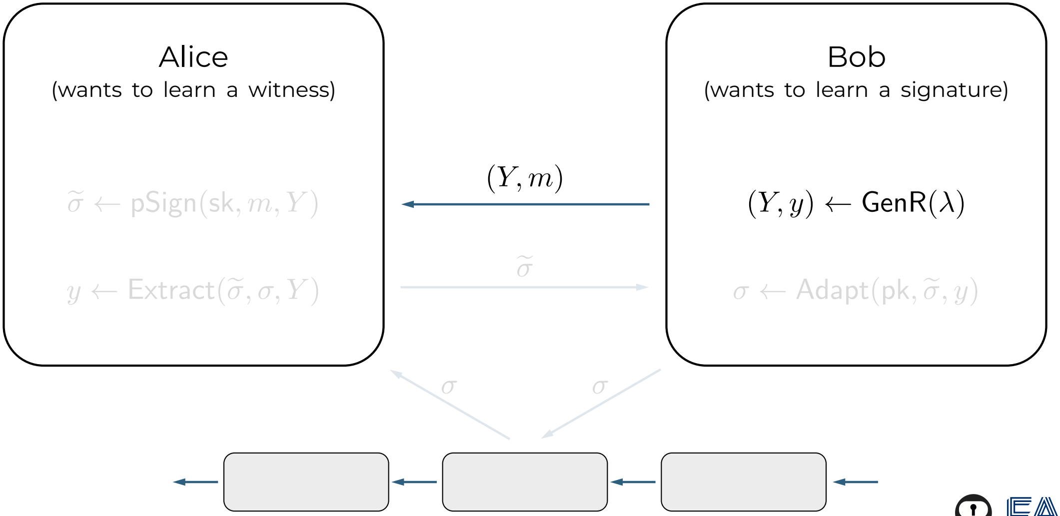
Bob
(wants to learn a signature)

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

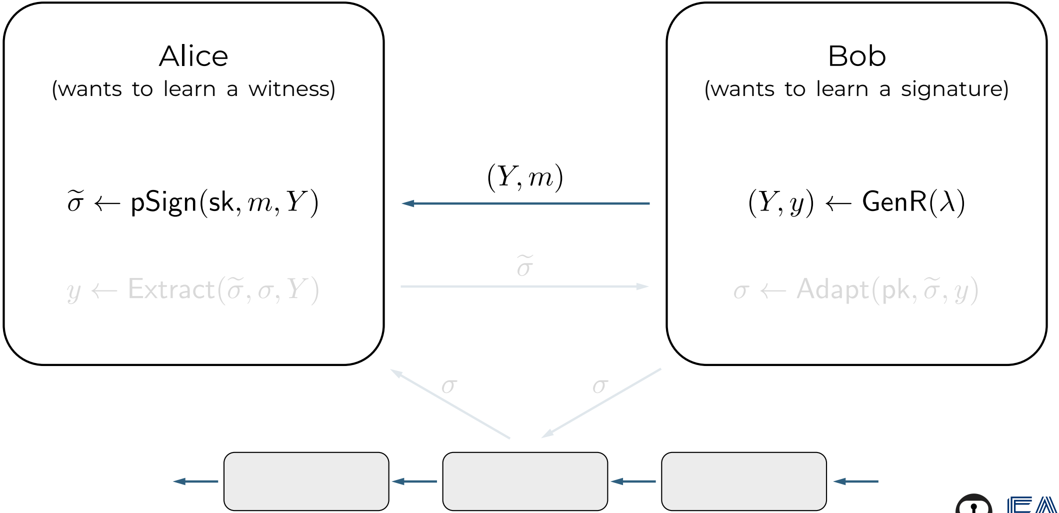
$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

σ σ

Fair Exchange using Adaptor Signatures



Fair Exchange using Adaptor Signatures



Fair Exchange using Adaptor Signatures

Alice
(wants to learn a witness)

$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$y \leftarrow \mathsf{Extract}(\widetilde{\sigma}, \sigma, Y)$

(Y, m)

$\widetilde{\sigma}$

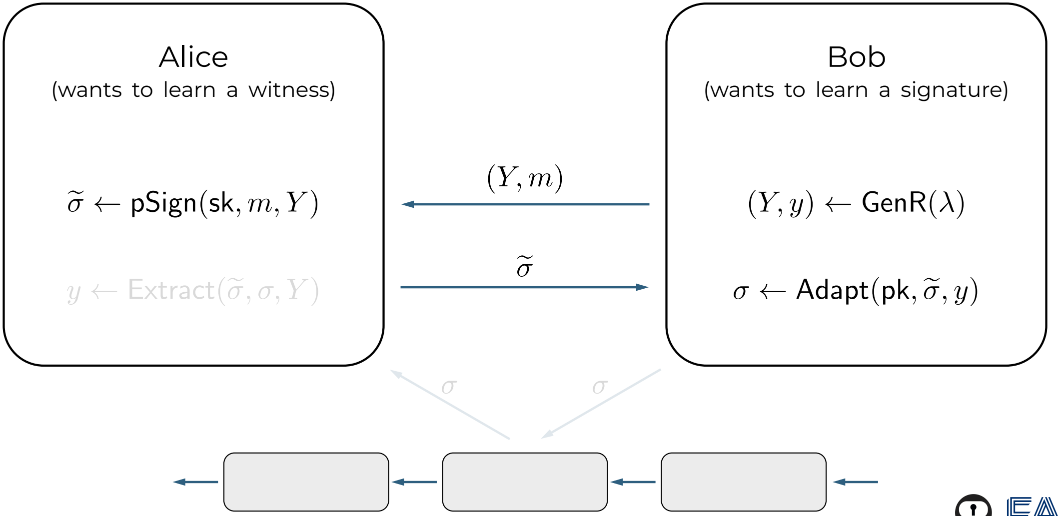
Bob
(wants to learn a signature)

$(Y, y) \leftarrow \mathsf{GenR}(\lambda)$

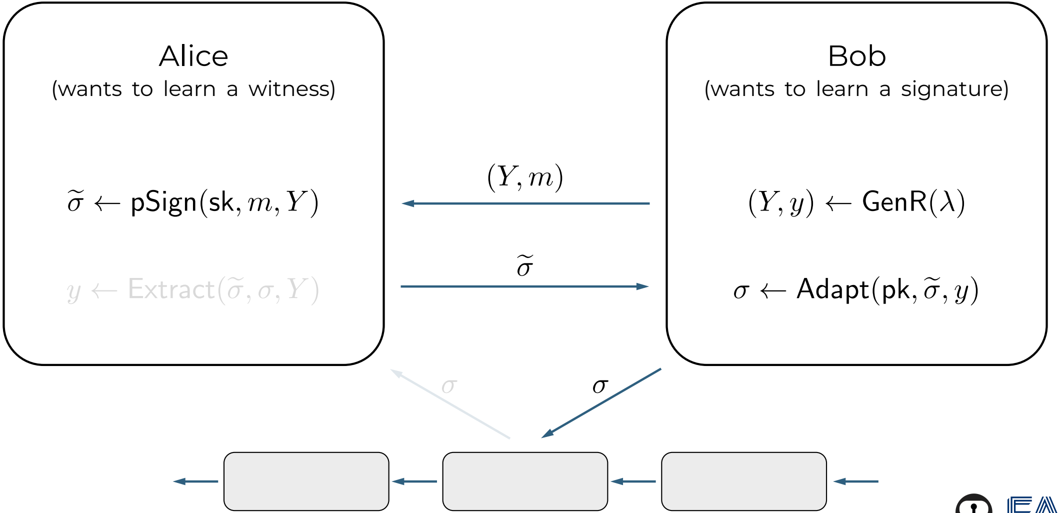
$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

σ σ

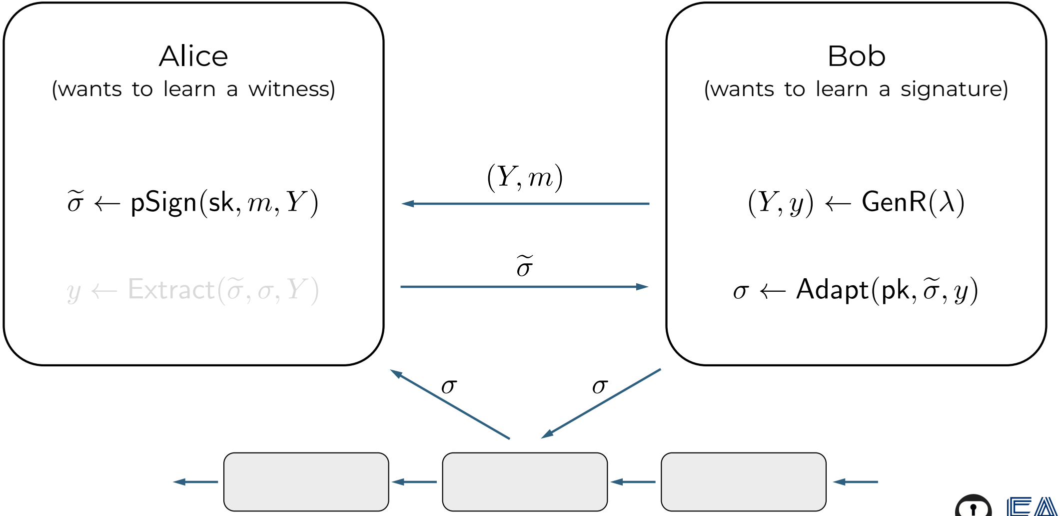
Fair Exchange using Adaptor Signatures



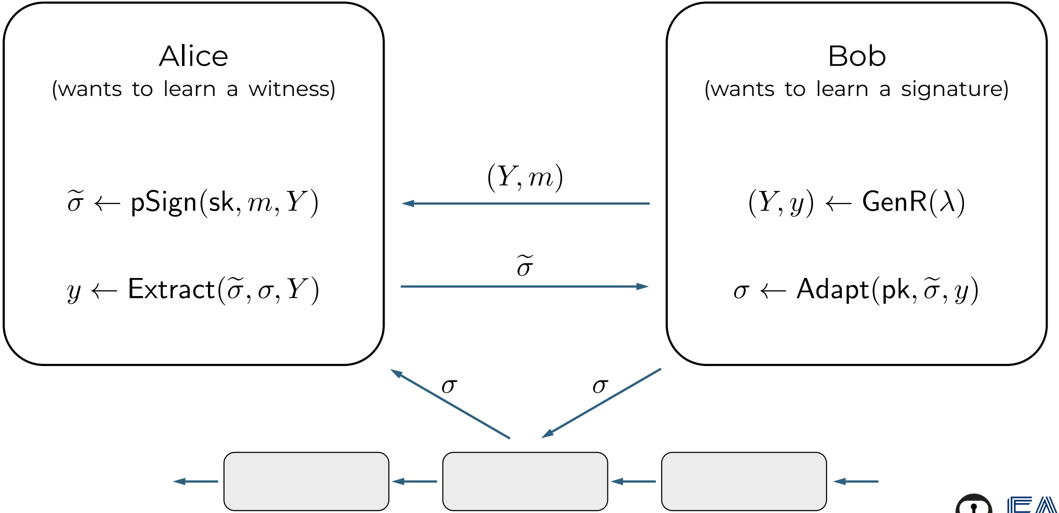
Fair Exchange using Adaptor Signatures



Fair Exchange using Adaptor Signatures



Fair Exchange using Adaptor Signatures



Adaptor Signatures in the Literature

- Introduced by Andrew Poelstra 2017
- Formally defined by Aumayr et al. [AEEFHMMR'21]
- Applications:
  - (Generalized) Payment Channels [AEEFHMMR'21]
  - (Blind) Coin Mixing [GMMMTT'22, QPMSESELYY'23]
  - Oracle-Based Payments [MTVFMM'23]
- Theory:
  - PQ Adaptors [TMM'20]
  - Stronger Definitions [DOY'22]

Adaptor Signatures in the Literature

- Introduced by Andrew Poelstra 2017
- Formally defined by Aumayr et al. [AEEFHMMR’21]
- Applications:
  - (Generalized) Payment Channels [AEEFHMMR’21]
  - (Blind) Coin Mixing [GMMMTT’22, QPMSESELYY’23]
  - Oracle-Based Payments [MTVFMM’23]
- Theory:
  - PQ Adaptors [TMM’20]
  - Stronger Definitions [DOY’22]

Adaptor Signatures in the Literature

- Introduced by Andrew Poelstra 2017
- Formally defined by Aumayr et al. [AEEFHMMR'21]
- Applications:
  - (Generalized) Payment Channels [AEEFHMMR'21]
  - (Blind) Coin Mixing [GMMMTT'22, QPMSESELYY'23]
  - Oracle-Based Payments [MTVFMM'23]
- Theory:
  - PQ Adaptors [TMM'20]
  - Stronger Definitions [DOY'22]

Theoretical Challenges

Given a signature scheme, building a secure adaptor signature is hard.

There is no secure adaptor signature in the standard model.

Theoretical Challenges

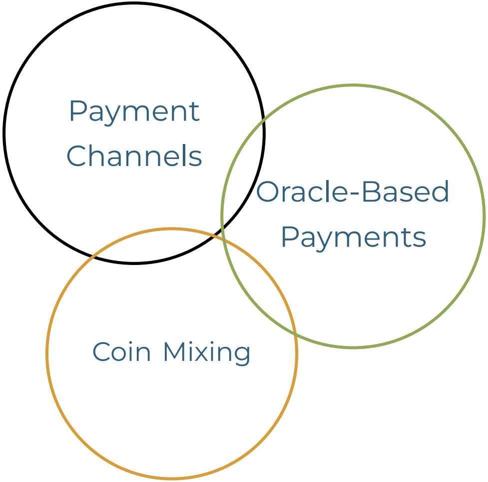
Given a signature scheme, building a secure adaptor signature is hard.

There is no secure adaptor signature in the standard model.

Practical Challenges

Adaptor signatures were formalized to build **payment channels**.

This formalization **does not match** the most recent applications.



Practical Challenges

Adaptor signatures were formalized to build **payment channels**.

This formalization **does not match** the most recent applications.

Payment Channels

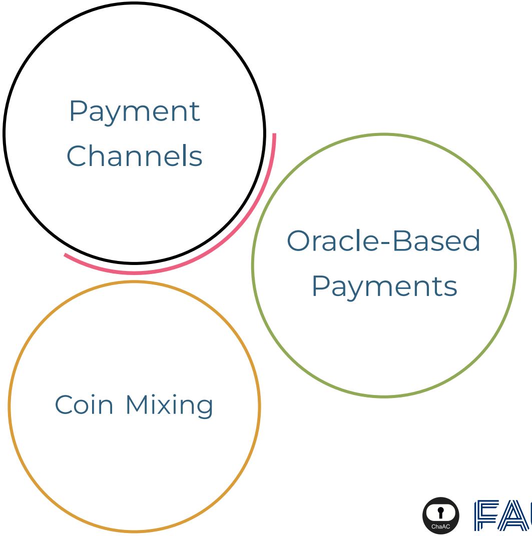
Oracle-Based Payments

Coin Mixing

Practical Challenges

Adaptor signatures were formalized to build **payment channels**.

This formalization **does not match** the most recent applications.



Our Contribution

Gaps

Definitions

Constructions

Transparent Reductions

Our Contribution

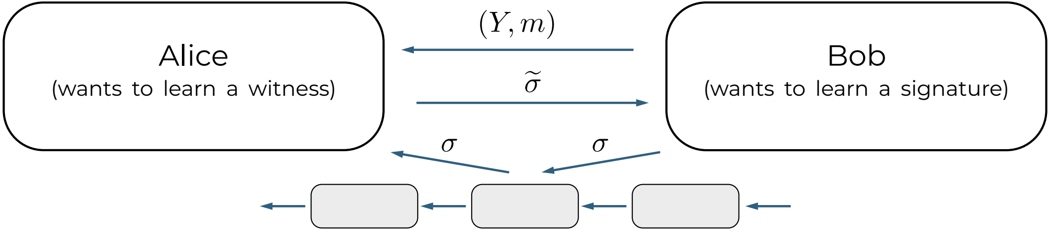
Gaps

Definitions

Constructions

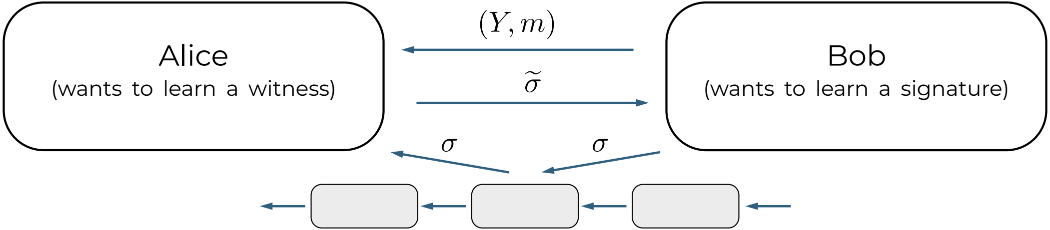
Transparent Reductions

Adaptor Signature Formalization



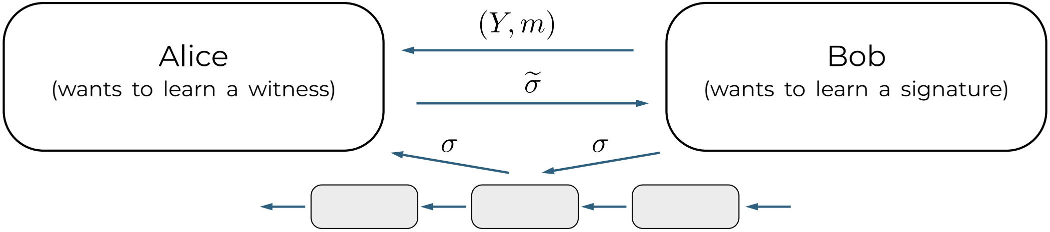
- The definition is a **one-shot experiment**
  - The adversary can only learn a single challenge pre-signature
- Adaptor signatures achieve only **existential unforgeability**, even if the signature scheme is strongly unforgeable
- The pre-signer **cannot influence** the statement

Adaptor Signature Formalization



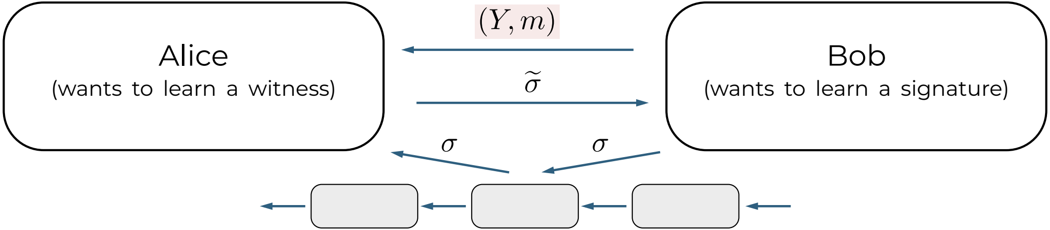
- The definition is a **one-shot experiment**
  - The adversary can only learn a single challenge pre-signature
- Adaptor signatures achieve only **existential unforgeability**, even if the signature scheme is strongly unforgeable
- The pre-signer **cannot influence** the statement

Adaptor Signature Formalization



- The definition is a **one-shot experiment**
  - The adversary can only learn a single challenge pre-signature
- Adaptor signatures achieve only **existential unforgeability**, even if the signature scheme is strongly unforgeable
- The pre-signer **cannot influence** the statement

Adaptor Signature Formalization



- The definition is a **one-shot experiment**
  - The adversary can only learn a single challenge pre-signature
- Adaptor signatures achieve only **existential unforgeability**, even if the signature scheme is strongly unforgeable
- The pre-signer **cannot influence** the statement

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \ldots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \ldots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \ldots, y_N)}$ **Oracles**

Oracles

testify for events

$\downarrow y_i$

$\xrightarrow{\widetilde{\sigma}_{1 \le i \le M}}$ **Bob**

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$$\forall i \in \{1, \ldots, M\}:$$
$$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$$

$$\forall i \in \{1, \ldots, M\}:$$
$$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$$

$(y_1, \ldots, y_N)$

$\widetilde{\sigma}_{1 \le i \le M}$

Oracles

testify for events

y_i

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \ldots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \ldots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \ldots, y_N)}$

$\xrightarrow{\widetilde{\sigma}_{1 \leq i \leq M}}$

Oracles

testify for events

$\downarrow y_i$

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \ldots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \ldots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \ldots, y_N)}$

Oracles

testify for events

$\downarrow y_i$

$\xrightarrow{\widetilde{\sigma}_{1 \leq i \leq M}}$

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \dots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \dots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \dots, y_N)}$

Oracles

testify for events

$\downarrow y_i$

$\xrightarrow{\widetilde{\sigma}_{1 \le i \le M}}$

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \ldots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \ldots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \ldots, y_N)}$ **Oracles**

testify for events

$\downarrow y_i$

$\xrightarrow{\widetilde{\sigma}_{1 \le i \le M}}$ **Bob**

obtains pre-signatures from Alice and requests the oracle for testimony

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \ldots, M\}$:

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \ldots, M\}$:

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \ldots, y_N)}$

Oracles

testify for events

$\downarrow y_i$

$\xrightarrow{\widetilde{\sigma}_{1 \le i \le M}}$

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \dots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \dots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$(y_1, \dots, y_N)$

$\widetilde{\sigma}_{1 \le i \le M}$

Oracles

testify for events

y_i

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\cancel{\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)}$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$\forall i \in \{1, \ldots, M\}:$

$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$

$\forall i \in \{1, \ldots, M\}:$

$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$

$\xrightarrow{(y_1, \ldots, y_N)}$

Oracles

testify for events

y_i

$\xrightarrow{\widetilde{\sigma}_{1 \le i \le M}}$

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$\cancel{\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)}$

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

$$\forall i \in \{1, \ldots, M\}:$$
$$(Y_i, y_i) \leftarrow \mathsf{RGen}(1^\lambda)$$

$$\forall i \in \{1, \ldots, M\}:$$
$$\widetilde{\sigma}_i \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y_i)$$

$$\xrightarrow{(y_1, \ldots, y_N)}$$

Oracles

testify for events

$$\cancel{y_i}$$

$$\xrightarrow{\widetilde{\sigma}_{1 \leq i \leq M}}$$

Bob

obtains pre-signatures from Alice and requests the oracle for testimony

$$\cancel{\sigma \leftarrow \mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}_i, y_i)}$$

$$\sigma \leftarrow \widetilde{\sigma}_1 \oplus \widetilde{\sigma}_2$$

Overview

Gaps

Definitions

Constructions

Transparent Reductions

Theoretical Challenges

Can we generically transform signatures into adaptor signatures?

Can we find an adaptor signature scheme in the standard model?

Theoretical Challenges

Can we generically transform signatures into adaptor signatures?

Can we find an adaptor signature scheme in the standard model?

Dichotomic Signatures: Pre-Signing

$\mathsf{pSign}(\mathsf{sk}, m, Y)$

$1: r \leftarrow\$ \mathbb{Z}_p; R \leftarrow g^r$

$2: h \leftarrow \mathcal{H}(\mathsf{pk}, R \cdot Y, m)$

$3: \mathbf{return}\ (R \cdot Y, \mathsf{sk} \cdot h + r)$

- The signature consists of two parts

$$\sigma = (\sigma_1, \sigma_2)$$

- The signature uses a homomorphic one-way function

$$R = \mathsf{OWF}(r); Y = \mathsf{OWF}(y); r, y \in \mathbb{Z}_p$$

- One part can be computed using

$$\sigma_1 = \Sigma_1(\mathsf{sk}, m; \mathsf{OWF}(r) \cdot \mathsf{OWF}(y))$$

- The other part can be computed using

$$\sigma_2 = \Sigma_2(\mathsf{sk}, m; r)$$

Dichotomic Signatures: Pre-Signing

$\mathsf{pSign}(\mathsf{sk}, m, Y)$

$1: r \leftarrow\!\!\$\ \mathbb{Z}_p; R \leftarrow g^r$

$2: h \leftarrow \mathcal{H}(\mathsf{pk}, R \cdot Y, m)$

$3: \textbf{return } (R \cdot Y, \mathsf{sk} \cdot h + r)$

- The signature consists of two parts

$$\sigma = (\sigma_1, \sigma_2)$$

- The signature uses a homomorphic one-way function

$$R = \mathsf{OWF}(r); Y = \mathsf{OWF}(y); r, y \in \mathbb{Z}_p$$

- One part can be computed using

$$\sigma_1 = \Sigma_1(\mathsf{sk}, m; \mathsf{OWF}(r) \cdot \mathsf{OWF}(y))$$

- The other part can be computed using

$$\sigma_2 = \Sigma_2(\mathsf{sk}, m; r)$$

Dichotomic Signatures: Pre-Signing

$\mathsf{pSign}(\mathsf{sk}, m, Y)$

1 : $r \leftarrow\!\$\; \mathbb{Z}_p; R \leftarrow g^r$

2 : $h \leftarrow \mathcal{H}(\mathsf{pk}, R \cdot Y, m)$

3 : $\textbf{return}\ (R \cdot Y, \mathsf{sk} \cdot h + r)$

- The signature consists of two parts

$$\sigma = (\sigma_1, \sigma_2)$$

- The signature uses a homomorphic one-way function

$$R = \mathsf{OWF}(r); Y = \mathsf{OWF}(y); r, y \in \mathbb{Z}_p$$

- One part can be computed using

$$\sigma_1 = \Sigma_1(\mathsf{sk}, m; \mathsf{OWF}(r) \cdot \mathsf{OWF}(y))$$

- The other part can be computed using

$$\sigma_2 = \Sigma_2(\mathsf{sk}, m; r)$$

Dichotomic Signatures: Pre-Signing

$\mathsf{pSign}(\mathsf{sk}, m, Y)$

1 : $r \leftarrow\$ \mathbb{Z}_p; R \leftarrow g^r$

2 : $h \leftarrow \mathcal{H}(\mathsf{pk}, R \cdot Y, m)$

3 : $\textbf{return } (R \cdot Y, \mathsf{sk} \cdot h + r)$

- The signature consists of two parts

$$\sigma = (\sigma_1, \sigma_2)$$

- The signature uses a homomorphic one-way function

$$R = \mathsf{OWF}(r); Y = \mathsf{OWF}(y); r, y \in \mathbb{Z}_p$$

- One part can be computed using

$$\sigma_1 = \Sigma_1(\mathsf{sk}, m; \mathsf{OWF}(r) \cdot \mathsf{OWF}(y))$$

- The other part can be computed using

$$\sigma_2 = \Sigma_2(\mathsf{sk}, m; r)$$

Dichotomic Signatures: Adapt/Extract

$\mathsf{Adapt}(\mathsf{pk}, \widetilde{\sigma}, y)$

1 : parse $\widetilde{\sigma}$ as $(\widetilde{\sigma}_1, \widetilde{\sigma}_2)$

2 : **return** $(\widetilde{\sigma}_1, \widetilde{\sigma}_2 + y)$

$\mathsf{Extract}(Y, \widetilde{\sigma}, \sigma)$

1 : parse $\widetilde{\sigma}$ as $(\widetilde{\sigma}_1, \widetilde{\sigma}_2)$

2 : parse σ as (σ_1, σ_2)

3 : **return** $\sigma_2 - \widetilde{\sigma}_2$

- The second part of the signature is homomorphic in the randomness

$$\Sigma_2(\mathsf{sk}, m; r) + y = \Sigma_2(\mathsf{sk}, m; r + y)$$

Dichotomic Signatures: A Definition

A signature scheme w.r.t. a homomorphic one-way function OWF is dichotomic; if

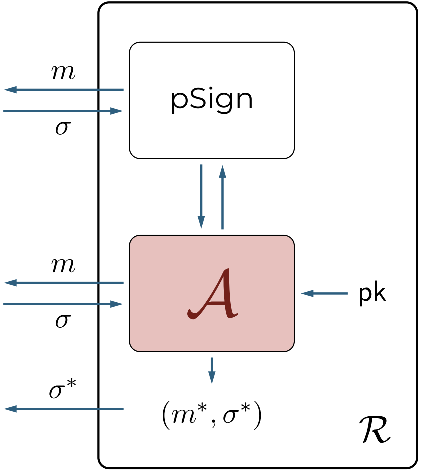
- It is decomposable

$$\sigma = (\sigma_1, \sigma_2) = (\Sigma_1(\mathsf{sk}, m; \mathsf{OWF}(r)), \Sigma_2(\mathsf{sk}, m; r))$$

- It is homomorphic in the randomness

$$\Sigma_2(\mathsf{sk}, m; r) + y = \Sigma_2(\mathsf{sk}, m; r + y)$$

Proving Security

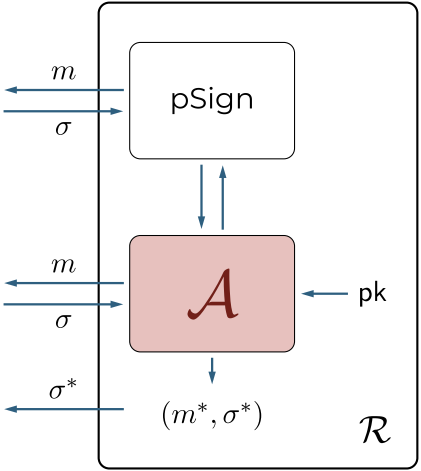


- We need to simulate pre-signatures to the adversary
- We cannot use the random oracle

Converting a signature into a pre-signature seems impossible

- We cannot reduce to the strong unforgeability directly

Proving Security

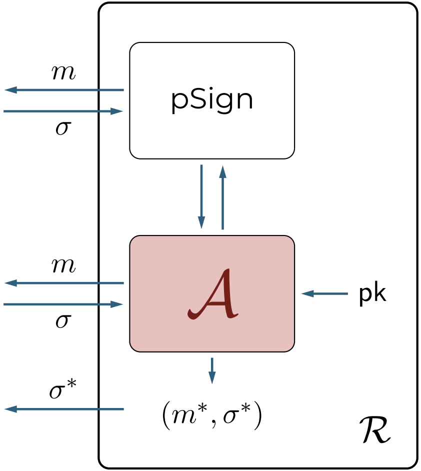


- We need to simulate pre-signatures to the adversary
- We cannot use the random oracle

Converting a signature into a pre-signature seems impossible

- We **cannot** reduce to the strong unforgeability directly

Proving Security

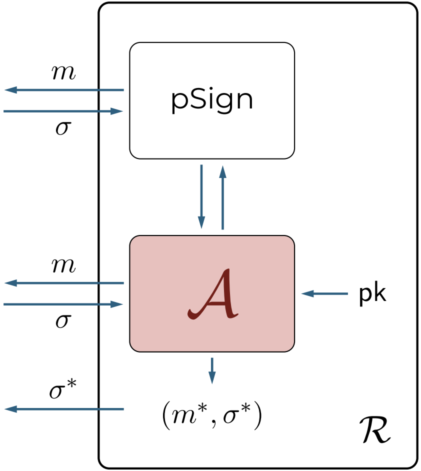


- We need to simulate pre-signatures to the adversary
- We cannot use the random oracle

Converting a signature into a pre-signature seems impossible

- We cannot reduce to the strong unforgeability directly

Proving Security



- We need to simulate pre-signatures to the adversary
- We cannot use the random oracle

Converting a signature into a pre-signature seems impossible

- We **cannot** reduce to the strong unforgeability directly

Overview

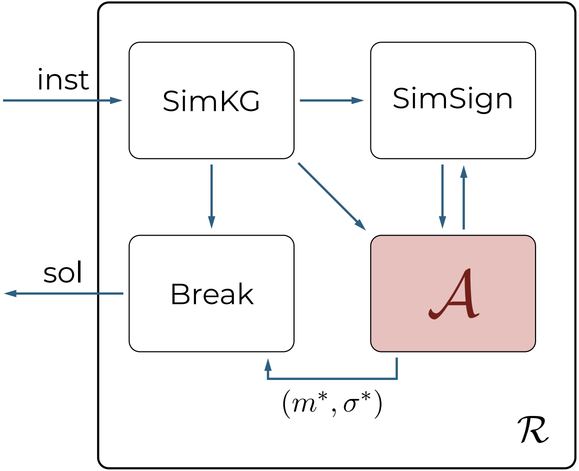
Gaps

Definitions

Constructions

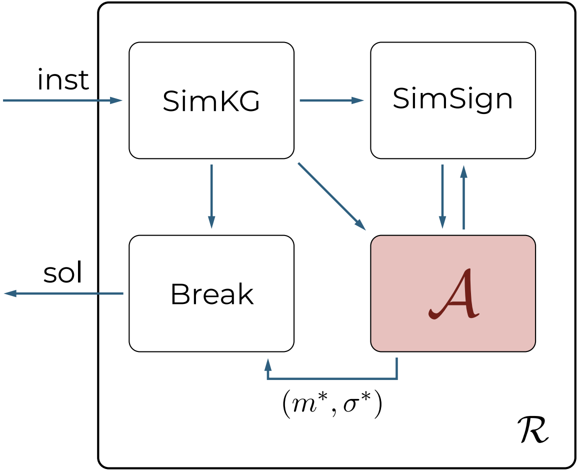
Transparent Reductions

Transparent Reductions



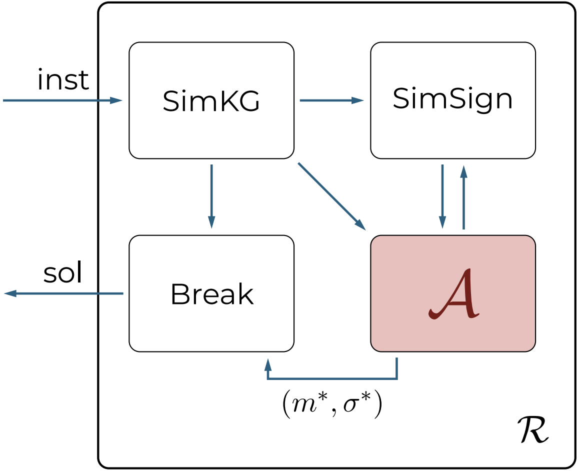
- **SimKG**:
  - Simulates keys $(\mathsf{simSK}, \mathsf{simPK})$
- **SimSign**:
  - Simulates signatures using simSK
- **Break**:
  - Solve problem instance using valid forgery

Transparent Reductions



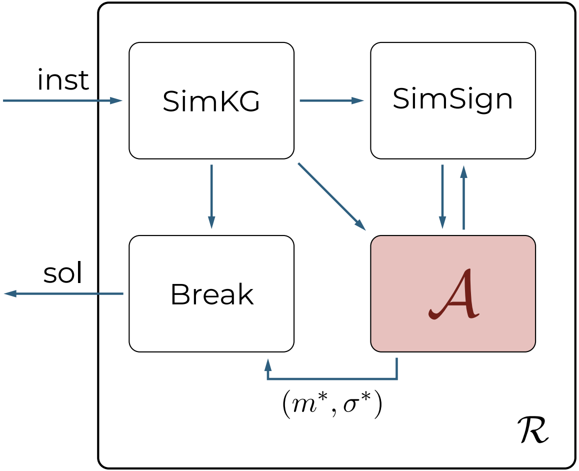
- **SimKG**:
  - Simulates keys $(\mathsf{simSK}, \mathsf{simPK})$
- **SimSign**:
  - Simulates signatures using simSK
- **Break**:
  - Solve problem instance using valid forgery

Transparent Reductions



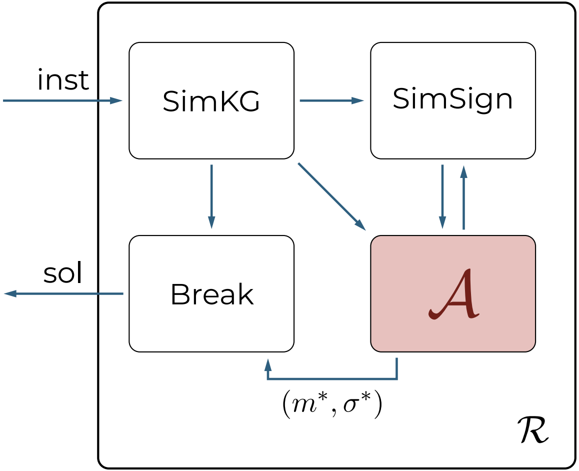
- **SimKG**:
  - Simulates keys $(\mathsf{simSK}, \mathsf{simPK})$
- **SimSign**:
  - Simulates signatures using simSK
- **Break**:
  - Solve problem instance using valid forgery

Simulating Pre Signatures



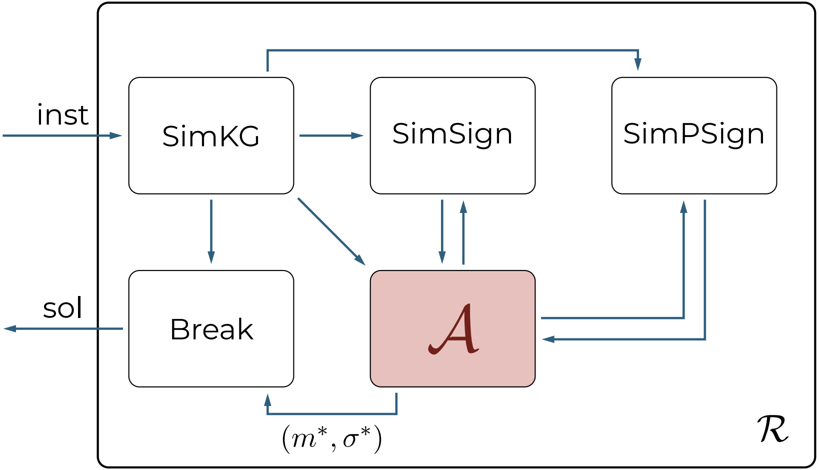
- So far, we **can**:
  - Simulate keys
  - Provide a signature oracle
  - Break the problem instance using a forgery
- So far, we **cannot**:
  - Provide a pre-signature oracle

Simulating Pre Signatures



- So far, we **can**:
  - Simulate keys
  - Provide a signature oracle
  - Break the problem instance using a forgery
- So far, we **cannot**:
  - Provide a pre-signature oracle

Simulatable Transparent Reductions



A Framework For Adaptor Signatures

A secure adaptor signature scheme requires the following three checks:

- The signature scheme is **dichotomic**
- There is a **transparent reduction** from the strong unforgeability to an underlying hard problem
- We can simulate a pre-signature oracle **(simulatability)**

Conclusion

Gaps

Definitions

Constructions

Transparent Reductions