Foundations of Adaptor Signatures

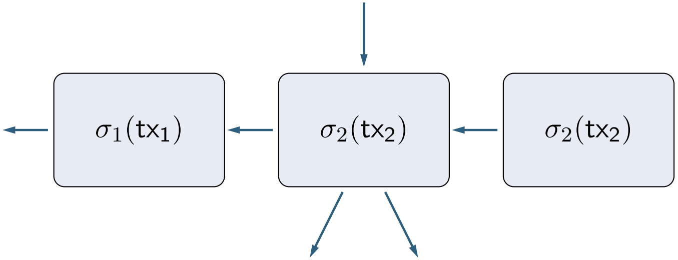
Paul Gerhart, Dominique Schröder, Pratik Soni, Sri AravindaKrishnan Thyagarajan

FAU

THE UNIVERSITY OF SYDNEY

Adaptor Signatures

Scriptless Scripts



- Execute a smart contract off-chain
- Only post a single singature on a transaction on-chain
- Benefit: there is no computational overhead on-chain and they are compatible to most existing blockchains

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

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

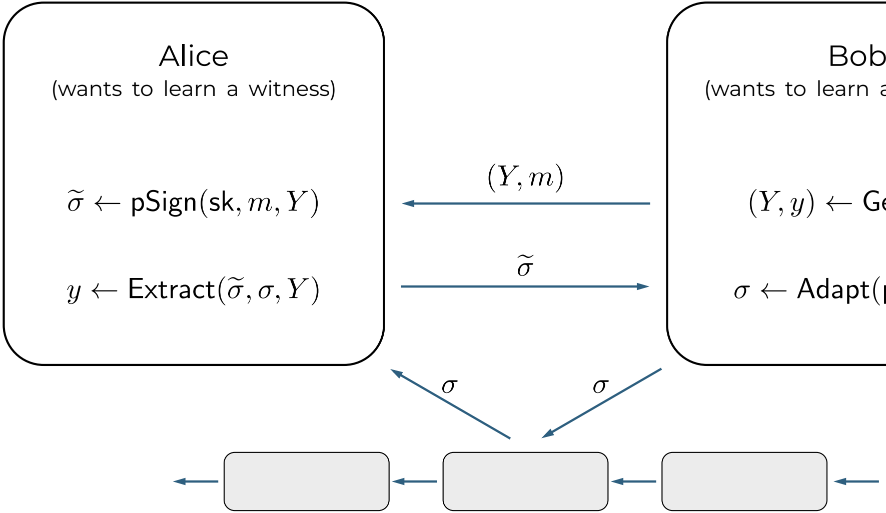
$\widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

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

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

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

Fair Exchange using Adaptor Signatures



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

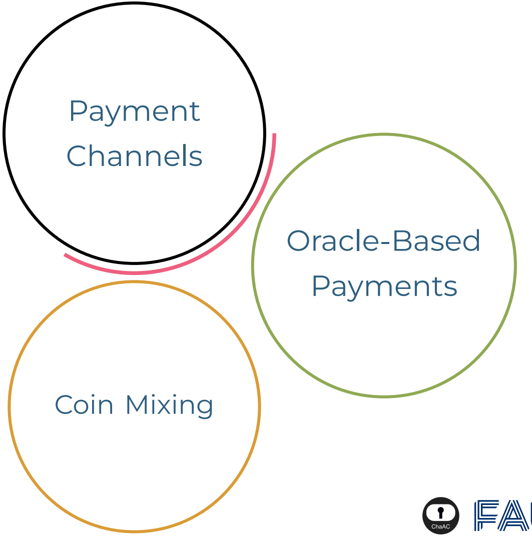
Given a signature scheme, building a secure adaptor signature is hard.

There is no secure adaptor signature in the standard model.

Practical Challenges

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

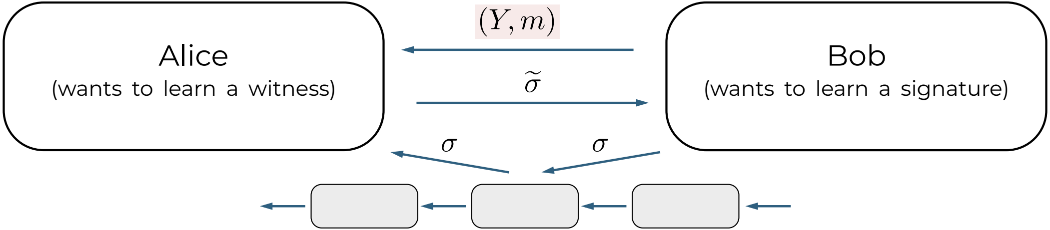
This formalization **does not match** the most recent applications.



Our Contribution

- Gaps
- Definitions
- Constructions
- Transparent Reductions

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 Unforgeability

$\mathsf{aSigForge}(\lambda)$

$1: (\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KGen}(\lambda); \mathcal{Q} := \emptyset$

$2: m^* \leftarrow \mathcal{A}^{\mathsf{Sign},\mathsf{pSign}}(\mathsf{pk})$

$3: (Y, y) \leftarrow \mathsf{RGen}(\lambda)$

$4: \widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m^*, Y)$

$5: \sigma^* \leftarrow \mathcal{A}^{\mathsf{Sign},\mathsf{pSign}}(\widetilde{\sigma}, Y)$

$6: \textbf{return } m^* \notin \mathcal{Q} \land \mathsf{Vrfy}(\mathsf{pk}, m^*, \sigma^*)$

$\mathsf{Sign}(m)$

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

$\mathcal{Q} := \mathcal{Q} \cup \{m\}$

$\textbf{return } \sigma$

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

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

$\mathcal{Q} := \mathcal{Q} \cup \{m\}$

$\textbf{return } \widetilde{\sigma}$

The adversary learns **a single** pre-signature on m^*.

Leaky Adaptor Signatures

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

$1: \widetilde{\sigma} \leftarrow \mathsf{pSign}(\mathsf{sk}, m, Y)$

$2: \sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$

$3: r_0 \leftarrow \mathcal{H}(\mathsf{sk}, m)$

$4: r_1 := r_0 \oplus \sigma$

$5: b \leftarrow\!\!\$\ \{0, 1\}$

$6: \mathbf{return}\ (\widetilde{\sigma}, r_b)$

- Learning a single pre-signature on the challenge message m does not reveal any information
- The second pre-signature on m **leaks a fresh valid signature** with a probability $1/2$.

This is not a problem for **payment channels** (only a single pre-signature per transaction is exchanged) but breaks other applications.

Oracle-Based Conditional Payments [MTVFMS'22]

Alice

sends a payment when the oracle testifies for an event

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

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

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

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

$(y_1, \ldots, y_N)$

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

Oracles

testify for events

y_i

Bob

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

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

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

Overview

Gaps

Definitions

Constructions

Transparent Reductions

Oracle-Based Conditional Payments (II)

Alice
sends a payment when the oracle testifies an event

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

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

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

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

- Alice computes both the **statement and pre-signature**
- This scenario is not covered by existing definitions
- A valid pre-signature w.r.t. a malicious statement generally **cannot** be adapted

Unadaptable Adaptor Signatures

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

$$
\begin{aligned}
&\textbf{if } Y \notin \mathcal{L}_{\mathsf{Rel}} \textbf{ then} \\
&\quad \tilde{\sigma} := \bot \\
&\textbf{else } \tilde{\sigma} := \mathsf{pSign}(\mathsf{sk}, m, Y) \\
&\textbf{return } \tilde{\sigma}
\end{aligned}
$$

$\mathsf{pVrfy}'(\mathsf{pk}, m, Y, \tilde{\sigma})$

$$
\begin{aligned}
&\textbf{if } Y \notin \mathcal{L}_{\mathsf{Rel}} \textbf{ then} \\
&\textbf{return } 1 \\
&\textbf{return } \mathsf{pVrfy}(\mathsf{pk}, m, Y, \tilde{\sigma})
\end{aligned}
$$

- This scheme achieves pre-signature adaptability (pre-signature adaptability is only defined w.r.t. $Y \in \mathcal{L}_{\mathsf{Rel}}$)
- A pre-verifying pre-signature w.r.t. a malicious $Y \notin \mathcal{L}_{\mathsf{Rel}}$ **cannot** be adapted

Pre-Verify Soundness

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

1 : **if** $Y \notin \mathcal{L}_{\mathsf{Rel}}$ **then**

2 : **return** 0

3 : **return** $\mathsf{pVrfy}(\mathsf{pk}, m, Y, \widetilde{\sigma})$

- A pre-signature w.r.t. $Y \notin \mathcal{L}_{\mathsf{Rel}}$ **does not pre-verify**
- This requires efficient language checking (**not always possible**)
- Pre-verify soundness is only needed if Alice computes Y

Overview

Gaps

Definitions

Constructions

Transparent Reductions

Theoretical Challenges

Can we generically convey signa-
tures into adaptor signatures?

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

Schnorr (Adaptor) Signatures

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

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

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

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

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

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

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

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

Pre-signing computes $(\sigma_1, \sigma_2 - y)$ implicitly

$\mathsf{Vrfy}(\mathsf{sk}, m, \sigma)$

1 : parse σ as (R, s)

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

3 : $\mathbf{return}\ \mathsf{pk}^h \cdot R = g^s$

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

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

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

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

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

2 : parse σ as (σ_1, σ_2)

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

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

$$R = \mathsf{OWF}(r)$$

- One part can be computed using

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

- 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

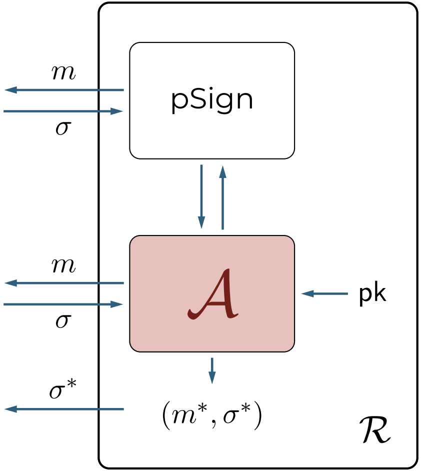
- It is decomposable

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

- 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

Overview

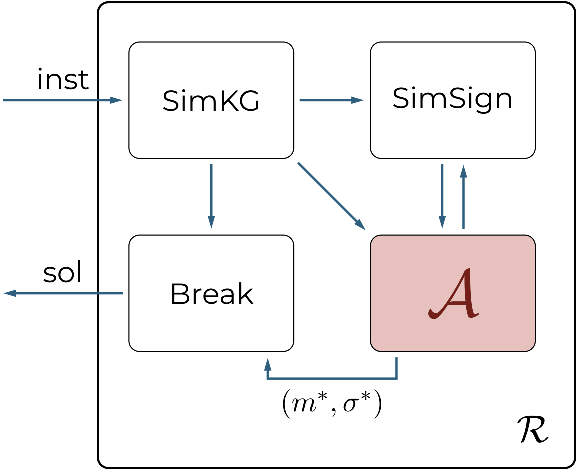
Gaps

Definitions

Constructions

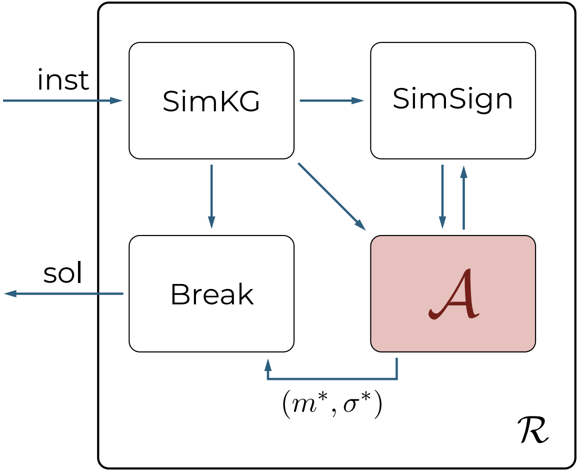
Transparent Reductions

Transparent Reductions



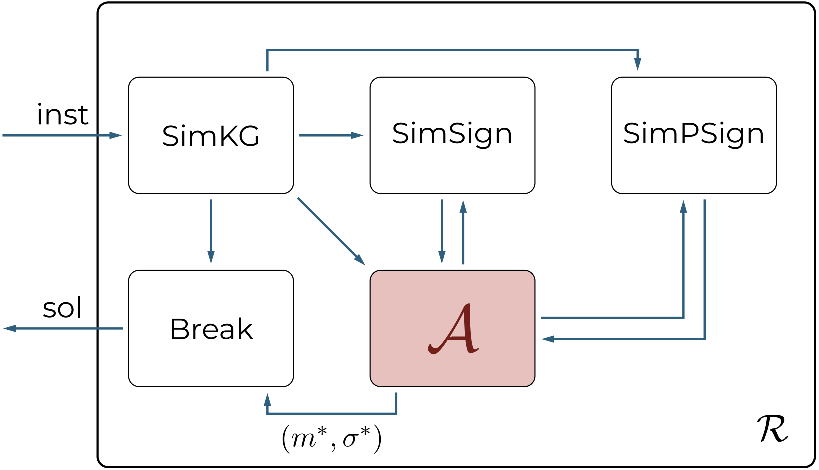
- **SimKG**:
  - Simulates keys (simSK, simPK)
- **SimSign**:
  - Simulates signatures using simSK
- **Break**:
  - Solve problem instance using valid forgery

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

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)**

Example: Secure Adaptor Signatures From BBS^+

BBS$^+$ Signatures Are Dichotomic

BBS$^+$.Sign(sk, m)

1 : $r \leftarrow\$ \mathbb{Z}_p$

2 : $e \leftarrow\$ \mathbb{Z}_p^*$

3 : $A = (g_0 \cdot g_1^r \cdot g_2^m)^{\frac{1}{e+\mathsf{sk}}}$

4 : **return** (A, e, r)

- Decomposability

$$\Sigma_1(\mathsf{sk}, m; \mathsf{OWF}(r)) = (A, e)$$

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

- Homomorphism

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

Transparent Reduction for BBS$^+$

- The reduction knows values (B_i, e_i), such that it can compute $A_i = (g_0 \cdot g_1^r \cdot g_2^m)^{\frac{1}{e+\mathsf{sk}}}$ **without** knowing sk.

$$
\begin{aligned}
A_i &= [g_0 g_1^{a_i}]^{\frac{1}{e_i+\mathsf{sk}}} \\
&= B_i [g_0^{\frac{a_i k^*(e^*+\mathsf{sk})-a_i}{(\mathsf{sk}+e_i)a^*}}] \\
&= (B_i^{(1-\frac{a_i}{a^*}-\frac{(e_i-e^*)a_i k^*}{a^*})})(g_0^{\frac{a_i k^*}{a^*}})
\end{aligned}
$$

- We set $\{(B_i, e_i)\}$ as simulated signing key simsk.

Simulatability for BBS$^+$: Pre-Signatures

$$A := (g_0 \cdot g_1^r \cdot g_2^m)^{\frac{1}{e+sk}}$$

$$\sigma := (A, e, r) \longrightarrow \widetilde{\sigma} := (A \cdot g_1^{\frac{y}{e+\mathsf{sk}}}, e, r)$$

We need to compute $g_1^{\frac{y}{e+\mathsf{sk}}}$

$$C_i = B_i^{\zeta_i} \longrightarrow (Y, y) = (g_1^y, y) \longrightarrow Y' = (g_1^y, \{C_i^y\})$$

$$C_i^{\frac{1}{\zeta_i} \cdot \frac{(e^*+x)k^*-1}{a^*}} = g_0^{\frac{\zeta_i \cdot y}{e_i+\mathsf{sk}} \cdot \frac{1}{\zeta_i} \cdot \frac{(e^*+x)k^*-1}{a^*}} = g_1^{\frac{\zeta_i \cdot y}{e_i+\mathsf{sk}} \cdot \frac{1}{\zeta_i}} = g_1^{\frac{y}{e_i+\mathsf{sk}}}$$

Conclusion BBS+ Adaptors

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

Conclusion

Gaps

Definitions

Constructions

Transparent Reductions

Foundations of Adaptor Signatures

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

https://www.paul-gerhart.de/publications/