

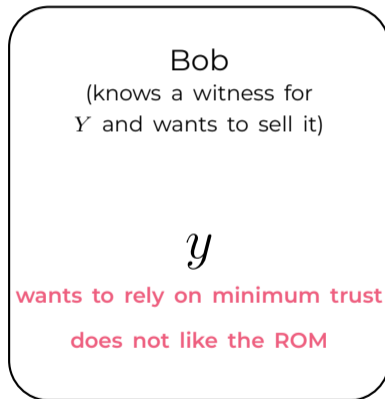
Foundations of Adaptor Signatures

Paul Gerhart, Dominique Schröder, Pratik Soni, Sri Aravinda Krishnan
Thyagarajan



THE UNIVERSITY OF
SYDNEY

Once Upon A Time



Adaptor Signatures

Adaptor Signature Interfaces

Signature
Scheme

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

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

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

Hard
Relation

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

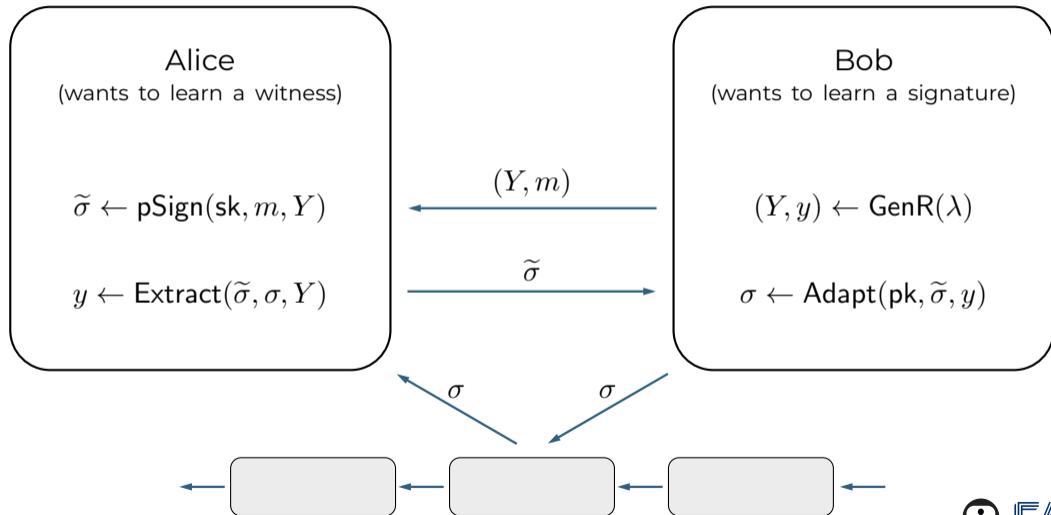
$$\tilde{\sigma} \leftarrow \text{pSign}(\text{sk}, m, Y)$$

$$b \leftarrow \text{pVrfy}(\text{pk}, m, \tilde{\sigma}, Y)$$

$$\sigma \leftarrow \text{Adapt}(\text{pk}, \tilde{\sigma}, y)$$

$$y \leftarrow \text{Extract}(\tilde{\sigma}, \sigma, Y)$$

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

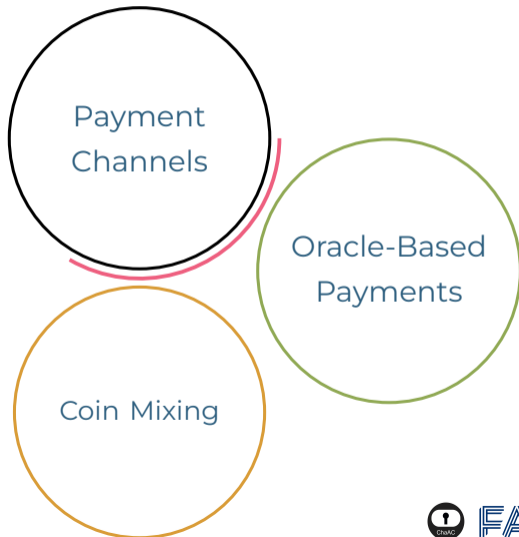
Given a signature scheme, building a secure adaptor signature is hard.

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

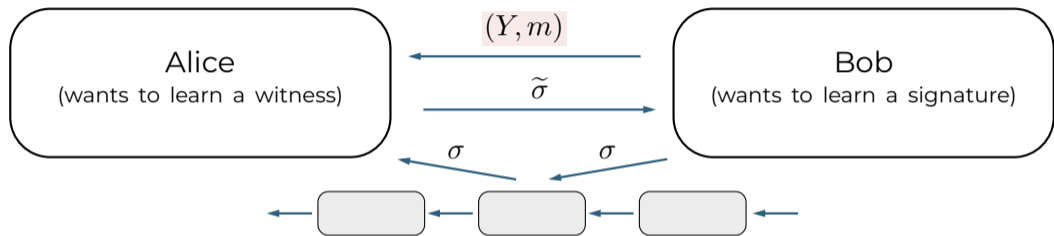


Constructions



**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

Leaky Adaptor Signatures

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

1 : $\tilde{\sigma} \leftarrow \text{pSign}(\text{sk}, m, Y)$

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

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

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

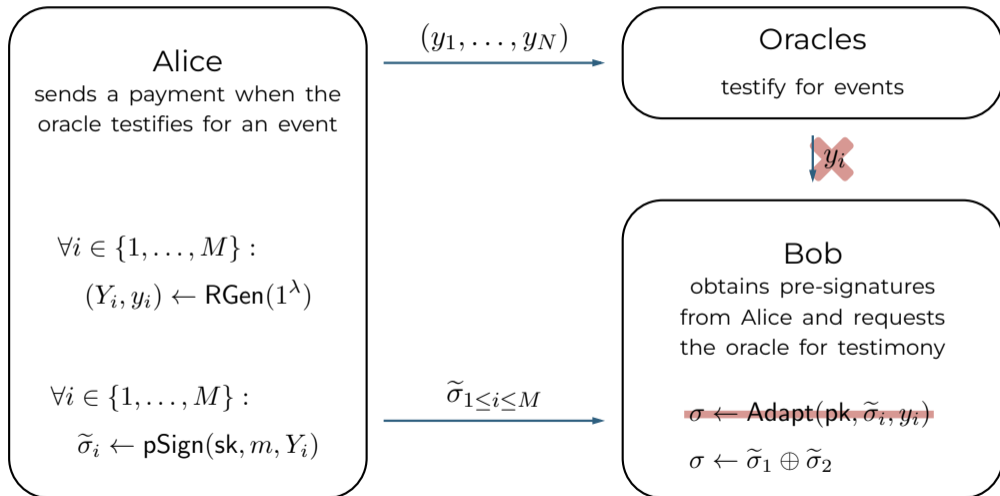
5 : $b \leftarrow_{\$} \{0, 1\}$

6 : **return** $(\tilde{\sigma}, r_b)$

- 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]



Overview



Gaps



Definitions



Constructions



Transparent
Reductions

Oracle-Based Conditional Payments (II)

Alice

sends a payment when the oracle testifies an event

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

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

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

$\tilde{\sigma}_i \leftarrow \text{pSign}(\text{sk}, m, Y_i)$

- 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

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

```
1 : if  $Y \notin \mathcal{L}_{\text{Rel}}$  then  
2 :    $\tilde{\sigma} := \perp$   
3 : else  $\tilde{\sigma} := \text{pSign}(\text{sk}, m, Y)$   
4 : return  $\tilde{\sigma}$ 
```

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

```
1 : if  $Y \notin \mathcal{L}_{\text{Rel}}$  then  
2 : return 1  
3 : return  $\text{pVrfy}(\text{pk}, m, Y, \tilde{\sigma})$ 
```

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

Pre-Verify Soundness

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

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

2 : **return** 0

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

- A pre-signature w.r.t. $Y \notin \mathcal{L}_{\text{Rel}}$ **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 transform signatures into adaptor signatures?

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

Dichotomic Signatures: Pre-Signing

- The signature consists of two parts

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

- The signature uses a homomorphic one-way function

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

- One part can be computed using

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

- The other part can be computed using

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

pSign(sk, m, Y)

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

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

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

Dichotomic Signatures: Adapt/Extract

Adapt(pk, $\tilde{\sigma}$, y)

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

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

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

Extract(Y, $\tilde{\sigma}$, σ)

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

2 : parse σ as (σ_1, σ_2)

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

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

Dichotomic Signatures: A Definition

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

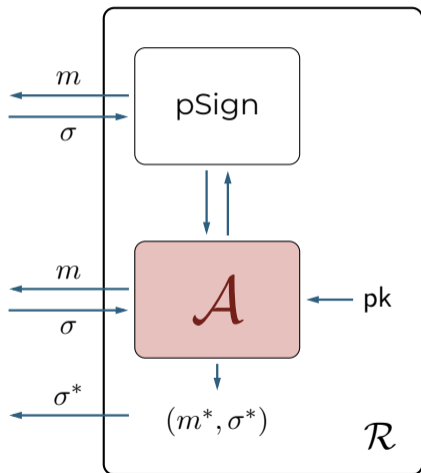
- It is decomposable

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

- It is homomorphic in the randomness

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

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

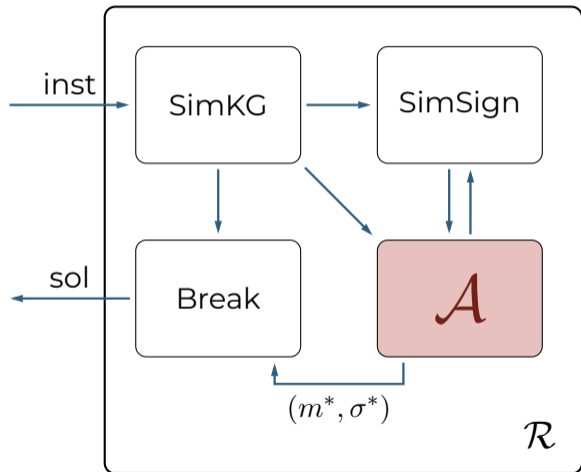


Constructions



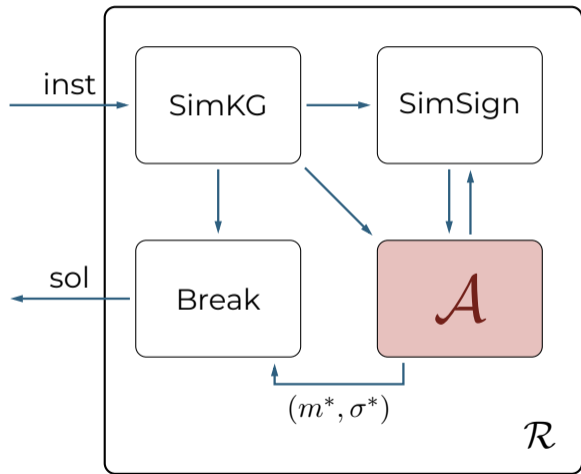
Transparent
Reductions

Transparent Reductions



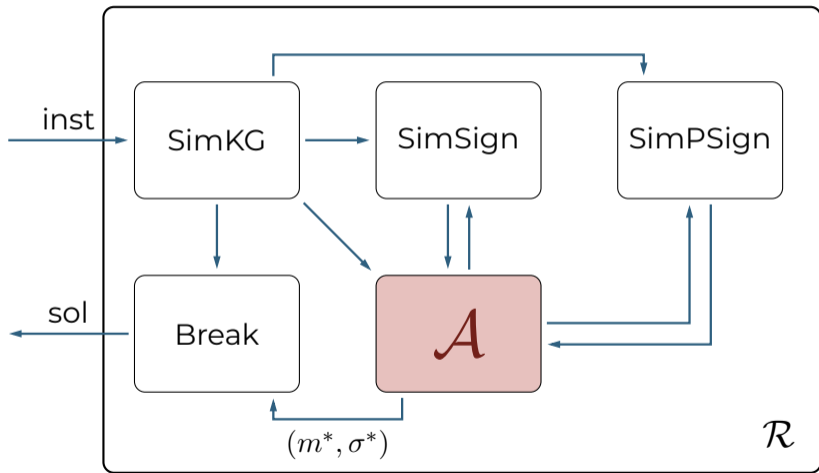
- **SimKG:**
 - Simulates keys ($\text{simSK}, \text{simPK}$)
- **SimSign:**
 - Simulates signatures using simSK
- **Break:**
 - Solve problem instance using valid forgery

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

- Decomposability

BBS⁺.Sign(sk, m)

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

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

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

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

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

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

- Homomorphism

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

Transparent Reduction for BBS⁺

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

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

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

Simulatability for BBS⁺: Pre-Signatures

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

$$\sigma := (A, e, r) \xrightarrow{\quad} \tilde{\sigma} := (A \cdot g_1^{\frac{y}{e+sk}}, e, r)$$

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

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

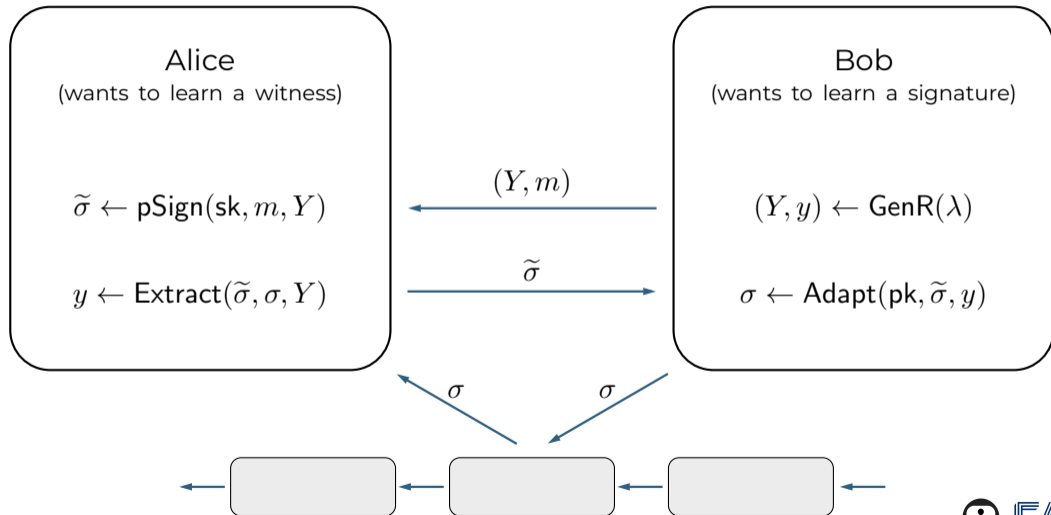
$$C_i^{\frac{1}{\zeta_i} \cdot \frac{(e^*+x)k^*-1}{a^*}} = g_0^{\frac{\zeta_i \cdot y}{e_i+sk} \cdot \frac{1}{\zeta_i} \cdot \frac{(e^*+x)k^*-1}{a^*}} = g_1^{\frac{\zeta_i \cdot y}{e_i+sk} \cdot \frac{1}{\zeta_i}} = g_1^{\frac{y}{e_i+sk}}$$

Conclusion BBS+ Adaptors

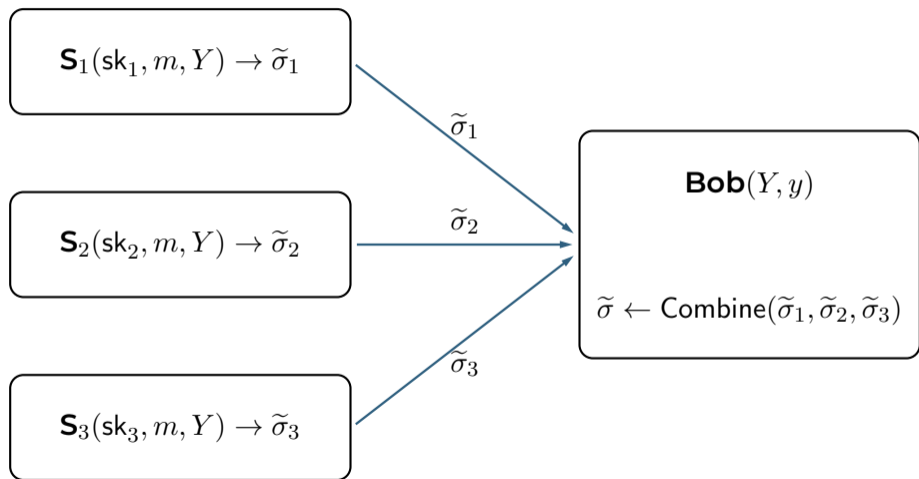
1. The signature scheme is **dichotomic** ✓
2. There is a **transparent reduction** from the strong unforgeability to an underlying hard problem ✓
3. We can simulate a pre-signature oracle (**simulatability**) ✓

Preview: Distributed Fair Exchange

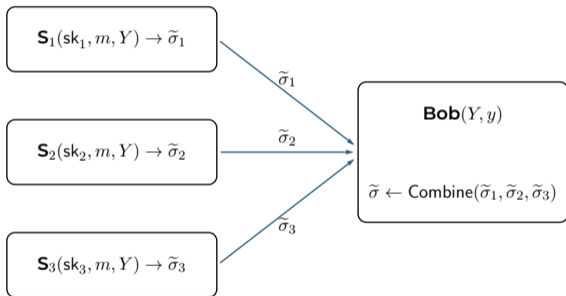
Fair Exchange using Adaptor Signatures



Preview: Threshold Adaptors

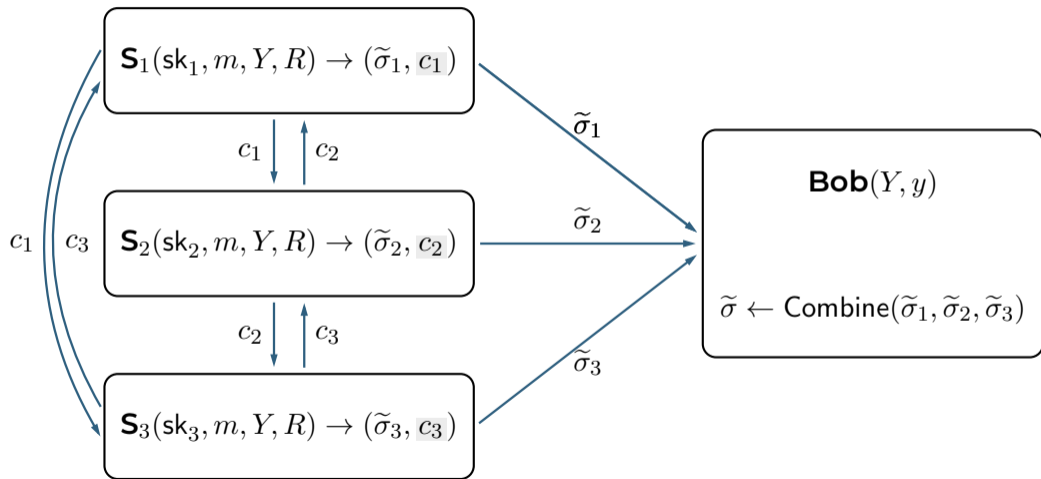


Threshold Adaptors: Extractability?



- To extract a witness, the signers need to know the *combined* pre-signature $\tilde{\sigma}$
- The signers just know their partial shares $\tilde{\sigma}_i$
- If Bob and all other signers are malicious, an honest signer never can extract the witness.

Threshold Adaptors



Certified Witness Encryption

- Goal: Find an encryption scheme such that we can compute
 - $c \leftarrow \text{Enc}(ek, \tilde{\sigma}_i)$
 - $\tilde{\sigma}_i \leftarrow \text{Dec}(\sigma, c)$

where σ is a valid signature and ek is known *prior* to sending the pre-signature shares.

- We found such construction for Schnorr, BBS, Waters, and many more

Conclusion



Gaps



Definitions



Constructions



**Transparent
Reductions**