ORCAS – *AN EFFICIENT ADAPTOR SIGNATURE BASED ON CSI-FISH*

Ryann Cartor, Nathan Daly, Giulia Gaggero, Jason LeGrow, Andrea Sanguineti, and Silvia Sconza

Swiss Crypto Day

ETH zürich, *4 September 2026*

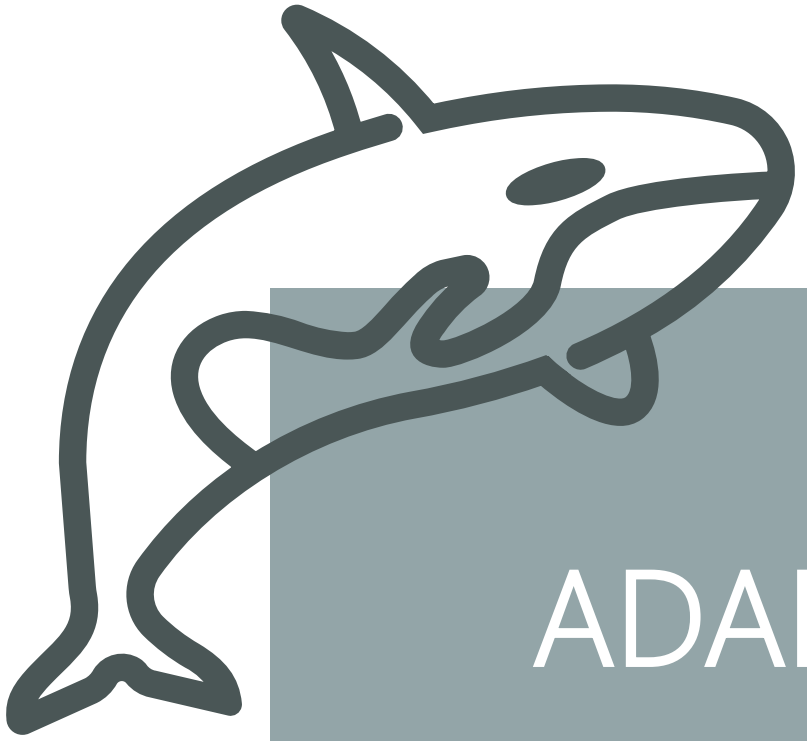
Universität Zürich[UZH]

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse Science and Technology
Cyber-Defence Campus

O
R
C
A
S

One Round Cheating Adaptor Signatures



ADAPTOR SIGNATURE SCHEMES

EXCHANGING CRYPTOCURRENCIES

Send 1 BTC to Bob.

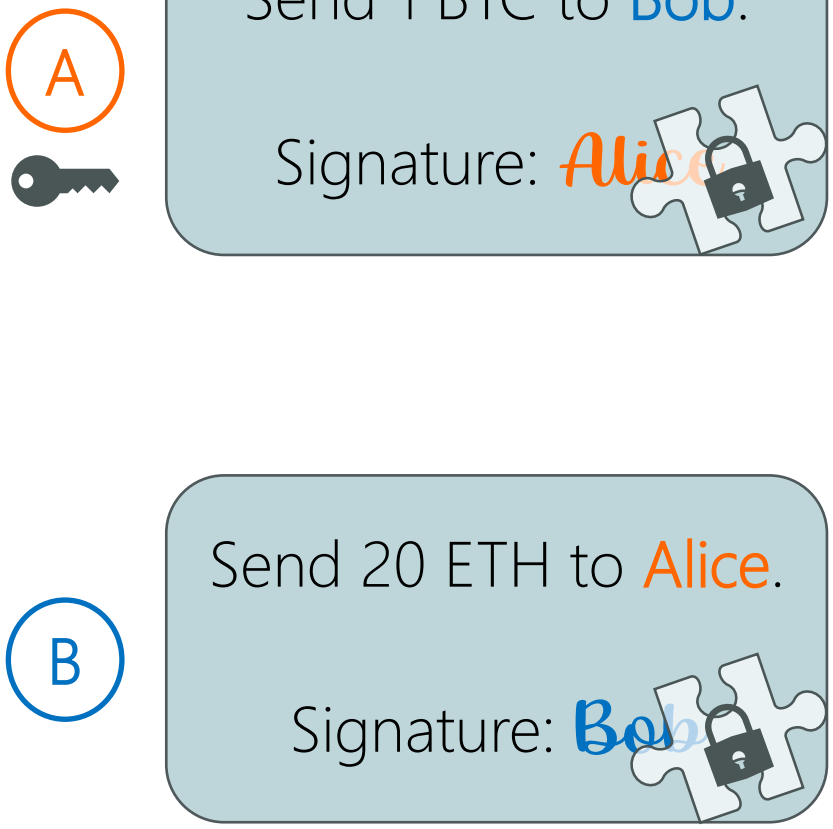
Signature: Alice

Send 20 ETH to Alice.

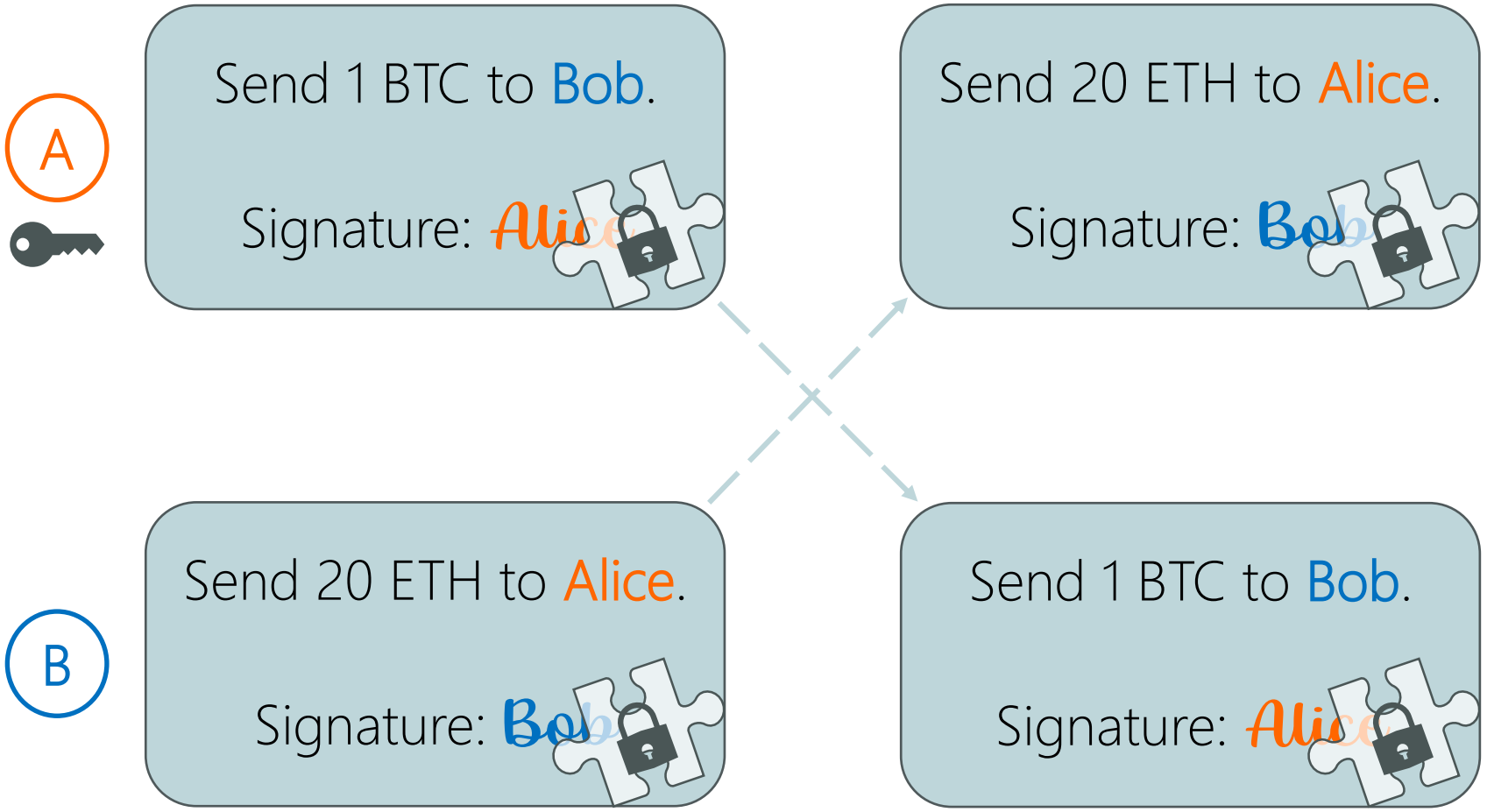
Signature: Bob

Who should send first?
They cannot trust each other.

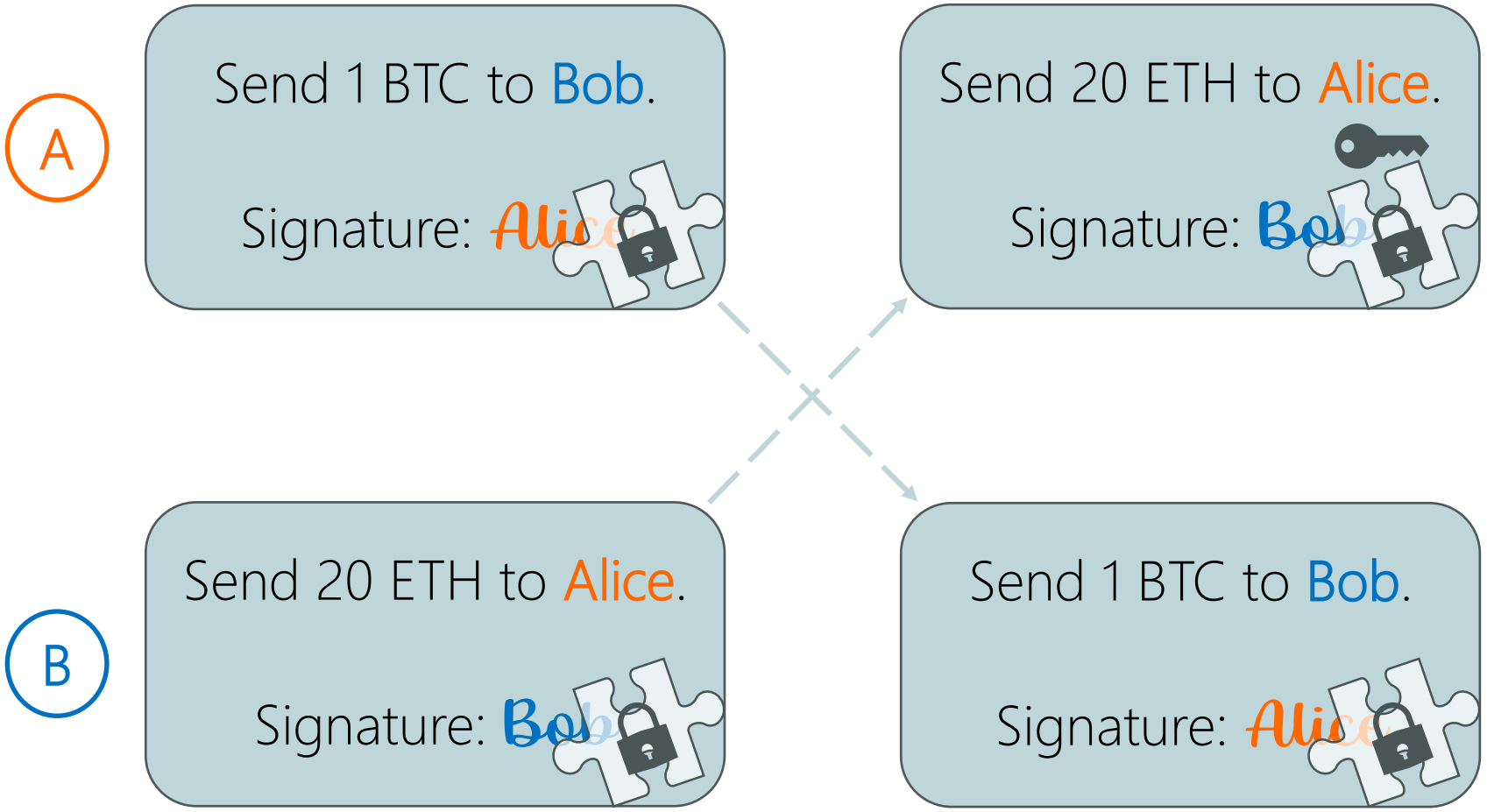
EXCHANGING CRYPTOCURRENCIES



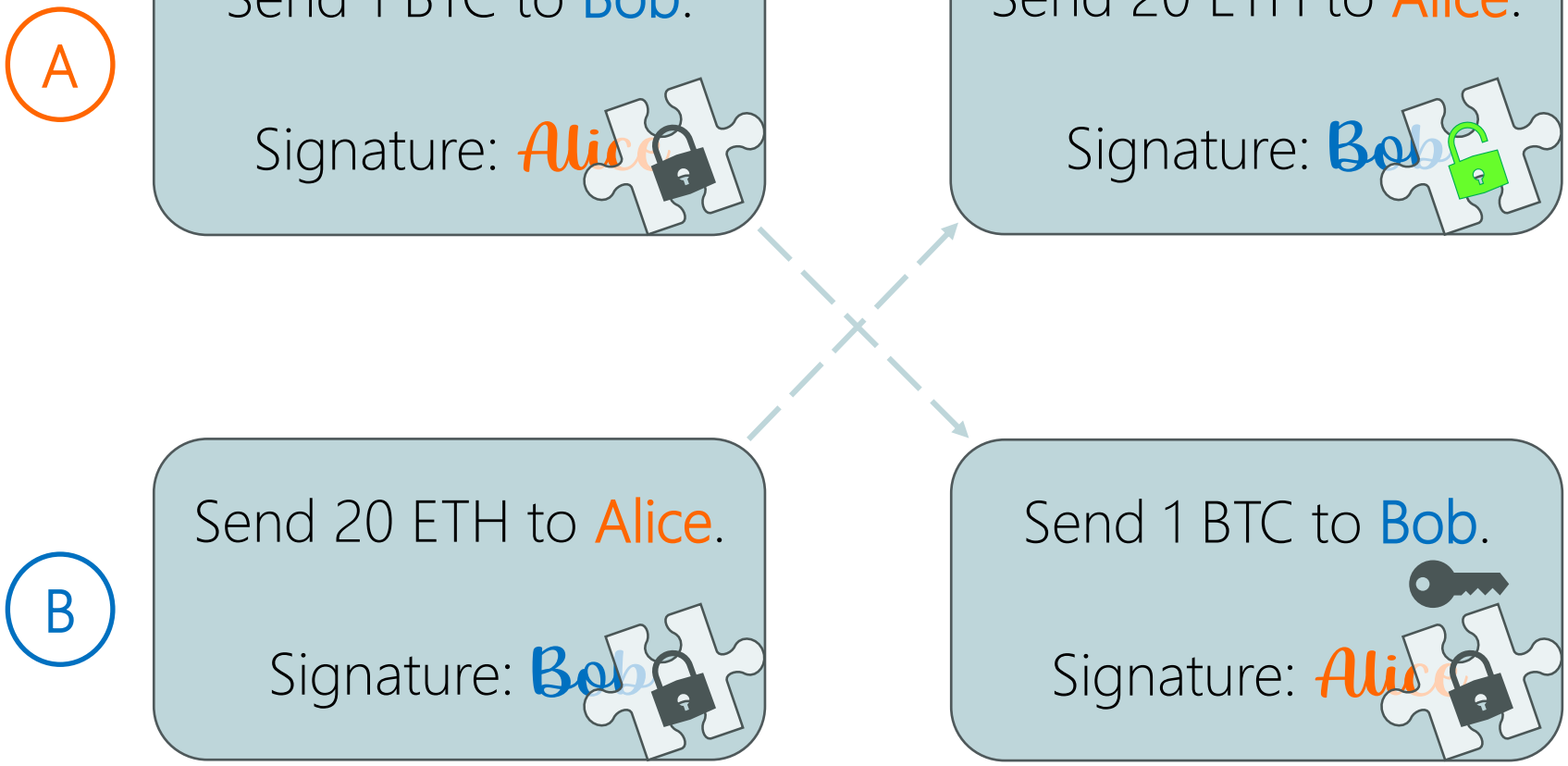
EXCHANGING CRYPTOCURRENCIES



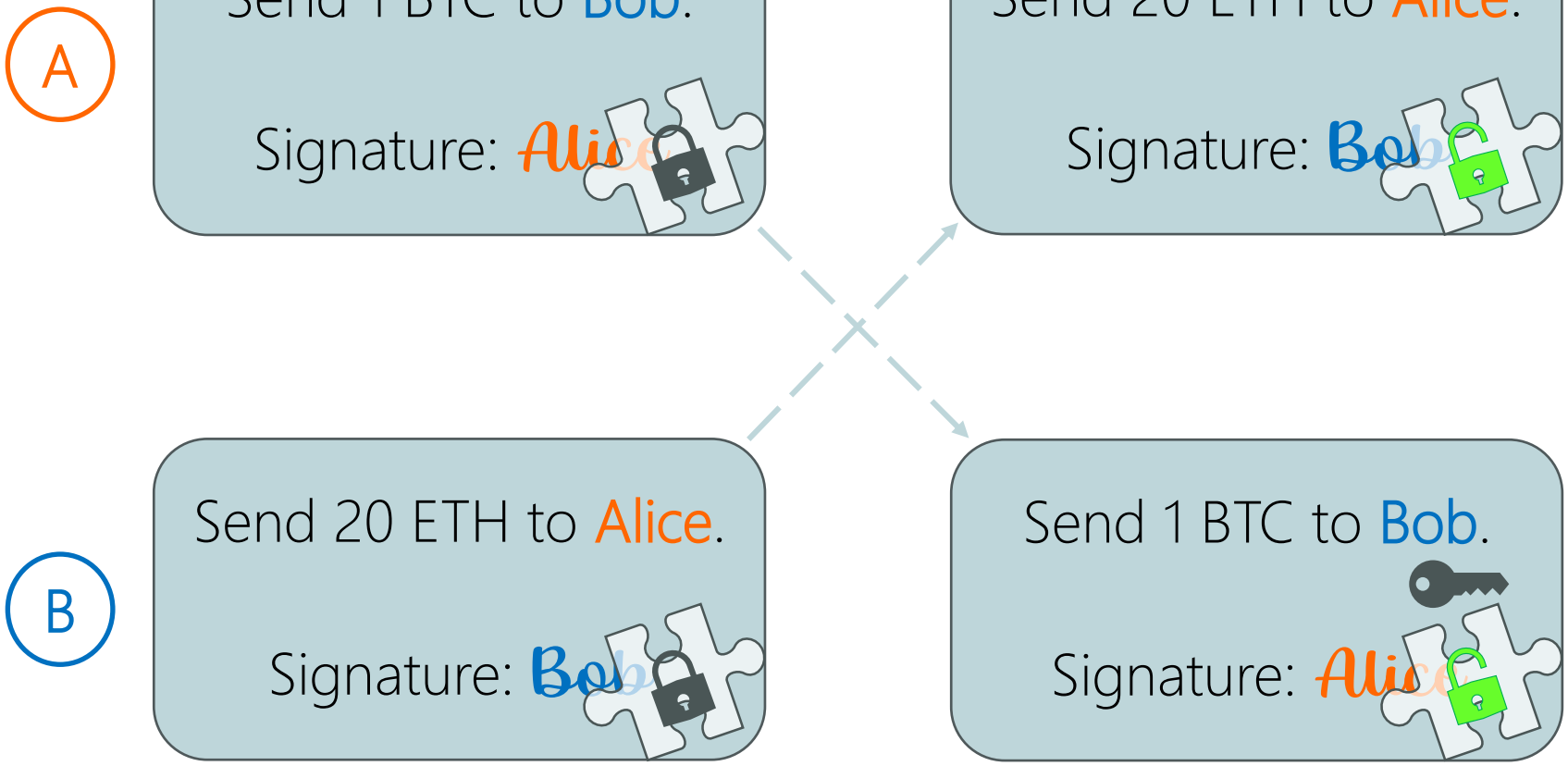
EXCHANGING CRYPTOCURRENCIES



EXCHANGING CRYPTOCURRENCIES



EXCHANGING CRYPTOCURRENCIES



EXCHANGING CRYPTOCURRENCIES

A

Send 1 BTC to Bob.

Signature: Alice

Send 20 ETH to Alice.

Signature: Bob

B

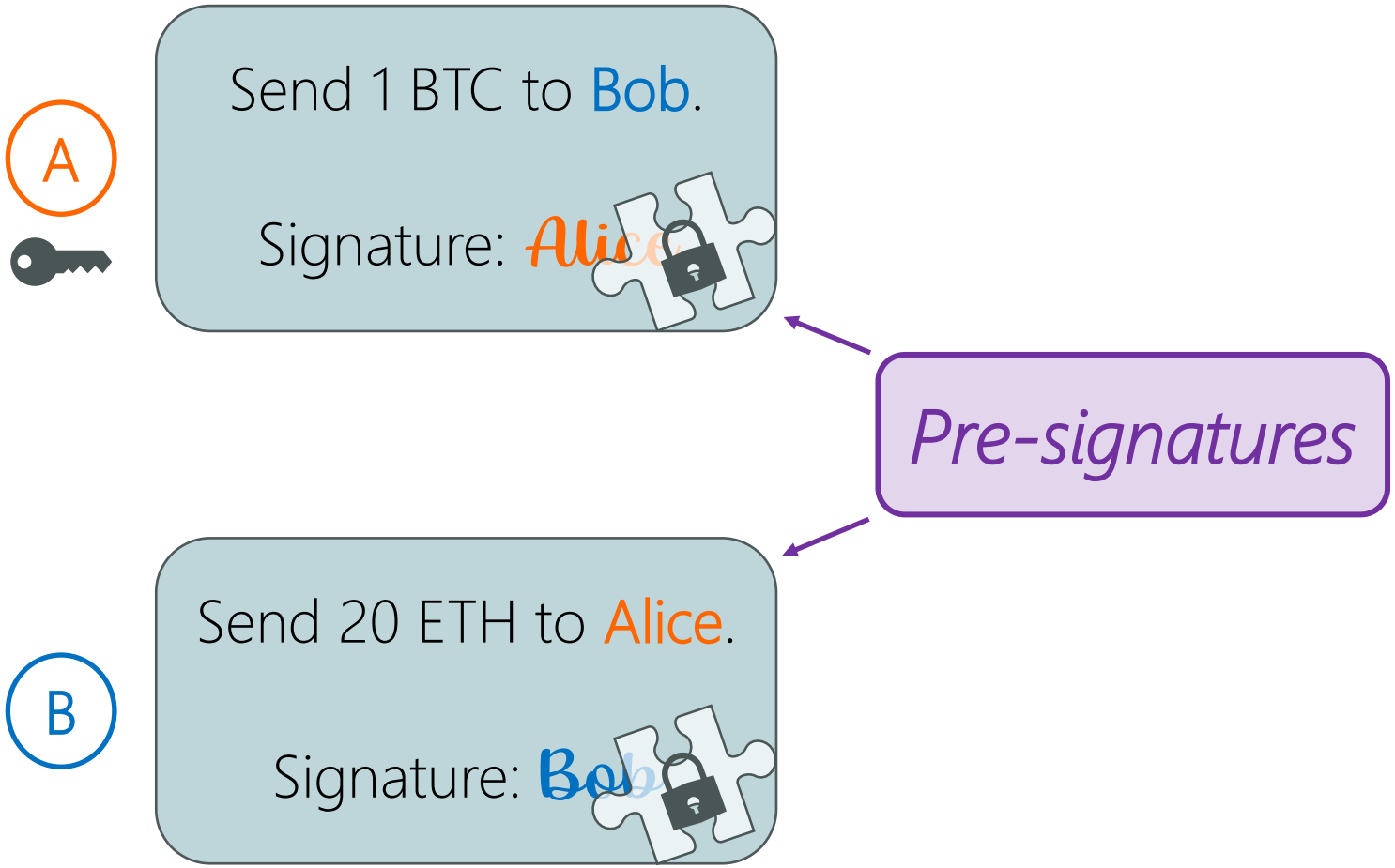
Send 20 ETH to Alice.

Signature: Bob

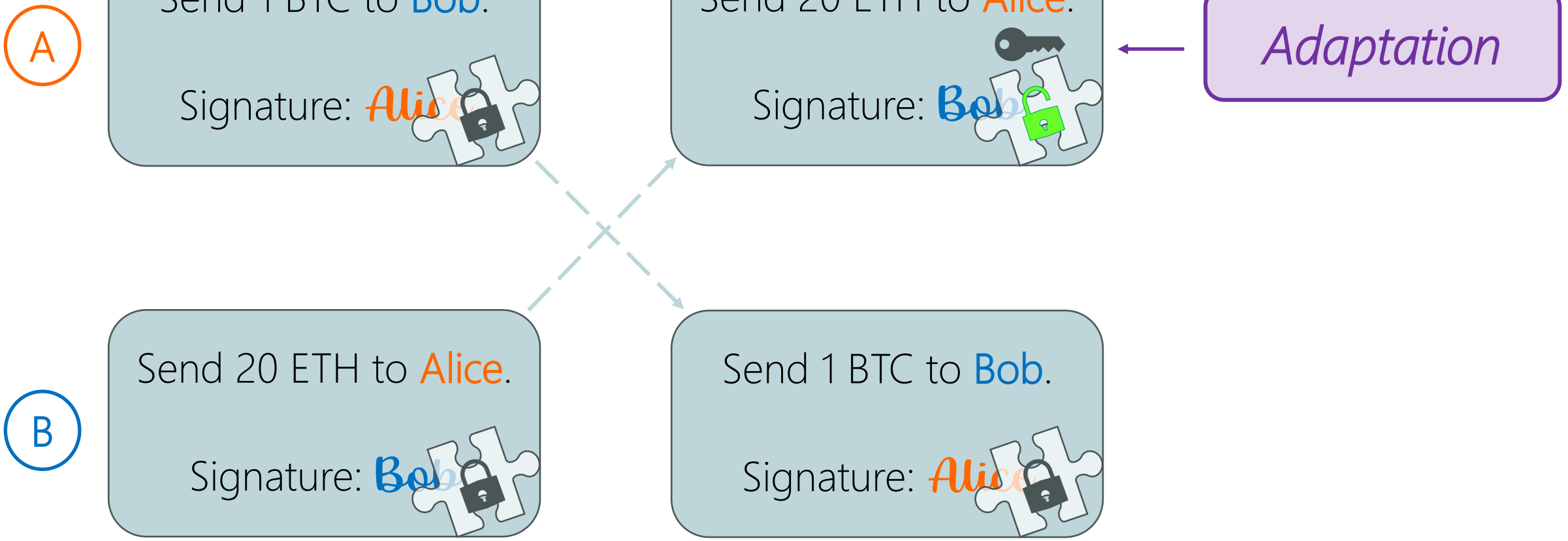
Send 1 BTC to Bob.

Signature: Alice

EXCHANGING CRYPTOCURRENCIES



EXCHANGING CRYPTOCURRENCIES



EXCHANGING CRYPTOCURRENCIES

A

Send 1 BTC to Bob.

Signature: Alice

Send 20 ETH to Alice.

Signature: Bob

B

Send 20 ETH to Alice.

Signature: Bob

Send 1 BTC to Bob.

Signature: Alice

Extraction

DEFINITION

SIGNATURE SCHEME

Algorithms:

- KeyGen $\rightarrow (\mathrm{sk}, \mathrm{pk})$
- Sign$(\mathrm{sk}, m) \rightarrow \sigma$
- Vfy$(\mathrm{pk}, m, \sigma) \rightarrow 0/1$

Keys:

- Secret key $\mathbf{sk}$ ⇝ used to sign
- Public key $\mathbf{pk}$ ⇝ used to verify

Hard Problems:

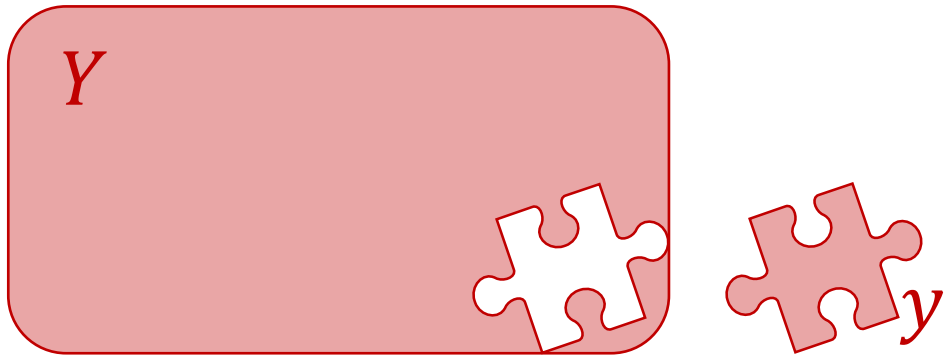
- Forge a σ for $\mathbf{pk}$ without knowing $\mathbf{sk}$

DEFINITION

ADAPTOR SIGNATURE SCHEME

Algorithms:

- KeyGen $\rightarrow (\mathrm{sk}, \mathrm{pk})$
- $\mathrm{Sign}(\mathrm{sk}, m) \rightarrow \sigma$
- $\mathrm{Vfy}(\mathrm{pk}, m, \sigma) \rightarrow 0/1$
- $\mathrm{RGen} \rightarrow (Y, y)$
- $\mathrm{PreSign}(\mathrm{sk}, Y, m) \rightarrow \hat{\sigma}$
- $\mathrm{PreVfy}(\mathrm{pk}, Y, m, \hat{\sigma}) \rightarrow 0/1$
- $\mathrm{Adapt}(y, \hat{\sigma}) \rightarrow \sigma$
- $\mathrm{Extract}(\sigma, \hat{\sigma}, Y) \rightarrow y$



Keys:

- Secret key $\mathbf{sk}$ ⇝ used to sign
- Public key $\mathbf{pk}$ ⇝ used to verify
- Statement Y ⇝ used to pre-sign
- Witness y ⇝ used to adapt

Hard Problems:

- Forge a σ for $\mathbf{pk}$ without knowing $\mathbf{sk}$
- Adapt $\hat{\sigma}$ to σ without knowing y

SIGNATURE SCHEMES FROM GROUP ACTIONS

DEFINITION

GROUP ACTION SIGNATURE SCHEME

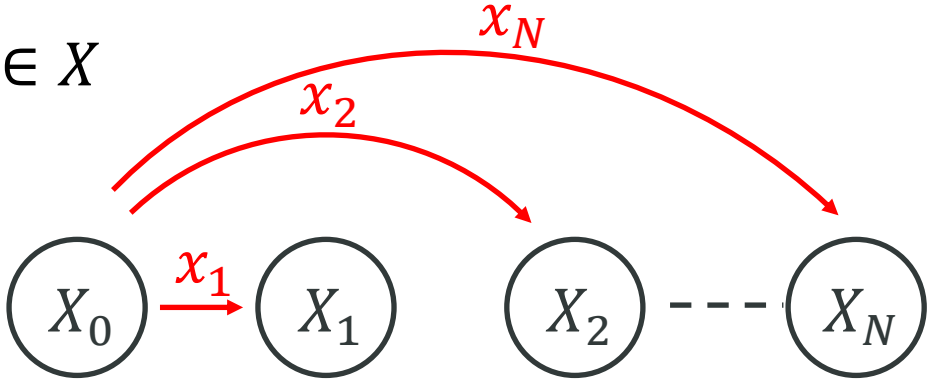
Setup: Group action $*: G \times X \to X$, base element $X_0 \in X$

DEFINITION

GROUP ACTION SIGNATURE SCHEME

Setup: Group action $*: G \times X \to X$, base element $X_0 \in X$

KeyGen: $\mathrm{sk} = x_1, \ldots, x_N \in G$,

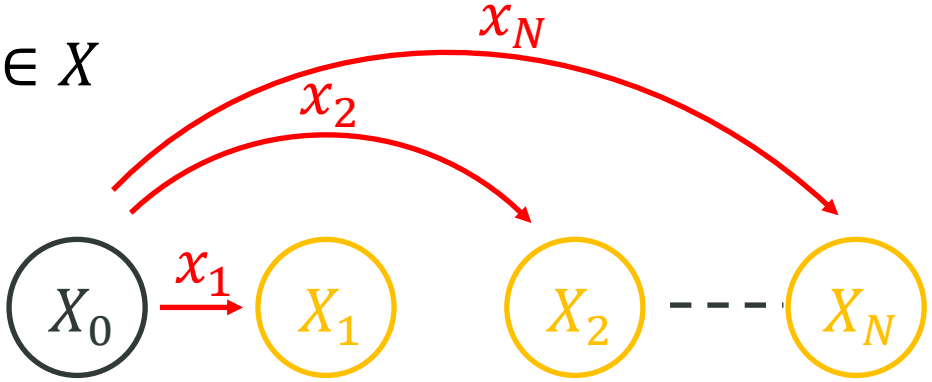


DEFINITION

GROUP ACTION SIGNATURE SCHEME

<u>Setup</u>: Group action $*: G \times X \to X$, base element $X_0 \in X$

<u>KeyGen</u>: $\mathrm{sk} = x_1, \ldots, x_N \in G$, $\mathrm{pk} = x_i * X_0 = X_i$



DEFINITION

GROUP ACTION SIGNATURE SCHEME

<u>Setup</u>: Group action $*: G \times X \to X$, base element $X_0 \in X$

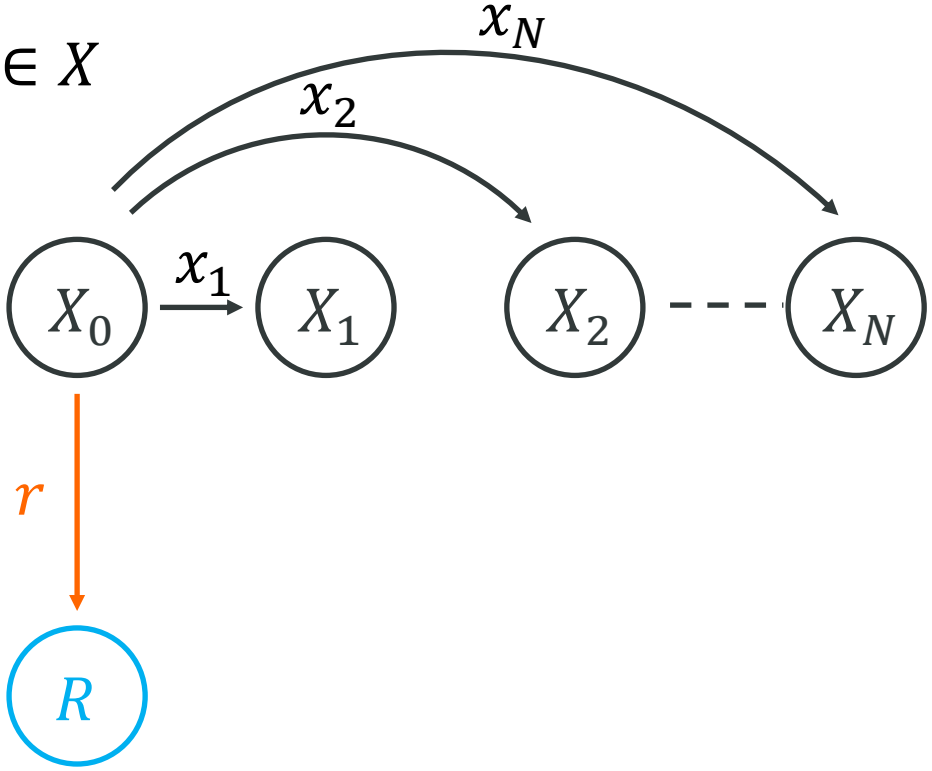
<u>KeyGen</u>: $\mathrm{sk} = x_1, \ldots, x_N \in G$, $\mathrm{pk} = x_i * X_0 = X_i$

Prover(X_i, x_i, m)

$r \in G$

$R = r * X_0$

Verifier(X_i, m)



DEFINITION

GROUP ACTION SIGNATURE SCHEME

Setup: Group action $*: G \times X \to X$, base element $X_0 \in X$

KeyGen: $\mathrm{sk} = x_1, \ldots, x_N \in G$, $\mathrm{pk} = x_i * X_0 = X_i$

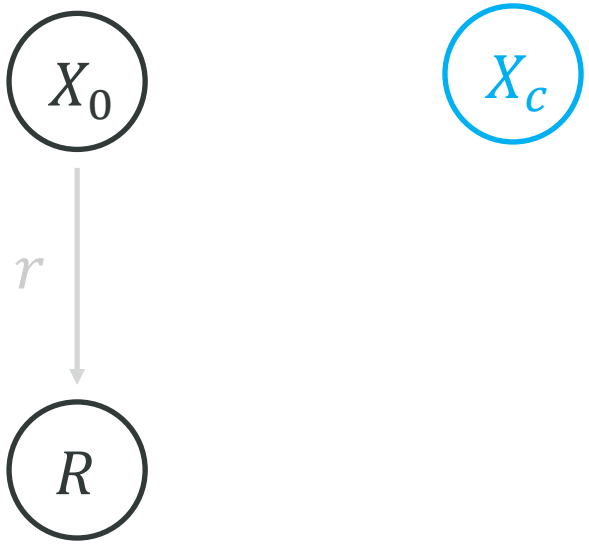
Prover(X_i, x_i, m)

$r \in G$

$R = r * X_0$

$\mathrm{c} = \mathrm{H}(m, R)$

Verifier(X_i, m)



DEFINITION

GROUP ACTION SIGNATURE SCHEME

<u>Setup</u>: Group action $*: G \times X \to X$, base element $X_0 \in X$

<u>KeyGen</u>: $\mathrm{sk} = x_1, \ldots, x_N \in G$, $\mathrm{pk} = x_i * X_0 = X_i$

Prover(X_i, x_i, m)

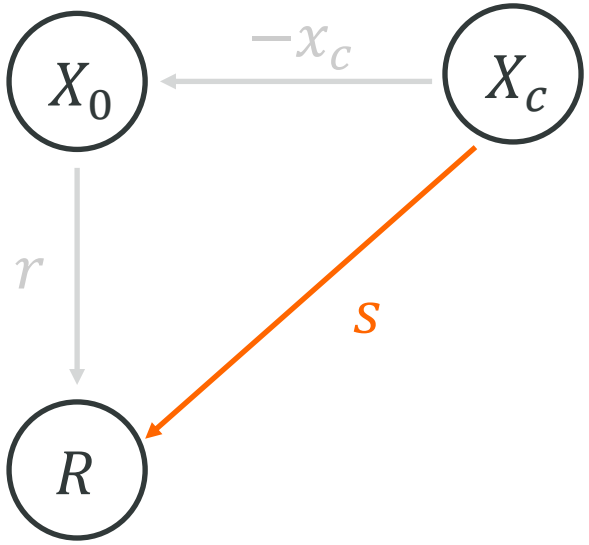
$r \in G$

$R = r * X_0$

$\mathrm{c} = \mathrm{H}(m, R)$

$s = r - x_c$

Verifier(X_i, m)



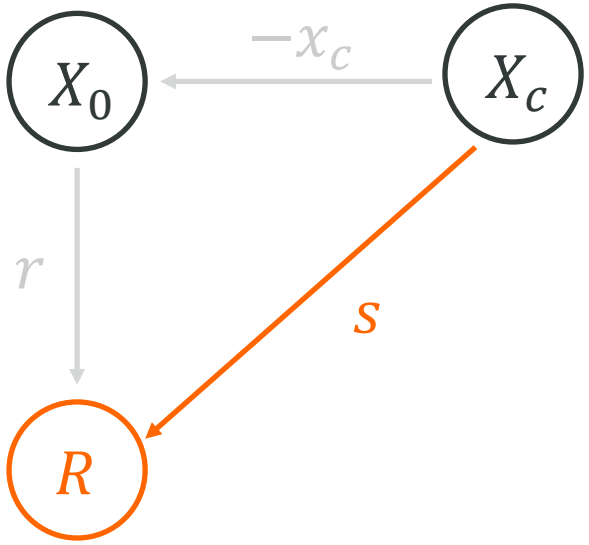
DEFINITION

GROUP ACTION SIGNATURE SCHEME

Setup: Group action $*: G \times X \to X$, base element $X_0 \in X$

KeyGen: $\mathrm{sk} = x_1, \ldots, x_N \in G$, $\mathrm{pk} = x_i * X_0 = X_i$

Prover(X_i, x_i, m)		Verifier(X_i, m)
$r \in G$		
$R = r * X_0$		
$\mathrm{c} = \mathrm{H}(m, R)$		
$s = r - x_c$		
	$\sigma = (R, s)$	



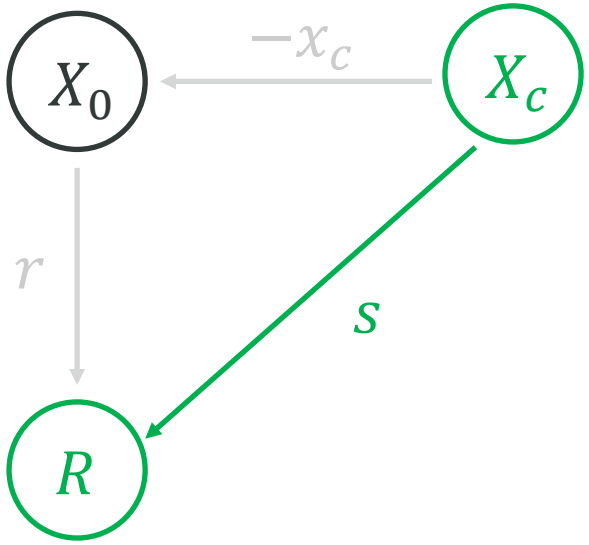
DEFINITION

GROUP ACTION SIGNATURE SCHEME

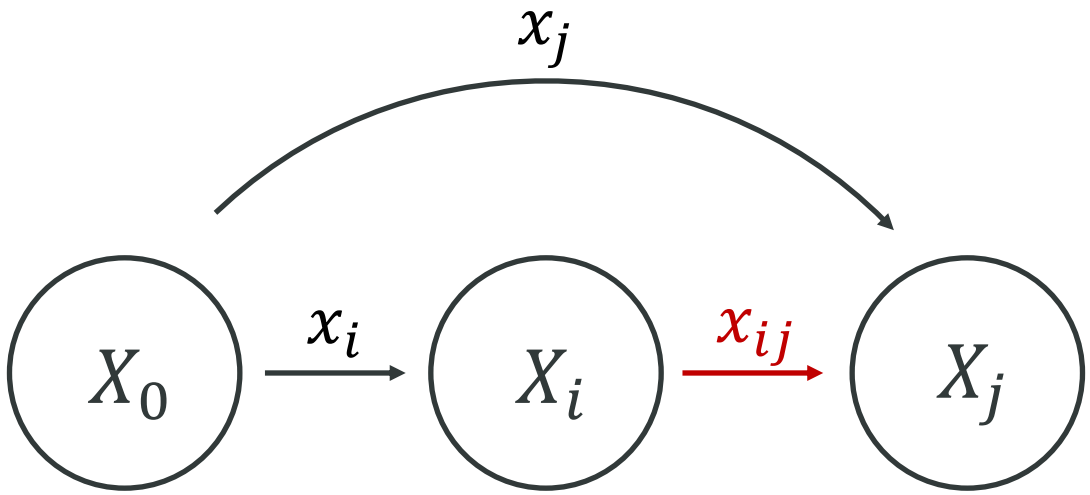
Setup: Group action $*: G \times X \to X$, base element $X_0 \in X$

KeyGen: $\mathrm{sk} = x_1, \ldots, x_N \in G$, $\mathrm{pk} = x_i * X_0 = X_i$

Prover(X_i, x_i, m)		Verifier(X_i, m)
$r \in G$		
$R = r * X_0$		
$\mathrm{c} = \mathrm{H}(m, R)$		
$s = r - x_c$		
	$\sigma = (R, s)$ ⇢	
		Check $s * X_c = R$
	⇠ $0/1$	



GROUP ACTION SIGNATURE SCHEME

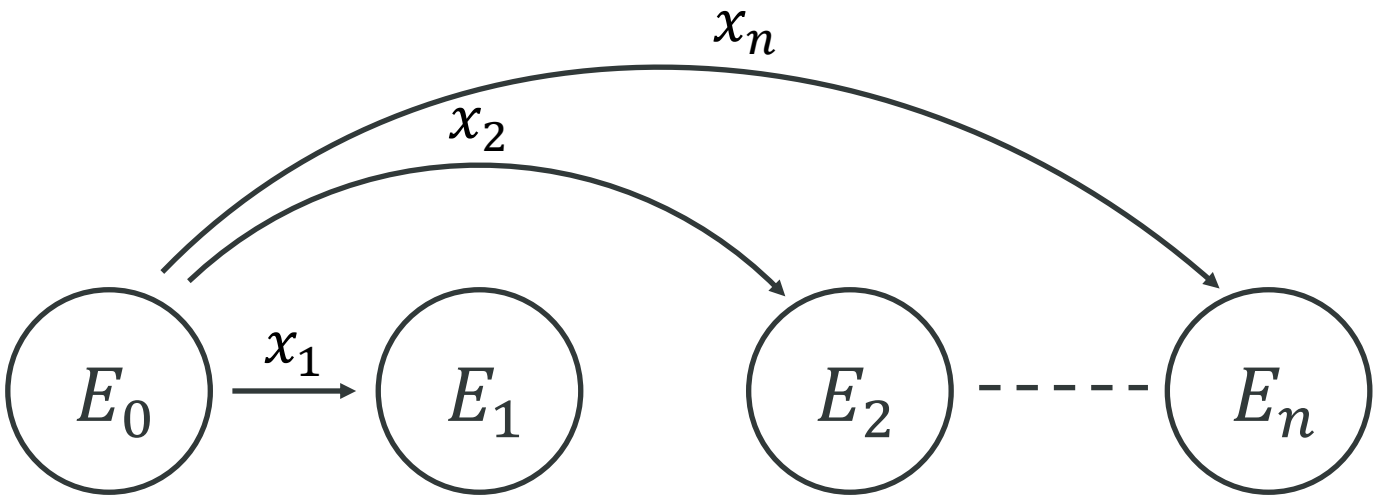


Multi-Target Group Action Inverse Problem

Given $(X_0, \ldots, X_N)$, find $i \neq j$ and $x_{ij} \in G$ such that $X_i = x_{ij} * X_j$.

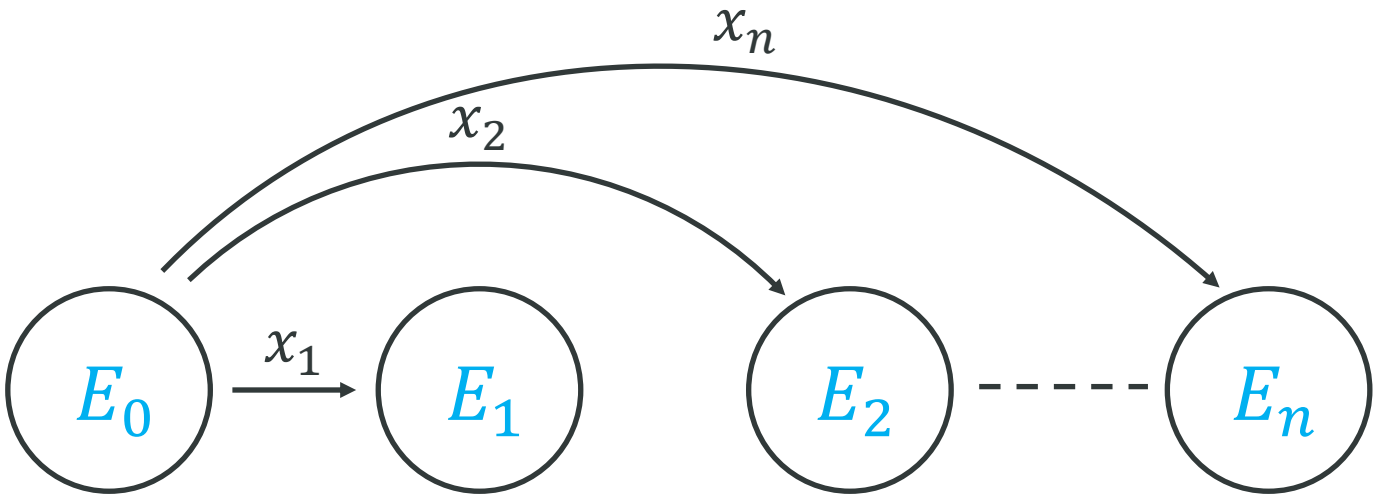
DEFINITION

CSI-FISH SIGNATURE SCHEME



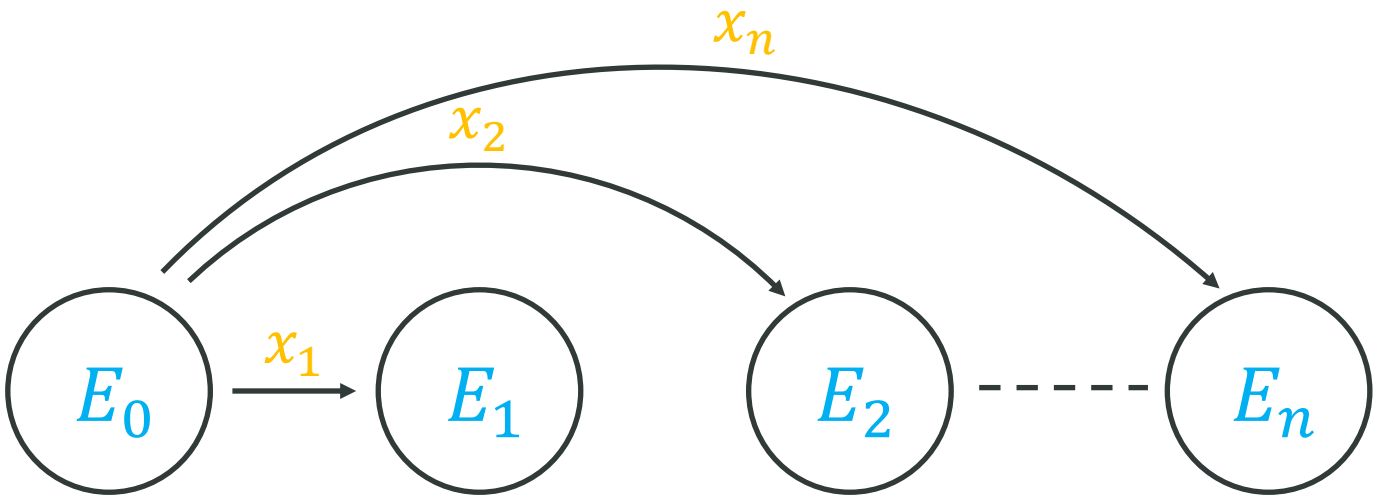
DEFINITION

CSI-FISH SIGNATURE SCHEME



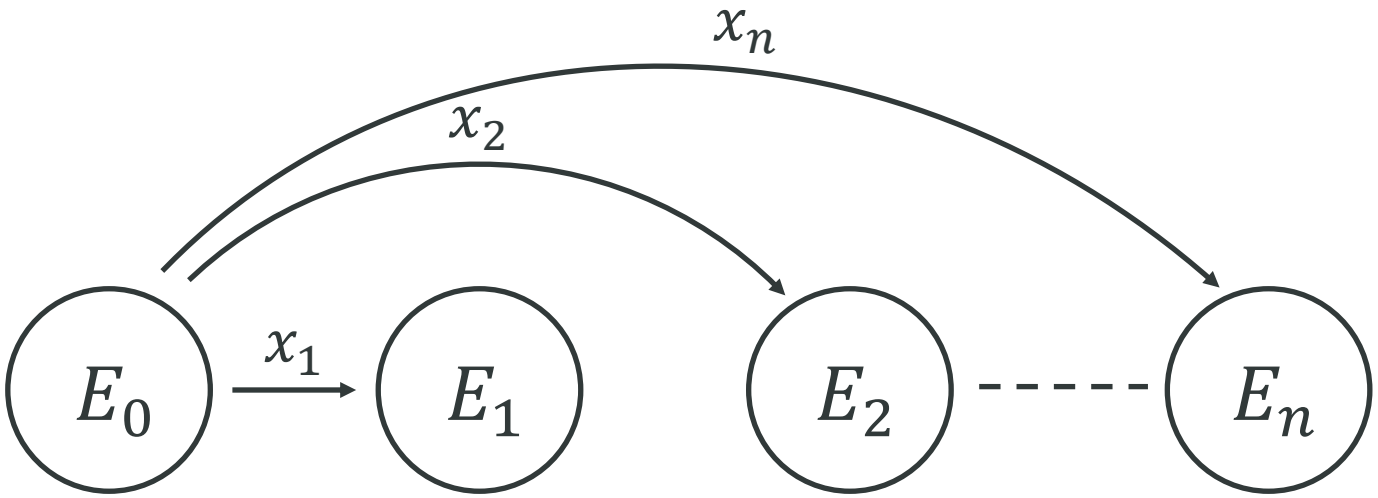
- E_i supersingular elliptic curves

CSI-FISH SIGNATURE SCHEME



- E_i supersingular elliptic curves
- x_i isogenies

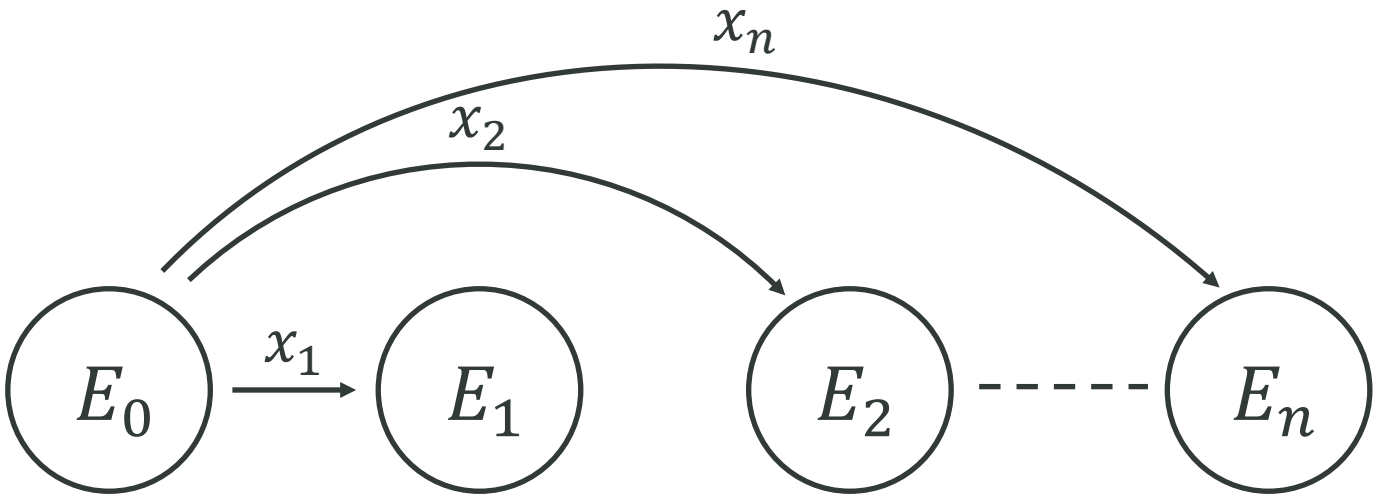
CSI-FISH SIGNATURE SCHEME



PROBLEM:

The size of the challenge space is NOT big enough.

CSI-FISH SIGNATURE SCHEME



PROBLEM:

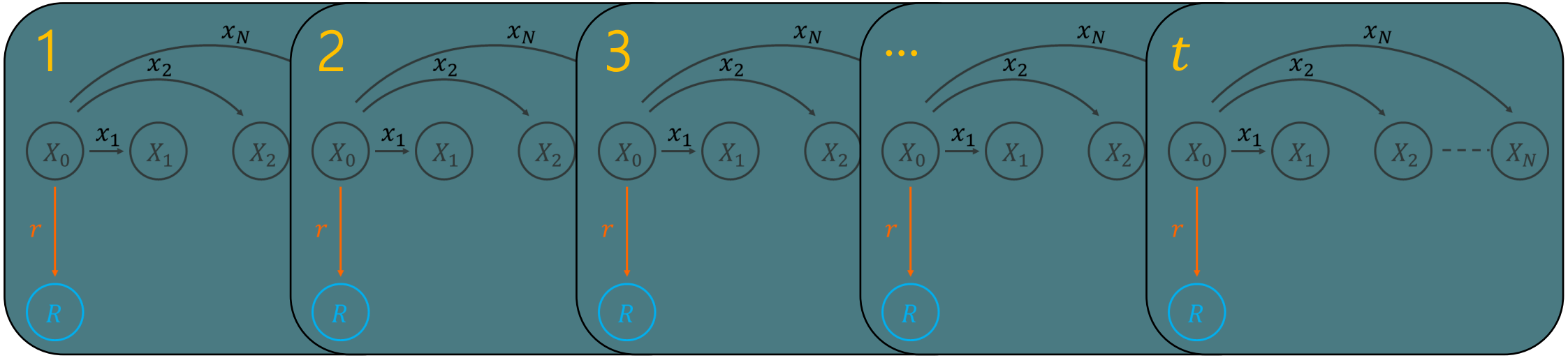
The size of the challenge space is NOT big enough.

SOLUTION:

Run the protocol for t rounds.

DEFINITION

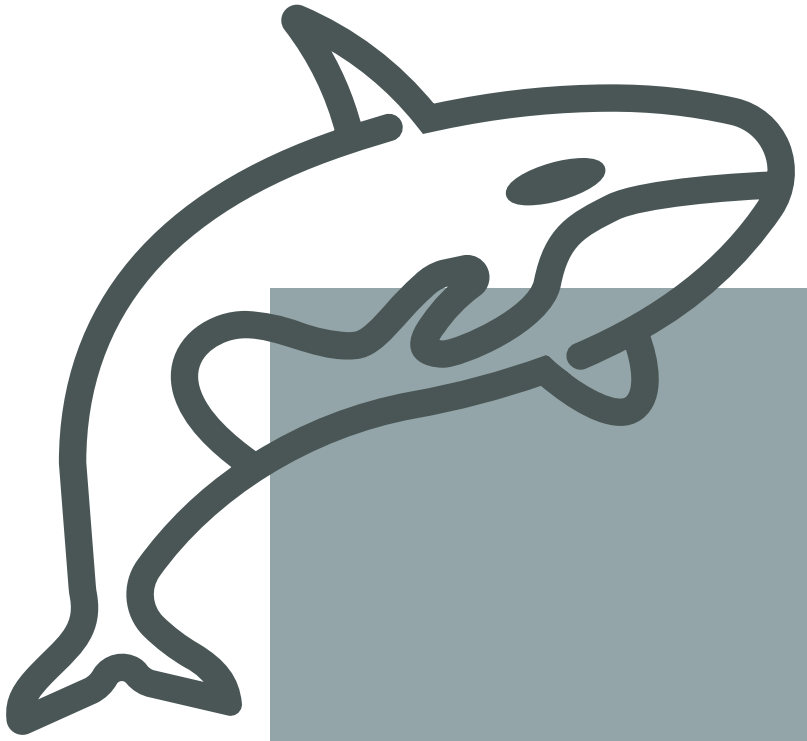
CSI-FISH SIGNATURE SCHEME



SOLUTION:

Run the protocol for t rounds.

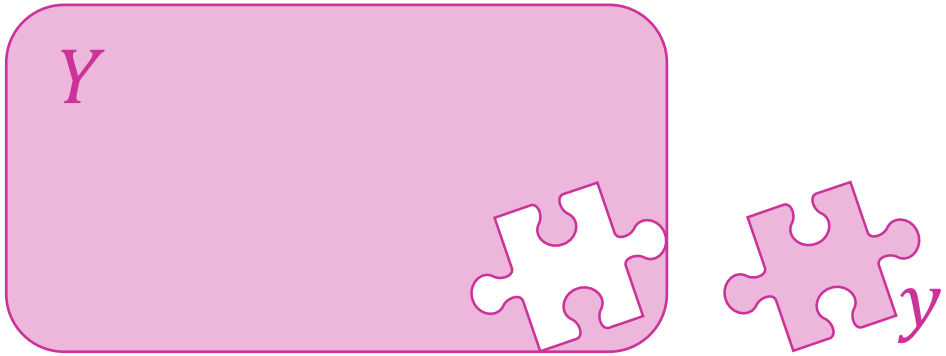
Challenge $(c_1, \ldots, c_t) = \mathrm{H}(m, R_1, \ldots, R_t)$

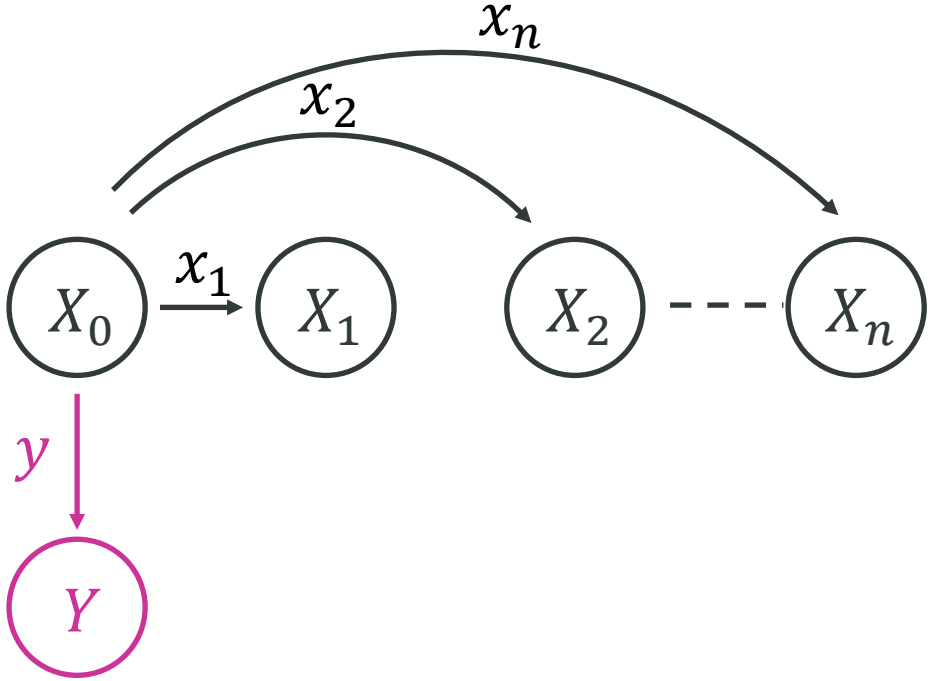


ORCAS

ADAPTOR CSI-FISH I

Setup: Statement Y for unknown witness y

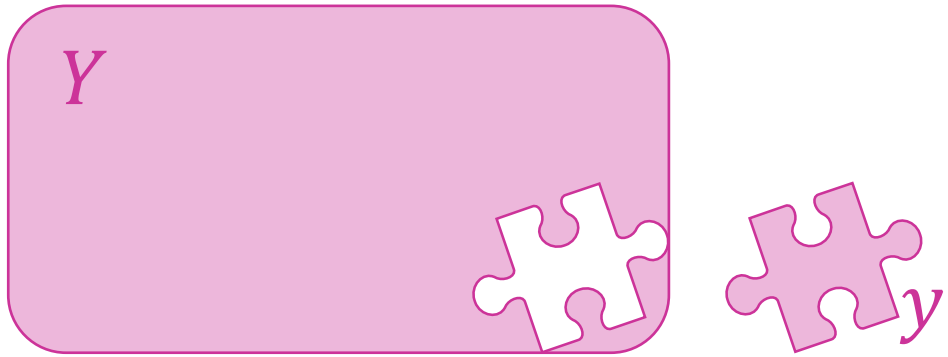


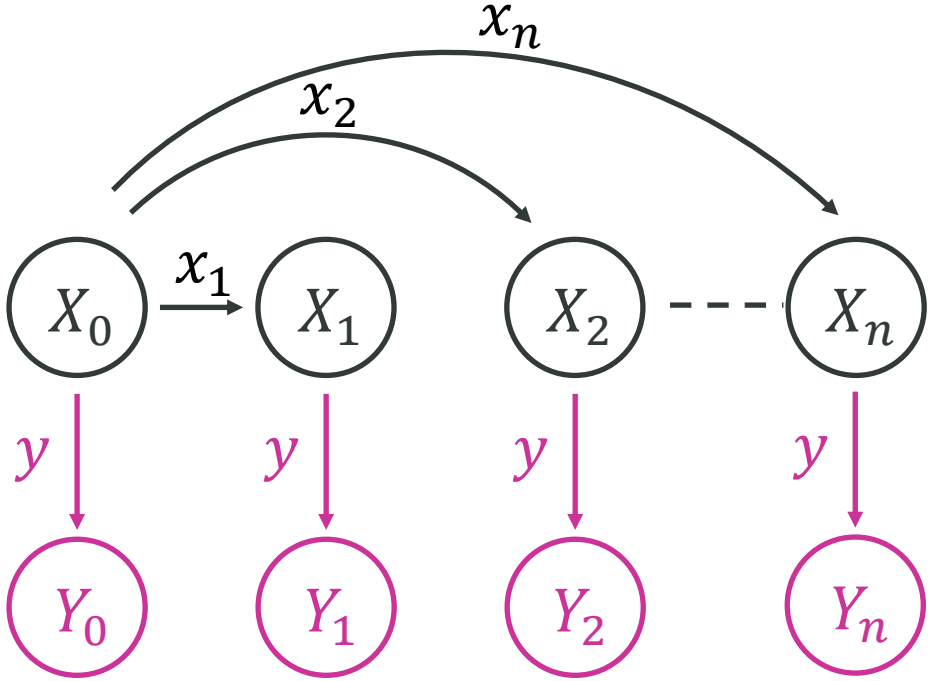


DEFINITION

ADAPTOR CSI-FISH I

<u>Setup</u>: Statement Y for unknown witness y





Define $Y_0 = Y$ and $Y_i = y * X_i$

ADAPTOR CSI-FISH I

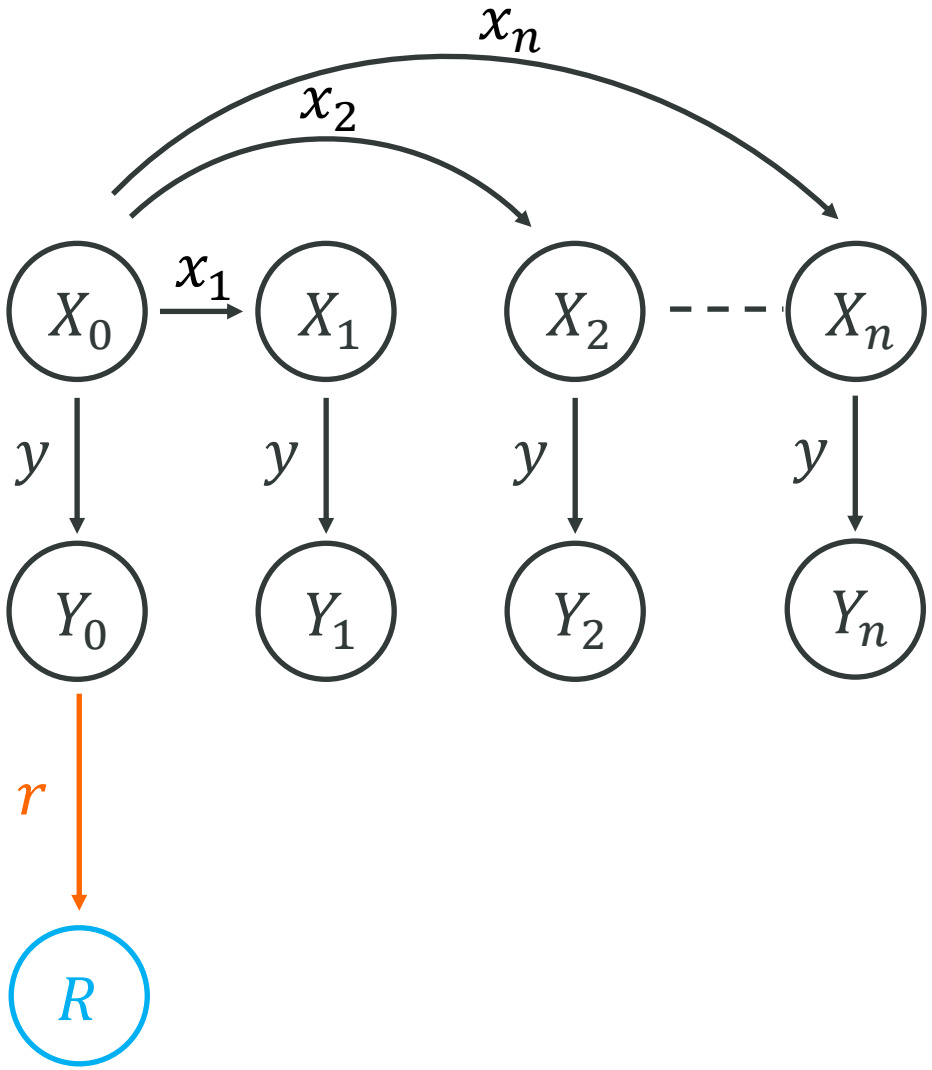
Pre-Signing and Pre-Verification

Prover(X_i, x_i, Y, y, m)

$r \in G$

$R = r * Y_0$

Verifier(X_i, Y, m)



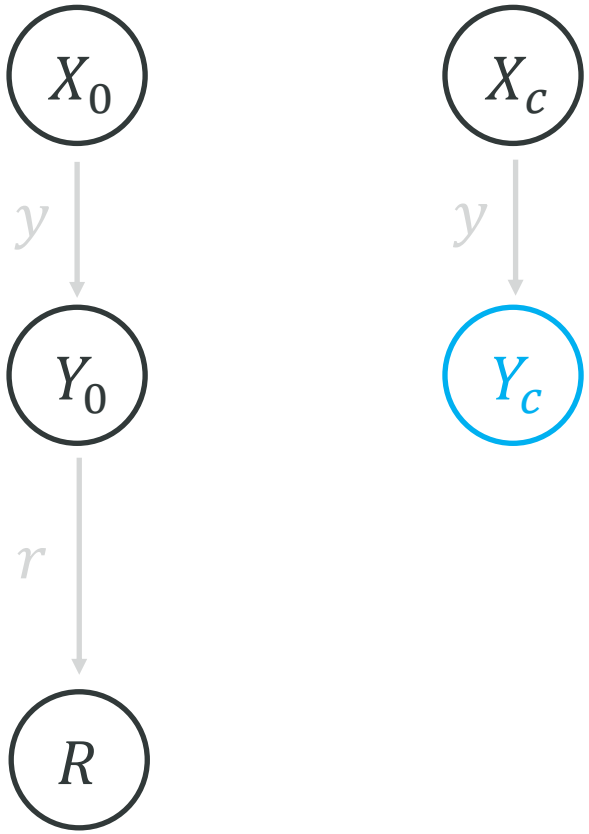
ADAPTOR CSI-FISH I

Pre-Signing and Pre-Verification

Prover(X_i, x_i, Y, y, m)

$$r \in G$$
$$R = r * Y_0$$
$$c = \mathrm{H}(m, R)$$

Verifier(X_i, Y, m)



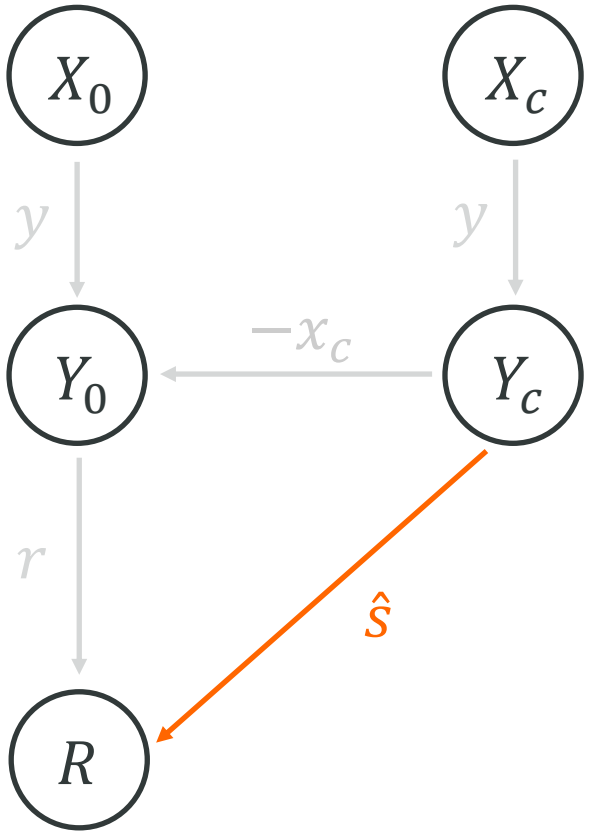
ADAPTOR CSI-FISH I

Pre-Signing and Pre-Verification

Prover(X_i, x_i, Y, y, m)

$$r \in G$$
$$R = r * Y_0$$
$$\mathrm{c} = \mathrm{H}(m, R)$$
$$\hat{s} = r - x_c$$

Verifier(X_i, Y, m)



ADAPTOR CSI-FISH I

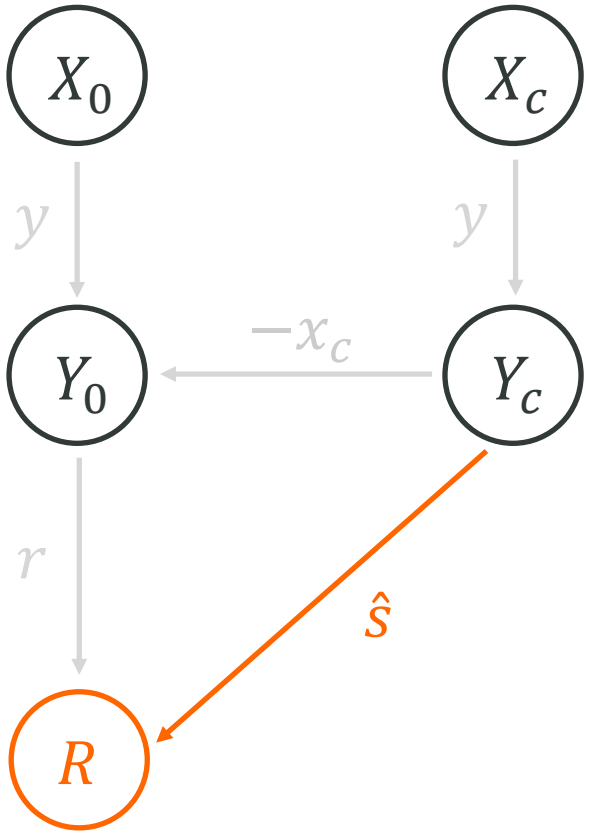
Pre-Signing and Pre-Verification

Prover(X_i, x_i, Y, y, m)

$$r \in G$$
$$R = r * Y_0$$
$$c = \mathrm{H}(m, R)$$
$$\hat{s} = r - x_c$$
$$\hat{\sigma} = (R, \hat{s})$$

Pre-signature

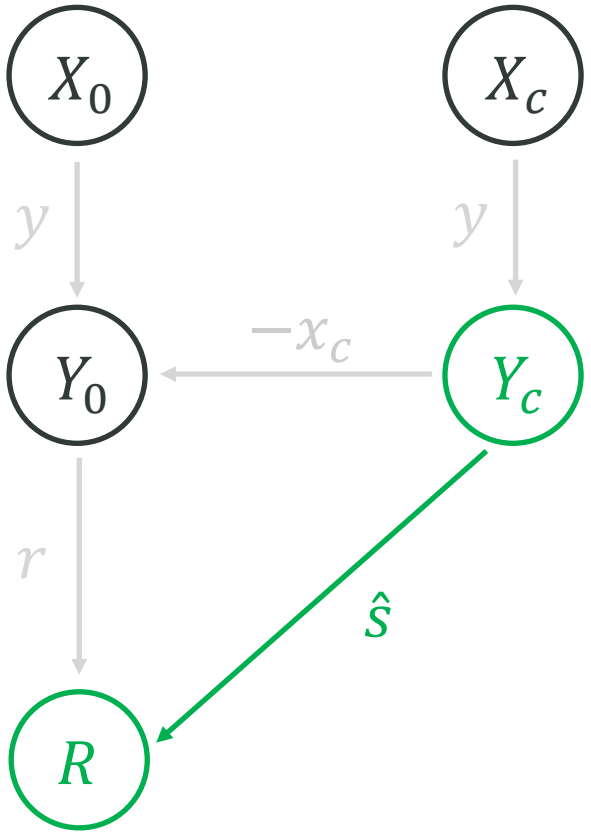
Verifier(X_i, Y, m)



ADAPTOR CSI-FISH I

Pre-Signing and Pre-Verification

Prover(X_i, x_i, Y, y, m)		Verifier(X_i, Y, m)
$r \in G$		
$R = r * Y_0$		
$c = \mathrm{H}(m, R)$		
$\hat{s} = r - x_c$		
	$\hat{\sigma} = (R, \hat{s})$ ⇢	
		Check $\hat{s} * Y_c = R$
	⇠ $0/1$	



ADAPTOR CSI-FISH I

Setup: Statement Y for unknown witness y

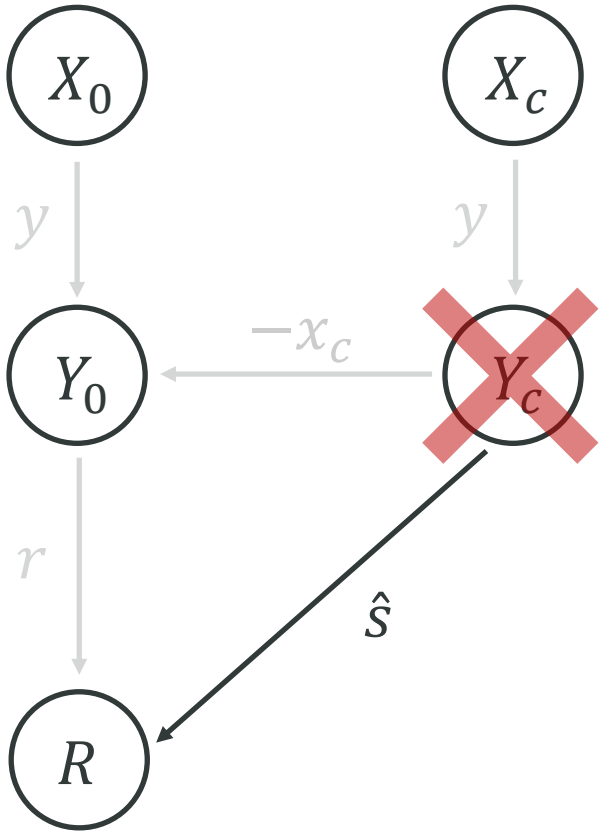
Prover(X_i, x_i, Y, y, m)

$r \in G$

$R = r * Y_0$

Verifier(X_i, Y, m)

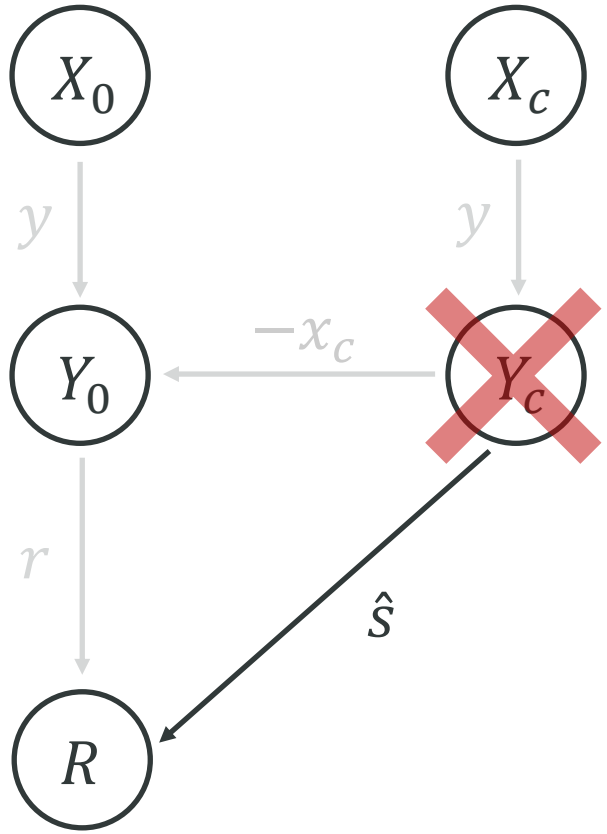
Verifier **CANNOT** compute Y_c for $c \neq 0$!



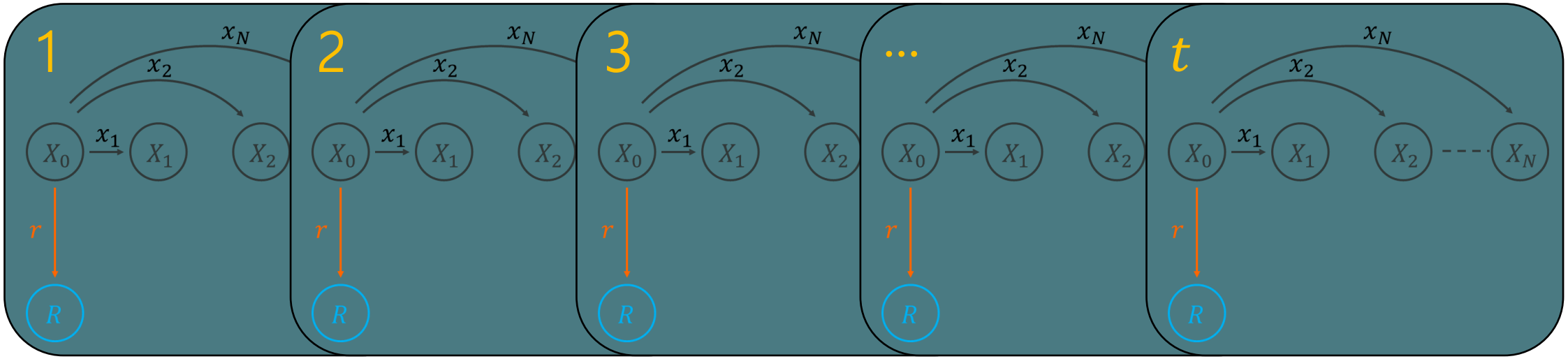
ADAPTOR CSI-FISH I

Verifier CANNOT compute Y_c for $c \neq 0$!

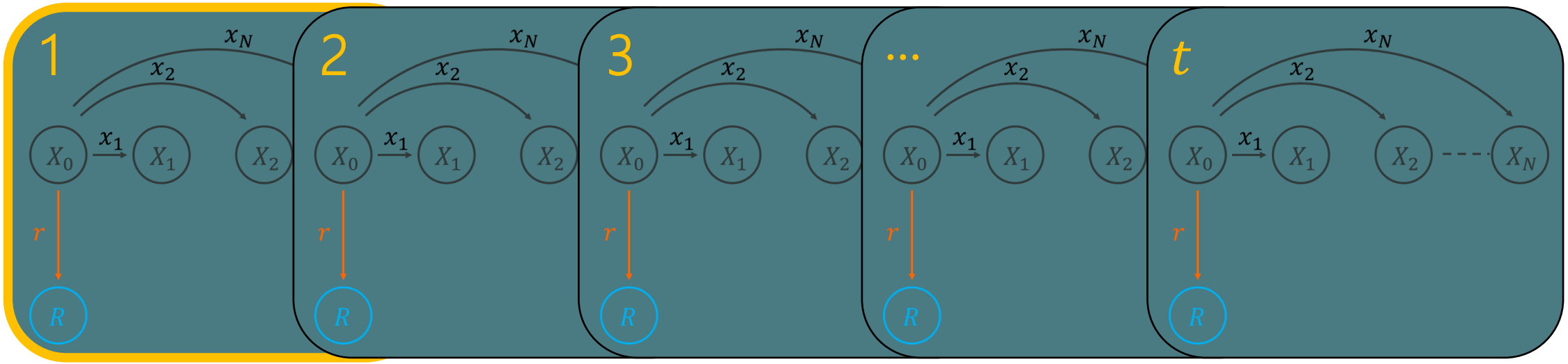
We cheat, forcing $c = 0$



ONE ROUND CHEATING



ONE ROUND CHEATING



It is enough to adapt and cheat just in one round

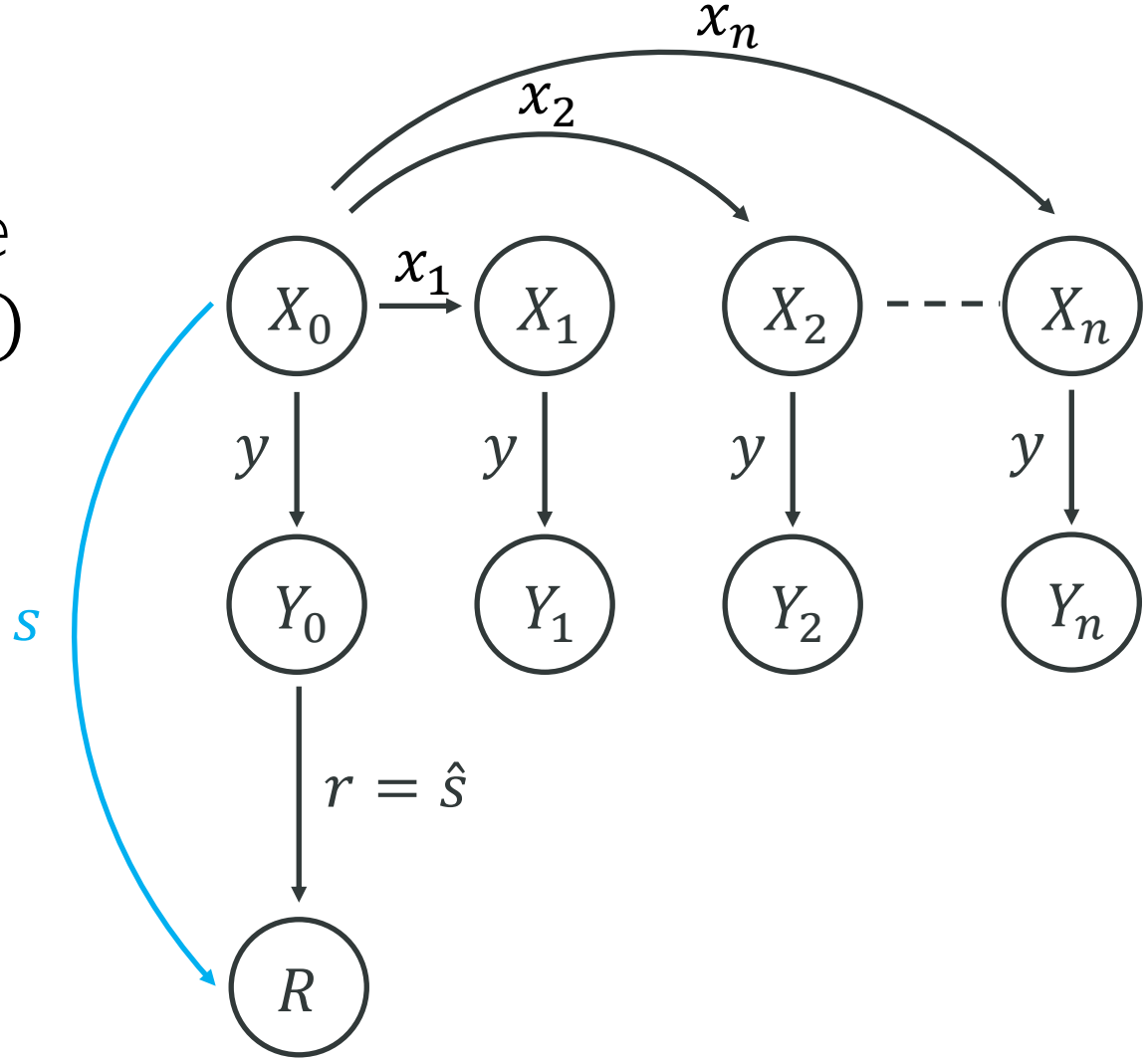
ORCAS: ADAPTOR CSI-FISH

Adapting

pre-signature
$\hat{\sigma} = (R, \hat{s})$

Compute
$s = \hat{s} + y$

signature
$\sigma = (R, s)$



CONSTRUCTION

ORCAS: ADAPTOR CSI-FISH

Adapting

pre-signature
$\hat{\sigma} = (R, \hat{s})$

$\longrightarrow$

signature
$\sigma = (R, s)$

Compute
$s = \hat{s} + y$

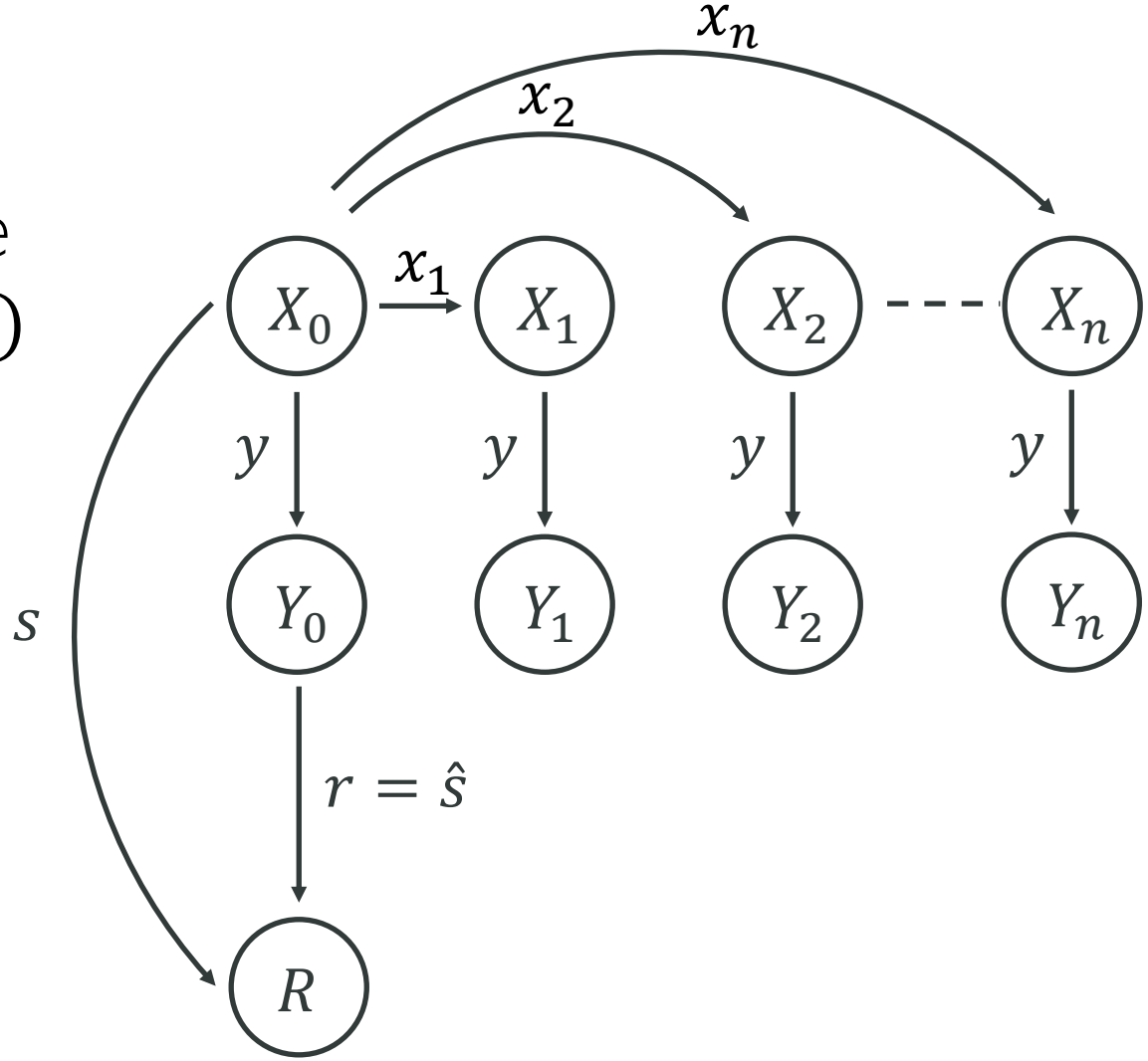
Extracting

pre-signature
$\hat{\sigma} = (R, \hat{s})$

signature
$\sigma = (R, s)$

$\longrightarrow$

witness
y



ORCAS: ADAPTOR CSI-FISH

Adapting

pre-signature
$\hat{\sigma} = (R, \hat{s})$

Compute
$s = \hat{s} + y$

signature
$\sigma = (R, s)$

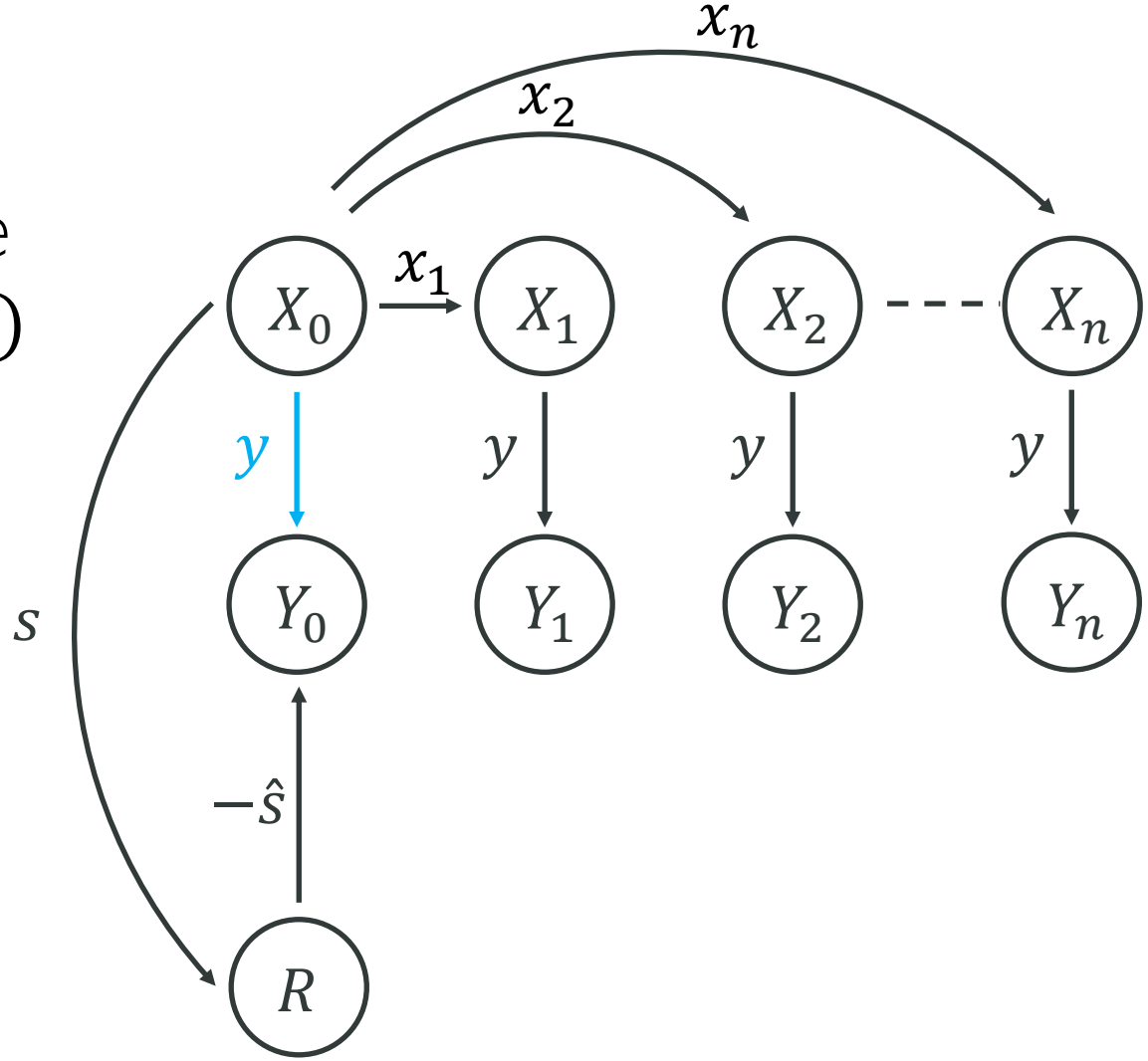
Extracting

pre-signature
$\hat{\sigma} = (R, \hat{s})$

signature
$\sigma = (R, s)$

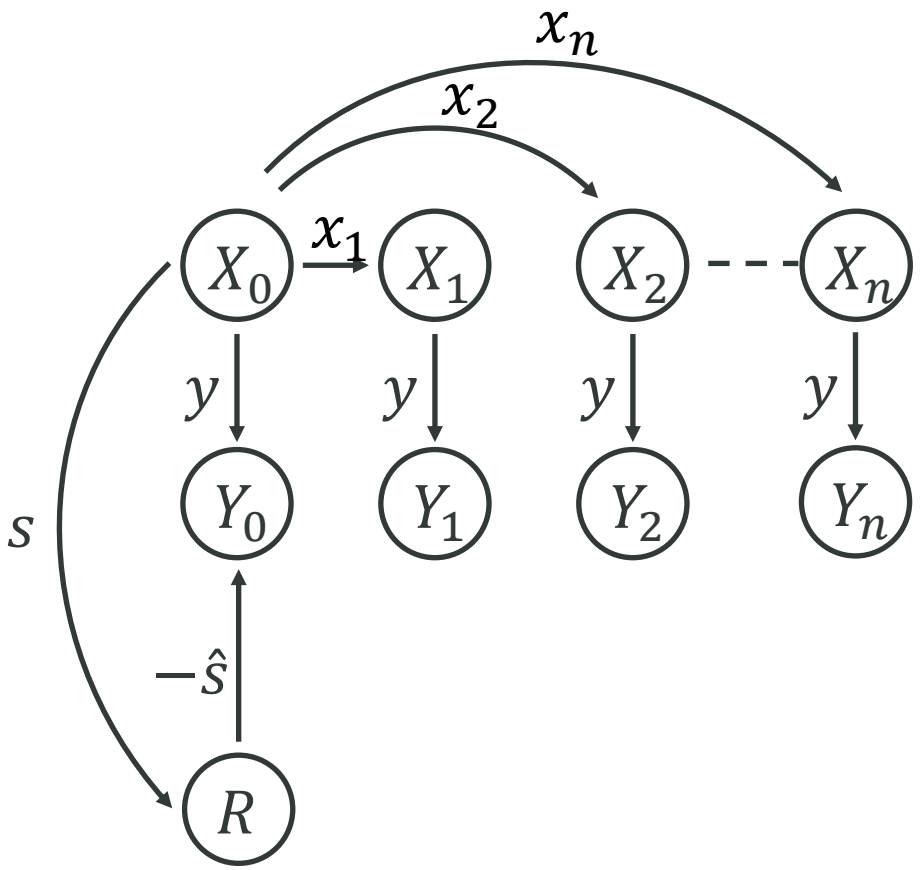
Compute
$y = s - \hat{s}$

witness
y



COMPARISON

PREVIOUS APPROACHES



THANK YOU!

Questions?

Universität Zürich[UZH]

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse Science and Technology
Cyber-Defence Campus

PERFORMANCE

BENCHMARK RESULTS

d	t	k	|sk|	|pk|	$\lvert\widehat{\sigma}\rvert$	$\lvert\sigma\rvert$	KeyGen	Sig	Ver	PreSig	PreVer	Adapt	Ext
1	49	15	16	128	1650	1649	0.11	2.00	2.01	2.03	2.00	0.005	0.005
2	36	14	16	256	1221	1220	0.19	1.46	1.47	1.49	1.47	0.005	0.005
4	23	12	16	1024	792	791	0.70	0.94	0.95	0.99	0.95	0.005	0.005
6	16	16	16	4096	561	560	2.65	0.68	0.68	3.40	0.68	0.005	0.005
8	13	11	16	16384	462	461	10.39	0.53	0.53	0.85	0.54	0.005	0.005
10	11	7	16	65536	396	395	40.67	0.45	0.45	0.55	0.44	0.005	0.005
12	9	11	16	262144	330	329	164.38	0.37	0.36	6.23	0.36	0.005	0.005

Table 1: ORCAS performance. Sizes are in bytes and times in seconds.

- Parameters: 2^d public keys, t rounds, 2^k times slower hash.
- Pre-signatures are just 1 byte bigger than signatures.
- Pre-signing overhead scales as approximately 2^{d+k}.

COMPARISON

PREVIOUS APPROACHES

[1] Tairi *et al.* (2021)

Use of non-interactive zero knowledge proofs.

- ✓ "Honest" adaptor signature scheme.
- ○ Slow pre-signing and large pre-signatures.

[2] Jana *et al.* (2024)

Add extra round to CSI-FiSh which always has $c = 0$.

- ✓ Fast pre-signing and short pre-signatures.
- ○ Adapted pre-signatures incompatible with standard CSI-FiSh.

ORCAS (2026)

Cheat only in one round, repeating until $c = 0$.

- ✓ Fast pre-signing and short pre-signatures.
- ✓ Adapted pre-signatures compatible with standard CSI-FiSh.
- ○ Slow pre-signing for certain parameters.

[1] Tairi, E., Moreno-Sanchez, P., Maffei, M.: Post-Quantum Adaptor Signature for Privacy-Preserving Off-Chain Payments. In: Borisov, N., Díaz, C. (eds.) FC 2021, Part II. LNCS, pp. 131–150. Springer, Berlin, Heidelberg (2021).

[2] Jana, P., Shaw, S., Dutta, R.: Compact Adaptor Signature from Isogenies with Enhanced Security. In: Kohlweiss, M., Di Pietro, R., Beresford, A.R. (eds.) CANS 2024, Part I. LNCS, pp. 77–100. Springer, Singapore (2024).

COMPARISON

PREVIOUS APPROACHES

[1] Tairi *et al.* (2021)

Use of non-interactive zero knowledge proofs.

- ✓ "Honest" adaptor signature scheme.
- ○ Slow pre-signing and large pre-signatures.

[2] Jana *et al.* (2024)

Add extra round to CSI-FiSh which always has $c = 0$.

- ✓ Fast pre-signing and short pre-signatures.
- ○ Adapted pre-signatures incompatible with standard CSI-FiSh.

ORCAS (2026)

Cheat only in one round, repeating until $c = 0$.

- ✓ Fast pre-signing and short pre-signatures.
- ✓ Adapted pre-signatures compatible with standard CSI-FiSh.
- ○ Slow pre-signing for certain parameters.

[1] Tairi, E., Moreno-Sanchez, P., Maffei, M.: Post-Quantum Adaptor Signature for Privacy-Preserving Off-Chain Payments. In: Borisov, N., Díaz, C. (eds.) FC 2021, Part II. LNCS, pp. 131–150. Springer, Berlin, Heidelberg (2021).

[2] Jana, P., Shaw, S., Dutta, R.: Compact Adaptor Signature from Isogenies with Enhanced Security. In: Kohlweiss, M., Di Pietro, R., Beresford, A.R. (eds.) CANS 2024, Part I. LNCS, pp. 77–100. Springer, Singapore (2024).