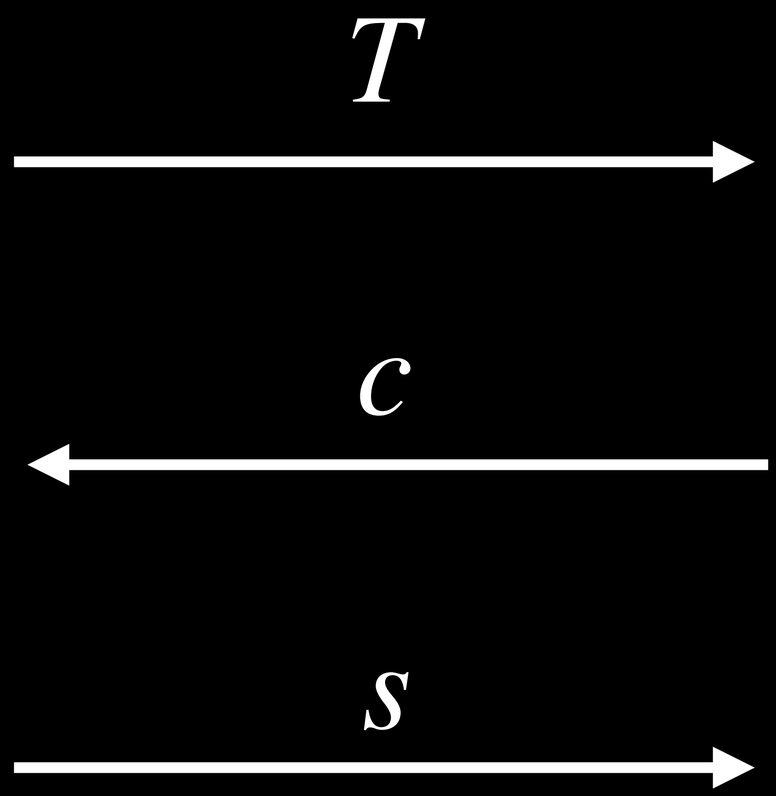
Proposal: Σ-protocols

Stephan Krenn, Michele Orrù

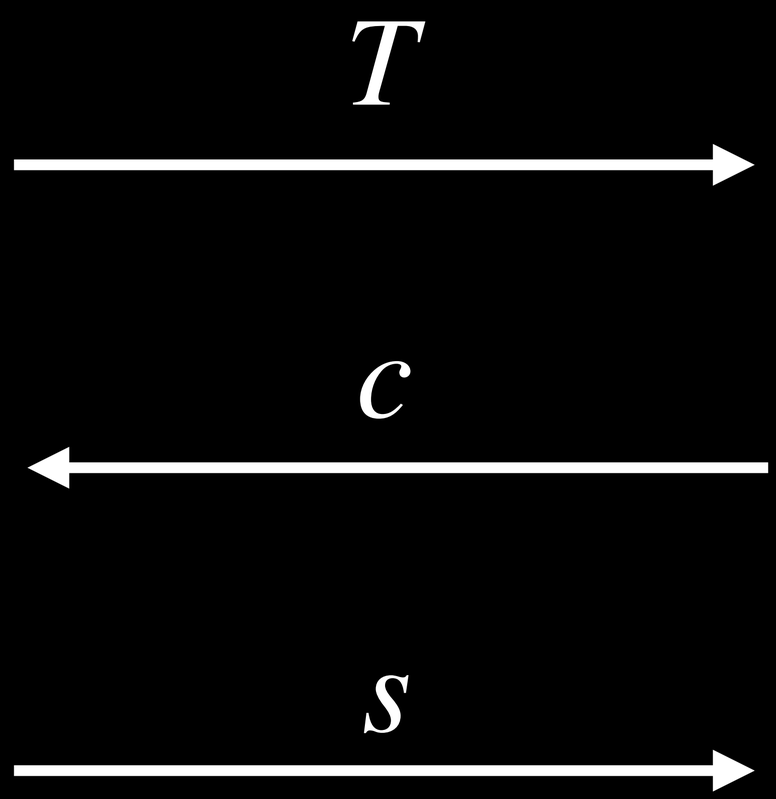
https://docs.zkproof.org/pages/standards/accepted-workshop4/proposal-sigma.pdf

AUSTRIAN INSTITUTE OF TECHNOLOGY

Σ-protocols



Σ-protocols



Special soundness, honest verifier zero-knowledge

In prime fields.

(Some) Σ-protocols

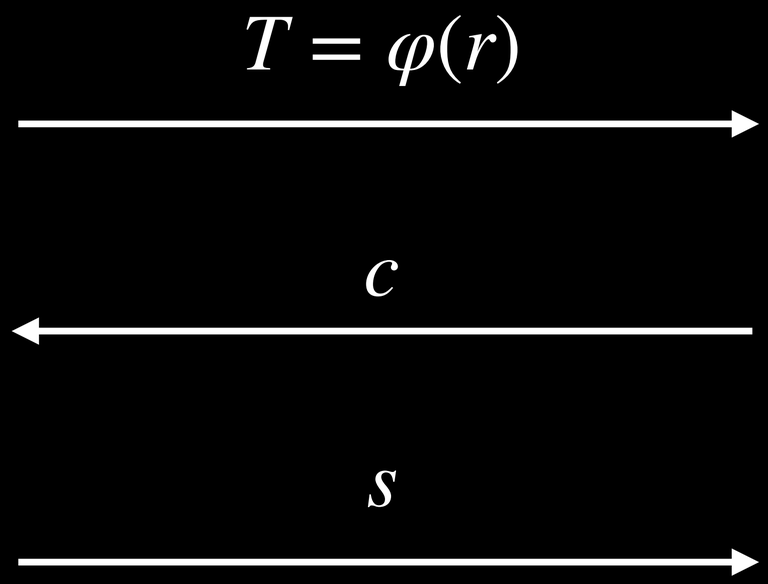
Consider $\varphi : \mathbb{Z}_p^n \to \mathbb{G}^m$.

(Some) Σ-protocols

Consider $\varphi : \mathbb{Z}_p^n \to \mathbb{G}^m$.

$\mathsf{R} = \{(w) : Y = \varphi(w)\}$

$r \leftarrow \mathbb{Z}_p^n$



$\varphi(s) = T + cY.$

(Some) Σ-protocols

Consider $\varphi : \mathbb{Z}_p^n \to \mathbb{G}^m$.

$\mathsf{R} = \{(w) : Y = \varphi(w)\}$

wG

$r \leftarrow \mathbb{Z}_p^n$

rG

$T = \varphi(r)$

c

s

sG

$\varphi(s) = T + cY.$

(Some) Σ-protocols

Consider $\varphi : \mathbb{Z}_p^n \to \mathbb{G}^m$.

$\mathsf{R} = \{(w) : Y = \varphi(w)\}$

$w\begin{bmatrix} G \\ H \end{bmatrix} = y$

$r \leftarrow \mathbb{Z}_p$

$T = \varphi(r)$ $\quad r\begin{bmatrix} G \\ H \end{bmatrix}$

$\longrightarrow$

c

$\longleftarrow$

s

$\longrightarrow$

$s\begin{bmatrix} G \\ H \end{bmatrix}$

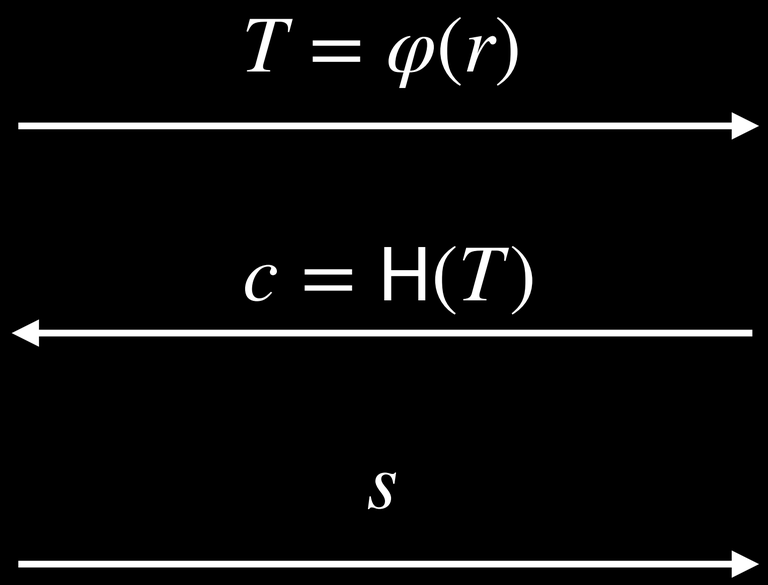
$\varphi(s) = T + cY.$

Non-interactive.

(Some) Σ-protocols

Consider $\varphi : \mathbb{Z}_p^n \to \mathbb{G}^m$.

$\mathsf{R} = \{(w) : Y = \varphi(w)\}$



$c = \mathsf{H}(T)$

$\varphi(s) = T + cY.$

Linear Relations

$$\mathsf{R} = \{(w) : Y = \varphi(w) \ \wedge \ Aw = b\}.$$

OR-composition

$$\mathsf{R} = \{(w) : Y_0 = \varphi(w) \ \vee \ Y_1 = \varphi(w)\}\,.$$

AND-composition

$$\mathsf{R} = \{(w_0, w_1) : Y_0 = \varphi(w_0) \ \wedge \ Y_1 = \varphi(w_1)\}.$$

Down to bytes

Choosing the group

Disclosure of a Major Bug in CryptoNote Based Currencies

Posted by: luigi1111 and Riccardo "fluffypony" Spagni

May 17, 2017

In Monero we've discovered and patched a critical bug that affects all CryptoNote-based cryptocurrencies, and allows for the creation of an unlimited number of coins in a way that is undetectable to an observer unless they know about the fatal flaw and can search for it.

Choosing the group

- Prime-order groups
- Prime-order group abstractions
- Pairing-friendly groups

Commitment

$$T := \varphi(r)$$

$$r \leftarrow \mathbb{Z}_p^n$$

$$r := \mathsf{H}(w, Y)$$

[c.f. RFC 6979. This is not the right way to do it. Don't do it in practice.]

- More prone to errors

- Incompatible with OR
- Deterministic prover

Challenge

$$\mathrm{H}(T, \qquad)$$

Challenge

$$\mathrm{H}(T, Y, \quad)$$

Challenge

$$\mathrm{H}(T, Y, \mathrm{gen}, \quad)$$

Challenge

$$\mathrm{H}(T, Y, \mathrm{gen}, \mathrm{curve}, \quad)$$

Challenge

$$\mathrm{H}(T, Y, \mathrm{gen}, \mathrm{curve}, \mathrm{ds})$$

Challenge Challenges

$$\mathrm{H}(T, Y, \text{gen, curve, ds}) \in \mathbb{Z}_p$$

- Length extension attacks
- "*Chop off at 256bits*", draft-irtf-cfrg-hash-to-curve, STROBE

Response

Short

(c, s)

Batchable

(T, s)

Response

Short

(c, s)

Batchable

(T, s)

$(T_1, s_1), \ldots, (T_k, s_k)$ for y

$\varphi\left(\sum_i e_i s_i\right) = \sum_i e_i T_i + \left(\sum_i e_i c_i\right) y$

[for random e_i's.]

What's out there

Project	Language	AND	OR	INT	FS
Cashlib	C++	✓			✓
Emmy	Go			✓	
Kyber	Go	✓	✓	✓	✓
SCAPI	C++	✓	✓	✓	✓
YAZKC	C	✓	✓	✓	✓
zkp	Rust	✓			✓
zksk	Python	✓	✓	✓	✓

Most of this table's data comes from the work of Lueks et al. [LKF+19]. Thanks!

Σ-protocols: limits

- When Σ-protocols are useless;
- One thought about post-quantum resistance.

Missing something?

Looking Ahead

- R1CS compatibility;
- Shared proof computation;
- Designated verifier;
- Interactive protocol.

What next?

michele.orru@berkeley.edu