Snarky Signatures: Minimal Signatures of Knowledge from Simulation-Extractable SNARKs

Jens Groth – University College London
Mary Maller – University College London

Crypto Santa Barbara: 21/08/2017

How can a sender of a message prove themselves trustworthy without revealing who they are?

Example: Bitcoin

Bitcoin uses digital signatures:

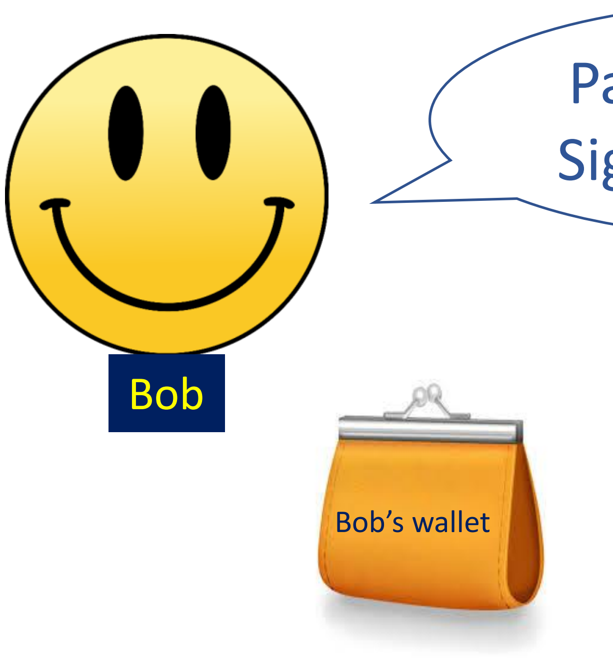
- Trustworthy?
- Anonymous?

Example: Bitcoin

Bitcoin uses digital signatures:

- Trustworthy? ✓
  - Bob can always convince the verifiers.
  - An adversary cannot forge Bob's signature.
  - An adversary cannot use Bob's signature on a different message.
- Anonymous?

Example: Bitcoin



A Fistful of Bitcoins: Characterizing Payments Among Men with No Names

Sarah Meiklejohn Marjori Pomarole Grant Jordan
Kirill Levchenko Damon McCoy[†] Geoffrey M. Voelker Stefan Savage

University of California, San Diego George Mason University[†]

ABSTRACT

Categories and Subject Descriptors

Keywords

1. INTRODUCTION

Bitcoin uses digital signatures:

- Trustworthy = 
- Anonymous = 
  - Pseudonymous – Bob's real name might be Roberta.
  - Can often uncover Bob's real world identity based on what he spends.

Example: Zerocoin

Owner of an unspent coin

Zerocoin uses signatures of knowledge:

- Trustworthy? ✓
  - The owner of an unspent coin can compute a signature.
  - A person without an unspent coin cannot compute a signature.
  - A signature cannot be adapted for use on a different message.
- Anonymous?

Example: Zerocoin

Zerocoin uses signatures of knowledge:

- Trustworthy =
- Anonymous =
    - Signature of knowledge provides no additional information as to who the spender is.

Example: Zerocoin

Owner of an unspent coin

Zerocoin uses signatures of knowledge:

- Trustworthy =
- Anonymous =
  - Signature of knowledge provides no additional information as to who the spender is.

However signatures of knowledge are large and take a long time to verify

=> Zerocoin not efficient.

Example: Zcash

Zcash uses zk-SNARKs:

- Trustworthy?
- Anonymous?
  - zk-SNARKs provides no additional information as to who the spender is.

zk-SNARKs are small and take a small time to verify

=> Zcash is efficient.

Example: Zcash

Zcash uses zk-SNARKs:

- Trustworthy =
  - The owner of an unspent coin can compute a proof.
  - A person without an unspent coin cannot compute a proof.
  - A proof cannot be adapted for use on a different message?????
- Anonymous = 

Standard zk-SNARKs do not provide this property. Zcash has to take additional steps to prevent transaction malleability.

Our Contributions

- We construct the first simulation-extractable SNARK (SE-SNARK).
- We exploit a link between signatures of knowledge and SE-SNARKs to also get the first succinct signature of knowledge.

Ingredients:

1. Asymmetric bilinear groups;
2. Square arithmetic programs;
3. External power knowledge of exponent assumption;
4. Computational assumption.

Plan

Definitions

Square Arithmetic Programs

Construction

Efficiency

Asymmetric Bilinear Groups

prime

$$bp = (p, \mathbb{G}, \mathbb{H}, \mathbb{T}, e)$$

Function $e: \mathbb{G} \times \mathbb{H} \rightarrow \mathbb{T}$

Groups of order p

- There are efficient algorithms for deciding group membership and computing group operations;

- No isomorphism between $\mathbb{G}$ and $\mathbb{H}$ is efficiently computable in either direction.

Properties:

Bilinearity:	$e(G^a, H^b) = e(G, H)^{ab}$
Non-degeneracy:	*if* $X \neq 1$ *and* $Y \neq 1$ *then* $e(X, Y) \neq 1$
Efficient:	*e is efficiently computable.*

SE-SNARK

Simulation-Extractable zero-knowledge Succinct Non-interactive ARgument of Knowledge

SE-SNARKs

“A person knows a witness for an instance Φ.”

Properties:

Correct:	*A person who knows a witness can always convince the verifier.*
Zero Knowledge:	*The verifier learns no information from the proof except that the instance is true.*
Sound:	*A false statement cannot be proven.*
Simulation-Extractable:	*Old proofs cannot be used to forge new proofs of false statements.*

Zero-Knowledge

Zero Knowledge: *The verifier learns no information from the proof except that the instance is true.*

$(CRS, \tau) \leftarrow Setup(R)$

CRS

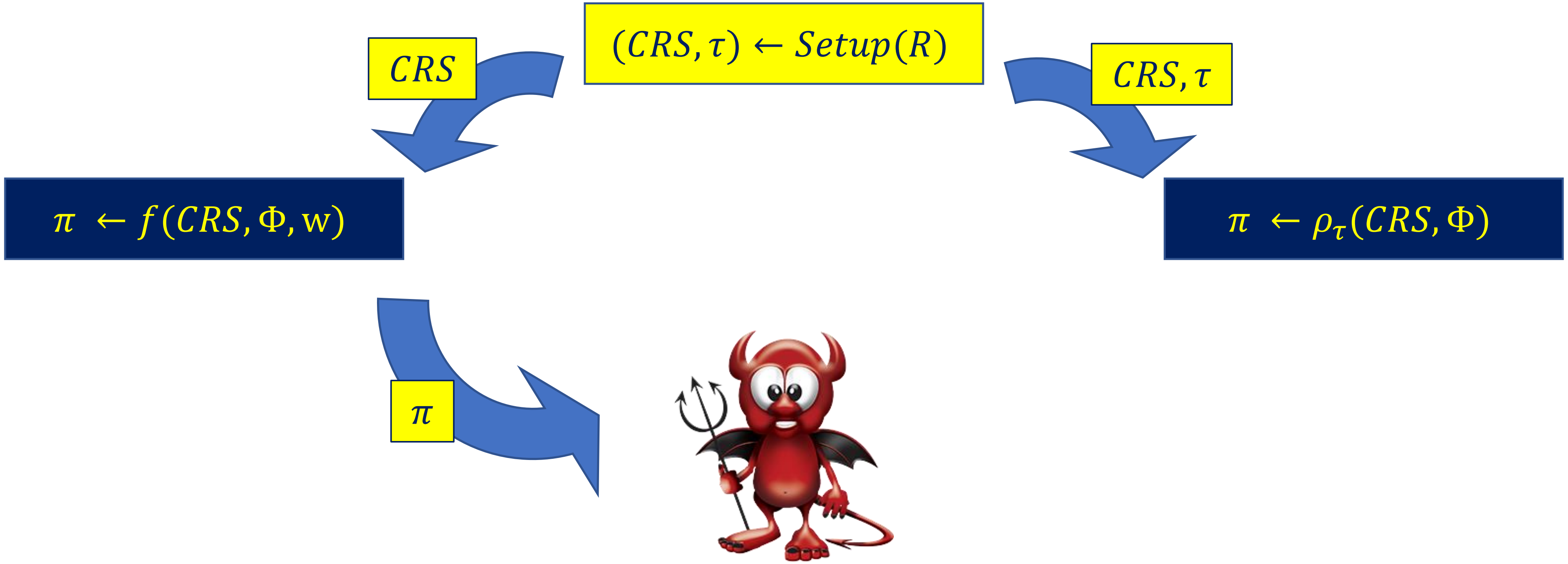
CRS, τ

$\pi \leftarrow f(CRS, \Phi, w)$

$\pi \leftarrow \rho_\tau(CRS, \Phi)$

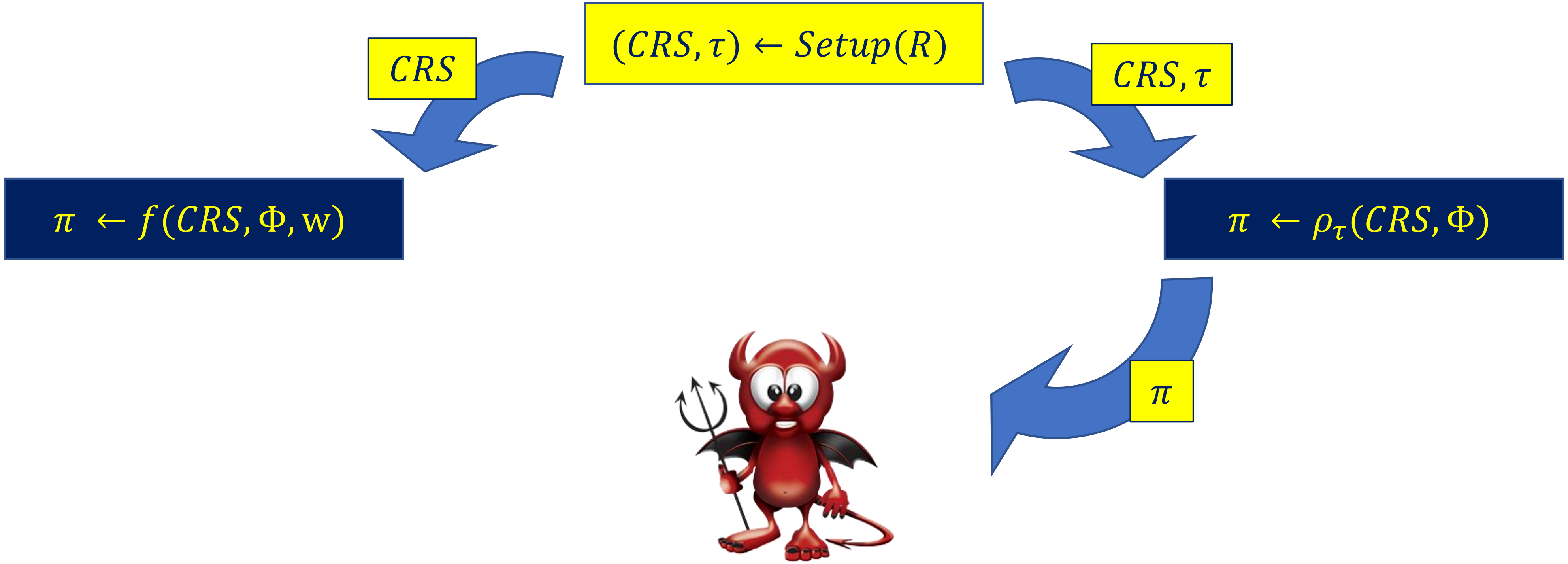
Zero-Knowledge

Zero Knowledge: *The verifier learns no information from the proof except that the instance is true.*



Zero-Knowledge

Zero Knowledge: *The verifier learns no information from the proof except that the instance is true.*



Zero-Knowledge

Zero Knowledge: *The verifier learns no information from the proof except that the instance is true.*

$(CRS, \tau) \leftarrow Setup(R)$

CRS

CRS, τ

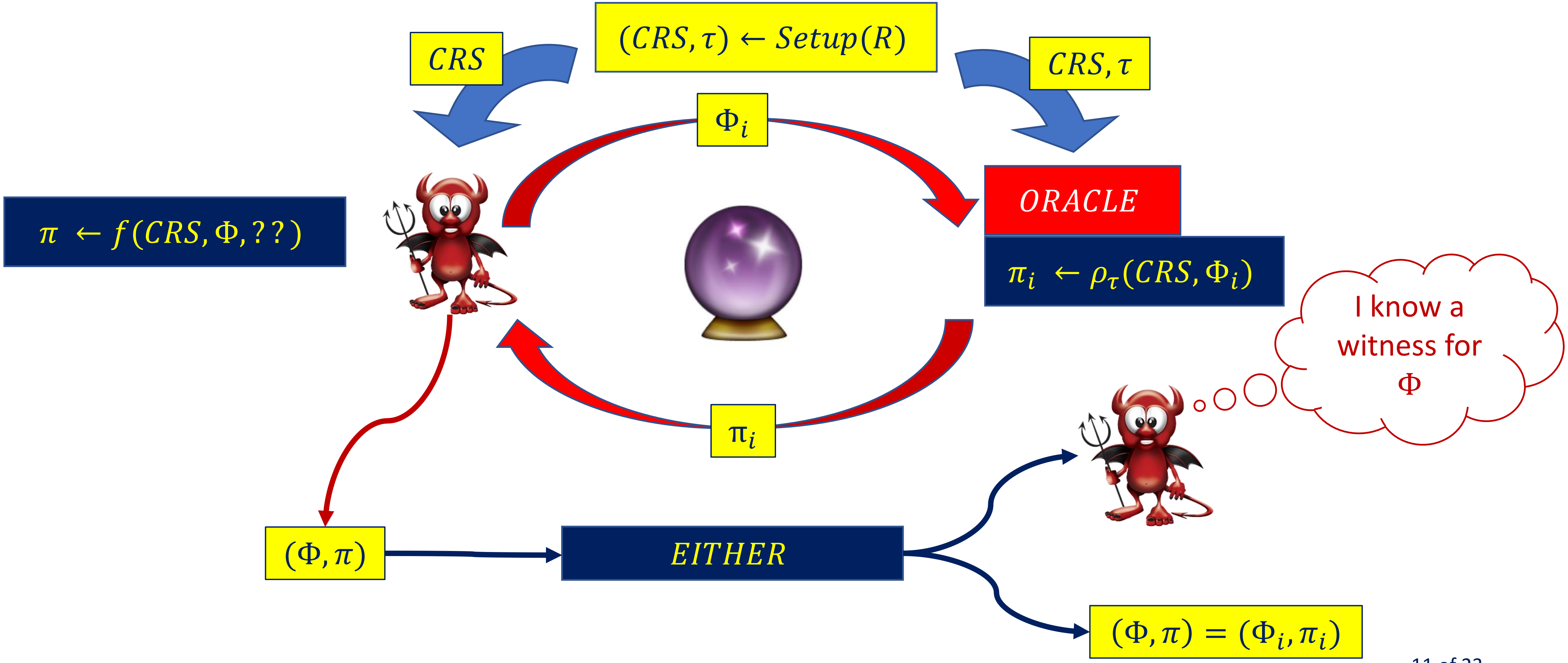
$\pi \leftarrow f(CRS, \Phi, w)$

$\pi \leftarrow \rho_\tau(CRS, \Phi)$

Did the prover use the witness or the trapdoor to compute π?

Simulation-Extractability

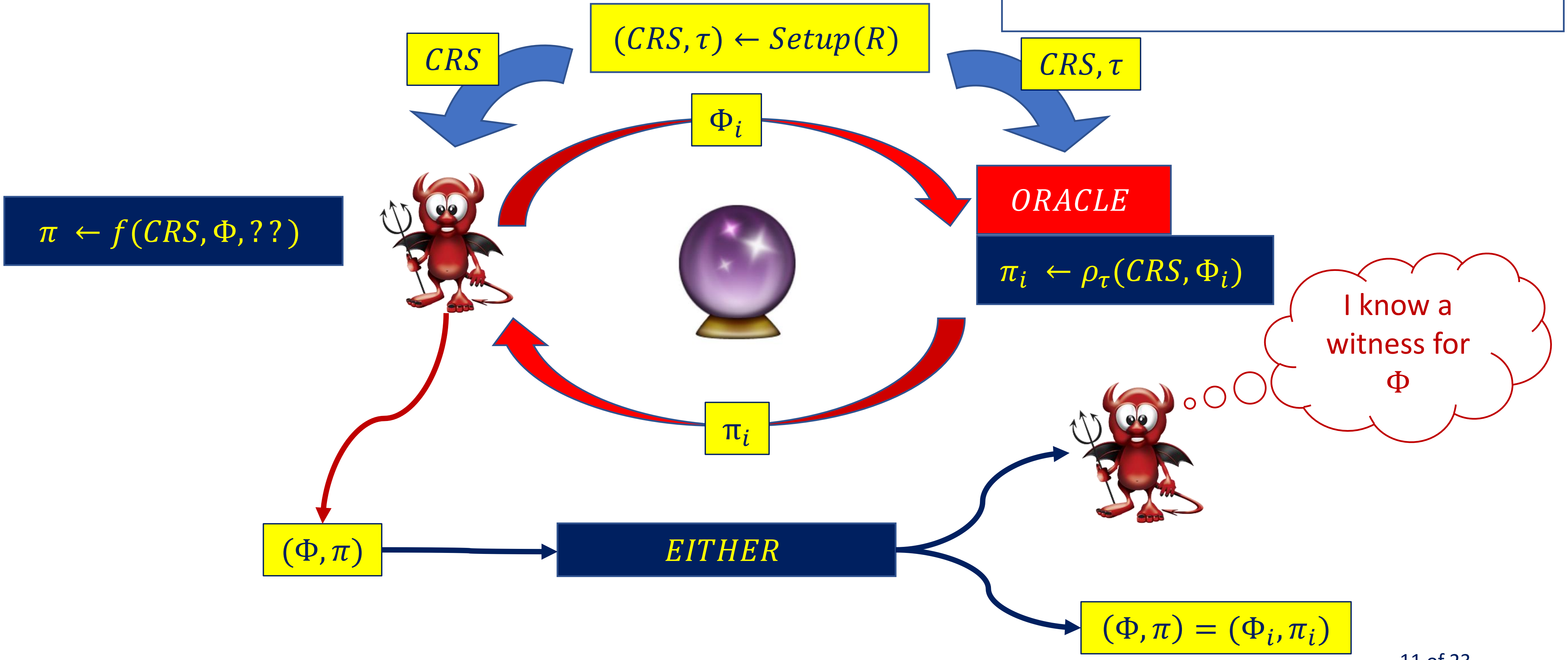
Simulation-Extractable: Old proofs cannot be used to forge new proofs of false statements.



Simulation-Extractability

Simulation-Extractable: *Old proofs cannot be used to forge new p*

Implies Soundness



Succinctness

The size of the proof and the time taken to verify a proof does not depend on the size of the witness.

Signature of Knowledge

"A person who knows a witness for an instance Φ has signed a message."

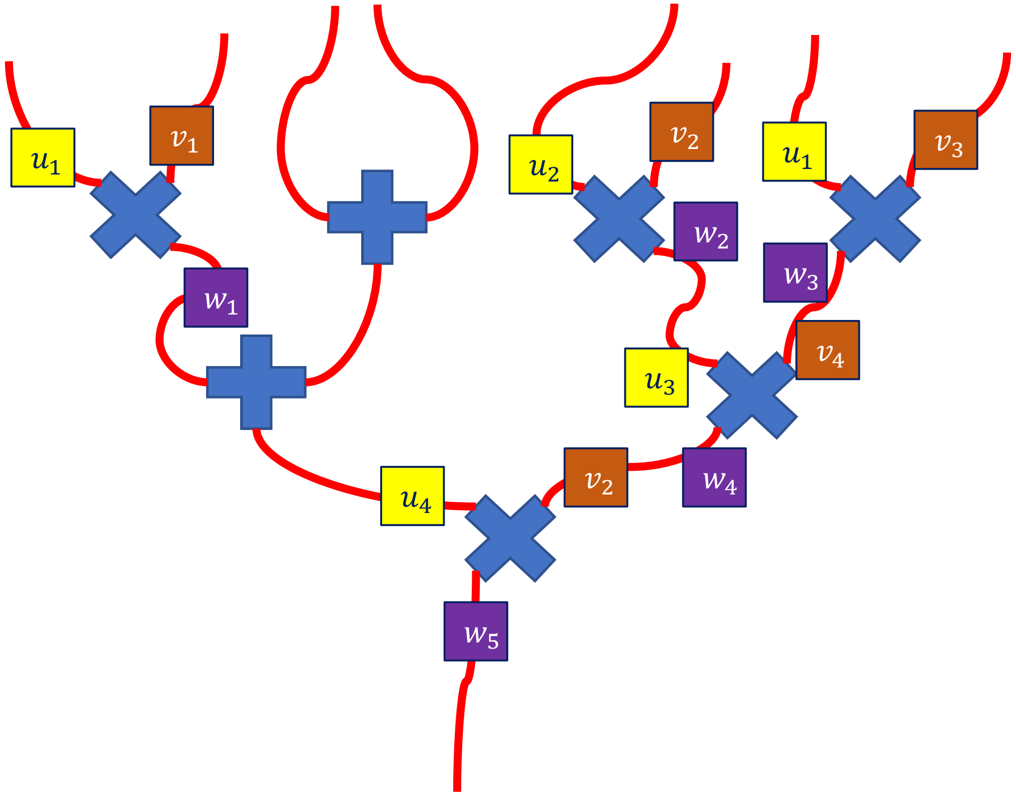
Properties:

Correct:	*A person who knows a witness can always convince the verifier.*
Zero Knowledge:	*The verifier learns no information from the signature except that the instance is true.*
Sound:	*A false statement cannot be signed.*
Simulation-Extractable:	*Old signatures cannot be used to forge new signatures of false statements.*

Plan

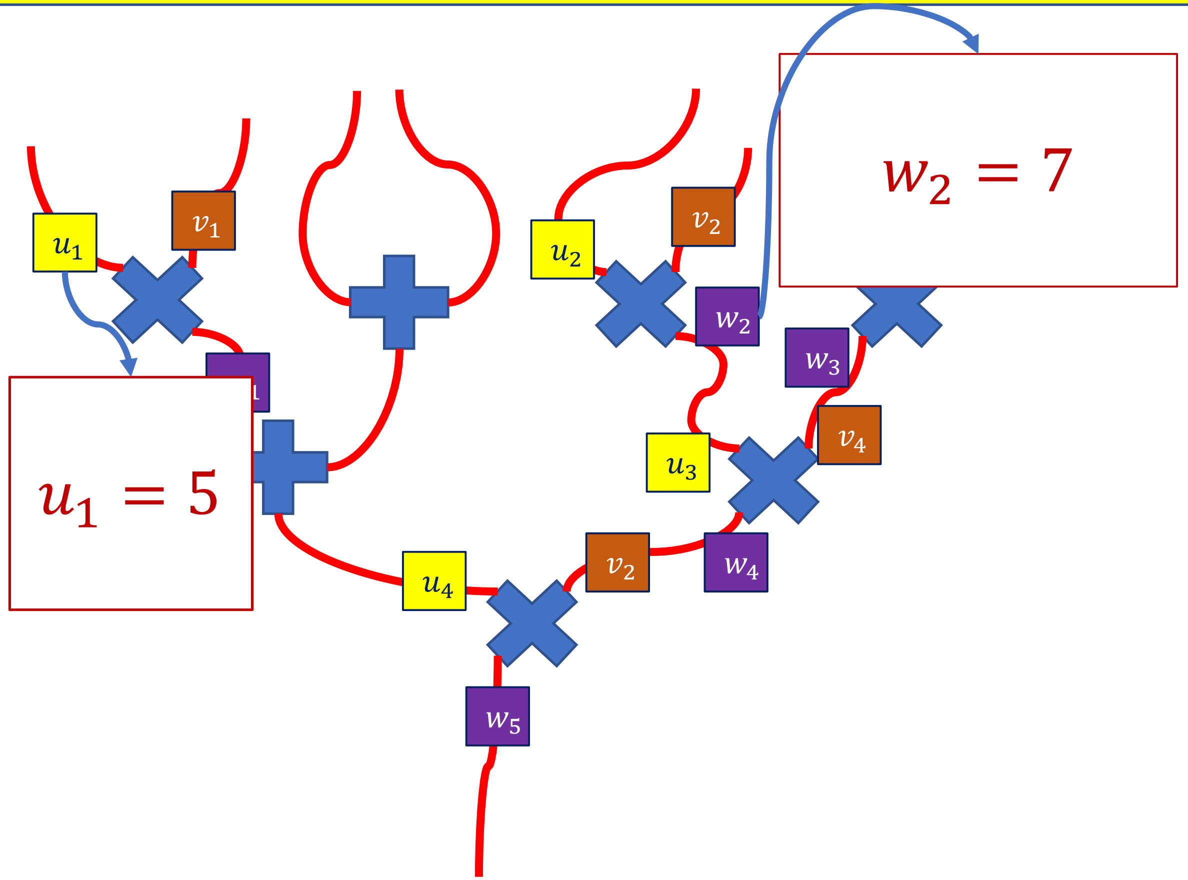
- Definitions
- Square Arithmetic Programs
- Construction
- Efficiency

Arithmetic Circuits



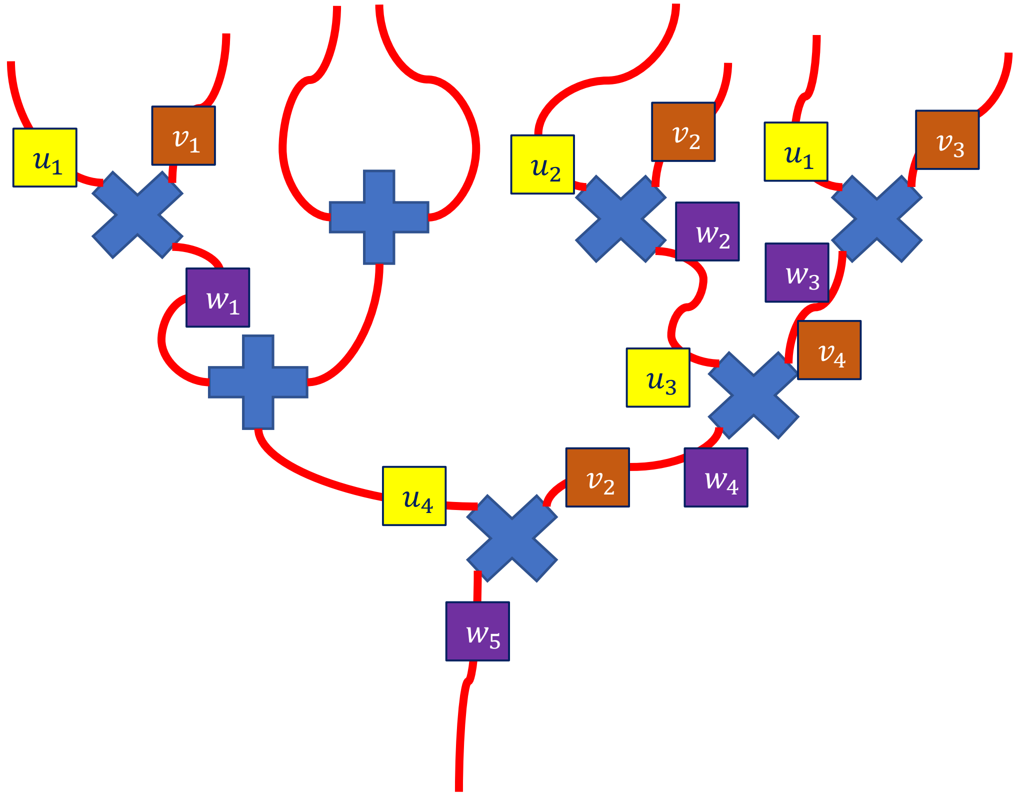
- Encoding of NP languages.
- The instance is some of the wire values that are revealed.
- The witness is the value of the remaining wires.

Arithmetic Circuits



- Encoding of NP languages.
- The instance is some of the wire values that are revealed.
- The witness is the value of the remaining wires.

Arithmetic Circuits



- Prover commits to values of wires.
- Prover shows
  - Output wires consistent with input wires.
  - Multiplication and addition gates calculated correctly.

Quadratic Arithmetic Programs [GGPREurocrypt13]

Relation described by

degree $n-1$ polynomials

degree n polynomial

$$R = \left(p, \ell, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}, t(X)\right)$$

Instance $\Phi = (s_1, \ldots, s_\ell)$ and witness $w = (s_{\ell+1}, \ldots, s_m)$ satisfy arithmetic circuit C if and only if

$$\left(\sum_{i=0}^{m} s_i u_i(X)\right)\left(\sum_{i=0}^{m} s_i v_i(X)\right) = \left(\sum_{i=0}^{m} s_i w_i(X)\right) + mod\; t(X)$$

Quadratic Arithmetic Programs [GGPREurocrypt13]

Relation described by

degree n − 1 polynomials

degree n polynomial

$$R = \left(p, \ell, \{u_i(X), v_i(X), w_i(X)\}_{i=0}^{m}, t(X)\right)$$

Instance $\Phi = (s_1, \ldots, s_\ell)$ and witness $w = (s_{\ell+1}, \ldots, s_m)$ satisfy arithmetic circuit C if and only if

$$\left(\sum_{i=0}^{m} s_i u_i(X)\right)\left(\sum_{i=0}^{m} s_i v_i(X)\right) = \left(\sum_{i=0}^{m} s_i w_i(X)\right) + mod\ t(X)$$

$s_0 = 1$

Square Arithmetic Programs

We can ensure that left input wires and right input wires are the same if we double the size of the circuit.

degree $2n-1$ polynomials

degree $2n$ polynomial

Relation described by

$$R = \left(p, \ell, \{u_i(X), w_i(X)\}_{i=0}^{m}, t(X)\right)$$

$(s_1, \dots, s_m)$ satisfy circuit C if and only if

$$\left(\sum_{i=0}^{m} s_i u_i(X)\right)^2 = \left(\sum_{i=0}^{m} s_i w_i(X)\right) + mod\ t(X)$$

$s_0 = 1$

Plan

Definitions

Square Arithmetic Programs

Construction

Efficiency

Groth Eurocrypt 2016 Construction

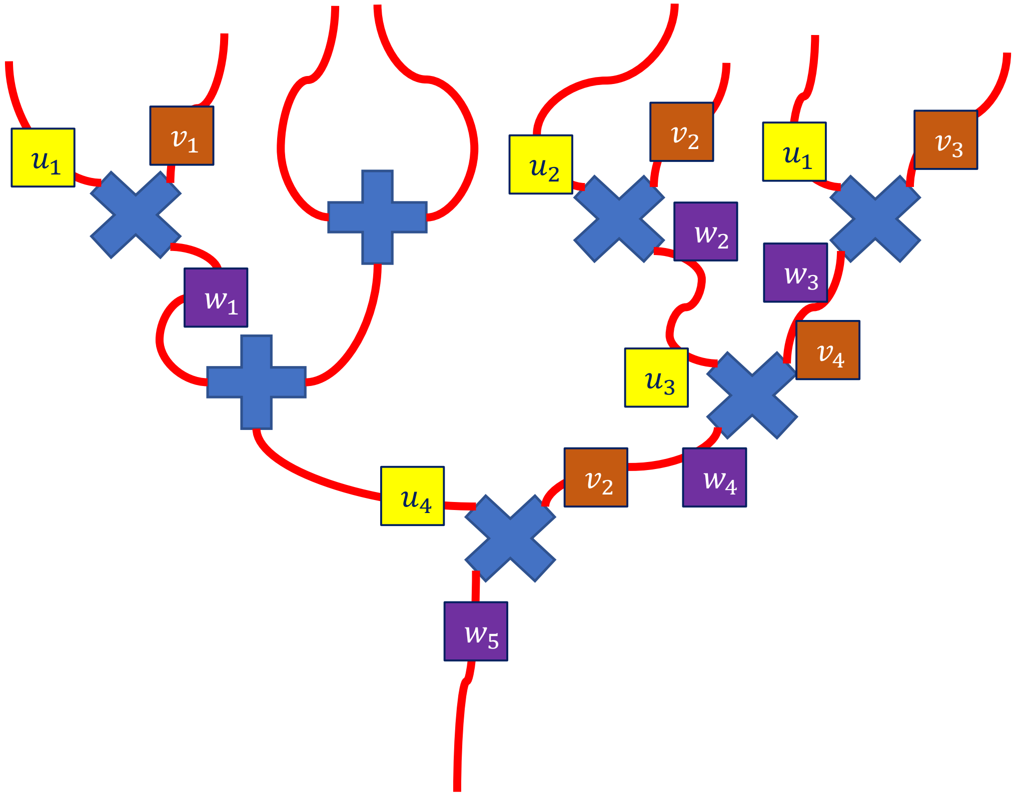
Instance = Φ

Commitment to left input wires

Commitment to right input wires

Proof = (A, B, C) group elements.

Commitment to output wires



Groth Eurocrypt 2016 Construction

Instance = Φ

Proof = (A, B, C) group elements.

Verification Equation:

Known function of Φ

$$e(A,B) = e(G^{\alpha}, H^{\beta})\, e(G^{f(\Phi)\frac{1}{\delta_1}}, H^{\delta_1})\, e(C, H^{\delta})$$

secret

Each of these pairings contribute towards knowledge soundness

Groth Eurocrypt 2016 is Sound

Multiplication and addition gates evaluated correctly

$$e(A,B) = e(G^{\alpha}, H^{\beta})\, e\left(G^{f(\Phi)\frac{1}{\delta_1}}, H^{\delta_1}\right) e(C, H^{\delta})$$

α, β ensure internal wires are consistent

Hard to find $G^{f(\Phi)}$ or $H^{f(\Phi)}$
=
prover must use their witness.

Groth Eurocrypt 2016 not SE

Suppose A, B, C satisfy

$$e(A,B) \;=\; e(G^{\alpha}, H^{\beta})\,e(G^{f(\Phi)\frac{1}{\delta_1}}, H^{\delta_1})\,e(C, H^{\delta})$$

Then so does
$A^r, B^{\frac{1}{r}}, C$

Then so does
$A, B \cdot H^{r\partial}, A^r \cdot C$

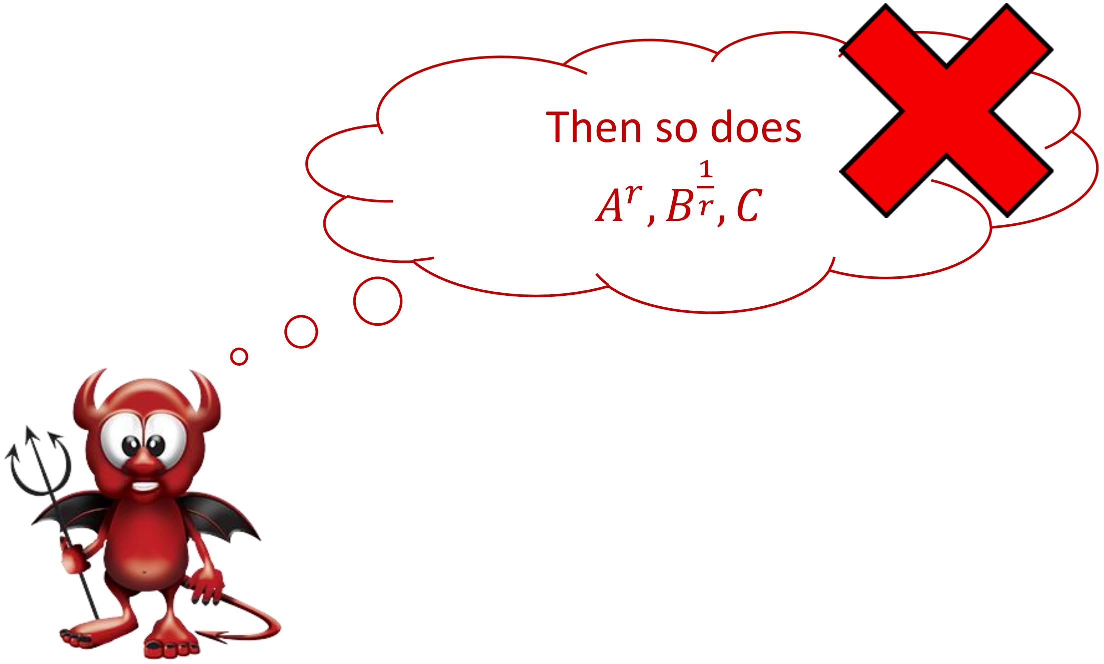
Our Techniques: 1

Suppose A, B, C satisfy

$$e(AG^{\alpha}, BH^{\beta}) = e(G^{\alpha}, H^{\beta})\, e(G^{f(\Phi)\frac{1}{\delta_1}}, H^{\delta_1})\, e(C, H^{\delta})$$

Second verification equation

$$e(A, H^{\gamma}) = e(G^{\gamma}, B)$$



Our Techniques: 2

Suppose A, B, C satisfy

$$e(AG^{\alpha}, BH^{\beta}) = e(G^{\alpha}, H^{\beta})e(G^{f(\Phi)}, H^{\gamma})e(C, H)$$

$$e(A, H^{\gamma}) = e(G^{\gamma}, B)$$

CRS contains $H, G^{\gamma}, H^{\gamma}$ but not G

Then so does $A', B' = BH^{r}, C'???$

Need second verification equation

Cannot calculate C' from the CRS

Implies $A' = AG^{r}$

Implies r depends on γ

Implies C' contains a factor of γ^2

Our Techniques: 2

Suppose A, B, C satisfy

$$e(AG^{\alpha}, BH^{\beta}) = e(G^{\alpha}, H^{\beta})e(G^{f(\Phi)}, H^{\gamma})e(C, H)$$

$$e(A, H^{\gamma}) = e(G^{\gamma}, B)$$

CRS contains $H, G^{\gamma}, H^{\gamma}$ but not G

Cannot calculate C' from the CRS

Then so does $A', B' = BH^{r}, C'$???

Need second verification equation

Implies C' contains a factor of γ^2

Implies $A' = AG^{r}$

Implies r depends on γ

Our Techniques: 2

Suppose A, B, C satisfy

$$e(AG^{\alpha}, BH^{\beta}) = e(G^{\alpha}, H^{\beta})e(G^{f(\Phi)}, H^{\gamma})e(C, H)$$

$$e(A, H^{\gamma}) = e(G^{\gamma}, B)$$

CRS contains $H, G^{\gamma}, H^{\gamma}$ but not G

Cannot calculate C' from the CRS

Then so does $A', B' = BH^{r}, C'???$

Need second verification equation

Implies $A' = AG^{r}$

Implies r depends on γ

Implies C' contains a factor of γ^2

Our Techniques: 2

Suppose A, B, C satisfy

$$e\left(AG^{\alpha}, BH^{\beta}\right) = e\left(G^{\alpha}, H^{\beta}\right)e\left(G^{f(\Phi)}, H^{\gamma}\right)e(C, H)$$

$$e(A, H^{\gamma}) = e(G^{\gamma}, B)$$

CRS contains $H, G^{\gamma}, H^{\gamma}$ but not G

Cannot calculate C' from the CRS

Then so does $A', B' = BH^{r}, C'???$

Need second verification equation

Implies C' contains a factor of γ^2

Implies $A' = AG^{r}$

Implies r depends on γ

Our Techniques: 2

Suppose A, B, C satisfy

$$e(AG^{\alpha}, BH^{\beta}) = e(G^{\alpha}, H^{\beta})e(G^{f(\Phi)}, H^{\gamma})e(C, H)$$

$$e(A, H^{\gamma}) = e(G^{\gamma}, B)$$

CRS contains $H, G^{\gamma}, H^{\gamma}$ but not G

Cannot calculate C' from the CRS

The … es $A', B' =$ … $C'???$

Need second verification equation

Implies C' contains a factor of γ^2

Implies $A' = AG^r$

Implies r depends on γ

Plan

Definitions

Square Arithmetic Programs

Construction

Efficiency

Efficiency

	Groth	BCTV	This work
CRS size	$m + 2n + 3\ \mathbb{G}_1$ $n + 3\ \mathbb{G}_2$	$6m + n - \ell\ \mathbb{G}_1$ $m\ \mathbb{G}_2$	$m + 5n + 5\ \mathbb{G}_1$ $2n + 3\ \mathbb{G}_2$
Proof size	$2\ \mathbb{G}_1,\ 1\ \mathbb{G}_2$	$7\ \mathbb{G}_1,\ 1\ \mathbb{G}_2$	$2\ \mathbb{G}_1,\ 1\ \mathbb{G}_2$
Prover computation	$m + 3n - \ell + 3\ E_1$ $n + 1\ E_2$	$6m + n - \ell\ E_1$ $m\ E_2$	$m + 5n - \ell\ E_1$ $2n\ E_2$
Verifier computation	$\ell\ E_1,\ 3\ P$	$\ell\ E_1,\ 12\ P$	$\ell\ E_1,\ 5\ P$
Verification equations	1	5	2

- Public parameters and prover computation a bit higher than the others.
- Verifier computation is low
- Verifier equations are minimal for SE-SNARKs
- Proof size is minimal for SE-SNARKs

Proof in full version
eprint.iacr.org/2017/540

Implemented in libsnark by Popovs, Chiesa, and Virza
github.com/scipr-lab/libsnark/tree/master/libsnark/zk_proof_systems/ppzksnark/r1cs_se_ppzksnark

Thank-you for listening