Subversion Resistance of SNARKs

Georg Fuchsbauer

G. F.: **Subversion-zero-knowledge SNARKs** ia.cr/2017/587

Workshop on Blockchain Technology and Theory in connection with DISC 2017
Vienna, 16 October 2017

Zero-knowledge contingent payments

Seller: w

Buyer: BTC

Zero-knowledge contingent payments

Seller: w

Buyer: BTC

Pay-to-PubkeyHash:

- pay to y
- redeem by giving x s.t. $H(x) = y$

Zero-knowledge contingent payments

$Enc_k(w),\ y = H(k)$

Seller: w

pay to y

Buyer: BTC

Pay-to-PubkeyHash:

- pay to y
- redeem by giving x s.t. $H(x) = y$

Zero-knowledge contingent payments

$Enc_k(w),\ y = H(k)$

Seller: w

k

pay to y

Buyer: BTC

Pay-to-PubkeyHash:

- pay to y
- redeem by giving x s.t. $H(x) = y$

Zero-knowledge contingent payments

$Enc_k(w),\ y = H(k)$

proof π

Seller: w

Buyer: BTC

Pay-to-PubkeyHash:

- pay to y
- redeem by giving x s.t. $H(x) = y$

Non-interactive proofs

- let $L \in \mathcal{NP}$
- prove $x \in L$

π

✓ / ×

statement

witness

Prover: x, w

Verifier: x

Non-interactive proofs

- let $L \in \mathcal{NP}$
- prove $x \in L$

π

✓ / ×

Prover: x, w

Verifier: x

Non-interactive proofs

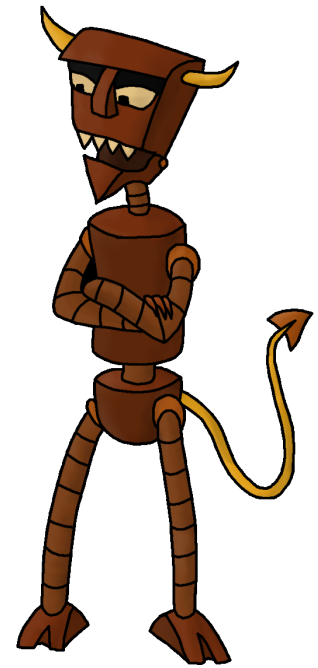
π

Zero-knowledge:

crs'

π'

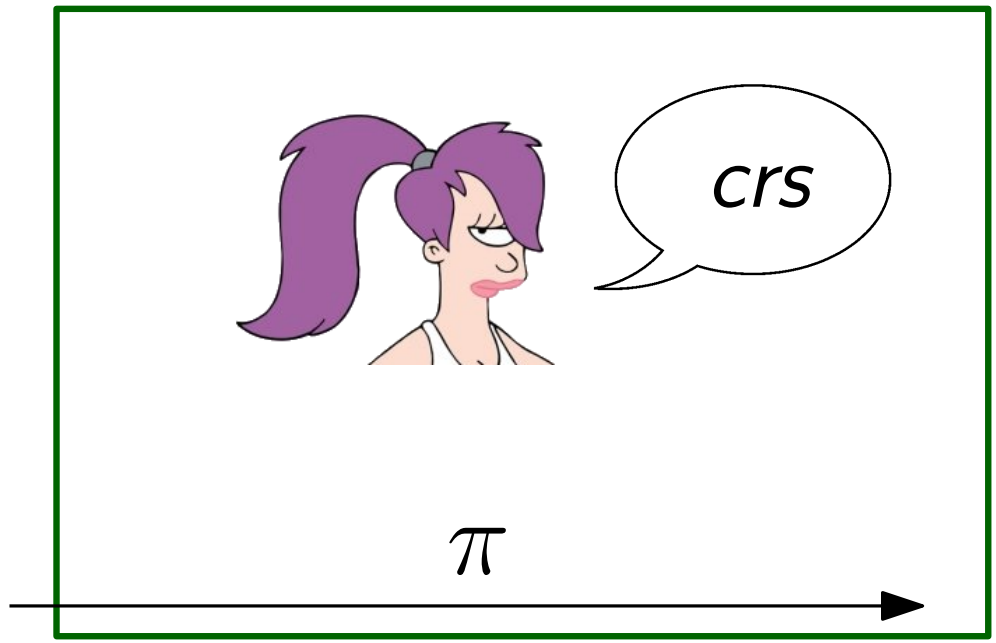
Simulator: $x, \not{w}$



Prover: x, w

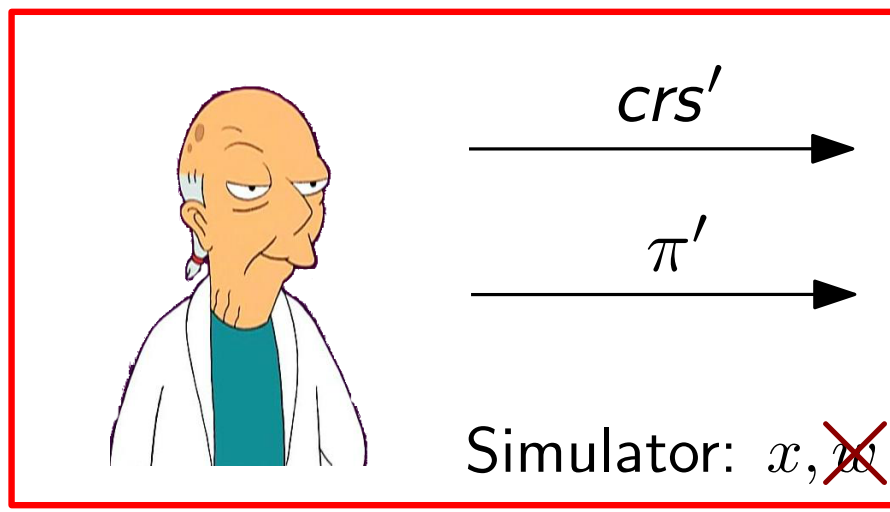
Verifier: x

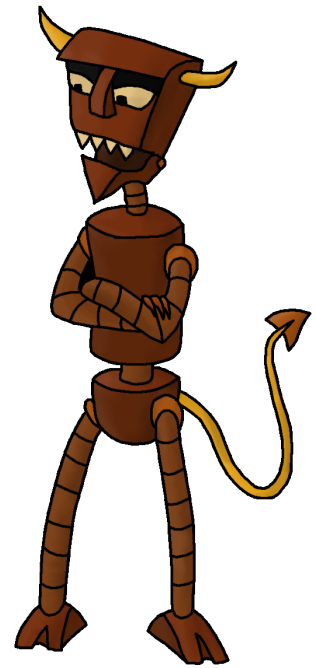
Non-interactive proofs



Prover: x, w

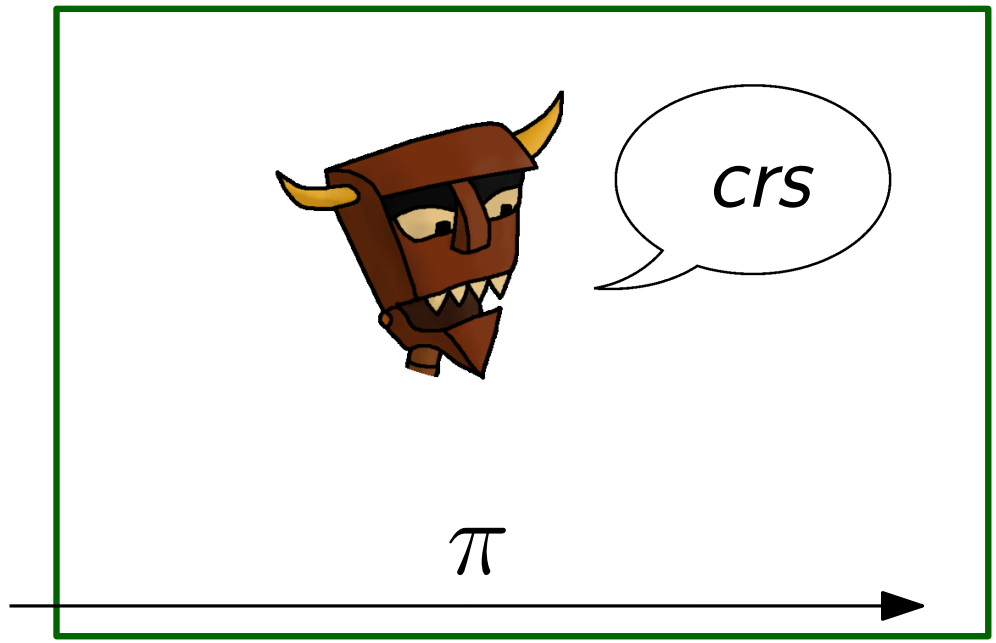
Zero-knowledge: $\approx$





Verifier: x

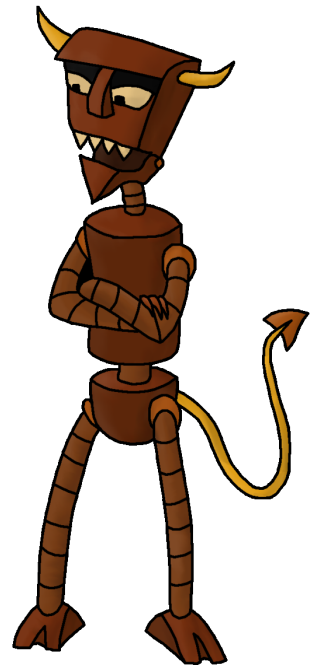
Subversion-zero-knowledge proofs [BFS16]



Prover: x, w

Subversion ZK: $\approx$

Simulator: $x, \not{w}$



Verifier: x

Subversion-zero-knowledge proofs [BFS16]

- Bellare, F, Scafuro (Asiacrypt'16):
  Subv.-ZK scheme
  using *knowledge* assumption

SNARKs

Succinct **N**on-interactive
ARgument of **K**nowledge

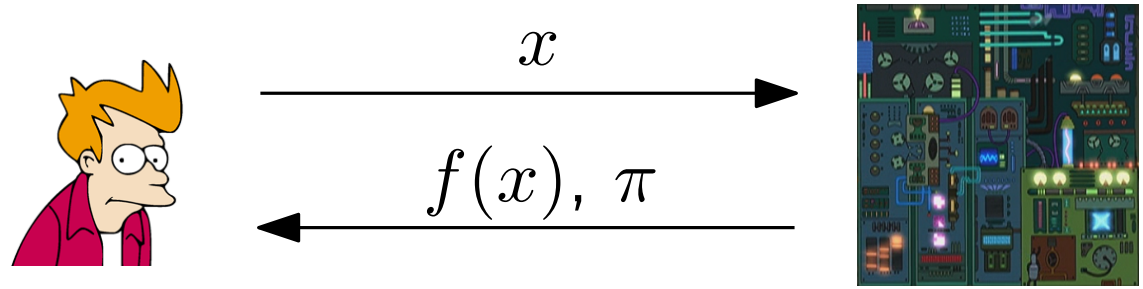
- succinct:

  $|\pi|$ independent of $|x|$ and $|w|$

- proves knowledge of w

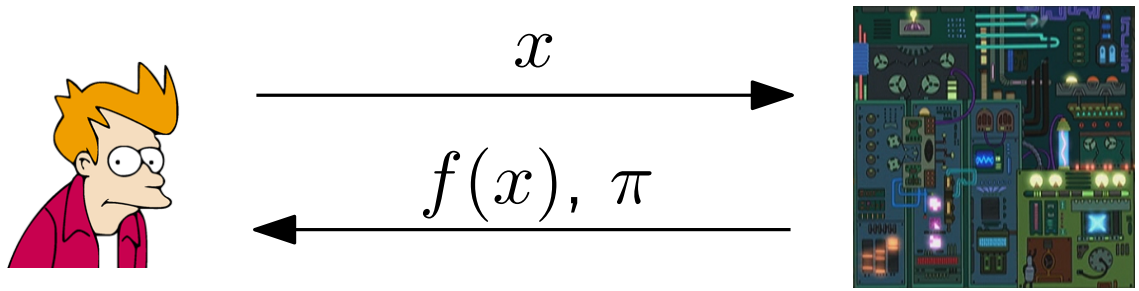
Applications of SNARKs

- Outsourcing of computation



Applications of SNARKs

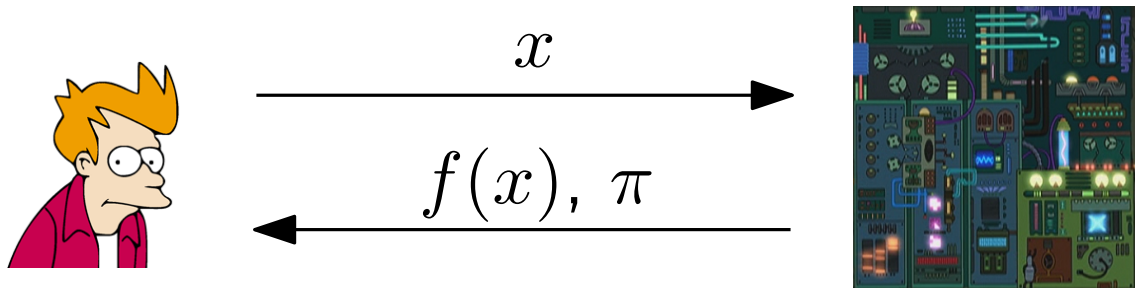
- Outsourcing of computation



- Cryptocurrencies: **Zerocash** [BCGGMTV'14]
  - fully **anonymous** payments
  - using **zero-knowledge** SNARKs
  - deployed as ZCASH

Applications of SNARKs

- Outsourcing of computation



- Cryptocurrencies: **Zerocash** [BCGGMTV'14]
  - fully **anonymous** payments
  - using **zero-knowledge** SNARKs
  - deployed as ZCASH
- ZK contingent payments [Maxwell'15]

Zero-knowledge contingent payments

$$Enc_k(w),\ y = H(k)$$

proof π

Seller: w

Buyer: BTC

Zero-knowledge contingent payments

$Enc_k(w),\ y = H(k)$

proof π

Seller: w

Buyer: BTC

Zero-knowledge contingent payments

crs

$Enc_k(w),\ y = H(k)$

proof π

Seller: w

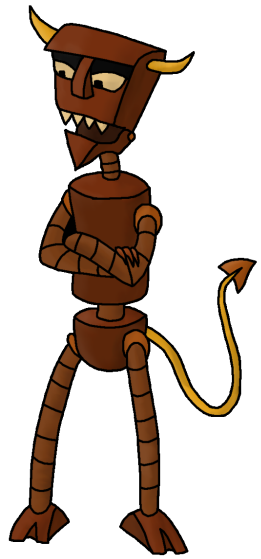
Buyer: BTC

Zero-knowledge contingent payments

crs

$Enc_k(w),\ y = H(k)$

proof π



Seller: w

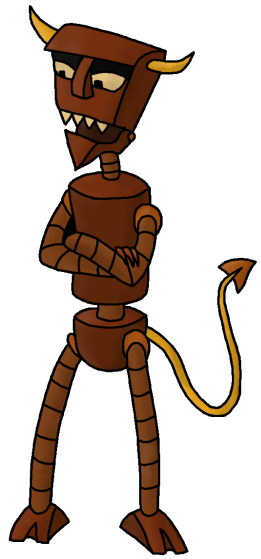
Buyer: BTC

Zero-knowledge contingent payments

crs

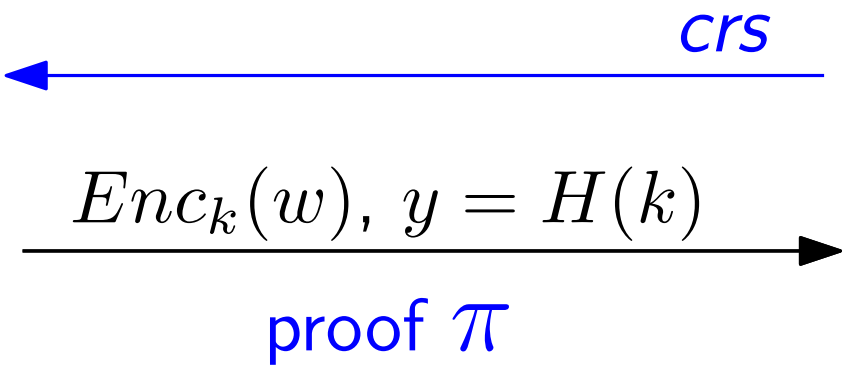
$Enc_k(w),\ y = H(k)$

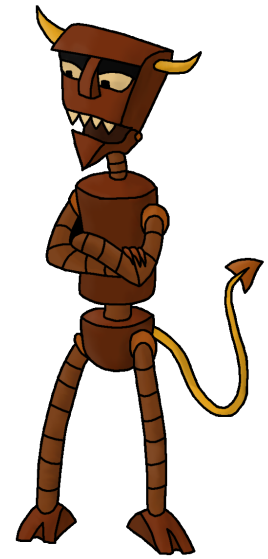
proof π



Subversion zero-knowledge?

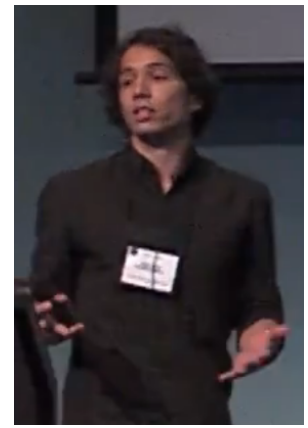
Zero-knowledge contingent payments



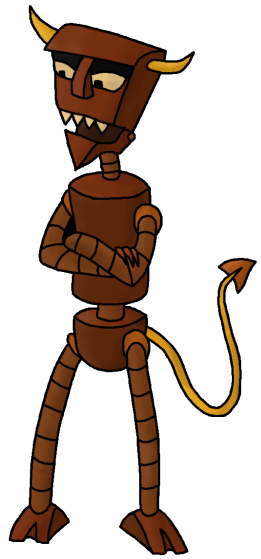


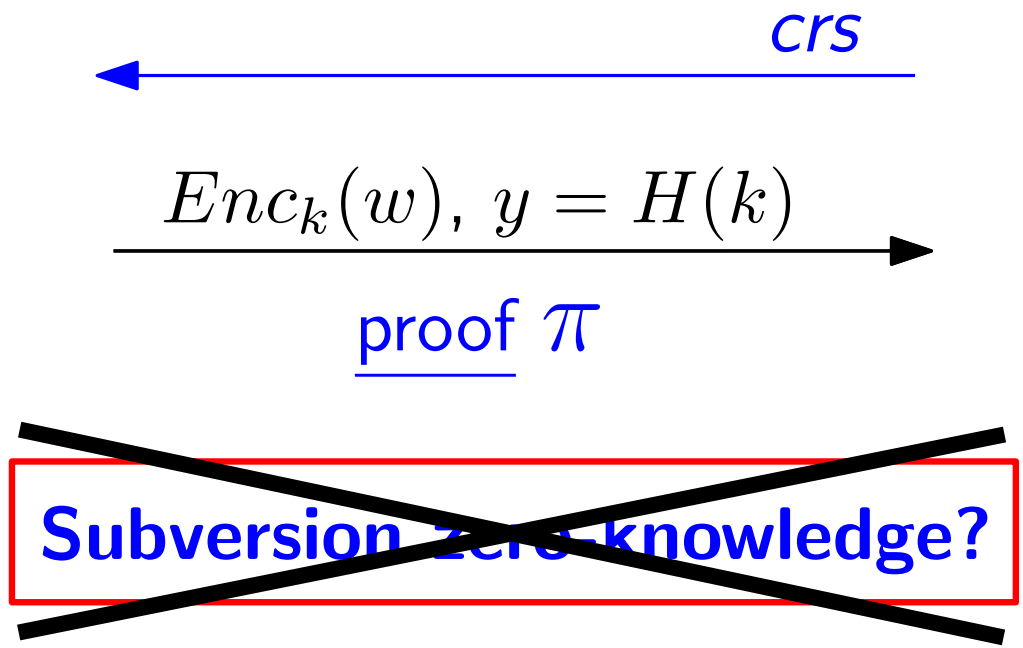
Subversion zero-knowledge?

- Campanelli et al. [CGGN17] ia.cr/2017/566
  show **attack**: – subvert CRS
  $\Rightarrow$ obtain information on w



Zero-knowledge contingent payments





- Campanelli et al. [CGGN17] ia.cr/2017/566
  show **attack**: – subvert CRS
  $\Rightarrow$ obtain information on w

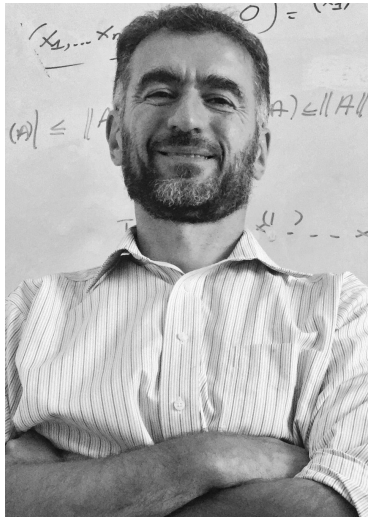
This work

Subversion zero knowledge of SNARKs

- Analysis of 5 important zk-SNARKs from literature
  - are they subversion-ZK?
  - if not, can they be made subversion-ZK?

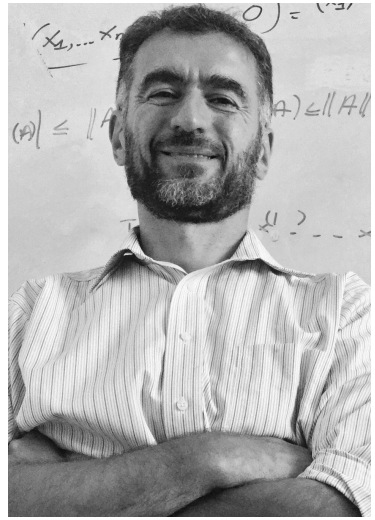
zk-SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]
  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)



zk-SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]
  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)

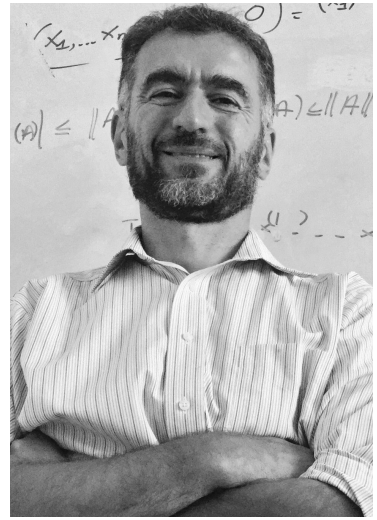


- Verfiy CRS well-formedness?

- Simulate proofs?

zk-SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]
  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)



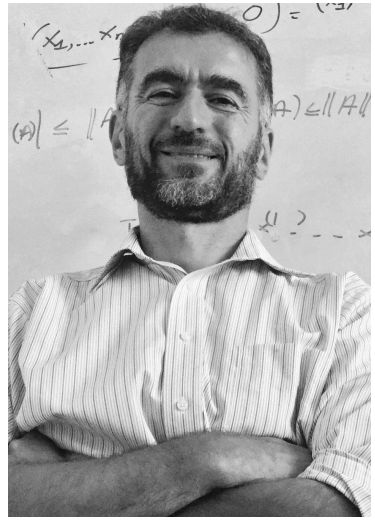
- Verfiy CRS well-formedness?

✓ using pairings

- Simulate proofs?

zk-SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]
  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)



- Verfiy CRS well-formedness?

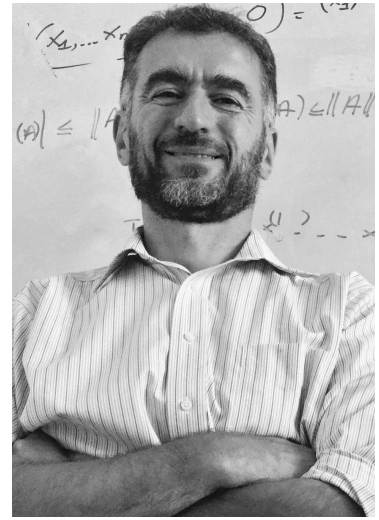
✓ using pairings

- Simulate proofs?

✓ under knowledge assumption

zk-SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]
  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)



- Verfiy CRS well-formedness?

✓ using pairings

- Simulate proofs? ⇒ **subversion zero knowledge**

✓ under knowledge assumption

zk-SNARK 3

- Optimized **Pinocchio** [PHGR13, BCTV14]

  underies – ZCASH

  – ZK contingent payments

zk-SNARK 3

- Optimized **Pinocchio** [PHGR13, BCTV14]

  underies – ZCASH

  – ZK contingent payments

- CRS verfiable? ✗

- Simulate proofs?

zk-SNARK 3

- Optimized **Pinocchio** [PHGR13, BCTV14]

  underies – ZCASH

  – ZK contingent payments

- CRS verfiable? ✗

  $\Rightarrow$ add 4 group elements ✓

- Simulate proofs?

zk-SNARK 3

- Optimized **Pinocchio** [PHGR13, BCTV14]

  underies – ZCASH

  – ZK contingent payments

- CRS verfiable? ✗

  $\Rightarrow$ add 4 group elements ✓

- Simulate proofs? ✓

$\Rightarrow$ **subversion zero knowledge** of modified scheme

zk-SNARK 4 & 5

- Danezis et al.'s SNARKs [DFGK14]

zk-SNARK 4 & 5

- Danezis et al.'s SNARKs [DFGK14]

  subversion zero knowledge (as is)

zk-SNARK 4 & 5

- Danezis et al.'s SNARKs [DFGK14]

  subversion zero knowledge (as is)

- Groth's SNARKs [Groth16]
  - most efficient scheme

zk-SNARK 4 & 5

- Danezis et al.'s SNARKs [DFGK14]

  subversion zero knowledge (as is)

- Groth's SNARKs [Groth16]
  - most efficient scheme

  subversion zero knowledge (as is)

zk-SNARK 4 & 5

- Danezis et al.'s SNARKs [DFGK14]

  subversion zero knowledge (as is)

- Groth's SNARKs [Groth16]
  - most efficient scheme

  subversion zero knowledge (as is)

Concurrently, [ABLZ17] show S-ZK of modified scheme under stronger assumption ia.cr/2017/599

Zcash

ZCASH

Is Zcash anonymous if parameters set up maliciously?

- uses SNARK **without** verifiable CRS
- parameters set up using MPC [BCGTV15]
  - uses ROM proofs to prove correctness

Zcash

Is Zcash anonymous if parameters set up maliciously?

- uses SNARK **without** verifiable CRS
- parameters set up using MPC [BCGTV15]
  - uses ROM proofs to prove correctness

$\Rightarrow$ CRS **verifiable** ✓  $\Rightarrow$ proofs **simulatable** ✓ 

Zcash is subversion-anonymous in the ROM

Subsequently also argued by [BGG17] ia.cr/2017/602

THANK YOU!

QUESTIONS?