Subversion-Resistant Zero Knowledge

Georg Fuchsbauer

joint work with **Mihir Bellare**

and **Alessandra Scafuro**

ECRYPT-NET Workshop on Crypto for the Cloud & Implementation
27 June 2017

Content of this talk

- M. Bellare, G. F., A. Scafuro:
  NIZKs with an Untrusted CRS:
  Security in the Face of Parameter Subversion
  ASIACRYPT '16 (eprint 2016/372)

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

Motivation

- 2013

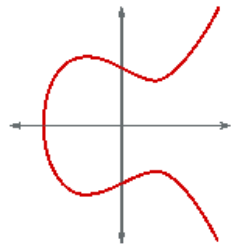
- compromised security not covered by standard model
- here: **parameter subversion**

Motivation

- 2013

- compromised security not covered by standard model
- here: **parameter subversion**

- Example: *Dual EC* RNG



  - "trusted" parameters P, Q
  - ISO standard; NSA paid RSA \$10 million
  - knowledge of $\log_Q P \Rightarrow$ predictable [ShuFer07]
    $\Rightarrow$ break TLS [CFN+14]

Motivation

- 2013

- compromised security not covered by standard model
- here: **parameter subversion**

- goal: **subversion resistance**
- this work: NIZK, relies on common reference string (CRS)
- example: zk-SNARK parameters
  for Zerocash (ZCASH) [BCG+14]

Related work

NIZK

- 2-move ZK protocols [BLV03, Pass03, BP04, BCPR14]
- NIZK in bare PK model [Wee07]
- CRS via multiparty computation [KKZZ14, BSCG$^+$15]
- UC w/ adv. CRS [CPs07], multiple CRSs [GO07, GGJS11]

Related work

NIZK

- 2-move ZK protocols [BLV03, Pass03, BP04, BCPR14]
- NIZK in bare PK model [Wee07]
- CRS via multiparty computation [KKZZ14, BSCG$^+$15]
- UC w/ adv. CRS [CPs07], multiple CRSs [GO07, GGJS11]

Subversion

- Algorithm-substitution attacks [BPR14, AMV15]
- Kleptography [YY96, YY97], cliptography [RTYZ16]
- Backdoored blockciphers [RP97, PG97, Pat99]

Non-interactive proofs

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

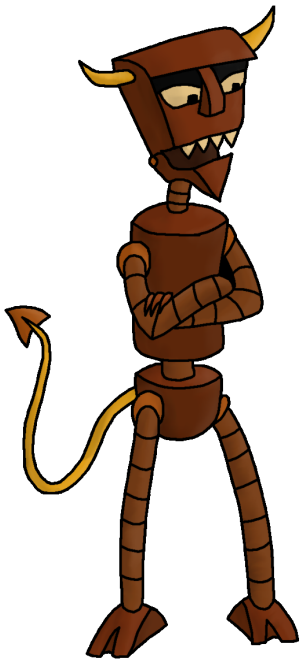
π

✓ / ×

Prover: x, w

Verifier: x

Non-interactive proofs



π

Soundness:

$\pi \checkmark \Rightarrow x \in L$

Prover: x, w

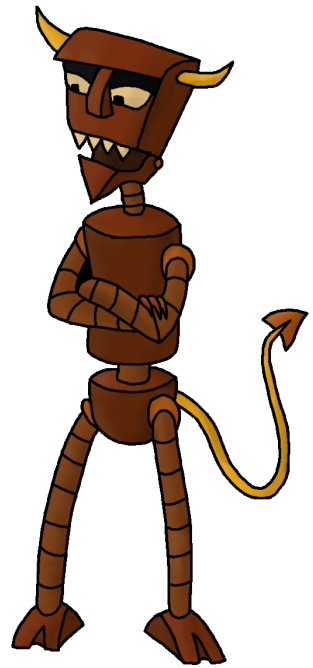
Verifier: x

Non-interactive proofs

π

Witness-indistinguishability:

$$\pi[w] \approx \pi[w']$$

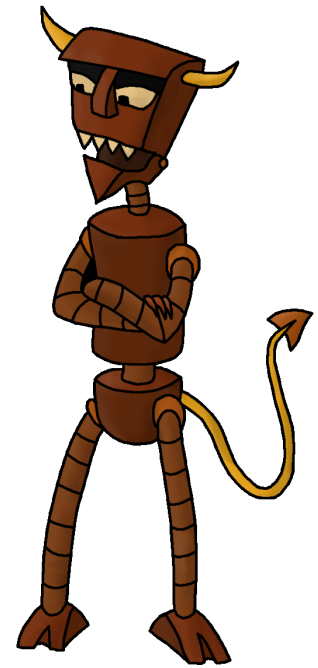


Prover: x, w

Verifier: x

Non-interactive proofs

π



Zero-knowledge:

crs'

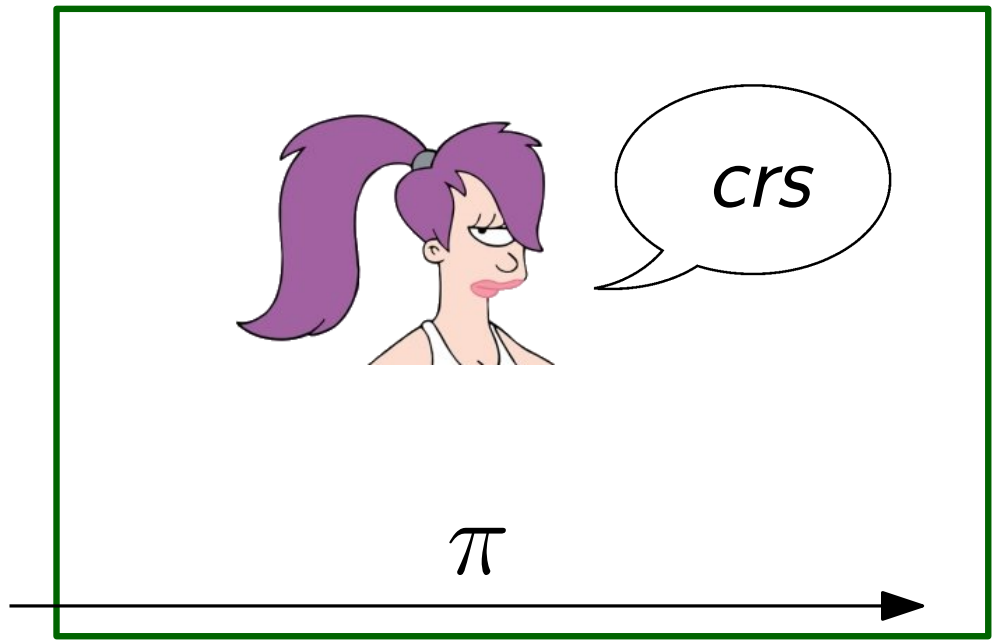
π'

Simulator: $x, \not{w}$

Prover: x, w

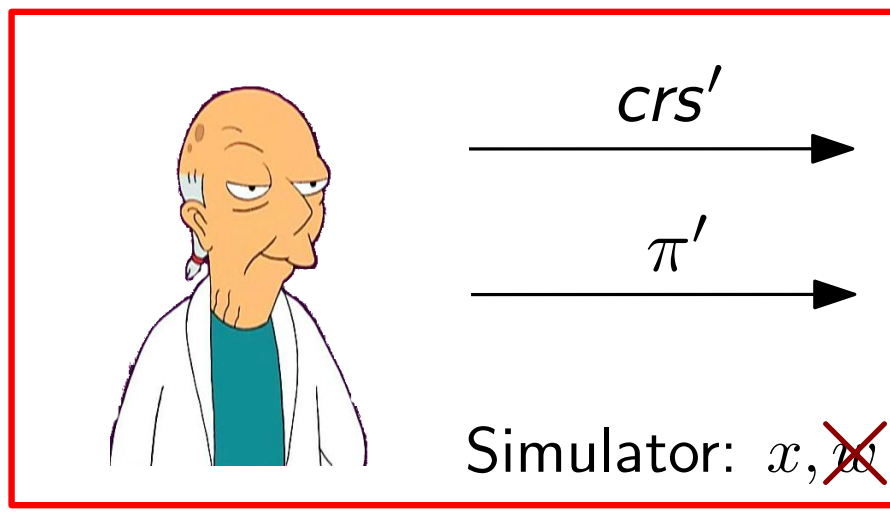
Verifier: x

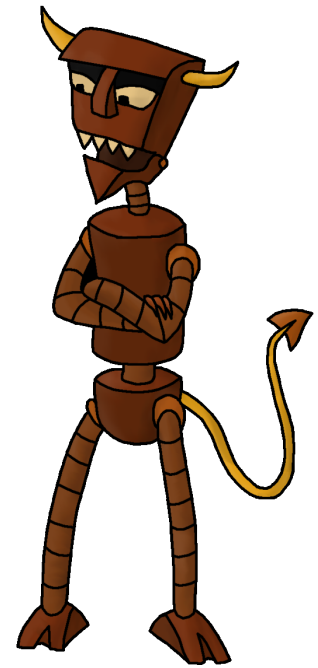
Non-interactive proofs



Prover: x, w

Zero-knowledge: $\approx$





Verifier: x

Subversion-resistant NI proofs

crs

π

Subversion Soundness:

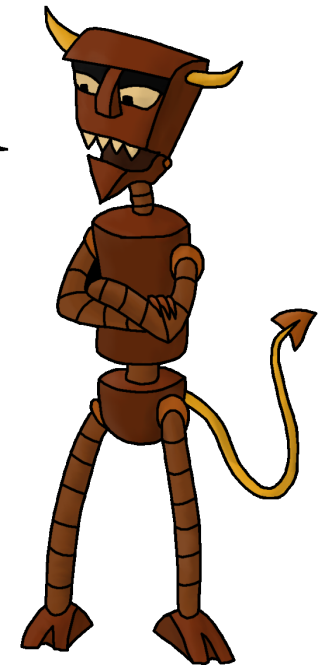
$\pi \checkmark \Rightarrow x \in L$

Prover: x, w

Verifier: x

Subversion-resistant NI proofs

crs



π

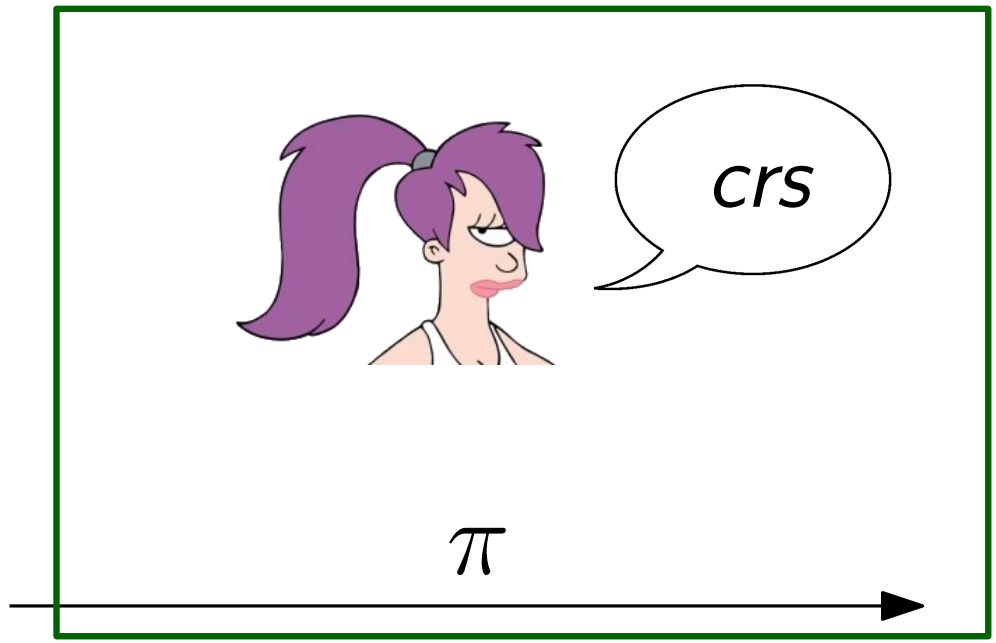
Subversion WI:

$$\pi[w] \approx \pi[w']$$

Prover: x, w

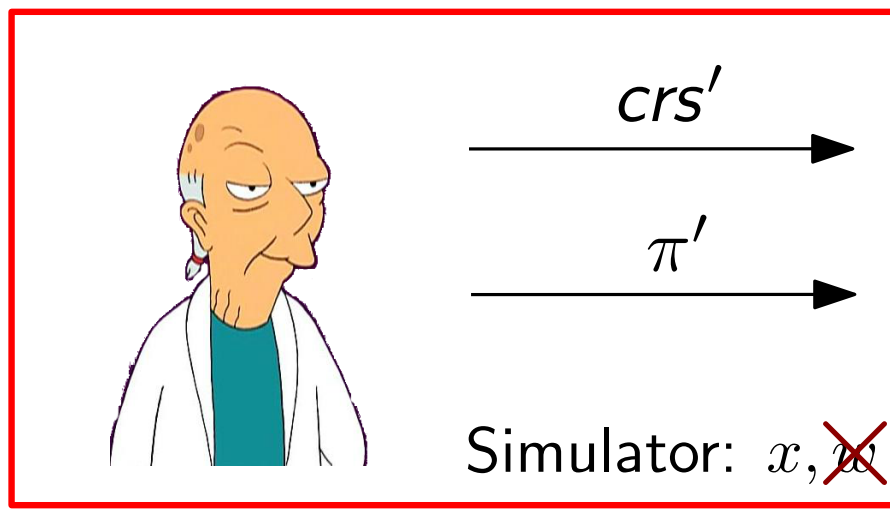
Verifier: x

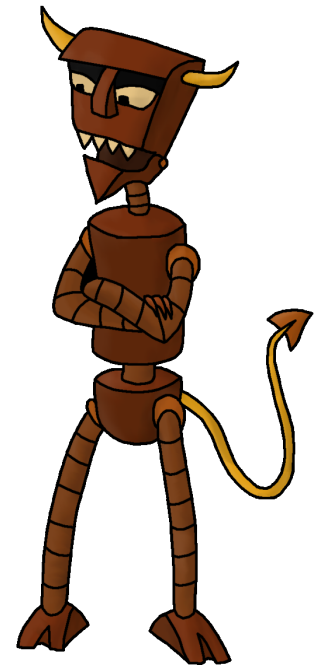
Non-interactive proofs



Prover: x, w

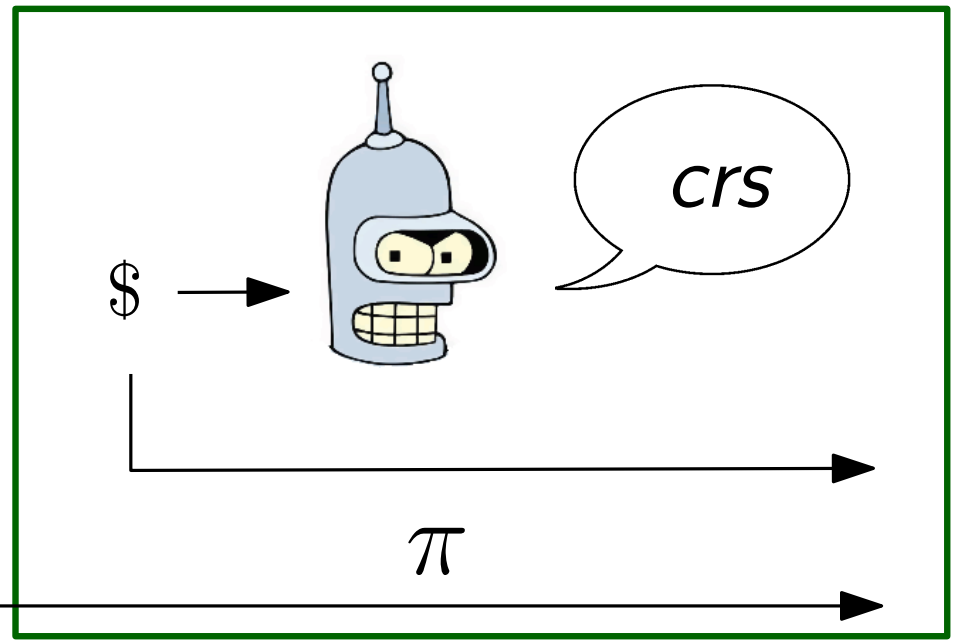
Zero-knowledge: $\approx$





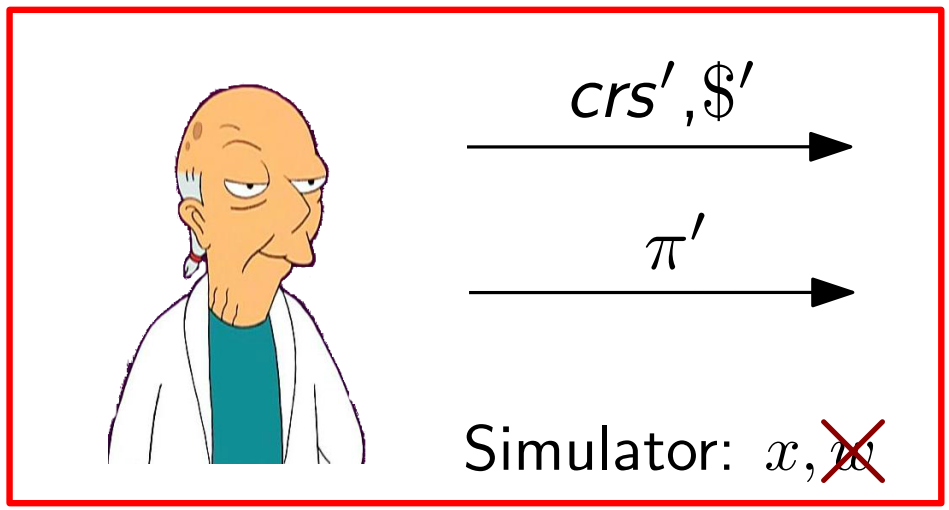
Verifier: x

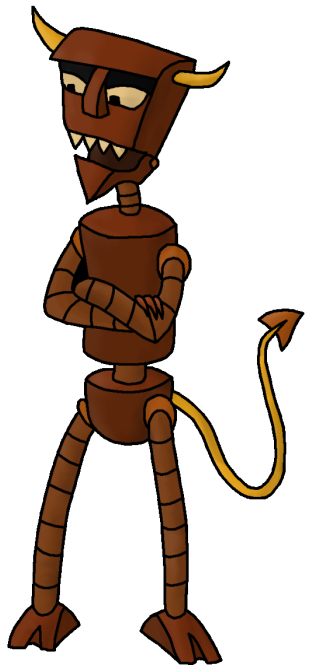
Subversion-resistant NI proofs



Prover: x, w

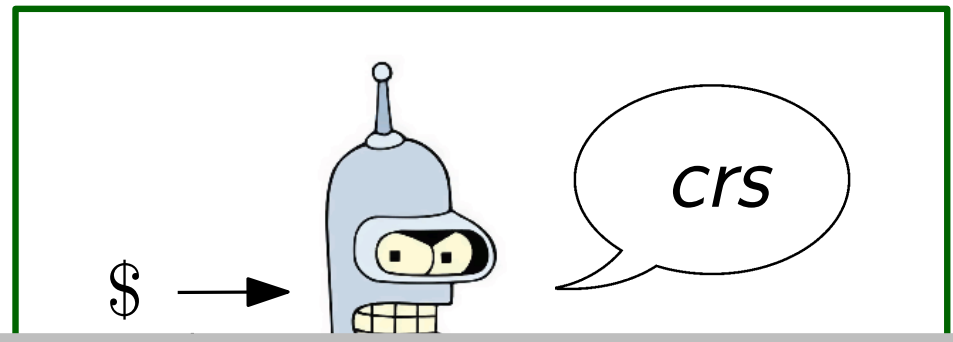
Subversion ZK: $\approx$

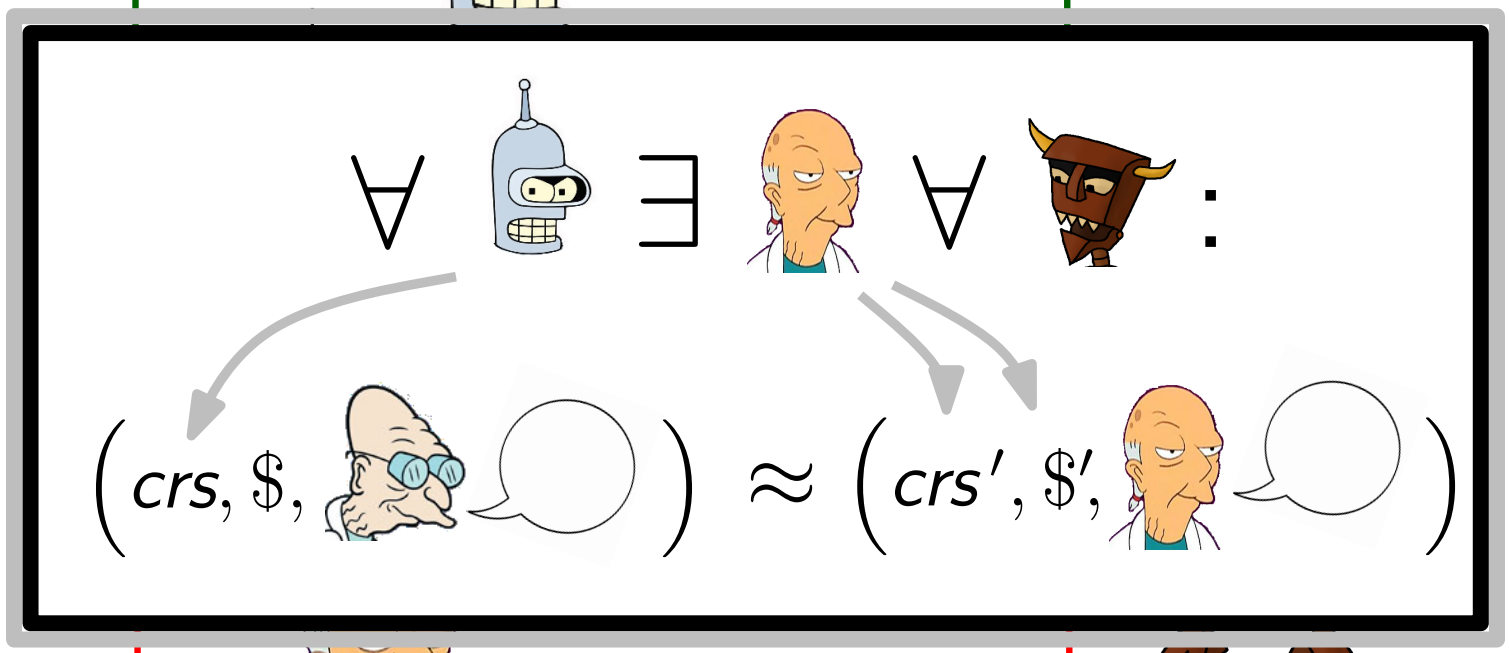


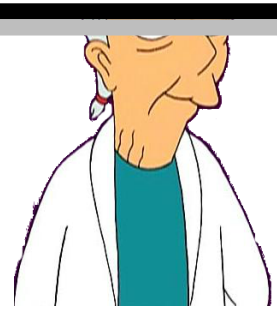


Verifier: x

Subversion-resistant NI proofs







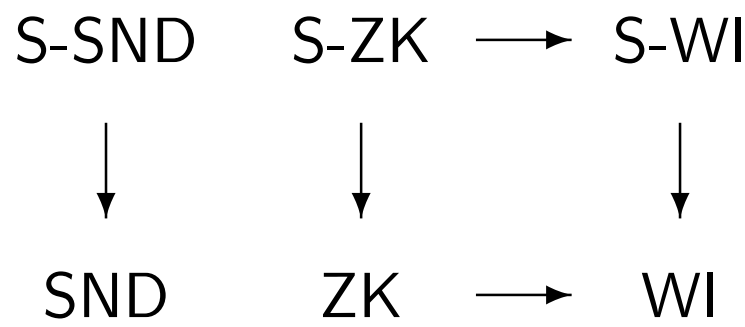
π'

Prover: x, w

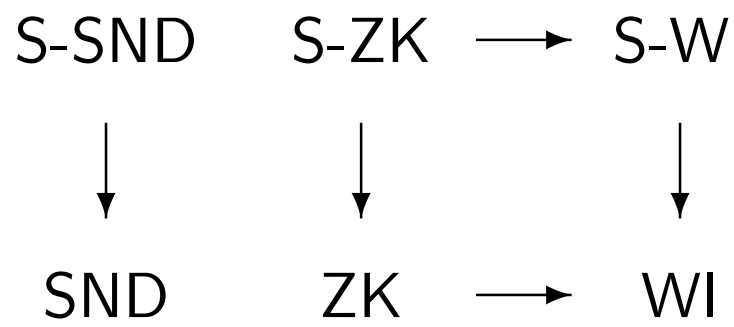
Simulator: x, ~~w~~

Verifier: x

Our results



Our results



Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		

Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
			•			✓	—

w

w witness for x?

Prover: x, w

Verifier: x

Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
				•		✓	—

$\mathcal{E}$

✓

Prover: x, w

Verifier: x

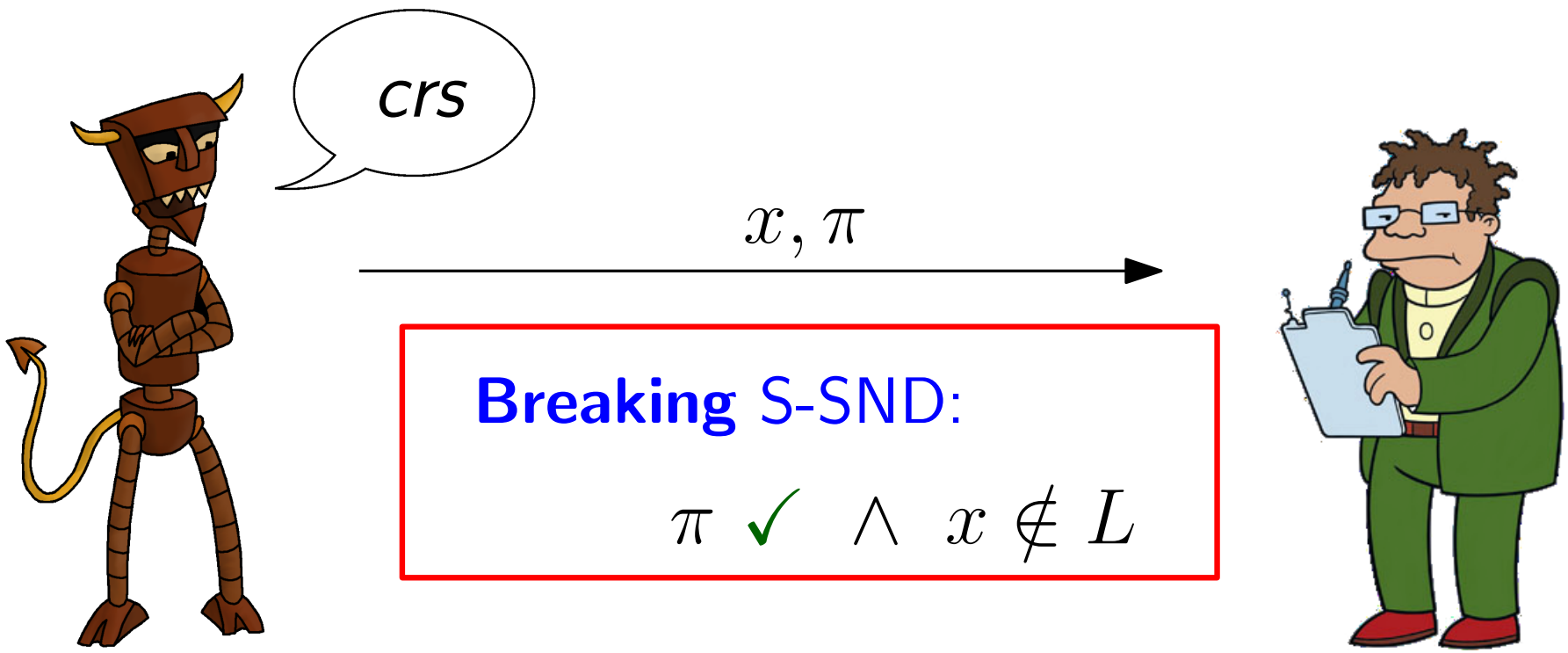
Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
•	•	•	**?**	**?**	**?**		

Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			×	

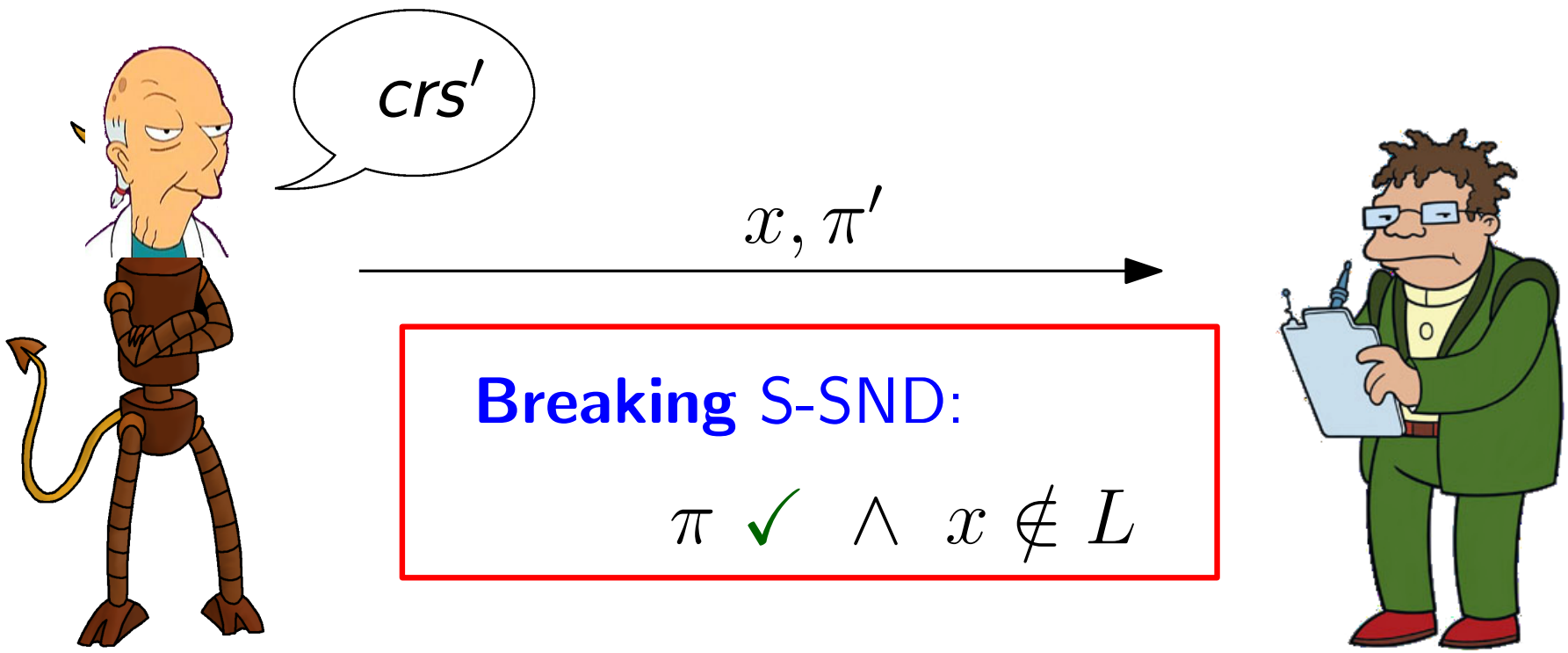
(if L is non-trivial)



Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			✗	

(if L is non-trivial)



Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	●		●			✗	
●		●	●		●	?	

Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			✗	
•		•	•		•	?	

Non-interactive Zaps [GOS06]

- NI WI proofs
- without CRS

No CRS $\Rightarrow$ subversion-resistant

Our results

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			✗	
•		•	•		•	✓	DLin
•	•	•		•	•	?	

Our results

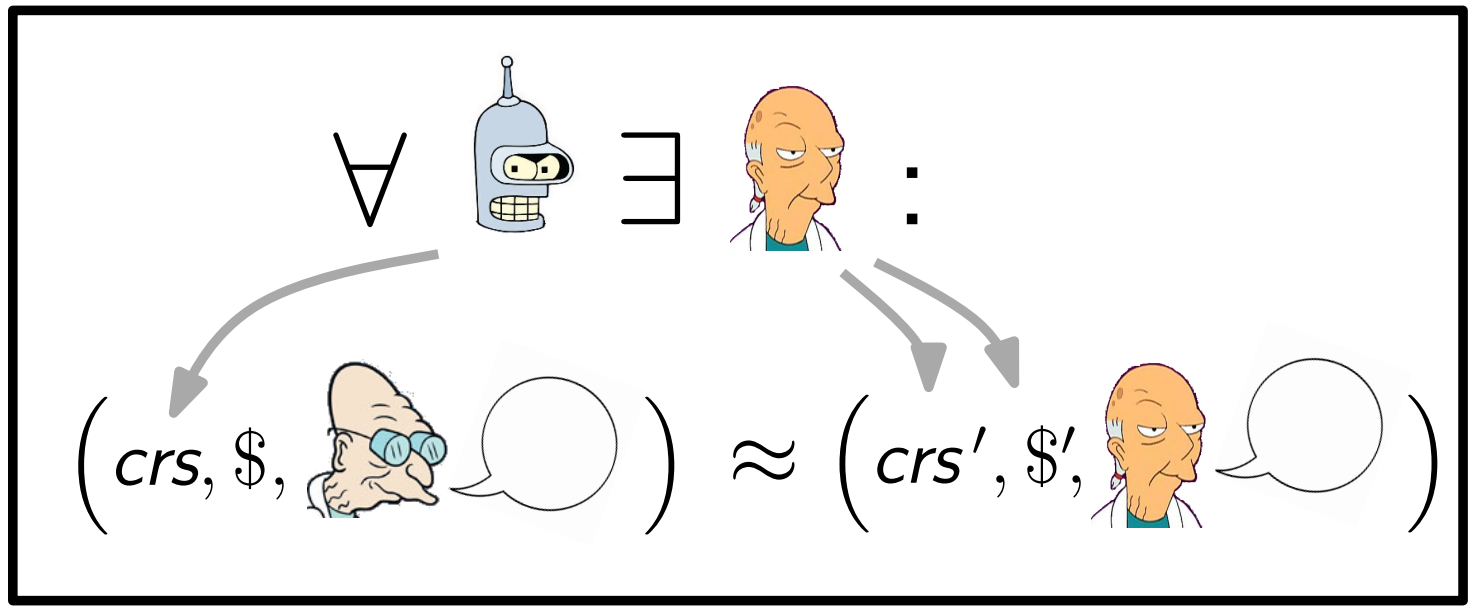
Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			✗	
•		•	•		•	✓	DLin
•	•	•		•	•	?	

- implies 2-move ZK (verifier chooses CRS)

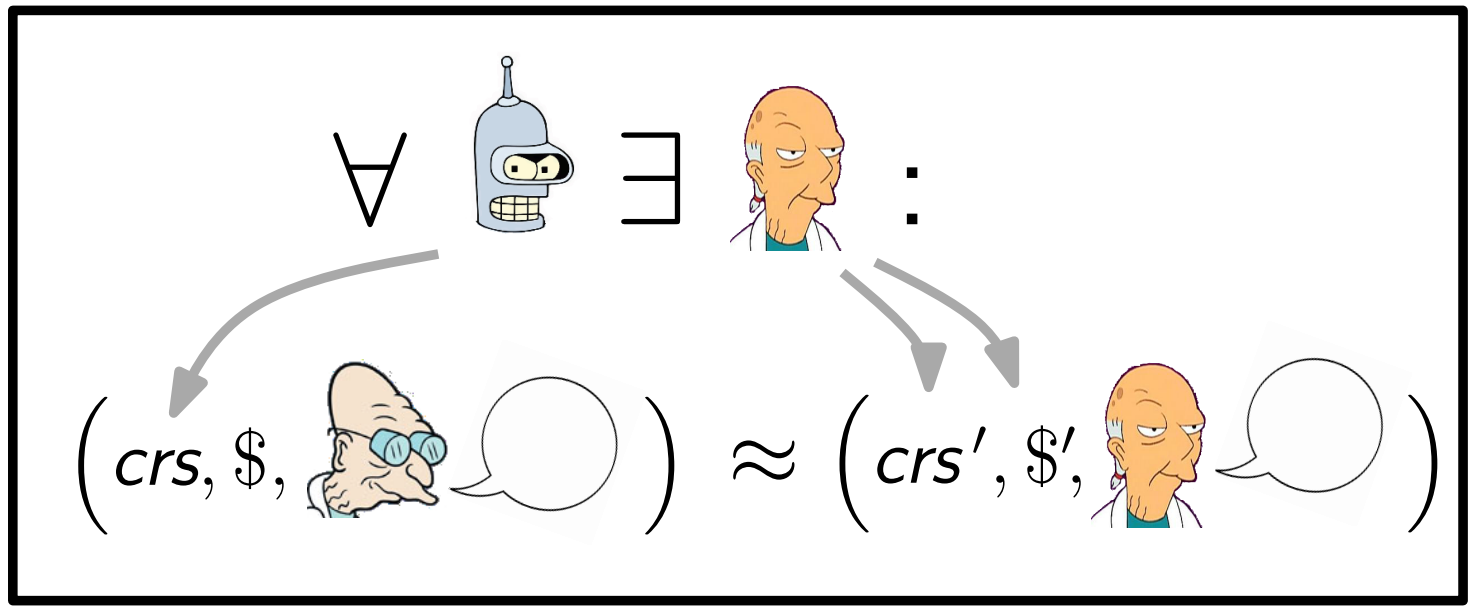
  $\Rightarrow$ only achieved under extractability assumpt's [BCPR14]

- construction under new *knowledge of exponent* assumption

Achieving SND + S-ZK

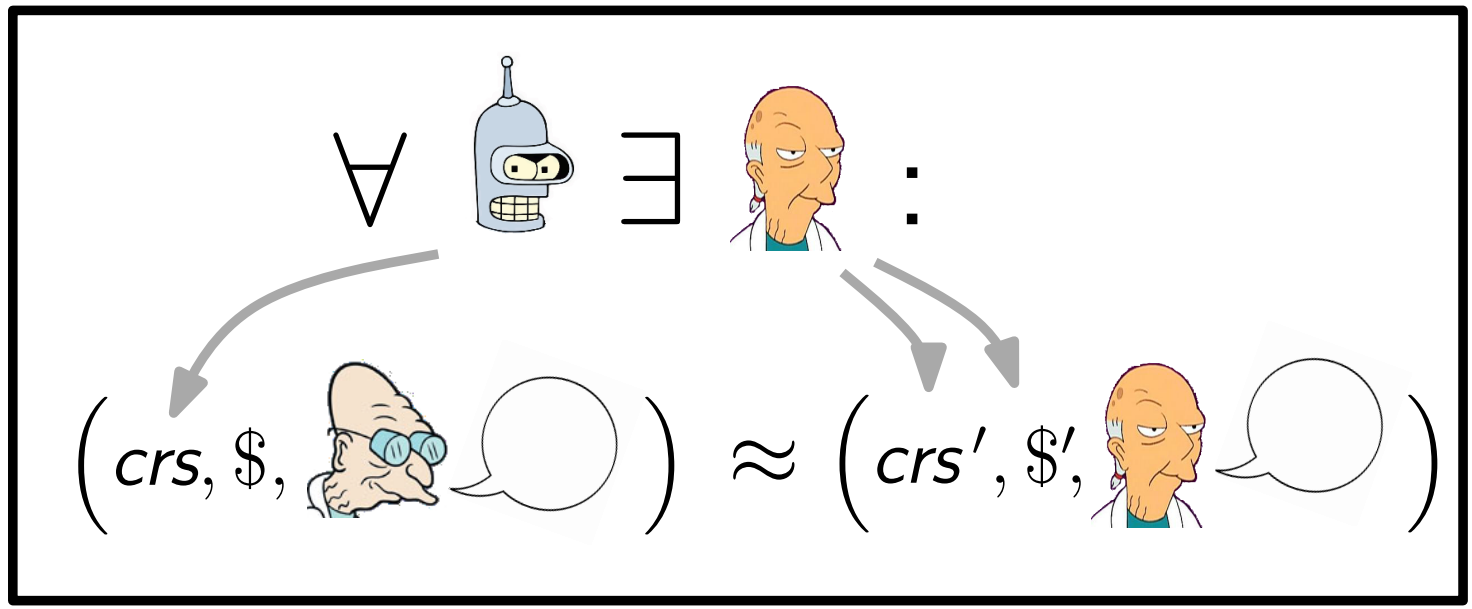


Achieving SND + S-ZK



KEA: $\forall$ $(g, h) \rightarrow$ $\rightarrow (g^s, h^s)$

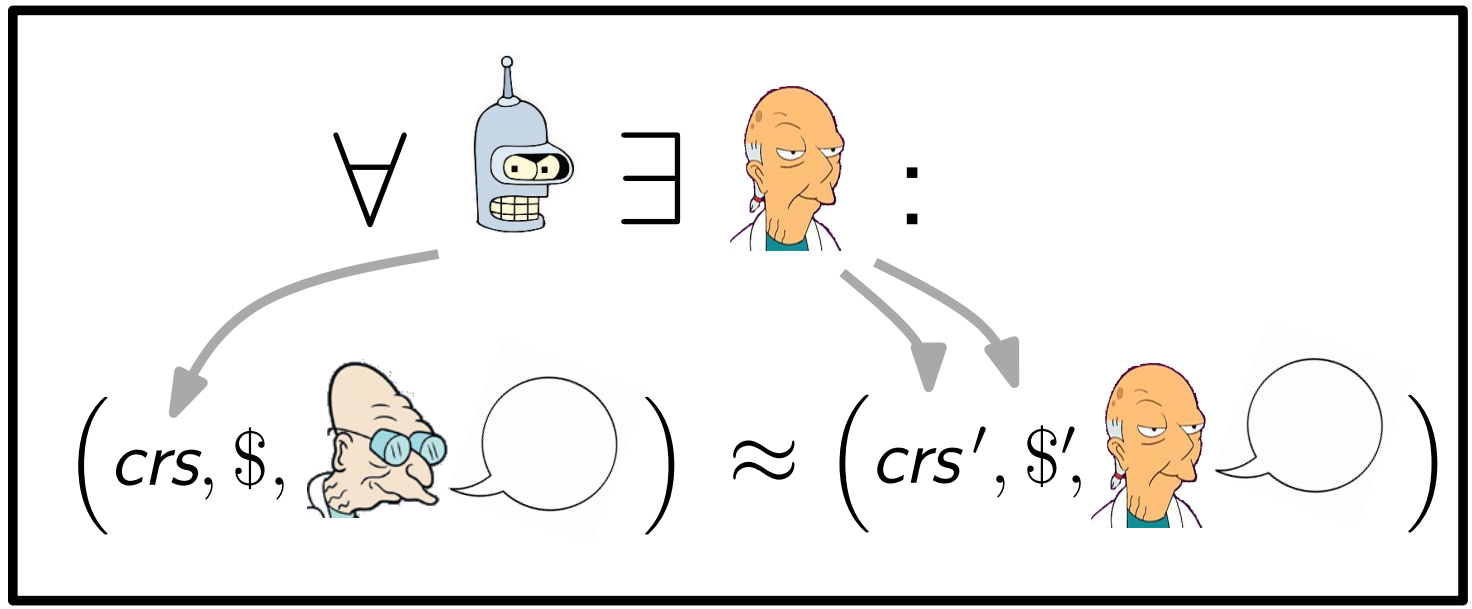
Achieving SND + S-ZK



KEA: $\forall$ $(g, h) \rightarrow$ $\rightarrow (g^s, h^s)$

$\exists$ $\rightarrow$ $\rightarrow s$

Achieving SND + S-ZK

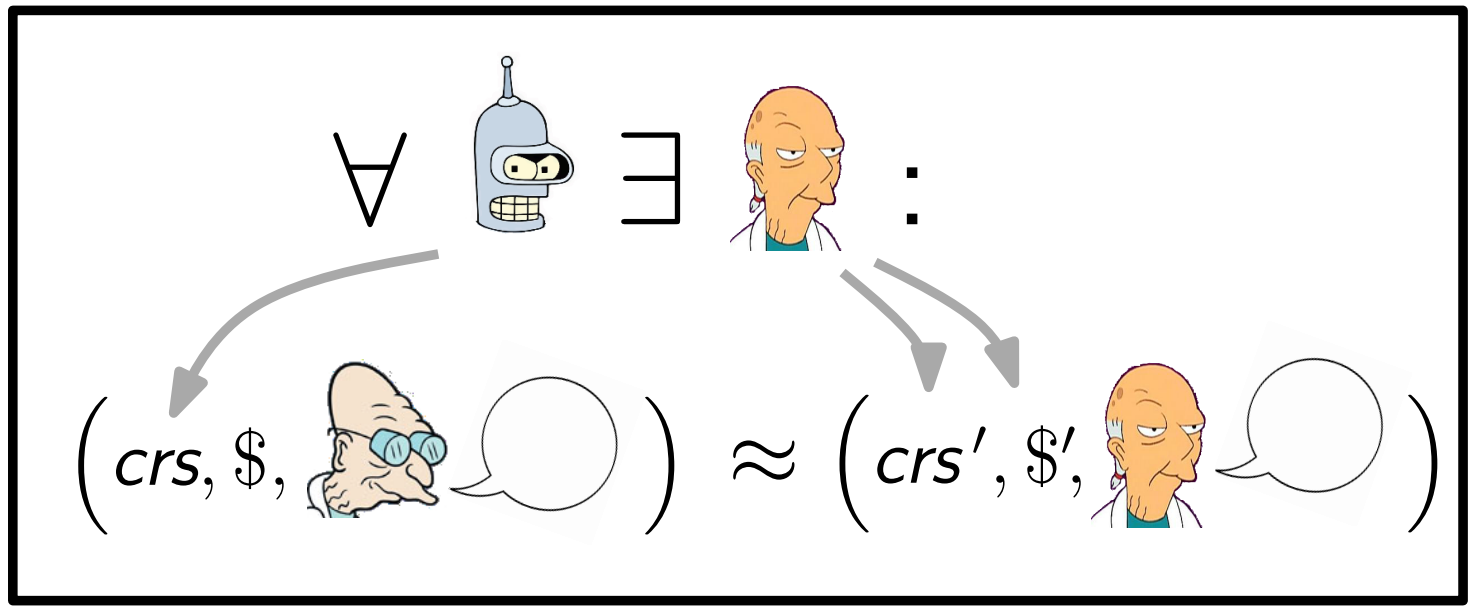


KEA: $\forall$ $(g, h) \rightarrow$ $\rightarrow (g^s, h^s)$

$\exists$ $\rightarrow$ $\rightarrow s$

checkable via pairing:
$\mathbf{e}(g^s, h) = \mathbf{e}(g, h^s)$

Achieving SND + S-ZK



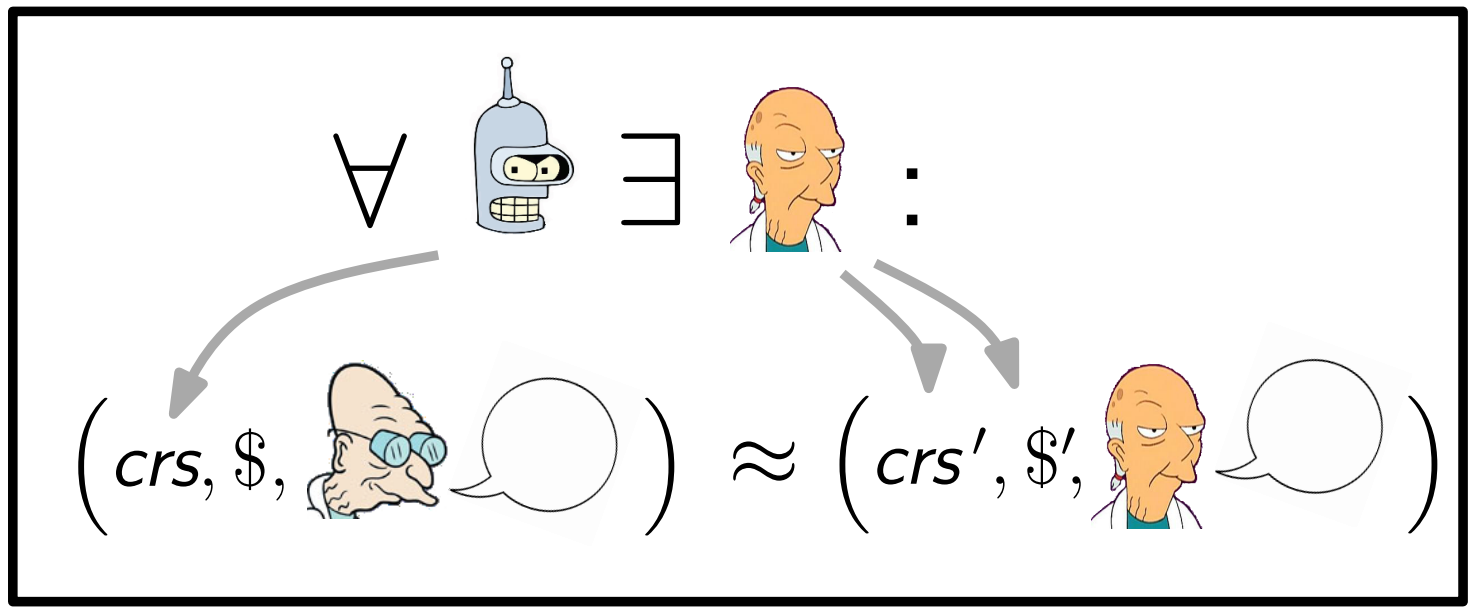
KEA: $\forall$ $(g, h) \rightarrow$ $\rightarrow (g^s, h^s)$

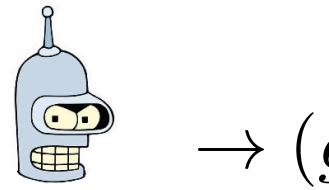
$\exists$ $\rightarrow s$

idea:

- *crs*
- trapdoor

Achieving SND + S-ZK





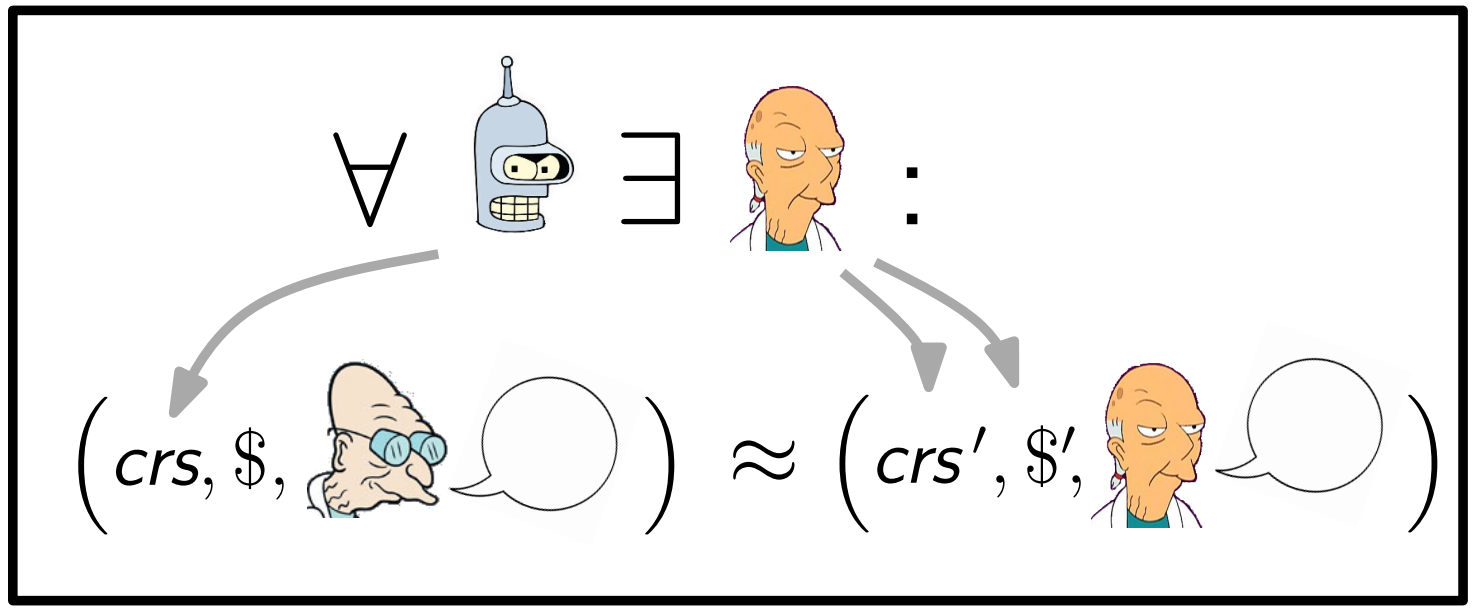
idea:

crs

trapdoor

Prove: $x \in L \vee$ "I know s"

Achieving SND + S-ZK



KEA: $\forall$ $(g, h) \rightarrow$ $\rightarrow (g^s, h^s)$

$\exists$ $\rightarrow s$

who chooses h?

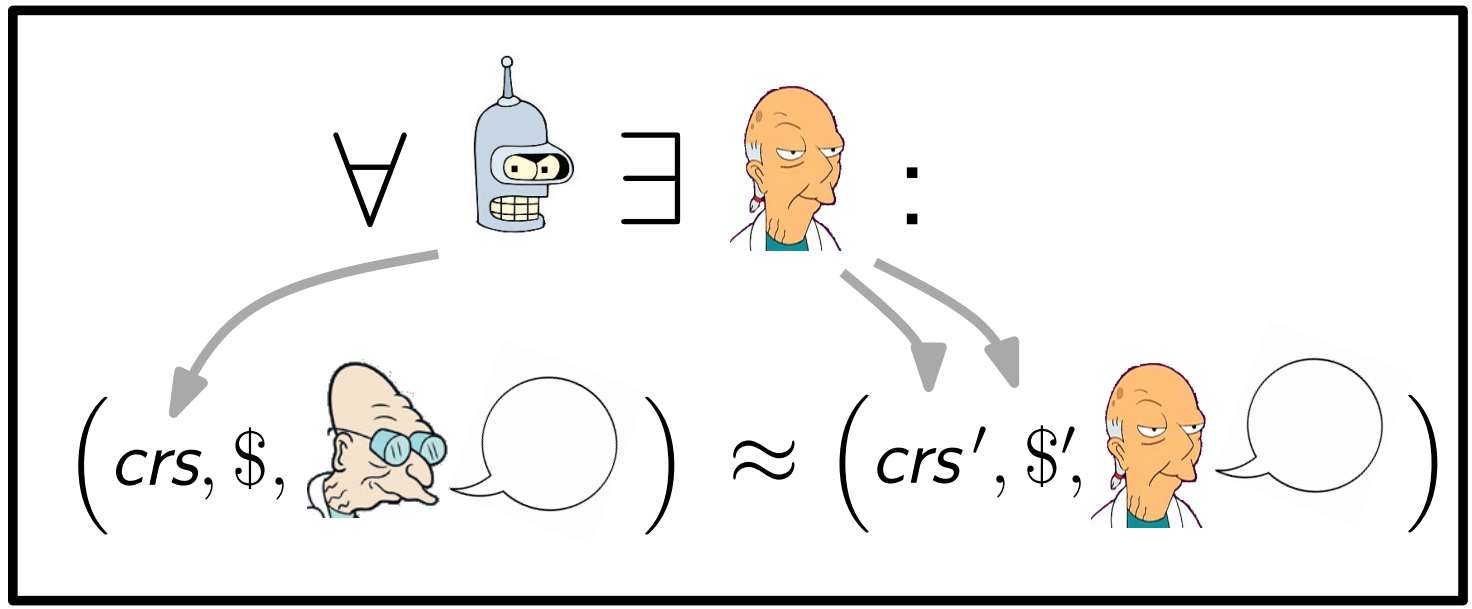
idea:

crs

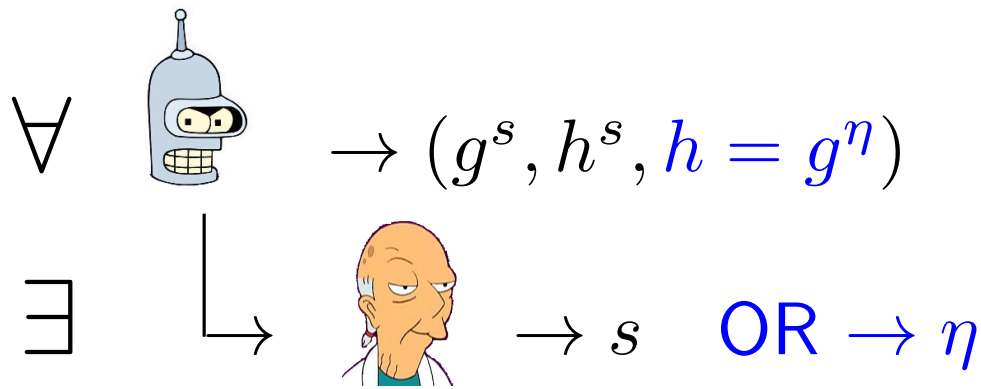
trapdoor

Prove: $x \in L \vee$ "I know s"

Achieving SND + S-ZK

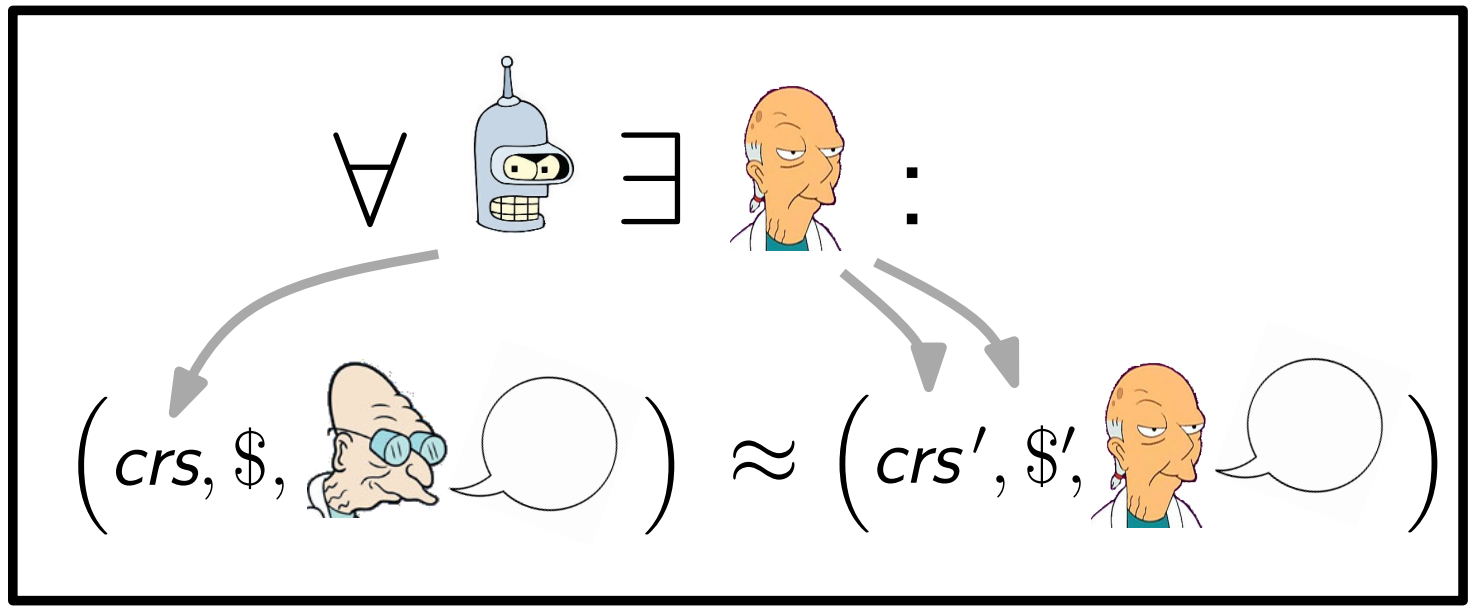


DH-KEA:



Prove: $x \in L \vee$ "I know s or η"

Achieving SND + S-ZK

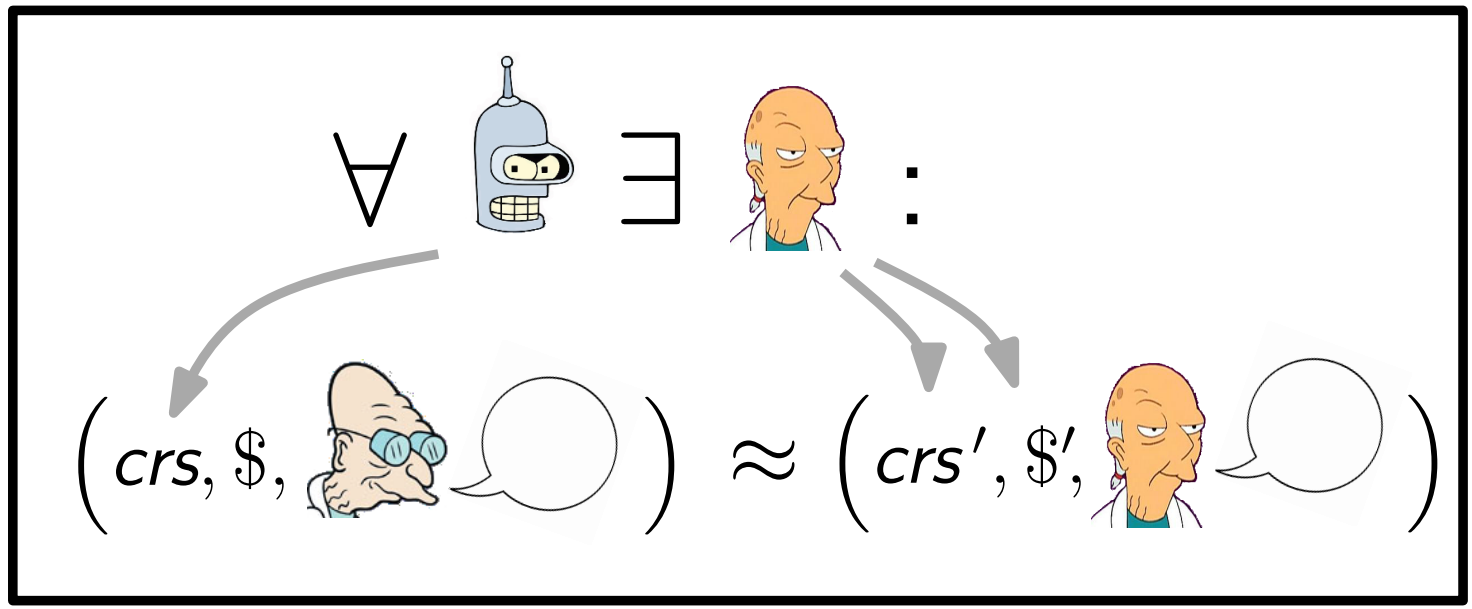


$$crs = (g^s, h^s, h = g^\eta)$$

prove knowledge how?

Prove: $x \in L \vee$ "I know s or η"

Achieving SND + S-ZK



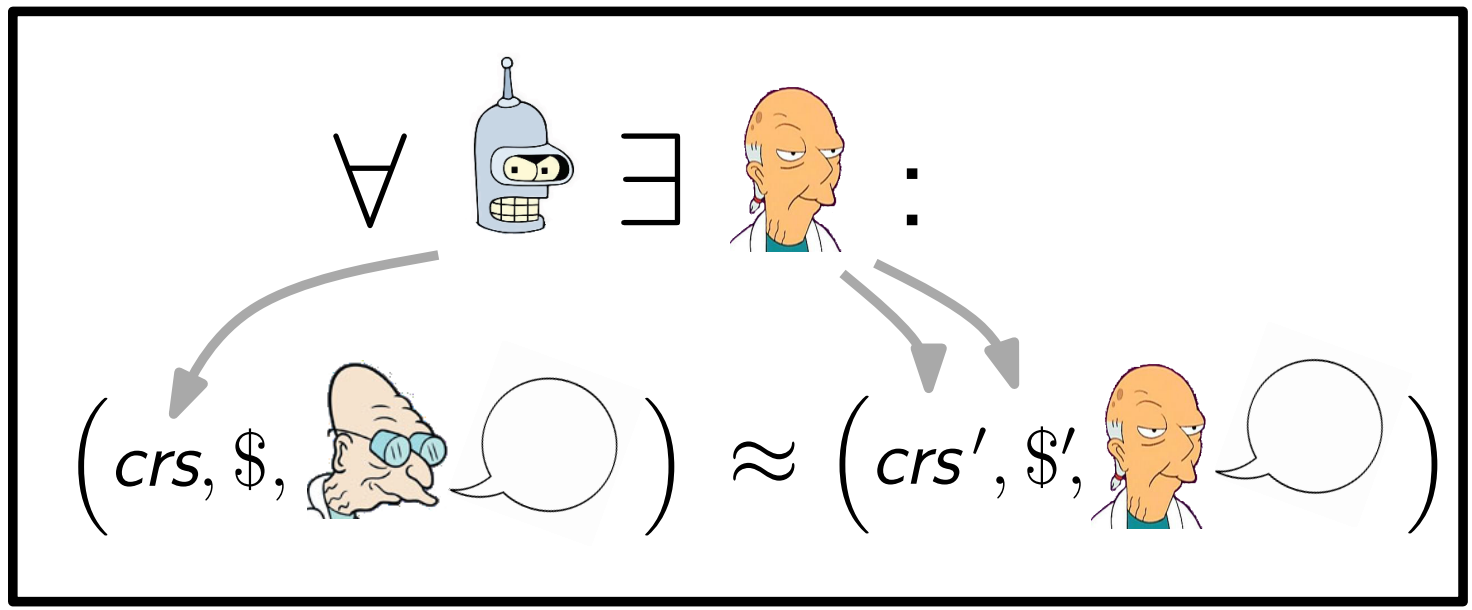
$$crs = (g^s, h^s, h = g^\eta)$$

$\mathsf{Enc}(pk, s)$

prove knowledge how?

Prove: $x \in L \vee$ "I know s or η"

Achieving SND + S-ZK



?

$crs = (g^s, h^s, h = g^\eta)$

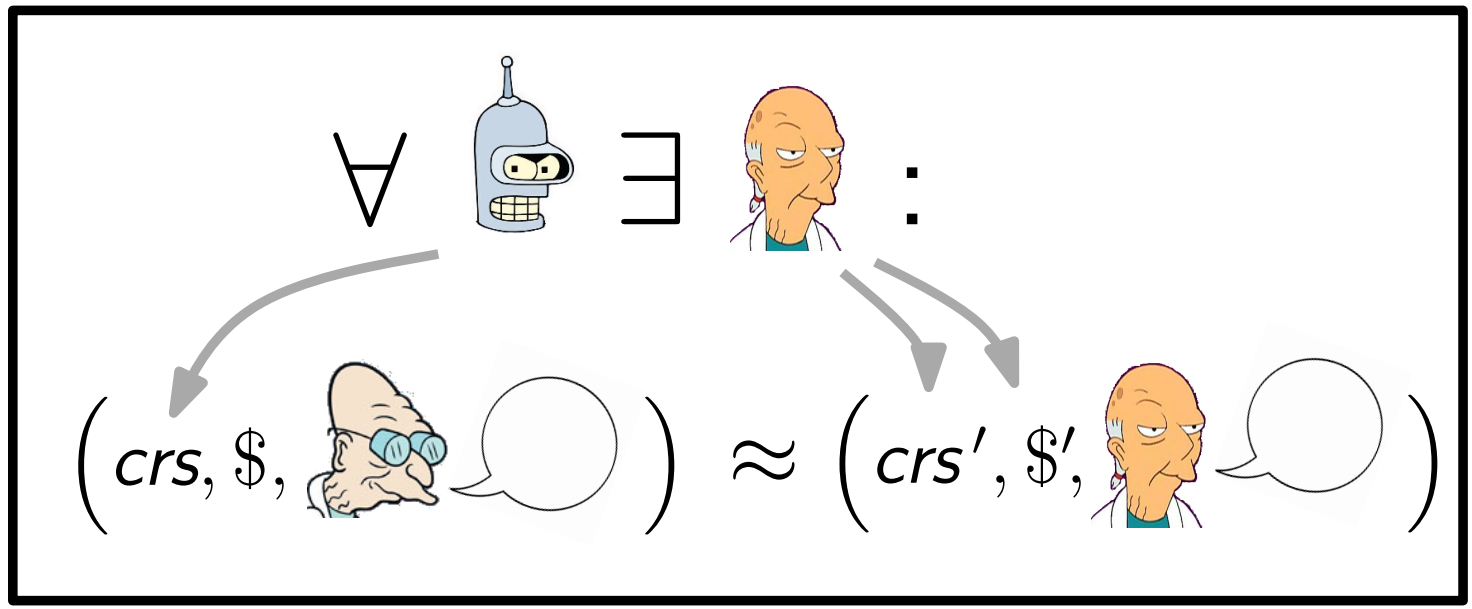
pk

$\mathsf{Enc}(pk, s)$

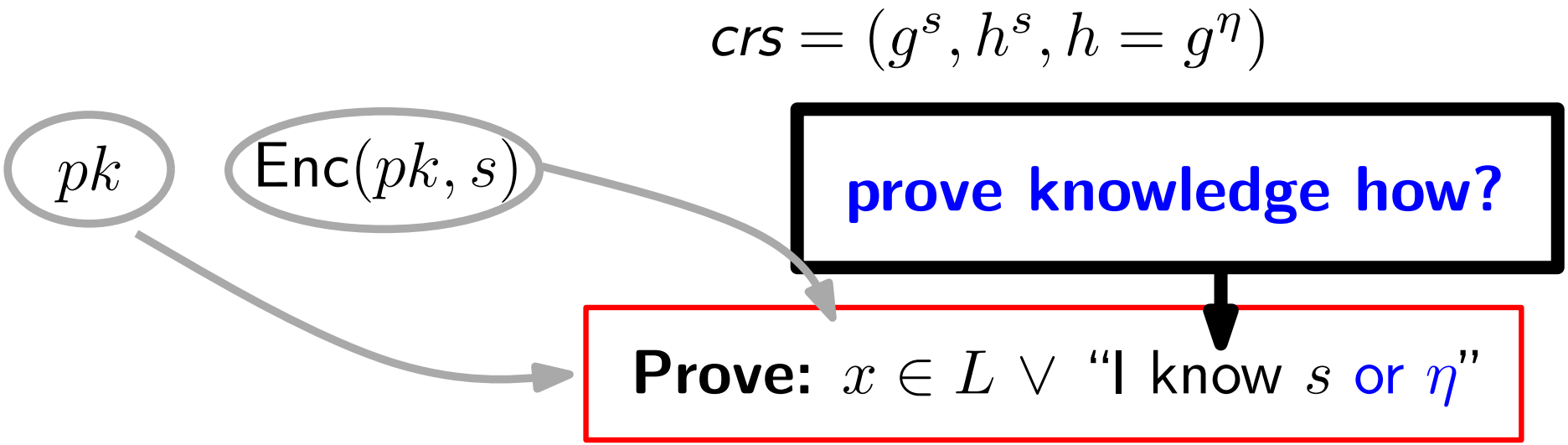
prove knowledge how?

Prove: $x \in L \vee$ "I know s or η"

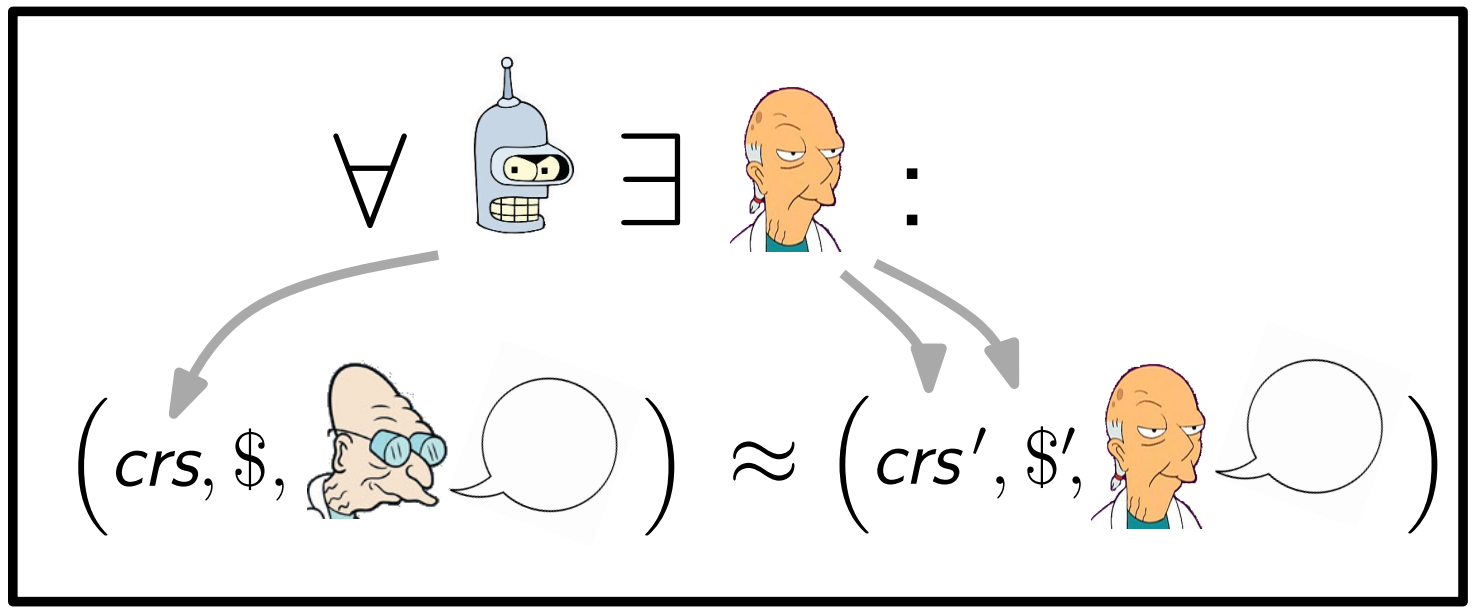
Achieving SND + S-ZK



$$crs = (g^s, h^s, h = g^\eta)$$



Achieving SND + S-ZK



$$crs = (g^s, h^s, h = g^\eta)$$

pk

$\mathsf{Enc}(pk, s)$

make sk extractable

prove knowledge how?

Prove: $x \in L \vee$ "I know s or η"

Results for NIZKs

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			✗	
•		•	•		•	✓	DLin
•	•	•		•	•	✓	DH-KEA

Results for NIZKs

Standard			Subversion-resistant			Possible?	Assumpt's:
SND	ZK	WI	S-SND	S-ZK	S-WI		
	•		•			✗	
•		•	•		•	✓	DLin
•	•	•		•	•	✓	DH-KEA
•	•	•			•	✓	NIZK

SNARKs

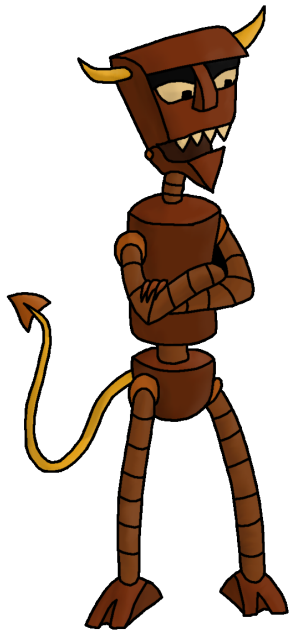
Succinct **N**on-interactive
ARgument of **K**nowledge

- succinct:

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

- proves knowledge of w

Arguments of knowledge



π

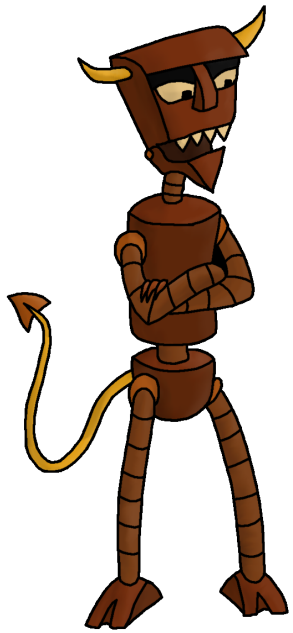
Soundness:

$$\pi \checkmark \Rightarrow x \in L$$

Prover: x, w

Verifier: x

Arguments of knowledge



π

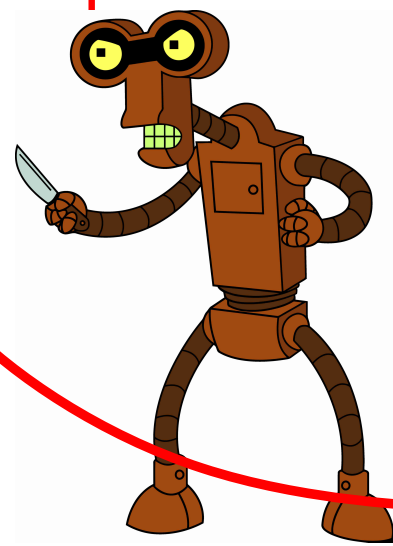
Prover: x, w

Knowledge Soundness:

$\exists$ extractor:

π ✓

$\Rightarrow w$ extracted

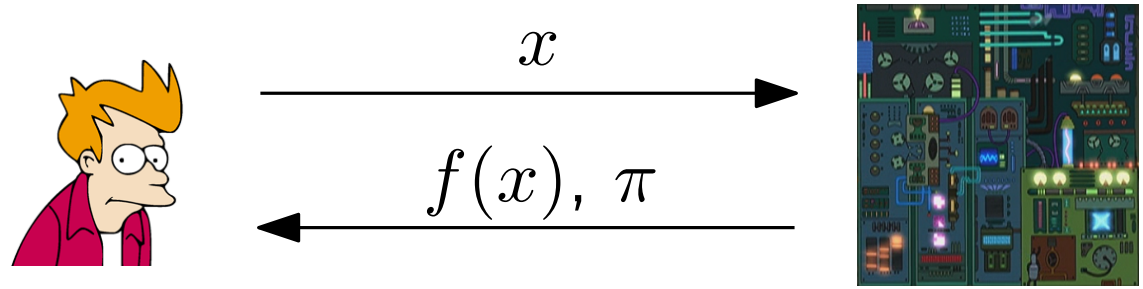


w

Verifier: x

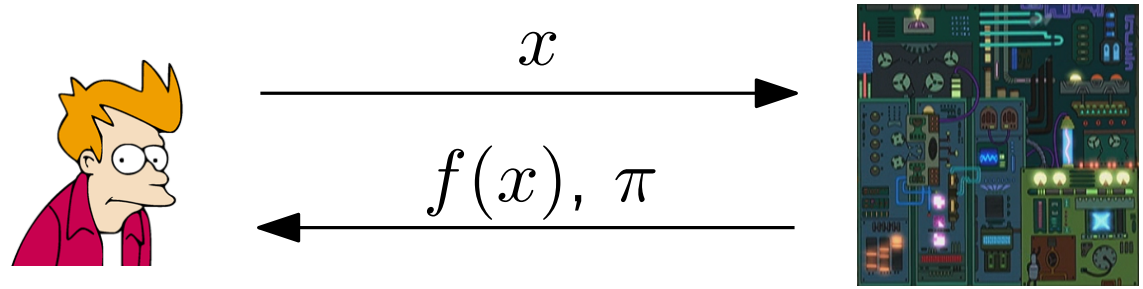
Applications of SNARKs

- Outsourcing of computation



Applications of SNARKs

- Outsourcing of computation

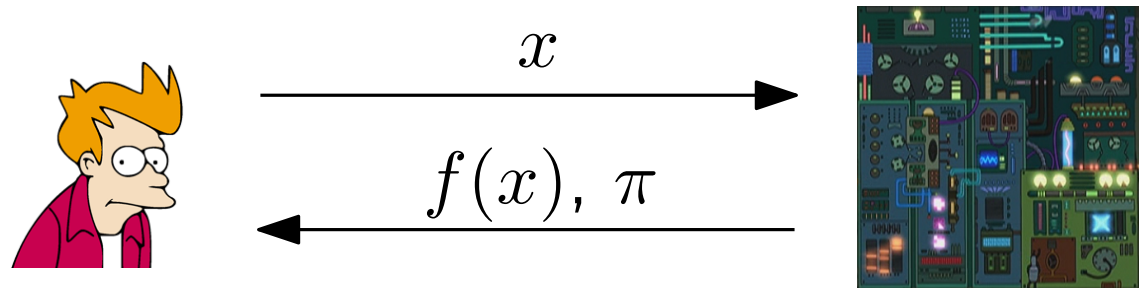


- Anonymous cryptocurrencies: **Zerocash** [BCGGMTV'14]
  - coin is commitment to **serial number**

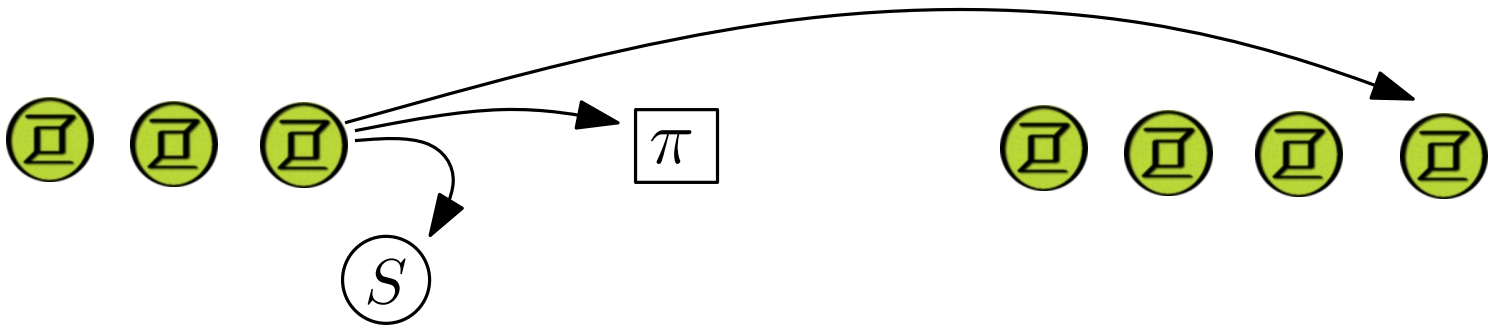
ZCASH

Applications of SNARKs

- Outsourcing of computation



- Anonymous cryptocurrencies: **Zerocash** [BCGGMTV'14]
  - coin is commitment to **serial number**
  - transaction – creates new coins; **reveals** spent serial no.'s
    – **proves** that everything done correctly



ZCASH

Subversion resistance

- SNARKs are perfect zero-knowledge
- but **not** subversion-sound

(CRS contains simulation trapdoor)

Subversion resistance

- SNARKs are perfect zero-knowledge
- but **not** subversion-sound

(CRS contains simulation trapdoor)

Subversion zero-knowledge?

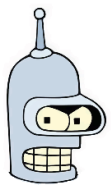
Subversion resistance

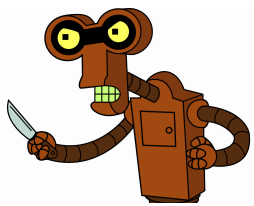
- SNARKs are perfect zero-knowledge
- but **not** subversion-sound

(CRS contains simulation trapdoor)

Subversion zero-knowledge?

Yes! under new KE assumption

DH-KEA: $\forall$  $\rightarrow (g^s, g^t, g^{st})$

$\exists$ $\hookrightarrow$  $\rightarrow s$ OR $\rightarrow t$

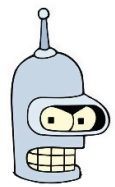
Subversion resistance

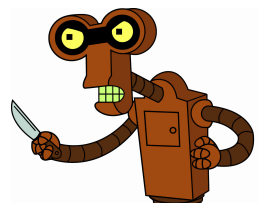
- SNARKs are perfect zero-knowledge
- but **not** subversion-sound

(CRS contains simulation trapdoor)

Subversion zero-knowledge?

Yes! under new KE assumption

SKE: $\forall$  $\rightarrow (g, g^s, g^{s^2})$

$\exists$ $\hookrightarrow$  $\rightarrow s$

Approach

- CRS for SNARKs: $\left(g, g^{s}, g^{s^2}, \ldots, g^{s^d}, \right.$

$$\left. \left\{ g^{p_j(s)} \right\}_j, \left\{ g^{\alpha_i \sum_k \beta_k p_{j,k}(s)} \right\}_{i,j}, \ldots \right)$$

for random $s, \alpha_i, \beta_k, \ldots$ 

Approach

- CRS for SNARKs: $\Big(g, g^{s}, g^{s^2}, \ldots, g^{s^d}, \{g^{p_j(s)}\}_j, \{g^{\alpha_i \sum_k \beta_k p_{j,k}(s)}\}_{i,j}, \ldots\Big)$

  for random $s, \alpha_i, \beta_k, \ldots$ 

- **Check of consistency?**

$$\mathbf{e}(g^{s^2}, h) = \mathbf{e}(g^{s}, h^{s})$$

$$\mathbf{e}(g^{\alpha p_j(s)}, h) = \mathbf{e}(\textstyle\prod_i (g^{s^i})^{p_{j,i}}, h^{\alpha})$$

Approach

- CRS for SNARKs: $\Big(g, g^{s}, g^{s^2}, \ldots, g^{s^d}, \{g^{p_j(s)}\}_j, \{g^{\alpha_i \sum_k \beta_k p_{j,k}(s)}\}_{i,j}, \ldots\Big)$

  for random $s, \alpha_i, \beta_k, \ldots$ 

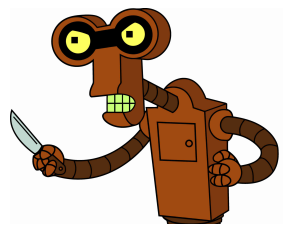
- **Check of consistency?**

$$\mathbf{e}\big(g^{s^2}, h\big) = \mathbf{e}\big(g^{s}, h^{s}\big)$$

$$\mathbf{e}\big(g^{\alpha p_j(s)}, h\big) = \mathbf{e}\big(\textstyle\prod_i (g^{s^i})^{p_{j,i}}, h^{\alpha}\big)$$

- **Simulation?** extraction of s ✓

  but no other  values



SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]

  symm. bilin. grps, $\pi \in \mathbb{G}^9$

  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)

SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]

  symm. bilin. grps, $\pi \in \mathbb{G}^9$

  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)

- CRS checkable?

- Proofs simulatable with s?

SNARK 1 & 2

- Gennaro et al.'s **original** SNARKs [GGPR13]

  symm. bilin. grps, $\pi \in \mathbb{G}^9$

  - QSP-based (boolean circuits)
  - QAP-based (arithmetic circuits)

- CRS checkable? ✓

- Proofs simulatable with s? ✓

$\Rightarrow$ **subversion zero knowledge**

SNARK 3

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

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^7 \times \mathbb{G}_2$

  – QAP-based (arithmetic circuits)

  – underly ZCASH

SNARK 3

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

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^7 \times \mathbb{G}_2$

  – QAP-based (arithmetic circuits)

  – underly ZCASH

- CRS checkable? ✗

- Proofs simulatable with s?

SNARK 3

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

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^7 \times \mathbb{G}_2$

  – QAP-based (arithmetic circuits)

  – underly ZCASH

- CRS checkable? ✗

  $\Rightarrow$ add 4 group elements ✓

- Proofs simulatable with s? ✓

$\Rightarrow$ **subversion zero knowledge**

SNARK 4

- Danezis et al.'s SNARKs [DFGK14]

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^3 \times \mathbb{G}_2$

  – **SSP**-based (boolean circuits)

SNARK 4

- Danezis et al.'s SNARKs [DFGK14]

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^3 \times \mathbb{G}_2$

  – **SSP**-based (boolean circuits)

- CRS checkable? ✓

- Proofs simulatable with s? ✓

$\Rightarrow$ **subversion zero knowledge**

SNARK 5

- Groth's SNARKs [Groth16]

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^{\mathbf{2}} \times \mathbb{G}_2$

  – **QAP**-based (arithmetic circuits)

  – knwl-snd in generic grp model

SNARK 5

- Groth's SNARKs [Groth16]

  asymm. bilin. grps, $\pi \in \mathbb{G}_1^{\mathbf{2}} \times \mathbb{G}_2$

  – **QAP**-based (arithmetic circuits)

  – knwl-snd in generic grp model

- CRS checkable? ✓

- Proofs simulatable with s? ✗

SNARK 5

- Groth's SNARKs [Groth16]

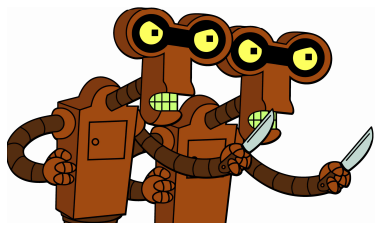
  asymm. bilin. grps, $\pi \in \mathbb{G}_1^{\mathbf{2}} \times \mathbb{G}_2$

  – **QAP**-based (arithmetic circuits)

  – knwl-snd in generic grp model

- CRS checkable? ✓

- Proofs simulatable with s? ✗



$\Rightarrow$ extract more  under SKE

$\Rightarrow$ simulate ✓

$\Rightarrow$ **subv. ZK**

Zcash

Is Zcash anonymous if parameters set up maliciously?

- uses SNARK w/o checkable CRS
- parameters set up using MPC [BCGTV15]
  - uses ROM proofs to prove correctness

Summary SNARKs

Assuming SKE:

- [GGPR13], QSP: subversion-ZK
- [GGPR13], QAP: subversion-ZK
- [BCTV14]: subversion-ZK *after extending CRS*
- [DFGK14]: subversion-ZK
- [Groth16]: subversion-ZK

Zcash

Is Zcash anonymous if parameters set up maliciously?

- uses SNARK w/o checkable CRS
- parameters set up using MPC [BCGTV15]
  - uses ROM proofs to prove correctness

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

Zcash

Is Zcash anonymous if parameters set up maliciously?

- uses SNARK w/o checkable CRS
- parameters set up using MPC [BCGTV15]
  - uses ROM proofs to prove correctness

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

Zcash is subversion-anonymous in the ROM
(if users verify CRS correctness)

THANK YOU!

QUESTIONS?