SNARKs:
An Introduction

Anca Nitulescu

Outline

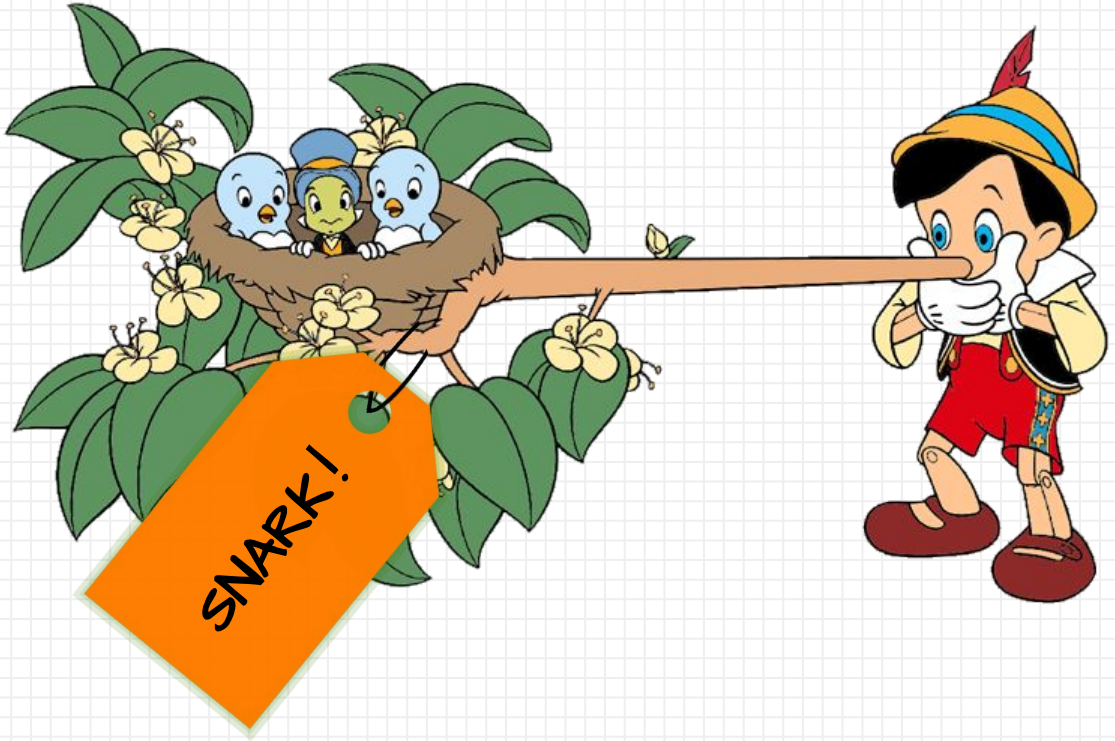
SNARKs
Background

Framework
from QAP

Challenges

Conclusion

Succinct **N**on-interactive **AR**gument of **K**nowledge



SNARK: Definitions and Properties

SNARKs: Proof system

Verifier

$y = f(x)$

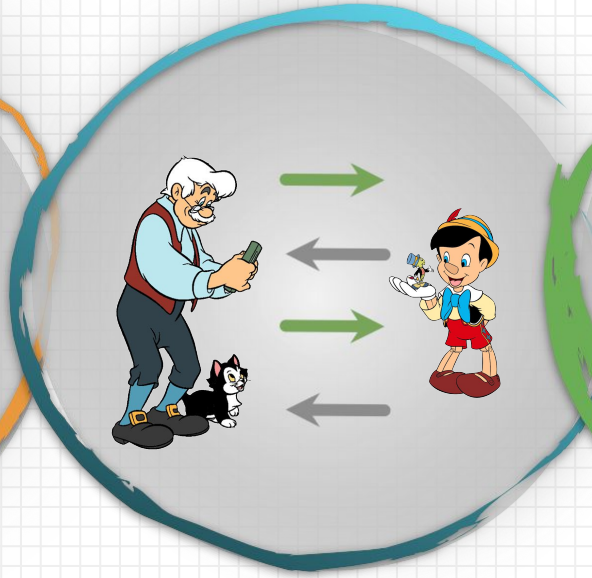
π

Prover

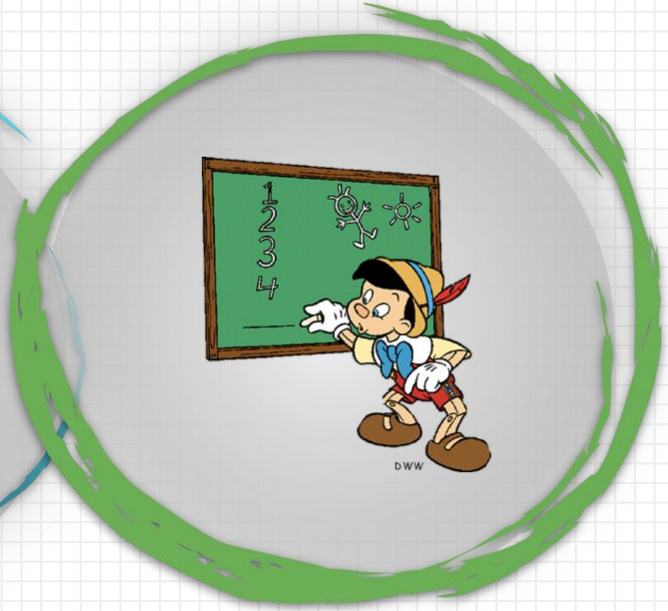
Properties

Succinct Proofs

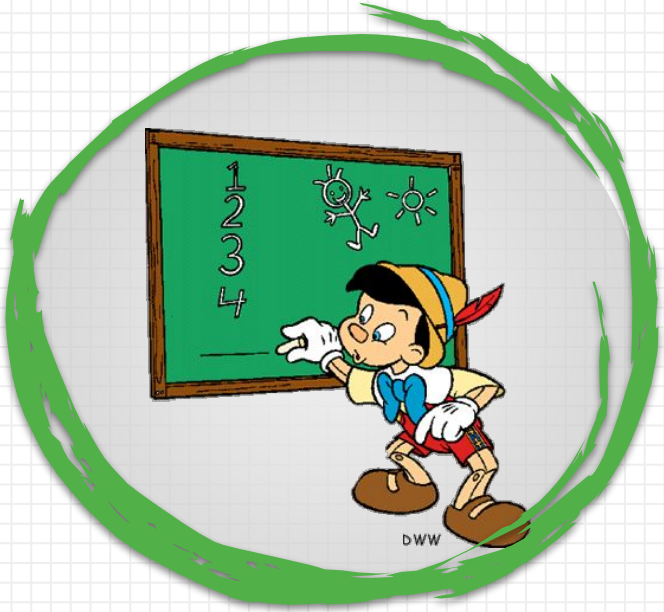
Non-Interactive



Knowledge Soundness



Knowledge Soundness



Knowledge Soundness

$\mathcal{A}$

crs

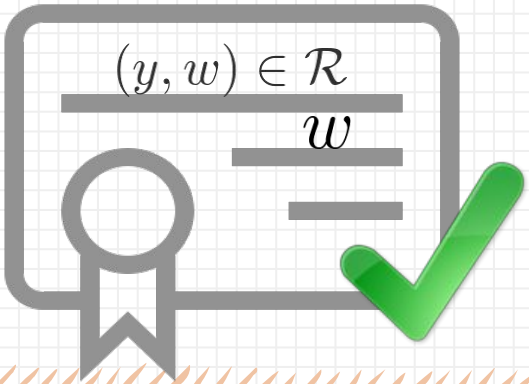
Adversary

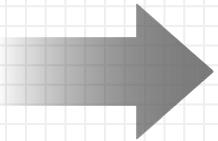
$\mathcal{E}$

crs

extractor

π





Zero-Knowledge: Verifier learns nothing about witness

Zero-Knowledge

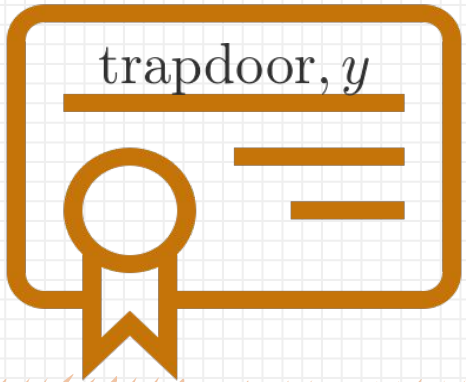
(crs, vk)
trapdoor

Simulator

(crs, vk)

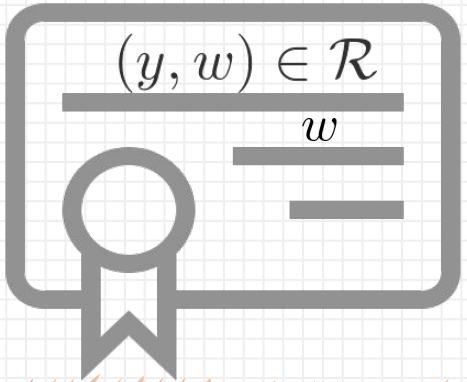
Prover

π'



$\approx$

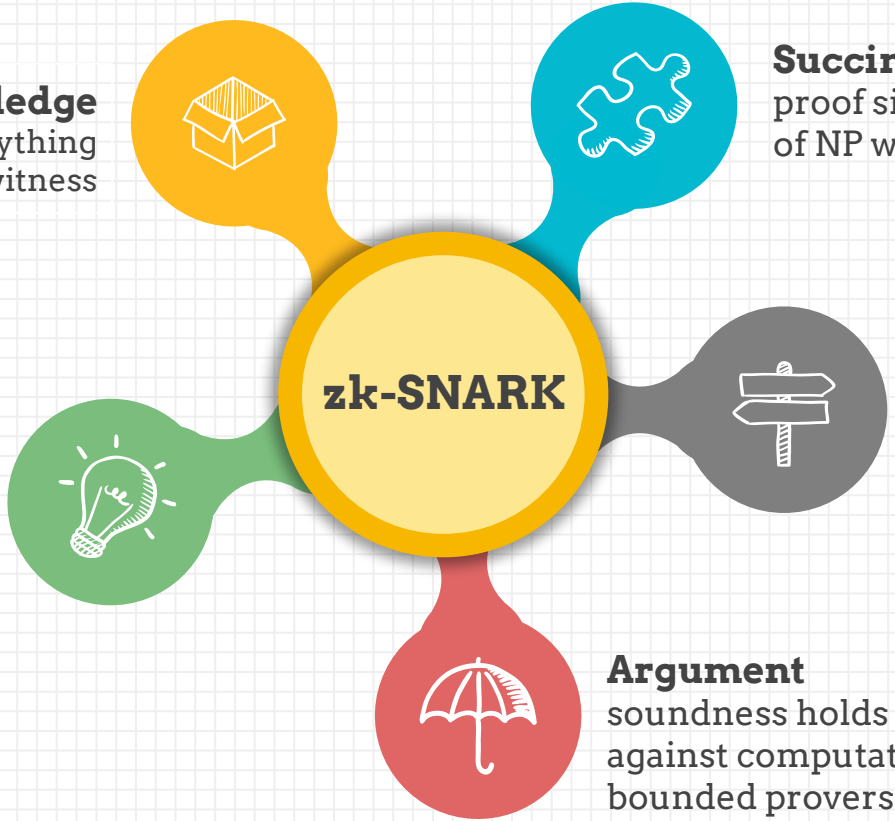
π



Zero-Knowledge SNARK

Zero-Knowledge
does not leak anything about the witness

Succinctness
proof size independent of NP witness size



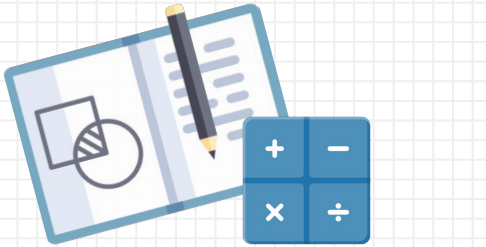
Non-Interactivity
no exchange between prover and verifier

Knowledge Soundness
a witness can be efficiently extracted from the prover

Argument
soundness holds only against computationally bounded provers

Usecase: Outsourcing Computation

Verifier



Prover

Prover claims a statement

Verifier

Claim

$y = f(x)$

Prover

Verifier is able to check

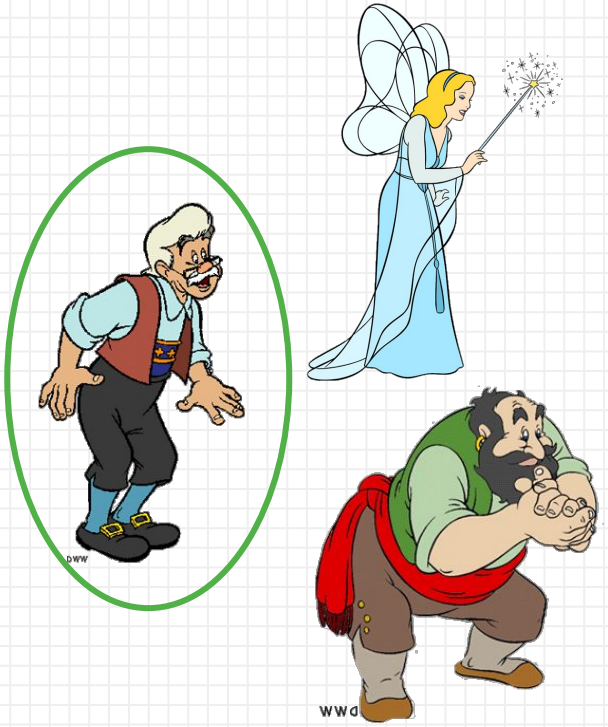
Verifier

Verify

$y' \neq f(x)$

Corrupted Prover

Publicly vs. Designated-Verifier



Verifiers

Claim

Proof π

Prover

SNARKs

PKE

ECRH

GGM

R. Gennaro, C. Gentry, B. Parno, M. Raykova

[GGPR13] QSP and succinct NIZKs without PCPs

B. Parno, J.Howell, C. Gentry, M. Raykova

[PHGR13] Pinocchio: Nearly practical verifiable computation

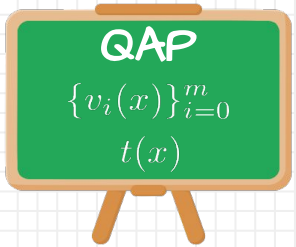
[BCI+13] SNARGs via linear interactive proofs

N.Bitansky, A. Chiesa, Y. Ishai, R. Ostrovsky, O Paneth

[Groth16] On the Size of Pairing-based Non-interactive Arguments

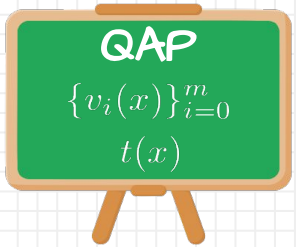
J. Groth

SNARK: Methodology



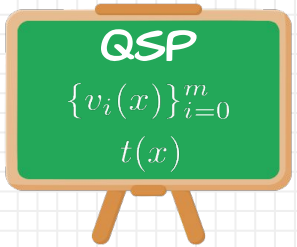
Target Statement R(y,w)=1	Computational Model (Representation)	Encodings Secure under Knowledge Assumptions
Arithmetic Circuit SAT	QAP / SAP over Field = Z_p	PKE Power Knowledge of Exponent GGM Generic Group Model

SNARK: Methodology



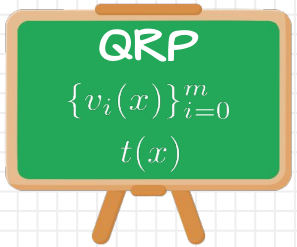
Target Statement R(y,w)=1	Computational Model (Representation)	Encodings Secure under Knowledge Assumptions
Arithmetic Circuit SAT	QAP / SAP over Field = Z_p	PKE Power Knowledge of Exponent GGM Generic Group Model
Boolean Circuit SAT		

SNARK: Methodology



Target Statement R(y,w)=1	Computational Model (Representation)	Encodings Secure under Knowledge Assumptions
Arithmetic Circuit SAT	QAP / SAP over Field = Z_p	**PKE** Power Knowledge of Exponent **GGM** Generic Group Model
Boolean Circuit SAT	QSP / SSP over Field = Z_p	**PKE:** Power Knowledge of Exponent

SNARK: Methodology



Target Statement R(y,w)=1	Computational Model (Representation)	Encodings Secure under Knowledge Assumptions
Arithmetic Circuit SAT	QAP / SAP over Field = Z_p	**PKE** Power Knowledge of Exponent **GGM** Generic Group Model
Boolean Circuit SAT	QSP / SSP over Field = Z_p	**PKE:** Power Knowledge of Exponent
General Circuits over Rings	QRP over Ring = R	**Augmented PKE:** Power Knowledge of Encoding

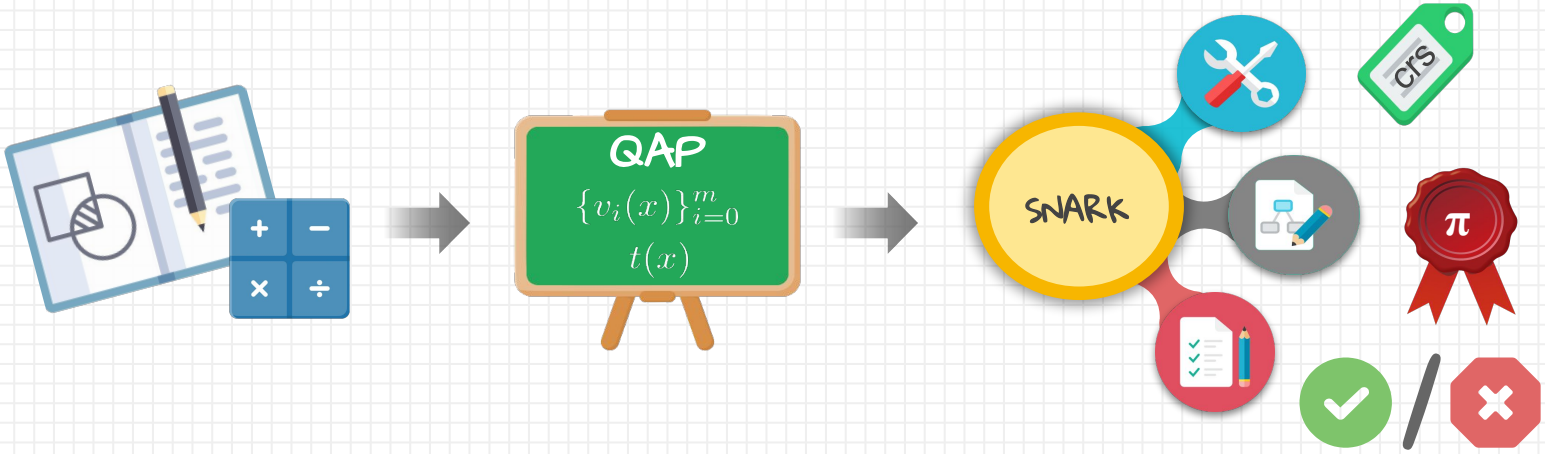
Key Steps to Build SNARKs

SNARK
Background

Framework
from QAP

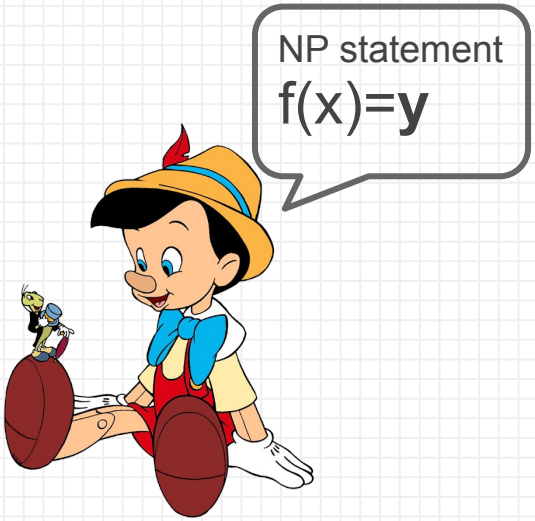
Challenges

Conclusion



Proving NP statements

Verifier

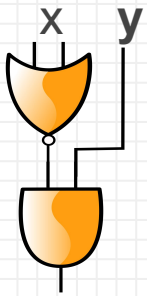


Prover

Prover solves equivalent problem instead

NP statement
$f(x)=y$

compilation



0/1

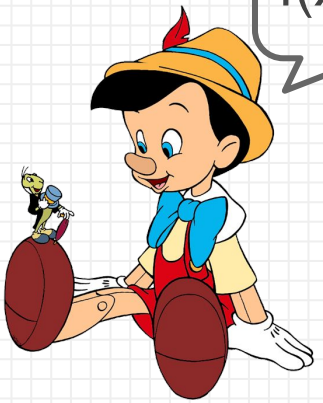
Circuit SAT solution

Polynomial problem

Given **v(x)**, **t(x)**.
Find **P(x)** such that
$\mathbf{P(x)}t(x) = v(x)$

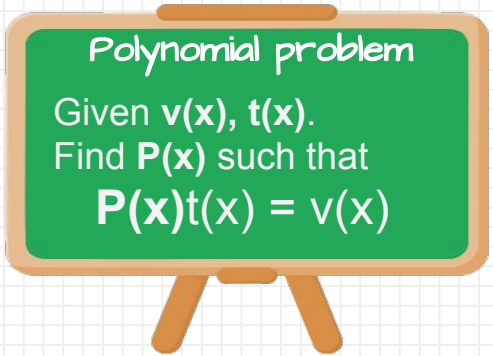
Verifier

NP statement
$f(x)=y$

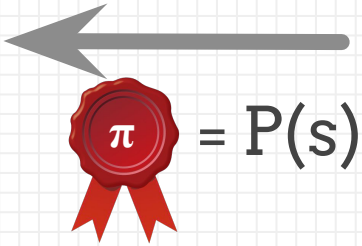


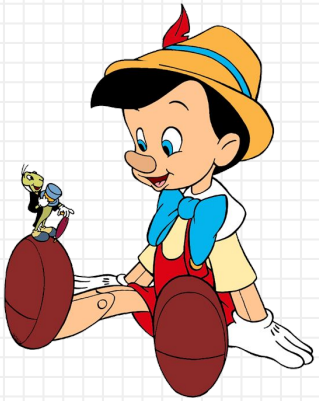
Prover

Evaluate solution at point s



Verifier

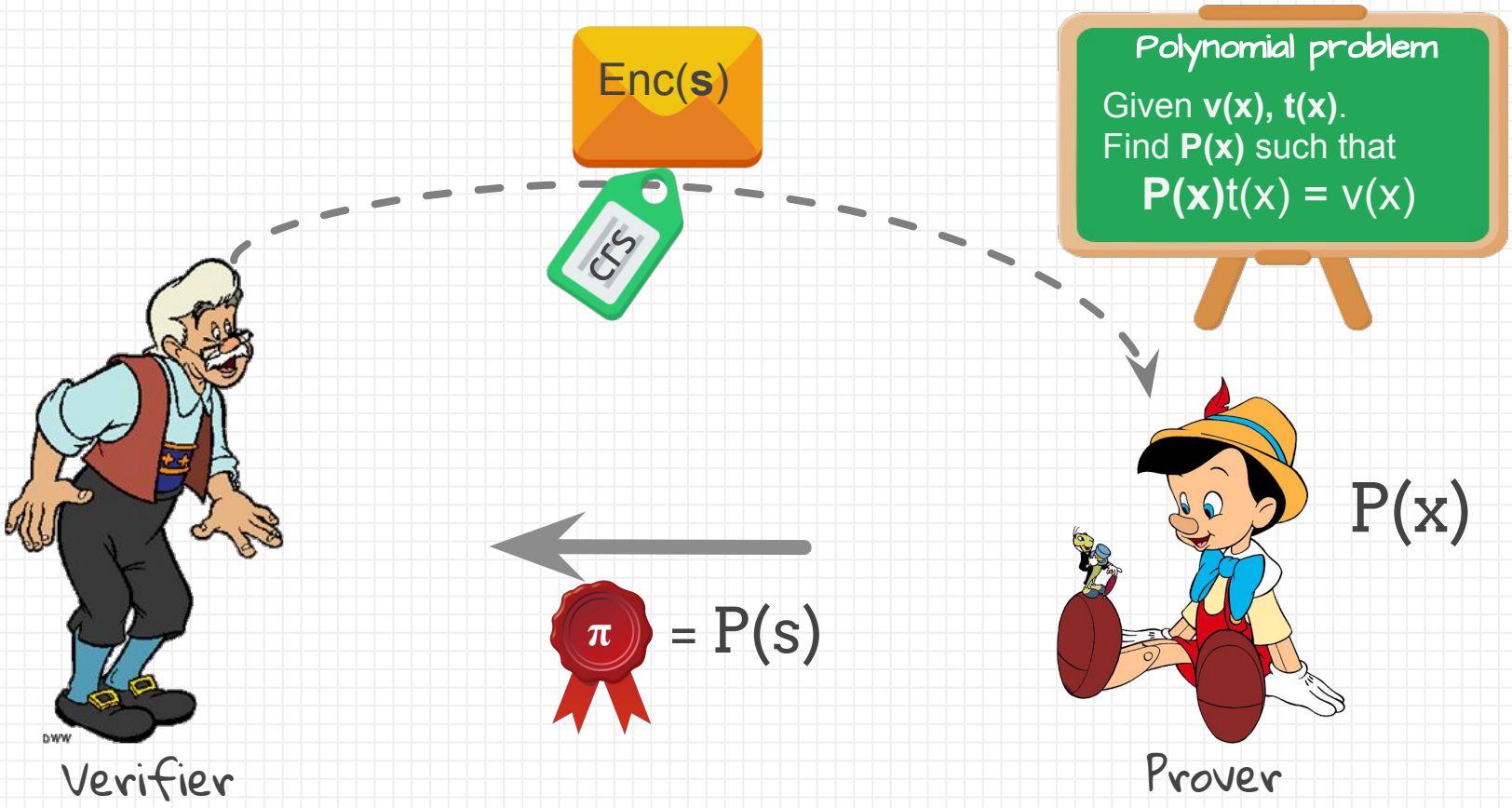




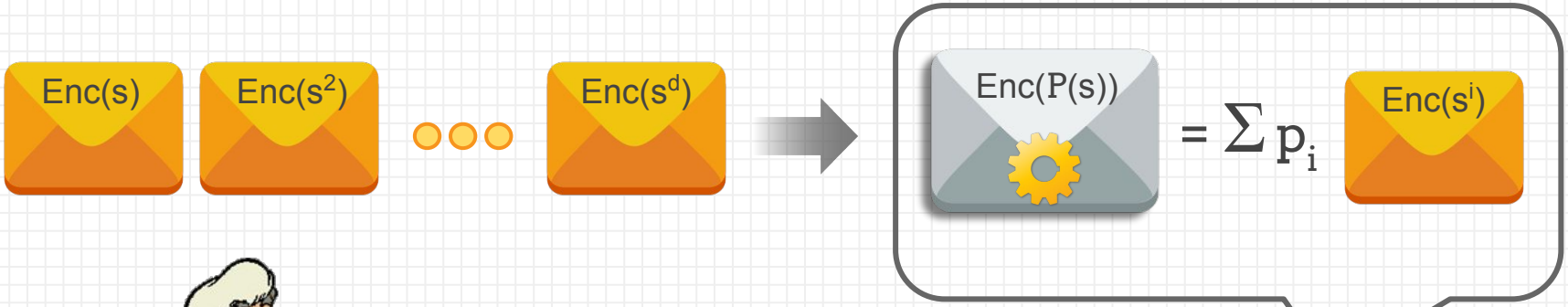
P(x)

Prover

Evaluate solution at point s



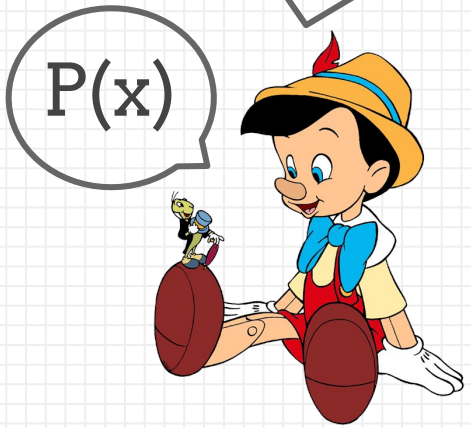
General SNARK framework



Verifier

Encoding:

- ***linearly*** homomorphic

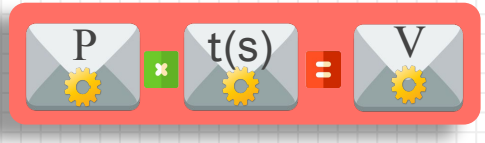


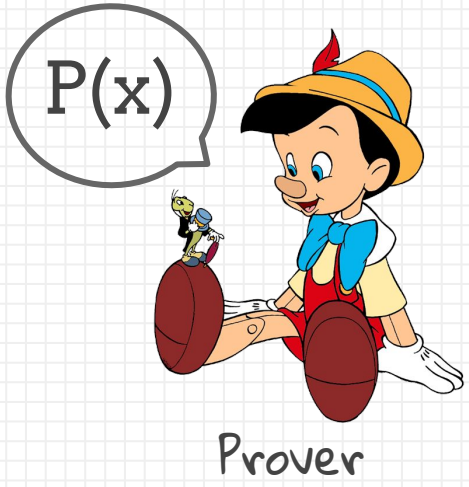
Prover

General SNARK framework

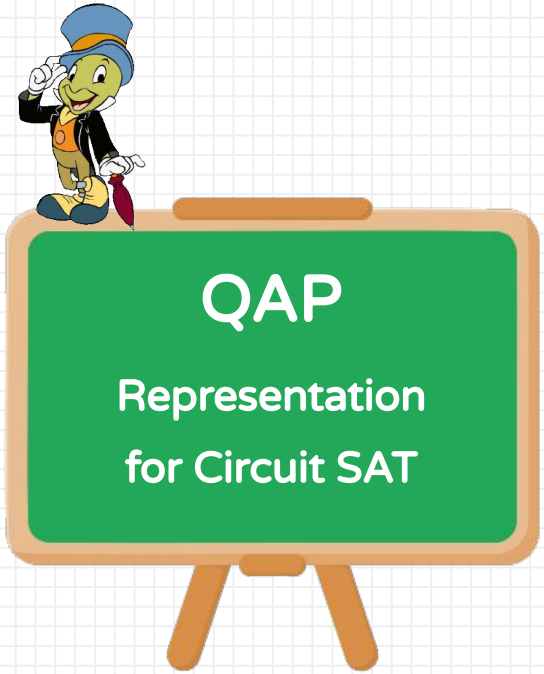
Encoding:

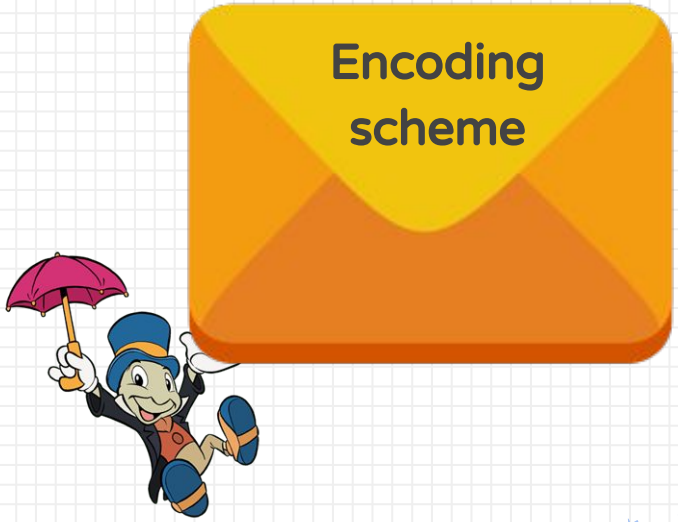
- ***linearly*** homomorphic
- quadratic root detection
- image verification



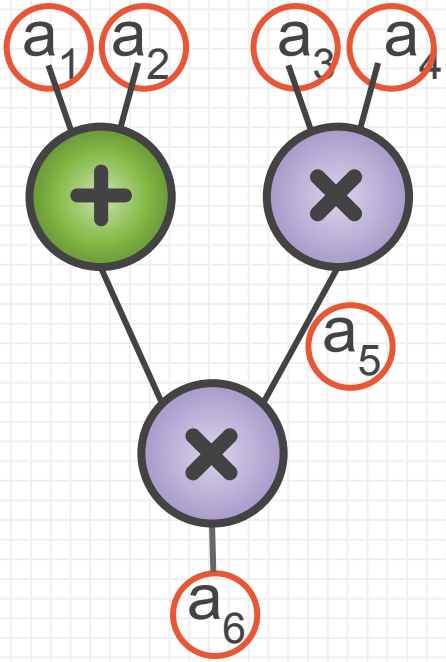


Main Ingredients for SNARKs





Quadratic Arithmetic Program



QAP

Given $\{v_i(x)\}_i, \{w_i(x)\}_i, \{y_i(x)\}_i, t(x)$

Find $V(x), W(x), Y(x), h(x)$ s.t.

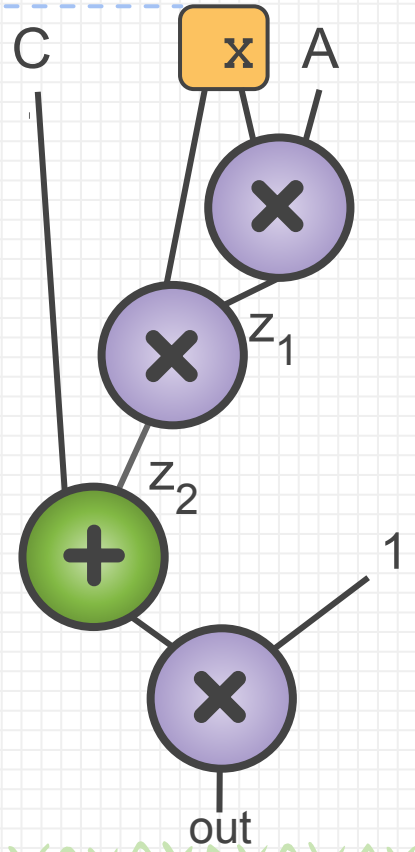
$$V(x) = \sum_{i=0}^{m} a_i v_i(x) \quad ...$$

and $t(x)h(x) = V(x)W(x) - Y(x)$

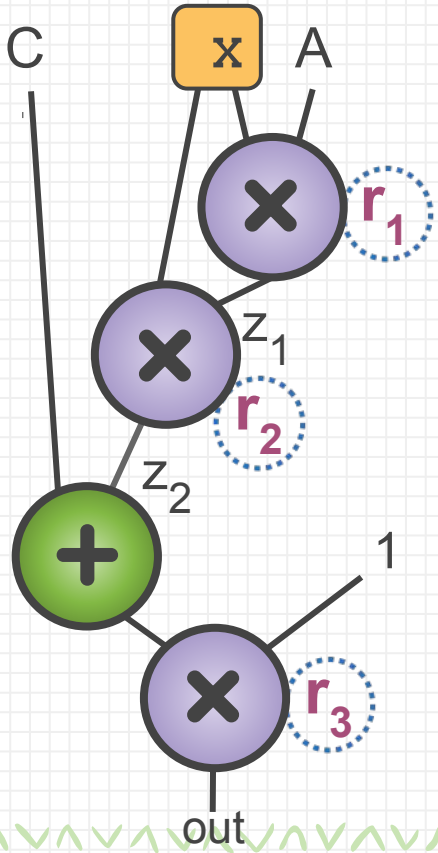
Example: Solution for equation $Ax^2 + C = 0$

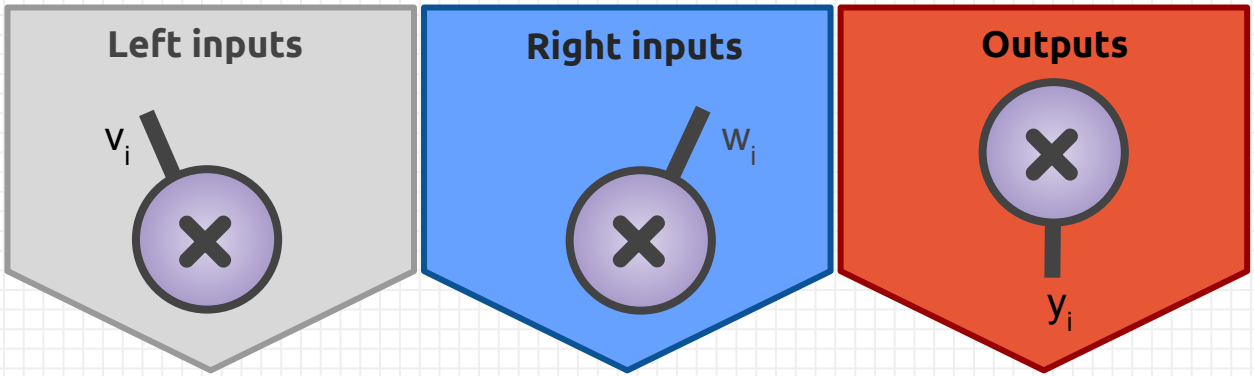
System:

$$\begin{cases} z_1 = A \cdot x \\ z_2 = z_1 \cdot x \\ out = (C + z_2) \cdot 1 \end{cases}$$



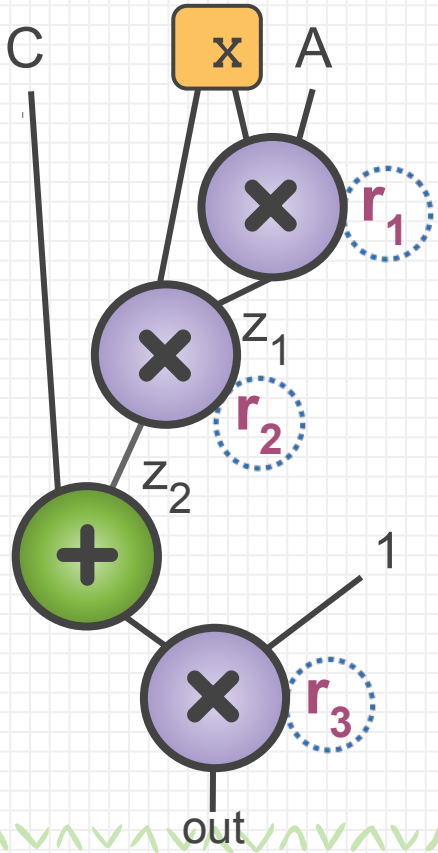
Indexes for wires: A=(0), C=(1), x=(2), z_1=(3), z_2=(4), out = (5)

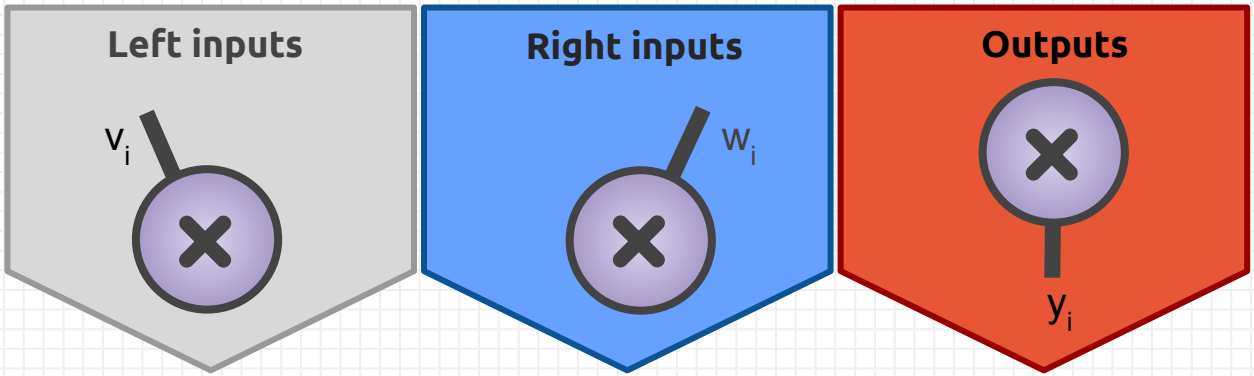




$v_i(r_1) = 1$ for i=2	$w_i(r_1) = 1$ for i=2	$y_i(r_1) = 1$ for i=3,
$v_i(r_2) = 1$ for i=2	$w_i(r_2) = 1$ for i=3	$y_i(r_2) = 1$ for i=4
$v_i(r_3) = 1$ for i=1,4	$w_i(r_3) = 0$ for all i	$w_i(r_3) = 1$ for i=5
$v_i(r_j) = 0$ for the rest	$w_i(r_j) = 0$ for the rest	$y_i(r_j) = 0$ for the rest

Indexes for wires: A=(0), C=(1), x=(2), z_1=(3), z_2=(4), out = (5)

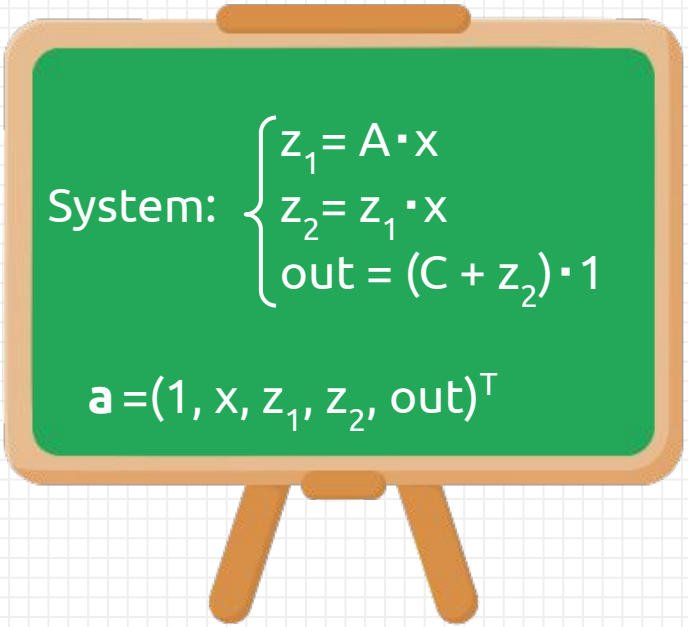




$v_i(r_1) = 1$ for i=2	$w_i(r_1) = 1$ for i=2	$y_i(r_1) = 1$ for i=3,
$v_i(r_2) = 1$ for i=2	$w_i(r_2) = 1$ for i=3	$y_i(r_2) = 1$ for i=4
$v_i(r_3) = 1$ for i=1,4	$w_i(r_3) = 0$ for all i	$w_i(r_3) = 1$ for i=5

$$\prod_{j=1}^{d}(x - r_j) \Big| \Big(\sum_{i=0}^{m} a_i v_i(x)\Big)\Big(\sum_{i=0}^{m} a_i w_i(x)\Big) - \Big(\sum_{i=0}^{m} a_i y_i(x)\Big)$$

R1CS for vector $\mathbf{a}=(1, A, C, x, z_1, z_2, out)$



$\mathbf{a}=(1, x, z_1, z_2, out)^T$

$(A, 0, 0, 0, 0) \cdot \mathbf{a} \circ (0, 1, 0, 0, 0) \cdot \mathbf{a} = (0, 0, 1, 0, 0) \cdot \mathbf{a}$

$(0, 0, 1, 0, 0) \cdot \mathbf{a} \circ (0, 1, 0, 0, 0) \cdot \mathbf{a} = (0, 0, 0, 1, 0) \cdot \mathbf{a}$

$(C, 0, 0, 1, 0) \cdot \mathbf{a} \circ (1, 0, 0, 0, 0) \cdot \mathbf{a} = (0, 0, 0, 0, 1) \cdot \mathbf{a}$

$$\left[V \; a \right] \circ \left[W \; a \right] = Y \; a$$

R1CS for vector **a** =(1,A, C, x, z_1, z_2, out)

System:
$$\begin{cases} z_1 = A \cdot x \\ z_2 = z_1 \cdot x \\ out = (C + z_2) \cdot 1 \end{cases}$$

a =$(1, x, z_1, z_2, out)^T$

$$v_i(r_j) = V_{ji} \qquad \forall \{r_j\} \in \mathbb{F}^d$$

$$\left(\sum_{i=0}^{m} a_i v_i(x)\right)\left(\sum_{i=0}^{m} a_i w_i(x)\right) = \left(\sum_{i=0}^{m} a_i y_i(x)\right)$$

R1CS for vector **a** =(1,A, C, x, z_1, z_2, out)

System:
$$\begin{cases} z_1 = A \cdot x \\ z_2 = z_1 \cdot x \\ out = (C + z_2) \cdot 1 \end{cases}$$

a $=(1, x, z_1, z_2, out)^T$

$$v_i(r_j) = V_{ji} \qquad \forall \{r_j\} \in \mathbb{F}^d$$

$$\left(\sum_{i=0}^{m} a_i v_i(x)\right)\left(\sum_{i=0}^{m} a_i w_i(x)\right) = \left(\sum_{i=0}^{m} a_i y_i(x)\right)$$

$$\prod_{j=1}^{d}(x - r_j) \Bigg| \left(\sum_{i=0}^{m} a_i v_i(x)\right)\left(\sum_{i=0}^{m} a_i w_i(x)\right) - \left(\sum_{i=0}^{m} a_i y_i(x)\right)$$

Schwartz-Zippel Lemma over Fields

$$t(x) = \prod_{j=1}^{d}(x - r_j) \;\Big|\; \Big(\sum_{i=0}^{m} a_i v_i(x)\Big)\Big(\sum_{i=0}^{m} a_i w_i(x)\Big) - \Big(\sum_{i=0}^{m} a_i y_i(x)\Big) = p(x)$$

Lemma: Let $f \in \mathbb{F}[X]$ be a **non-zero** poly.

$$\Pr_{s \leftarrow \mathbb{F}}[f(s) = 0] \leq \frac{\deg(f)}{|\mathbb{F}|}$$

Encodings over Fields

DLog Group $\mathbb{G}$

$$\langle g \rangle = \mathbb{G}, \;\; Enc(s) = g^s$$

$$Enc(p(s)) = g^{p(s)}$$

$$g^{\sum_i p_i s^i} = \prod (g^{s^i})^{p_i}$$

DLog vs General Encoding

DLog Group $\mathbb{G}$

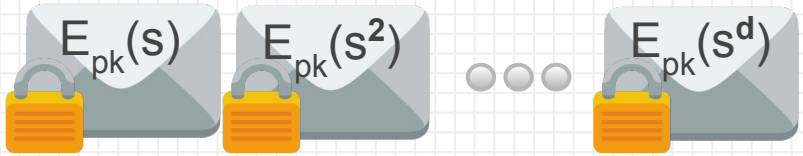
$$\langle g \rangle = \mathbb{G}, \quad Enc(s) = g^s$$

$$Enc(p(s)) = g^{p(s)}$$

$$g^{\sum_i p_i s^i} = \prod (g^{s^i})^{p_i}$$

Encode: $E_{pk}(m) = c$

Decode: $D_{sk}(c) = m$



$$Enc(p(s)) = E_{pk}(p(s))$$

$$E_{pk}(\sum p_i s^i) = \sum p_i E_{pk}(s^i)$$

Quadratic Root Detection – Pairings

$$\langle g \rangle = \mathbb{G}, \langle \tilde{g} \rangle = \tilde{\mathbb{G}}$$

$$Enc(s) = g^s \qquad e : \mathbb{G} \times \mathbb{G} \to \tilde{\mathbb{G}}$$

$$e(g^a, g^b) = \tilde{g}^{ab}$$

Quadratic root detection **public**

$$t(s)h(s) \stackrel{?}{=} p(s)$$

$$e(g^{t(s)}, g^{h(s)}) \stackrel{?}{=} e(g^{p(s)}, g)$$

Publicly Verifiable vs Designated Verifiable

$$\langle g \rangle = \mathbb{G}, \langle \tilde{g} \rangle = \tilde{\mathbb{G}}$$
$$Enc(s) = g^s \qquad e : \mathbb{G} \times \mathbb{G} \to \tilde{\mathbb{G}}$$
$$e(g^a, g^b) = \tilde{g}^{ab}$$

Quadratic root detection **public**

$$t(s)h(s) \stackrel{?}{=} p(s)$$

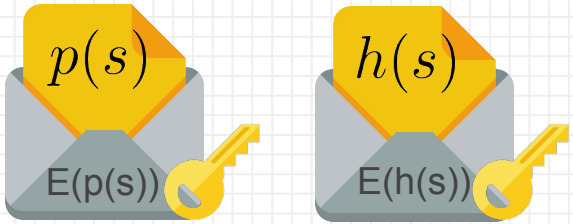
$$e(g^{t(s)}, g^{h(s)}) \stackrel{?}{=} e(g^{p(s)}, g)$$

Encode: $E_{pk}(m) = c$

Decode: $D_{sk}(c) = m$

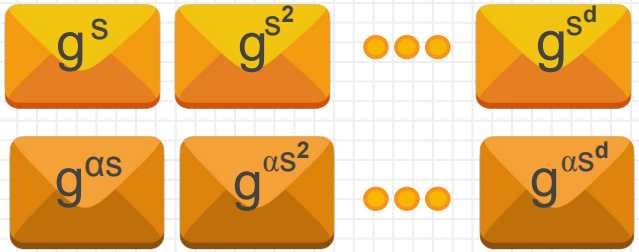
Quadratic root detection needs **sk**

$$t(s)h(s) \stackrel{?}{=} p(s)$$



Assumptions on Discrete Log Encoding for Fields

d-PKE



d-PDH

Technical Details

SNARK
Background

Framework
from QAP

Challenges

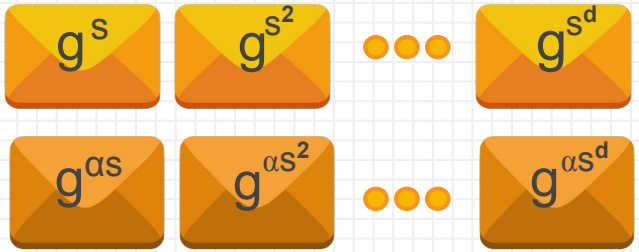
Conclusion

Encodings

Assumptions

Assumptions on Discrete Log Encoding for Fields

d-PKE

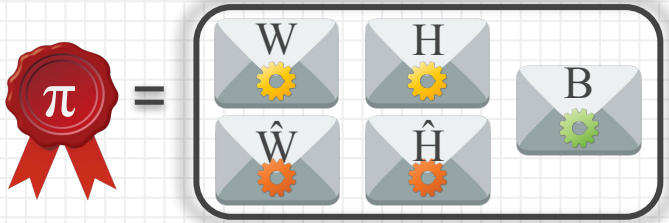


d-PDH

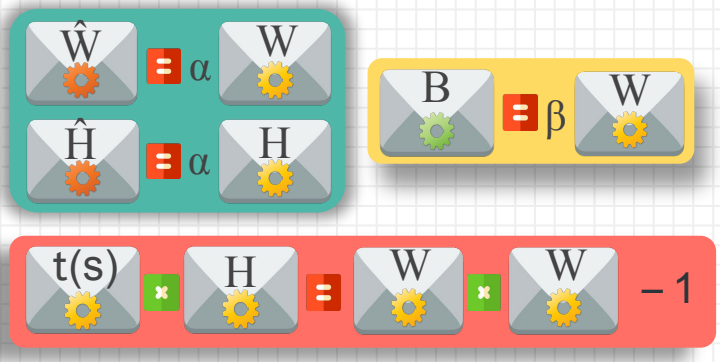
Review of the Protocol (Algorithms)

$\mathsf{Gen}(1^\lambda, \mathcal{R})$

$\mathsf{P}(\mathsf{crs}, y, w)$



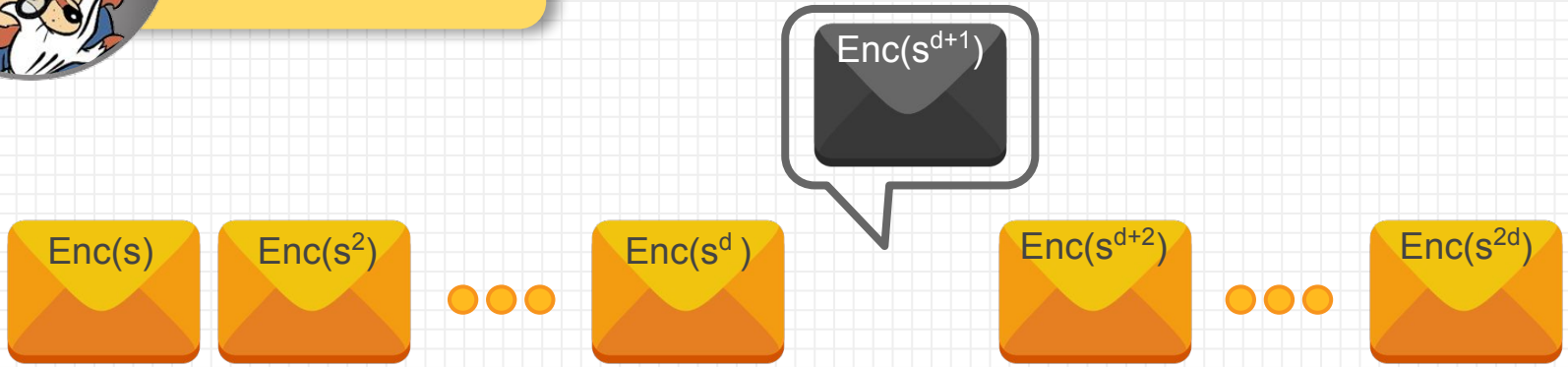
$\mathsf{V}(\mathsf{vk}, y, \pi)$



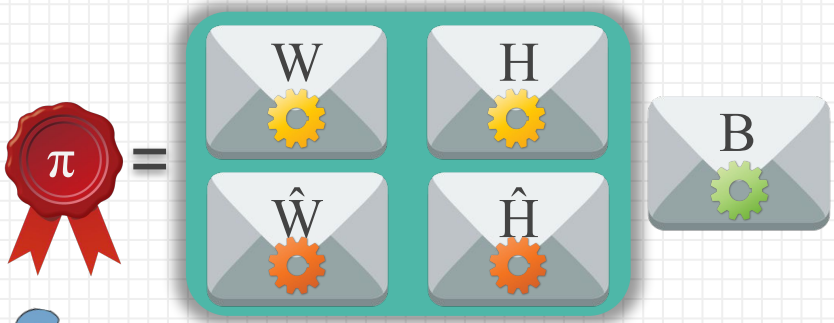
Security Analysis

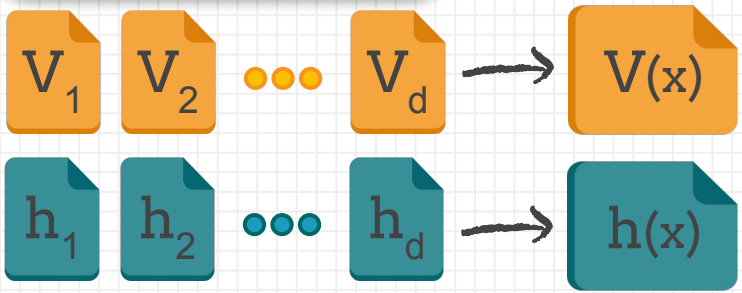
Reduction to PDH: Power Diffie-Hellman

d-PDH

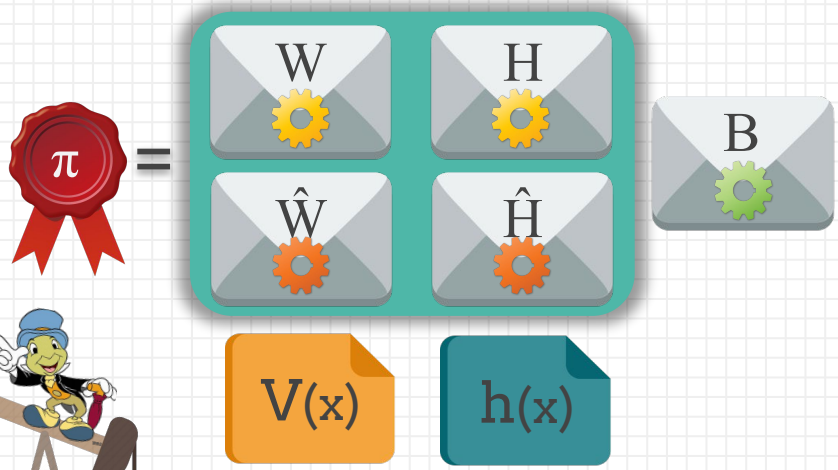


Security Reduction: Cheating Strategy

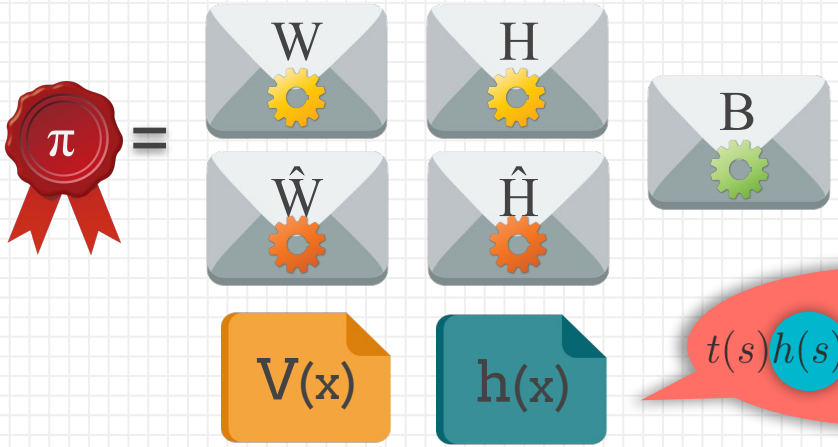




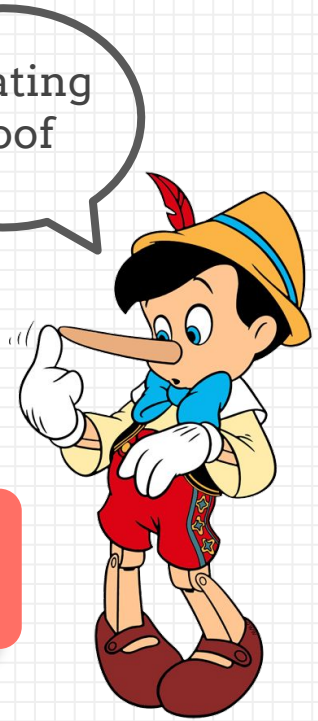
Security Reduction: Cheating Strategy



Polynomial Division does not Hold



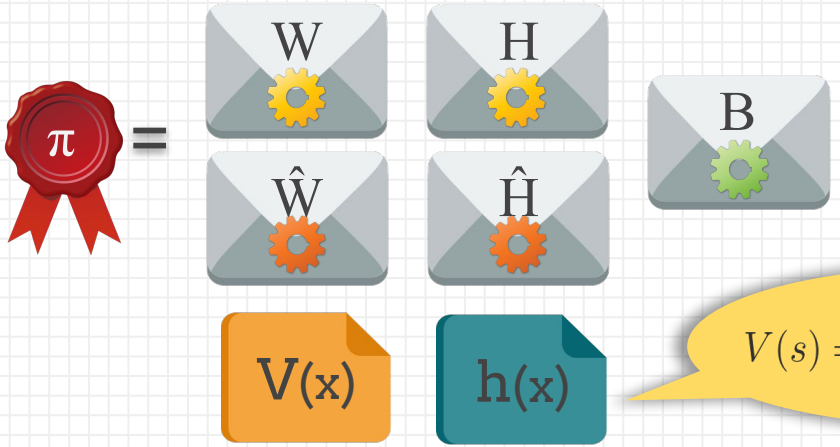
$t(s)h(s) = V(s)^2 - 1$



- $t(x)h(x) \neq V^2(x) - 1$, but

Enc(t(s)) × H = W^2 – 1

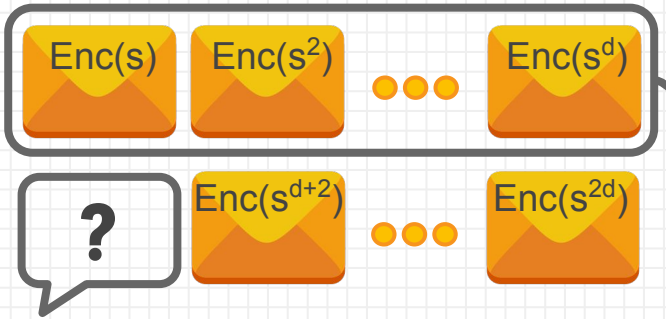
Not in the Proper Span



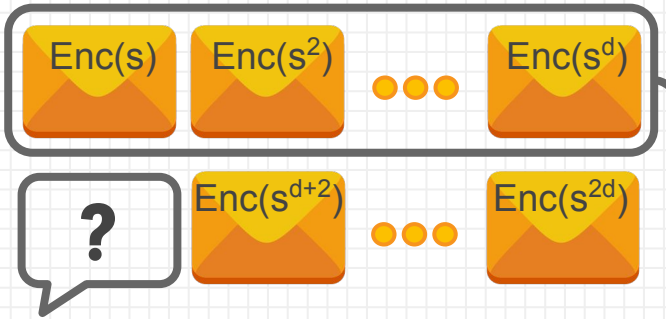
$$V(s) = v_0(s) + \sum_{i=1}^{m} a_i v_i(s)$$

- $t(x)h(x) \neq V^2(x) - 1$, but $\mathsf{Enc}(t(s))H = W^2 - 1$
- $V(x) \notin \mathsf{Span}(v_1, \ldots v_m)$, but

d-PDH



d-PDH



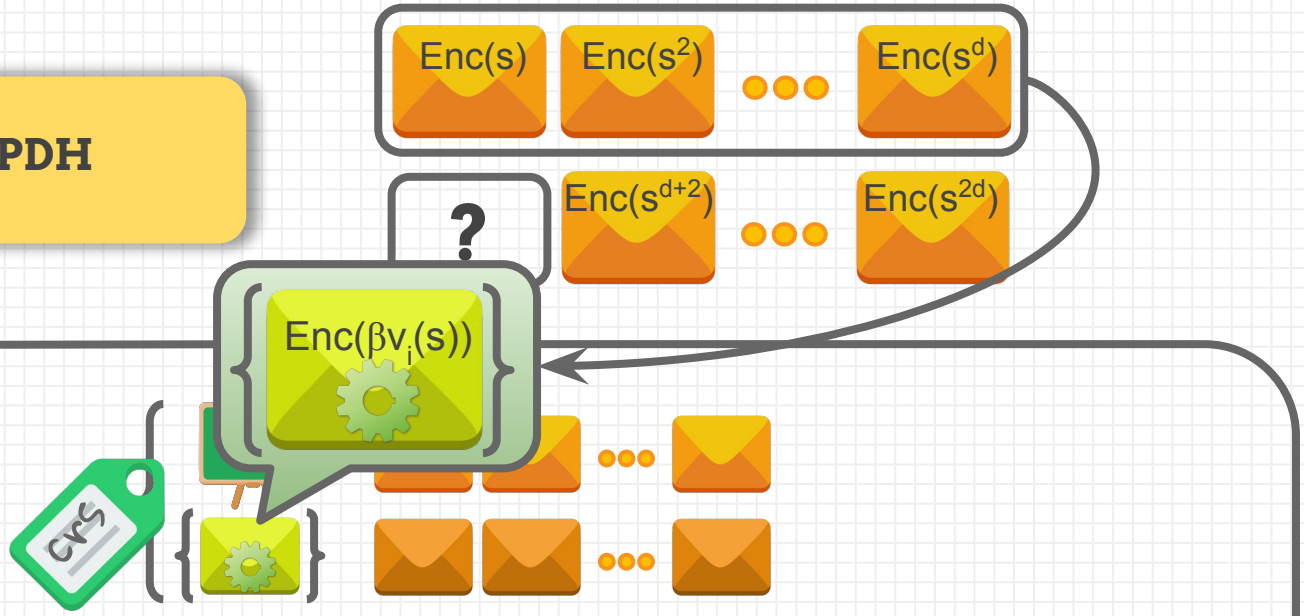
d-PDH

Enc(s) Enc(s^2) ••• Enc(s^d)

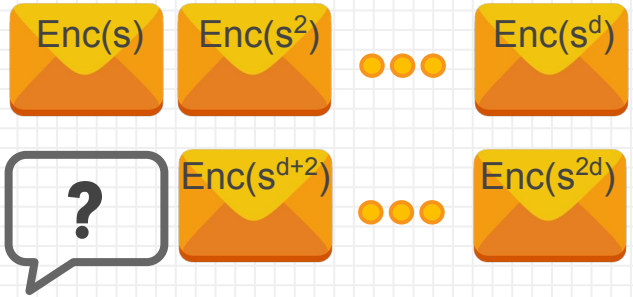
Enc(s^{d+2}) ••• Enc(s^{2d})

$$\{v_i(x)\}_{i=0,m}$$
$$t(x)$$

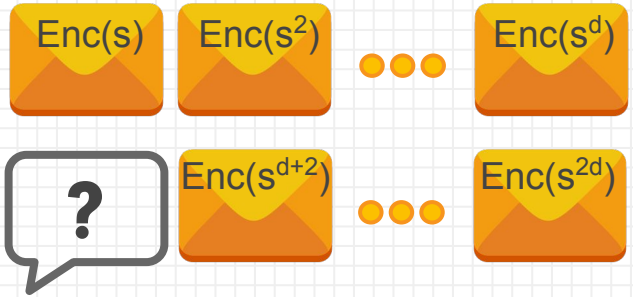
d-PDH

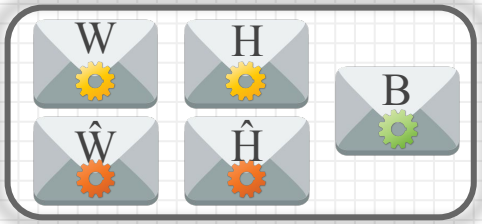


d-PDH

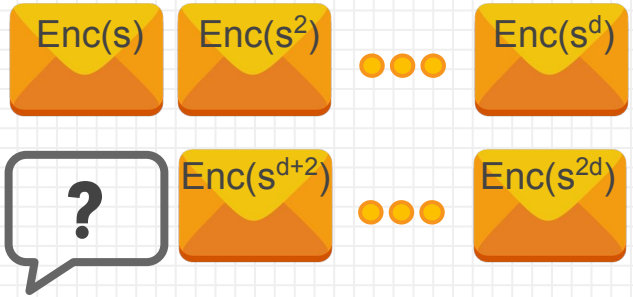


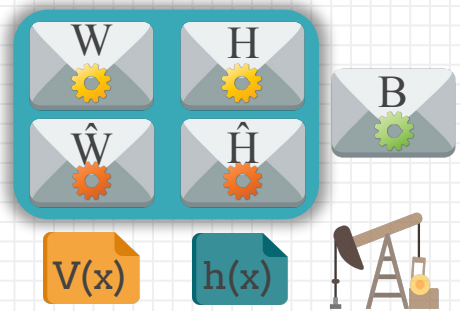
d-PDH





d-PDH





d-PDH

Enc(s) Enc(s^2) ... Enc(s^d)

? Enc(s^{d+2}) ... Enc(s^{2d})

W H B

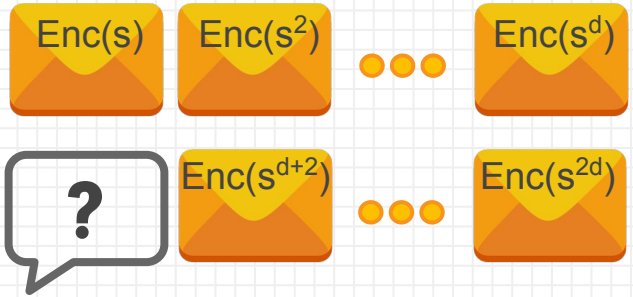
$\hat{W}$ $\hat{H}$

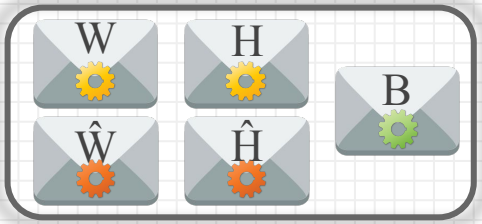
V(x)

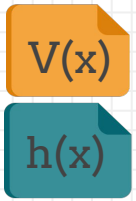
h(x)

$$t(x)h(x) \neq V(x)^2 - 1, \text{ but } t(s)h(s) = V(s)^2 - 1$$

d-PDH

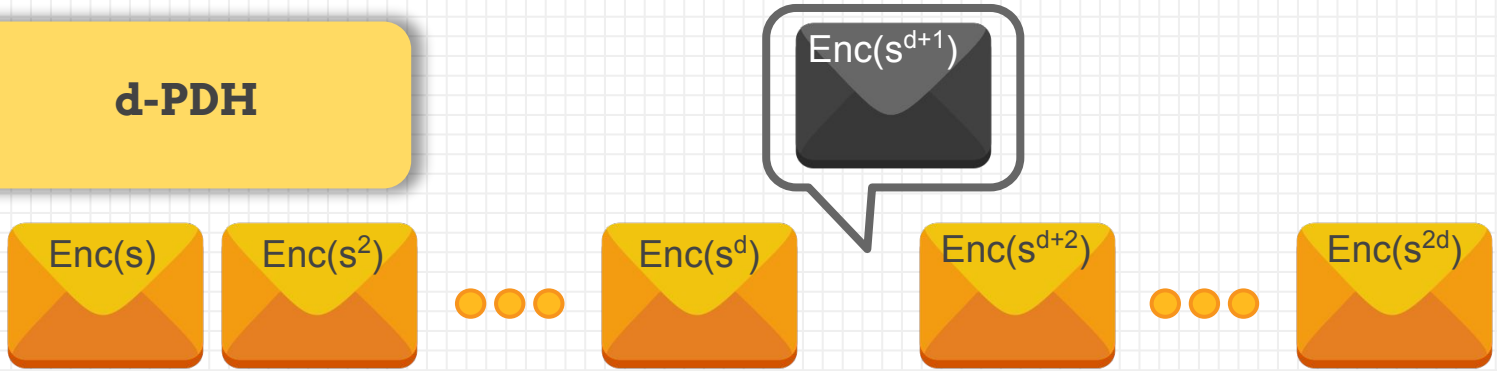






$$t(x)h(x) \neq V(x)^2 - 1, \text{ but } t(s)h(s) = V(s)^2 - 1$$
$$p(x) = t(x)h(x) - V(x)^2 + 1 \neq 0, \quad \text{but } p(s) = 0$$

d-PDH



$$p(x) = t(x)h(x) - V(x)^2 + 1 \neq 0, \quad \text{but } p(s) = 0$$

$$p_{d+1}\,\text{Enc}(s^{d+1}) = -\sum_{i=1,\ldots d}^{d+2,\ldots 2d} p_i\,\text{Enc}(s^i)$$

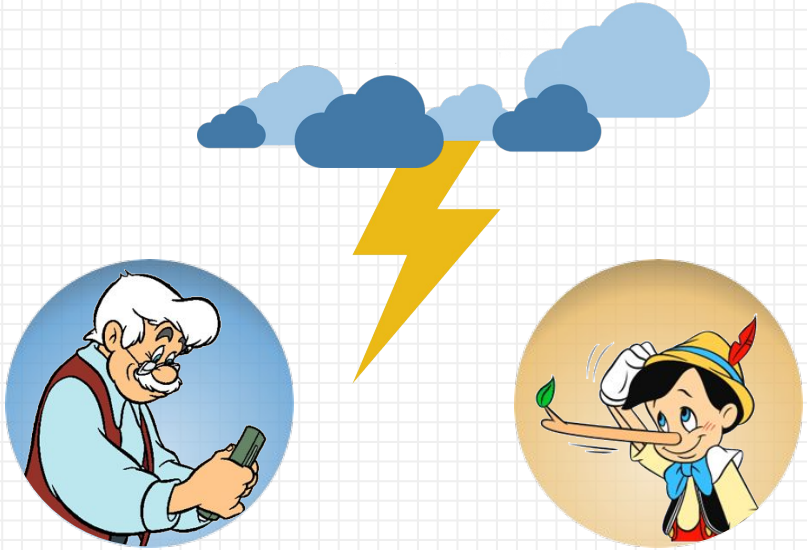
Conclusions

SNARK
Background

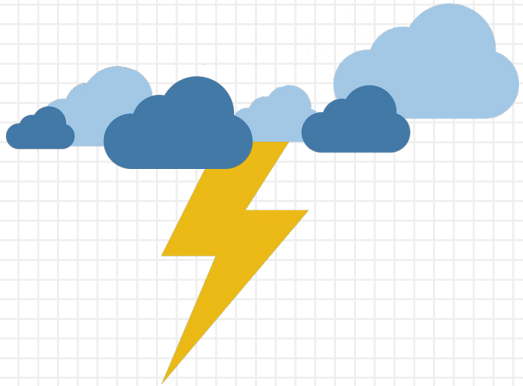
Framework
from QAP

Challenges

Conclusion



Conclusions



Pre-Processing:
(**crs**: common reference string)

- ✘ Secret coins
- ✘ Expensive
- ✘ Subversion

Subversion-Resistant Protocols

- ✘ Updatable **crs**
- ✘ Verifiable **crs**

THANK YOU

Credits

Special thanks to all those who made and released these resources for free:

- ✘ Presentation template by SlidesCarnival
- ✘ Illustrations by Disneyclips, Iconfinder and Flaticon