

Rinocchio: SNARKs for Ring Arithmetic

joint work with Chaya Ganesh,
Eduardo Soria-Vazquez

Anca Nitulescu

Protocol Labs



Outline

SNARKs

Background

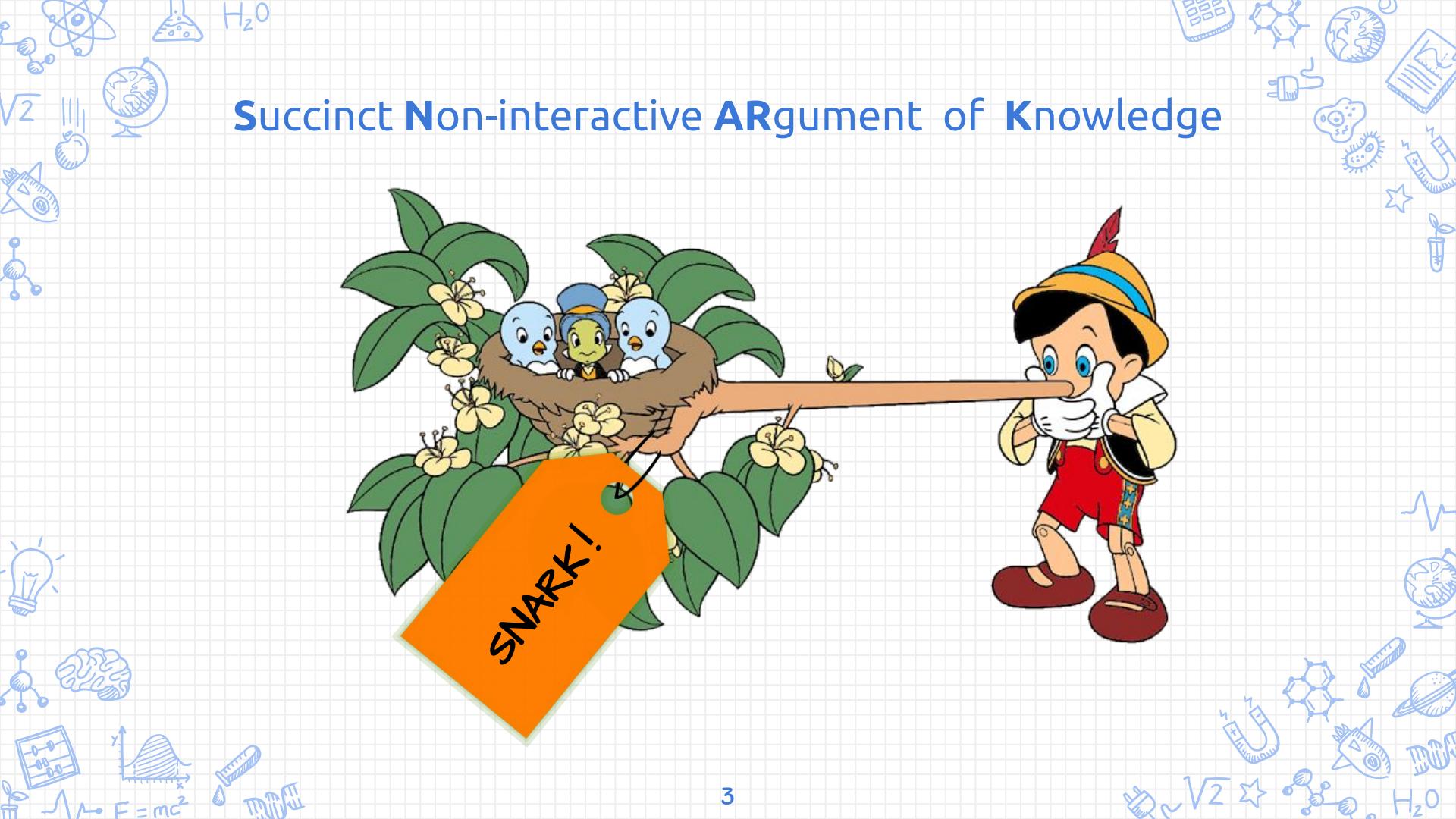
Framework
for Rinocchio

Challenges

New tools

Conclusion





Succinct Non-interactive ARgument of Knowledge

SNARK: Definitions and Properties

SNARK
Background

Framework
for Rinocchio

Challenges
New tools

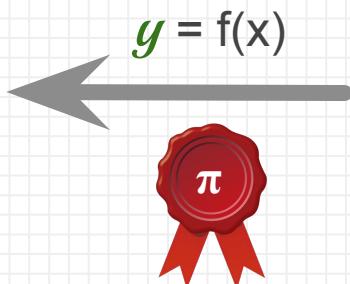
Conclusion



SNARKs: Proof system



Verifier



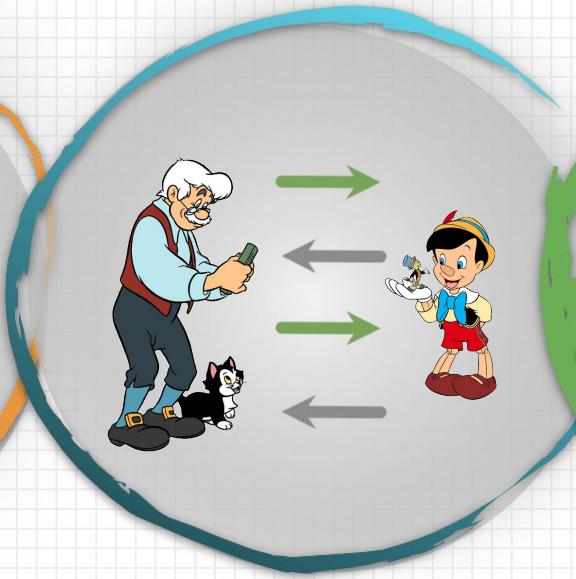
Prover

Properties

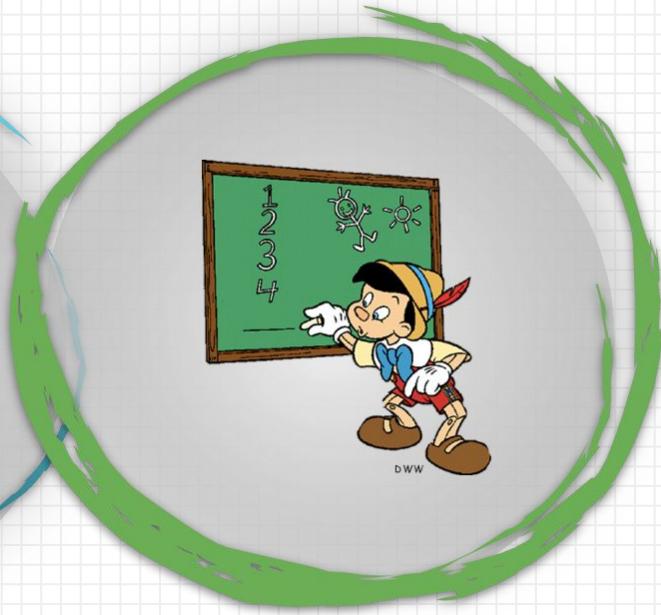
Succinct Proofs



Non-Interactive



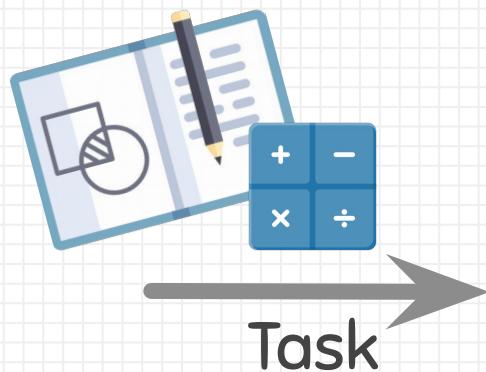
Knowledge Soundness



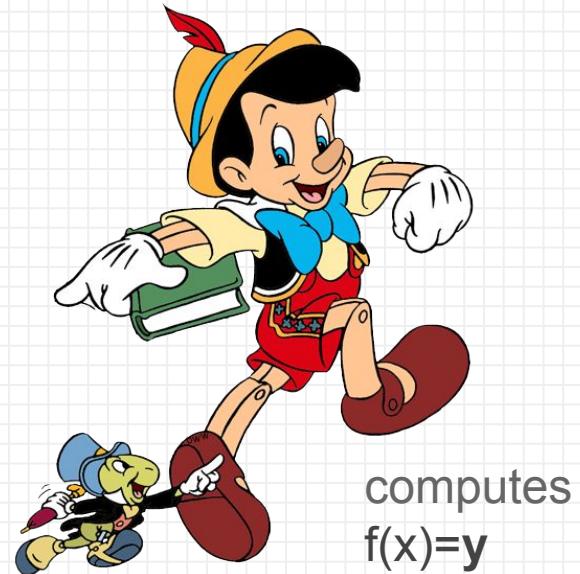
Usecase: Outsourcing Computation



Verifier



Task



computes
 $f(x)=y$

Prover

Prover claims a statement



Verifier

Claim
 $y = f(x)$



Prover

Verifier is able to check



Verify 

$$y' \neq f(x)$$





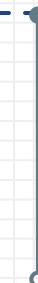
SNARKs

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



[GGPR13]
QSP and succinct NIZKs
without PCPs

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

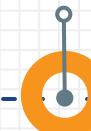


[PHGR13]
Pinocchio: Nearly practical
verifiable computation

[BCI+13] SNARGs via
linear interactive proofs



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



J. Groth

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

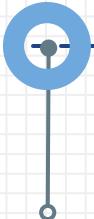




Assumptions



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



[GGPR13]
QSP and succinct NIZKs
without PCPs

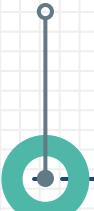
B. Parno,
J. Howell, C. Gentry,
M. Raykova



[PHGR13]
Pinocchio: Nearly practical
verifiable computation



[BCI+13] SNARGs via
linear interactive proofs



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

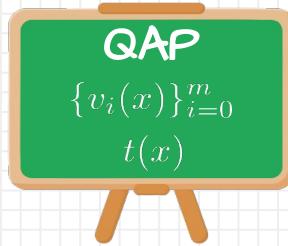


[Groth16] On the Size of
Pairing-based
Non-interactive Arguments



J. Groth

SNARK: Methodology



Target Statement
 $R(y,w)=1$

Computational Model
(Representation)

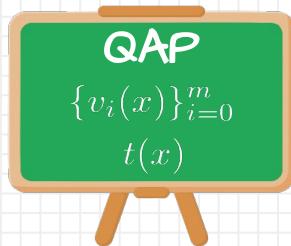
Encodings Secure
under
Knowledge Assumptions

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

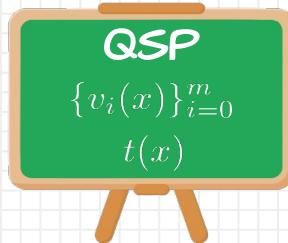
Arithmetic Circuit SAT

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

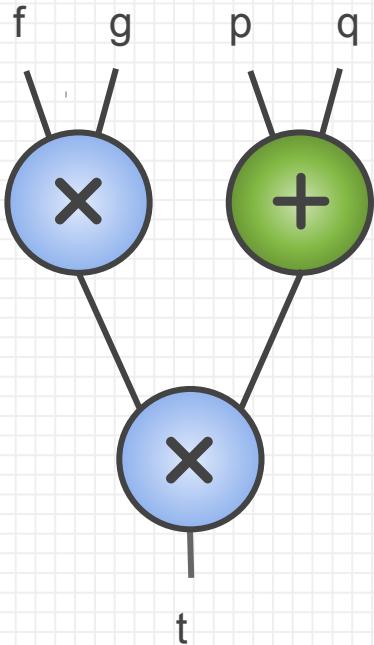
PKE Power Knowledge of Exponent
GGM Generic Group Model

Boolean Circuit SAT

QSP / SSP
over **Field** = Z_p

PKE: Power Knowledge of Exponent

Circuit over Field



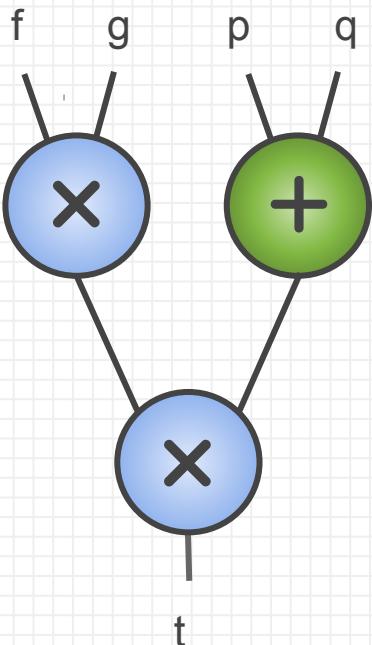
$p = 254$ -bit prime



Circuit over Field

vs

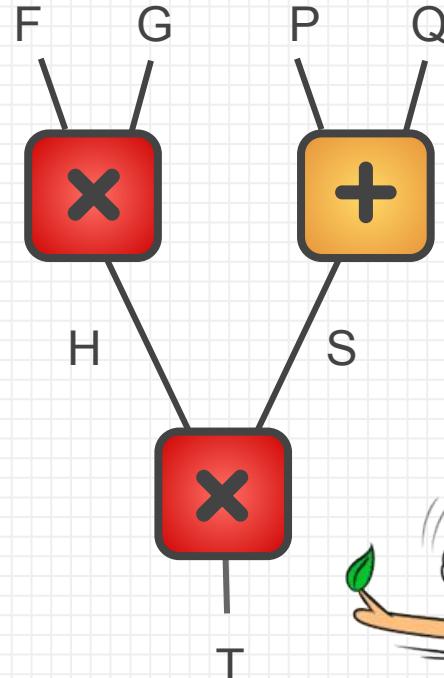
Circuit over Ring



$$\mathbb{Z}_p$$



$p = 254$ -bit prime



$$\mathbb{Z}_{2^k}$$

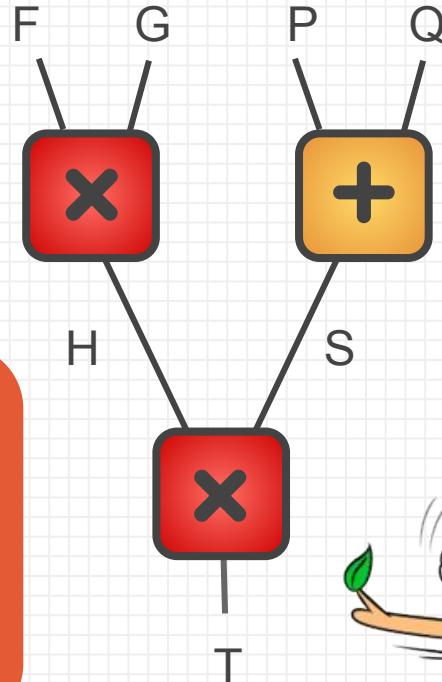
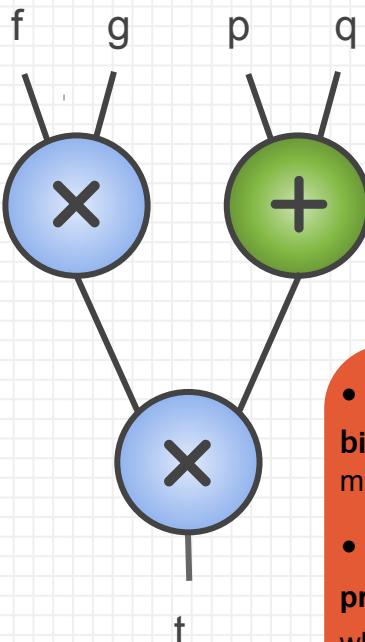
 RSA



Circuit over Field

vs

Circuit over Ring



Z_p



$p = 254\text{-bit prime}$

- modular reduction $x \bmod 2^k$
- **bit decomposition cost** = $m+1 \otimes$ where $m = \lceil \log(x_{\max}) \rceil$
- RSA groups are larger than p
- **product of ring elements** = $O(m^{1.58}) \otimes$ where ring element A = array of m words of 254 bits each

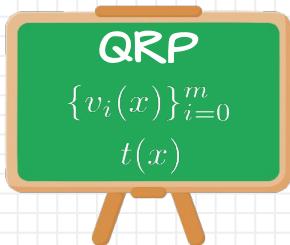
Z_{2^k}
RSA



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

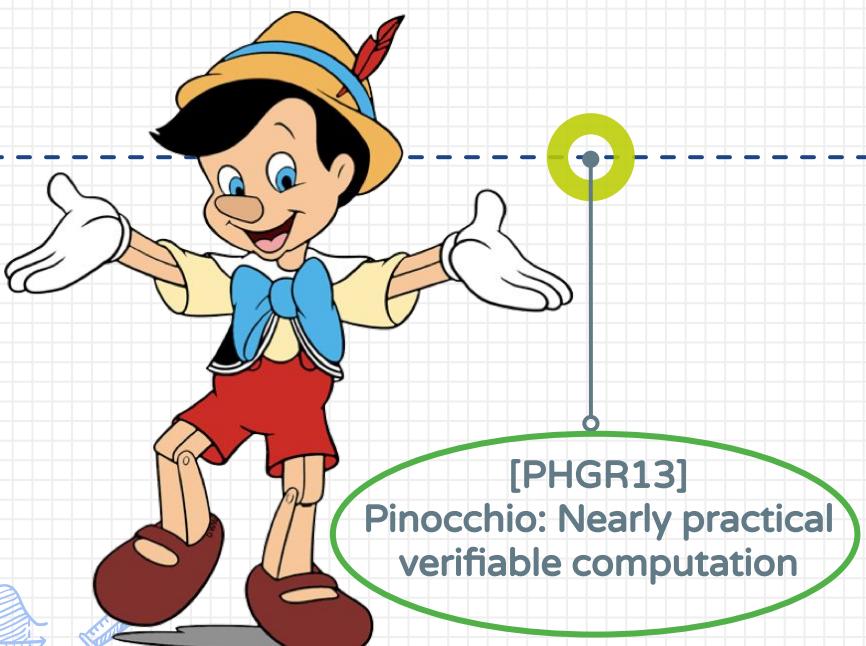
PKE: Power Knowledge of Exponent

General Circuits over Rings

QRP over Ring = R

Augmented PKE: Power Knowledge of Encoding

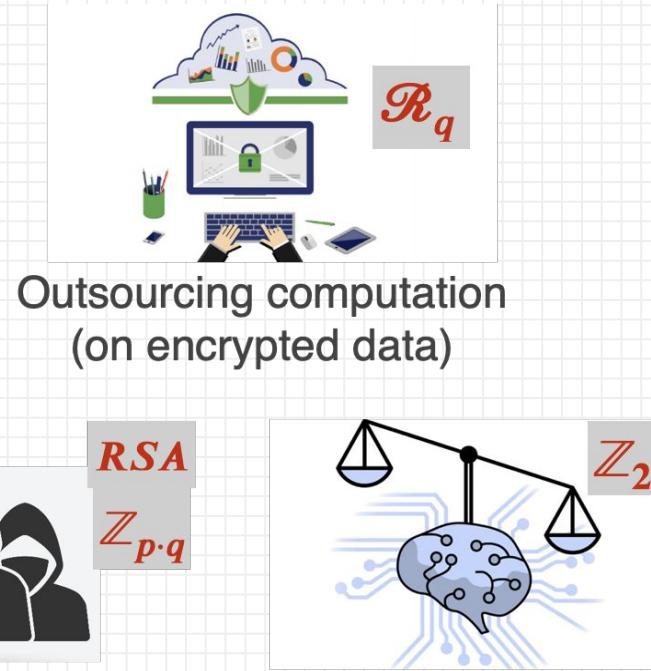
Contribution for short



More SNARKs applications



Anonymous Credentials



RSA
 $\mathbb{Z}_{p \cdot q}$



Fairness in
Machine Learning

More SNARKs applications



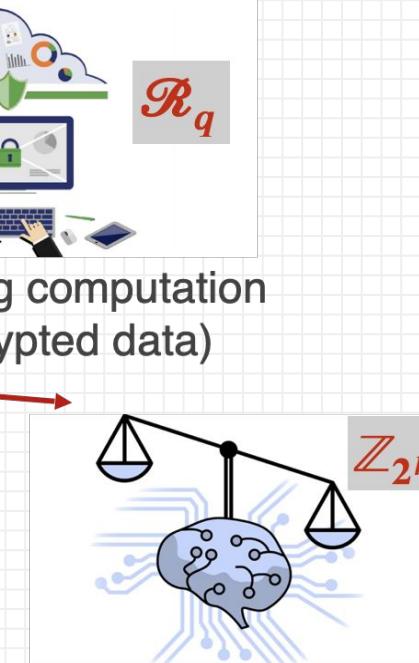
Rinocchio

Anonymous Credentials

Outsourcing computation
(on encrypted data)



RSA
 $\mathbb{Z}_{p \cdot q}$



Fairness in
Machine Learning

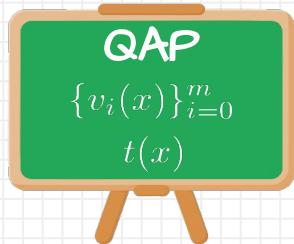
Key Steps to Build SNARKs

SNARK
Background

Framework
for Rinocchio

Challenges
New tools

Conclusion

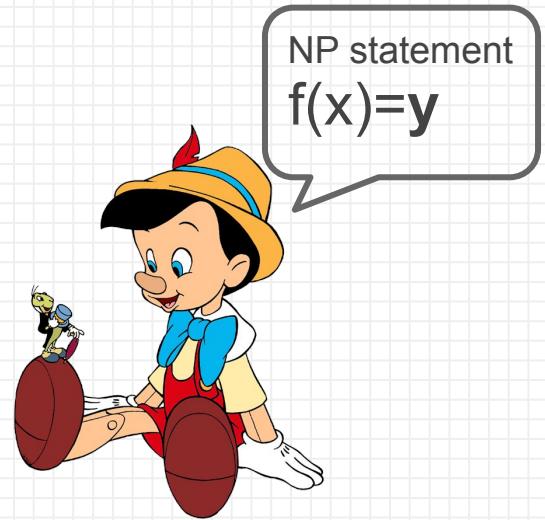


Proving NP statements



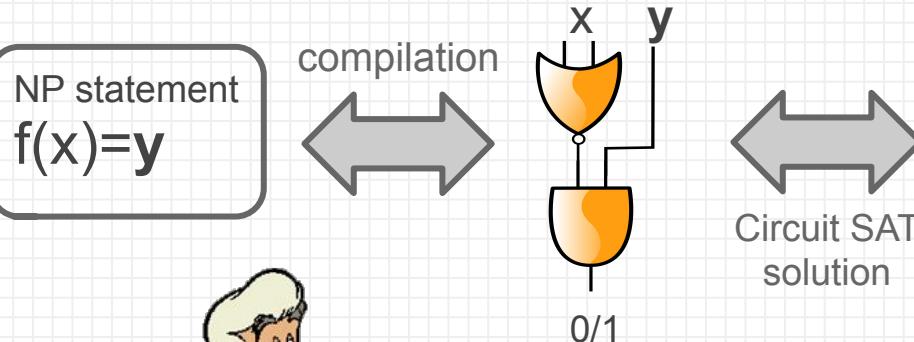
Verifier

DWW



Prover

Verifier solves equivalent problem instead



Verifier



Prover

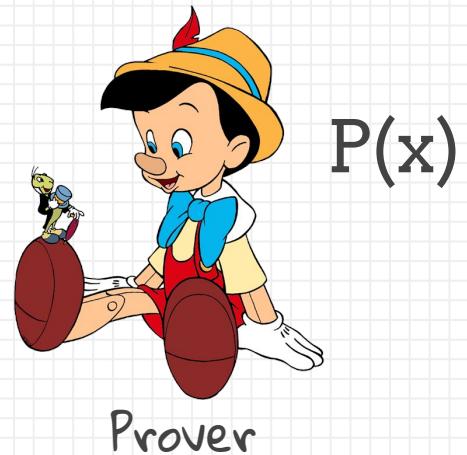
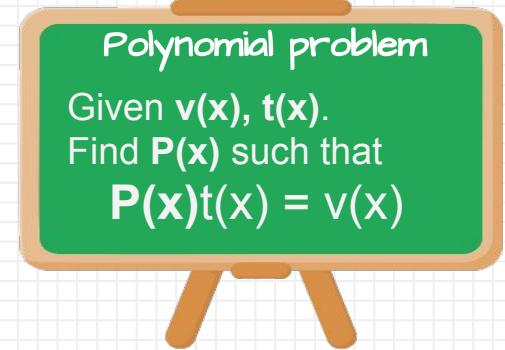
NP statement
 $f(x) = y$

Evaluate solution at point s



Verifier

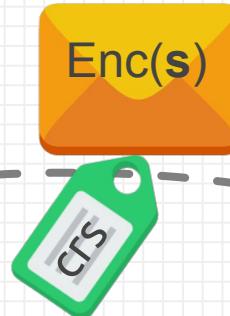
$$\pi = P(s)$$



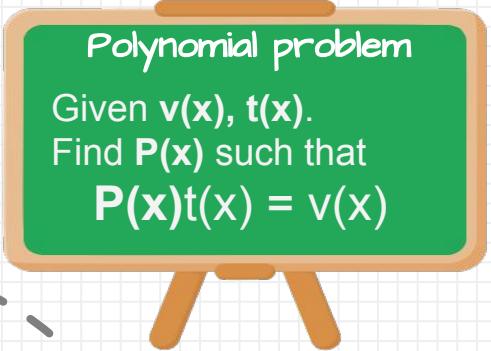
Evaluate solution at point s



Verifier

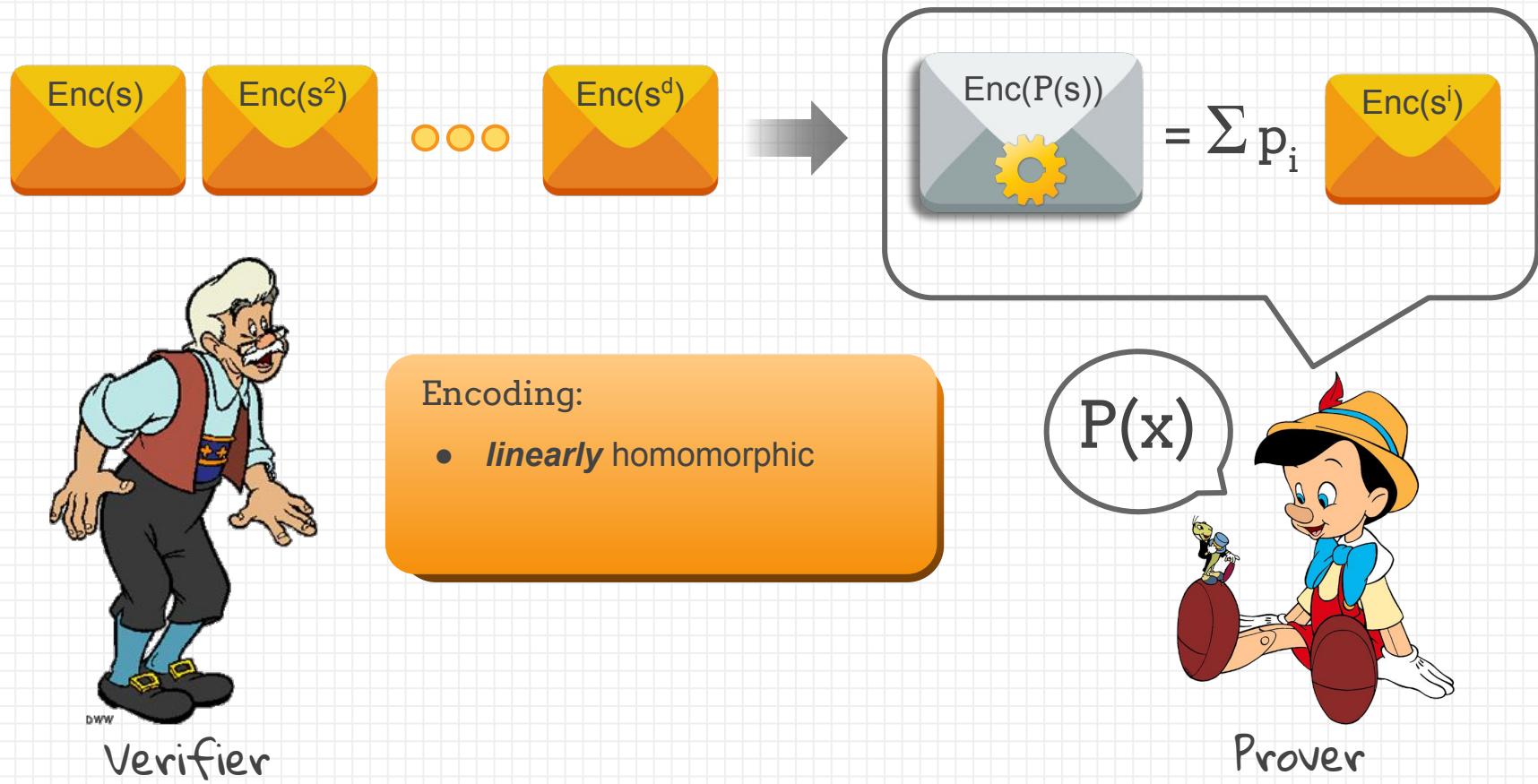


$$\pi = P(s)$$

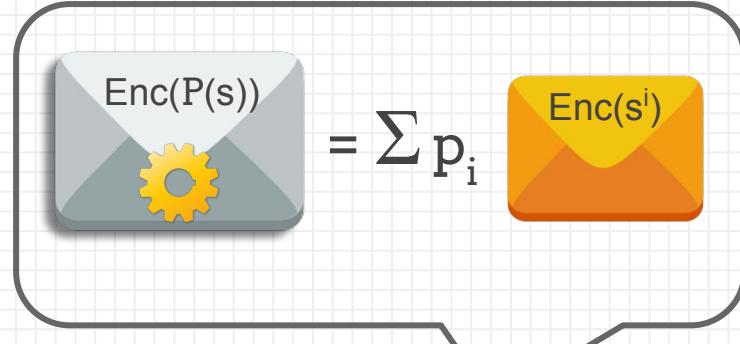
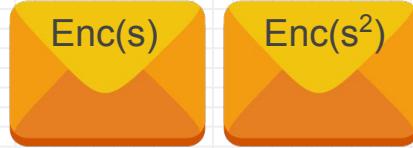


$P(x)$

General SNARK framework



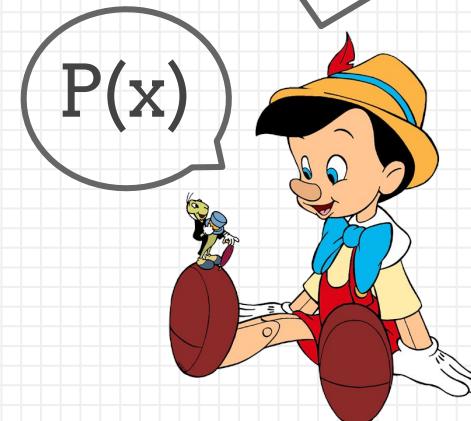
General SNARK framework



Verifier

Encoding:

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



Prover

Main Ingredients for Rinocchio



QRP

Representation
for Ring Circuits

Encodings

\mathbb{R}

over Rings



SNARKs for Ring Arithmetics



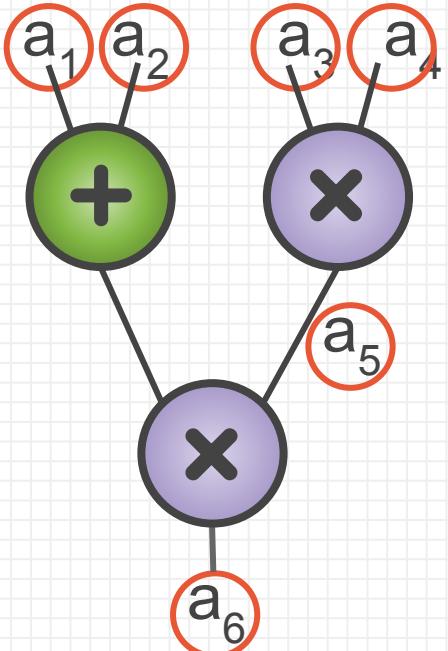
Generic framework for (zk)-SNARK over Ring

- ✖ Circuit-SAT for arithmetic circuits over commutative rings:
Quadratic Ring Programs (QRP)
- ✖ Suitable secure encodings over Rings

Properties

- ✖ Fits FHE schemes arithmetics
- ✖ Designated Verifier only

Quadratic Arithmetic Program



QAP

Given

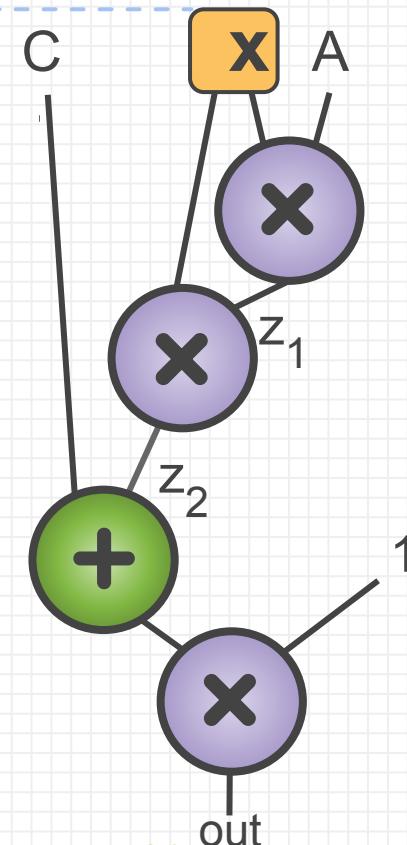
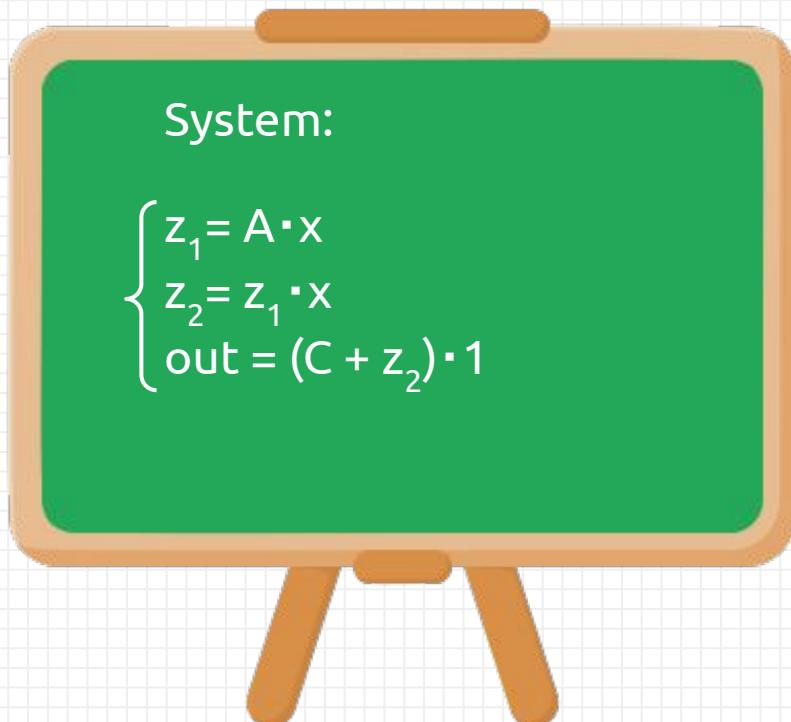
$$\{v_i(x)\}_i, \{w_i(x)\}_i,$$
$$\{y_i(x)\}_i, t(x)$$

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

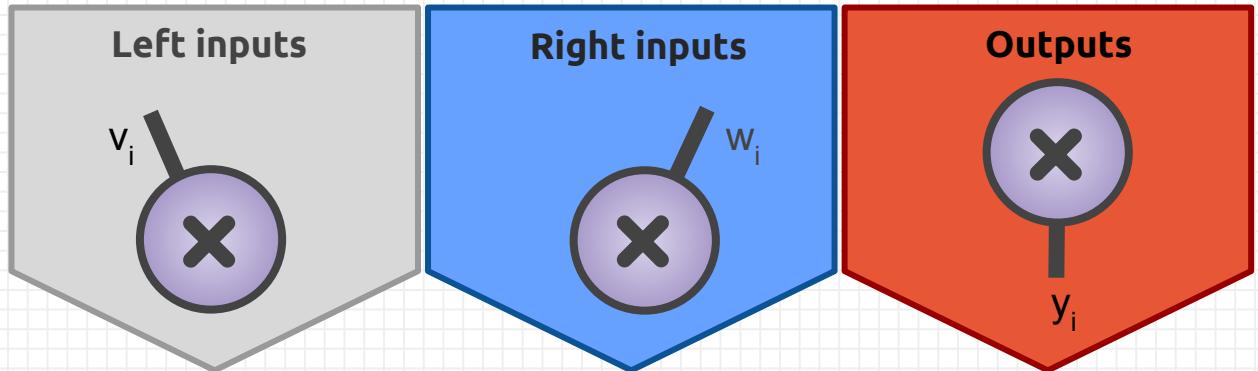
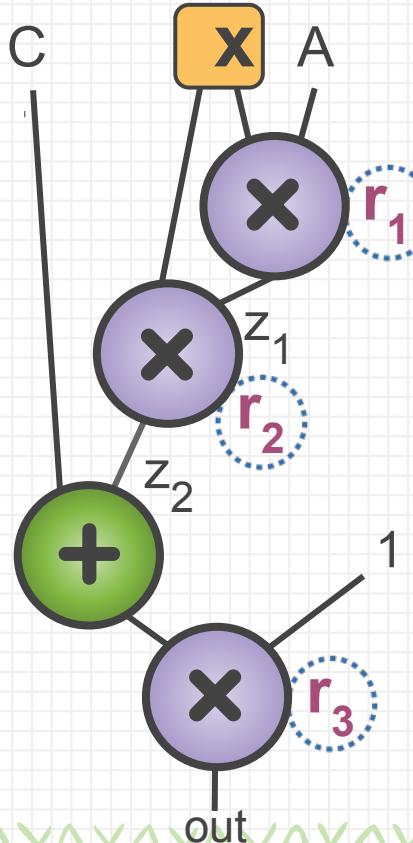
$$V(x) = \sum_{i=0}^m a_i v_i(x) \dots$$

$$\text{and } t(x)h(x) = V(x)W(x) - Y(x)$$

Example: Solution for equation $\mathbf{A}x^2 + \mathbf{C} = 0$

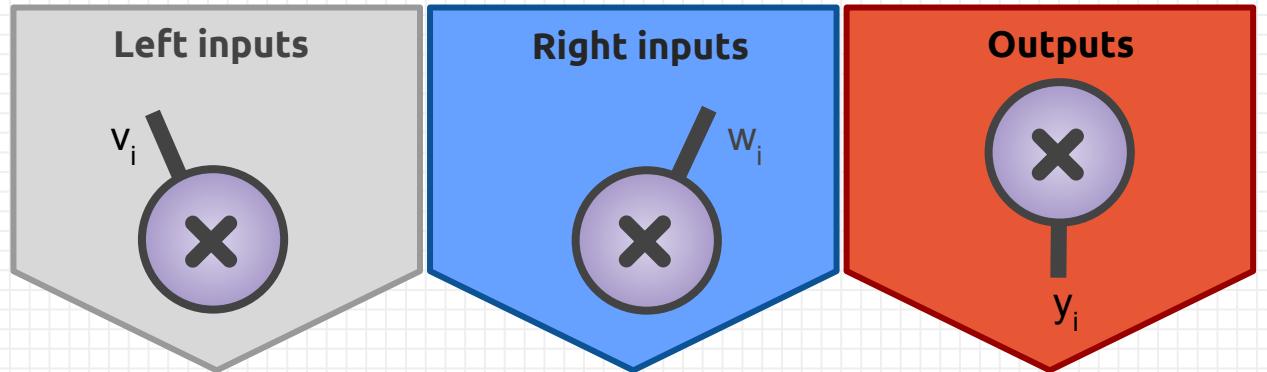
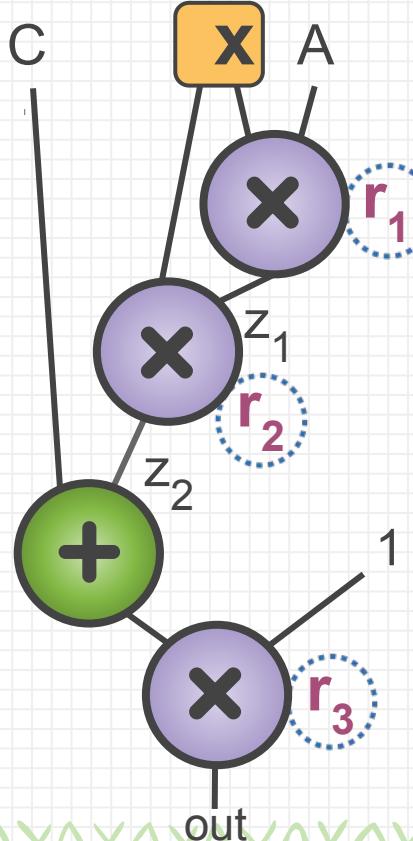


Indexes for wires: A=(0), C=(1), x=(2), z₁=(3), z₂=(4), out = (5)



$v_i(r_1) = 1$ 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), z₁=(3), z₂=(4), out = (5)



$$v_i(r_1) = 1 \text{ for } i=2$$

$$v_i(r_2) = 1 \text{ for } i=2$$

$$v_i(r_3) = 1 \text{ for } i=1,4$$

$$w_i(r_1) = 1 \text{ for } i=0$$

$$w_i(r_2) = 1 \text{ for } i=3$$

$$w_i(r_3) = 0 \text{ for all } i$$

$$y_i(r_1) = 1 \text{ for } i=3,$$

$$y_i(r_2) = 1 \text{ for } i=4$$

$$w_i(r_3) = 1 \text{ for } i=5$$

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

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

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

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

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

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

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

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

$$\left(\begin{array}{c|c} V & a \end{array} \right) \circ \left(\begin{array}{c|c} W & a \end{array} \right) = \left(\begin{array}{c|c} Y & a \end{array} \right)$$

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

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

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

$$\left[\begin{array}{c|c} \mathbf{V} & \mathbf{a} \\ \hline \mathbf{d} & \end{array} \right] \circ \left[\begin{array}{c|c} \mathbf{W} & \mathbf{a} \\ \hline \mathbf{d} & \end{array} \right] = \left[\begin{array}{c|c} \mathbf{Y} & \mathbf{a} \\ \hline \mathbf{d} & \end{array} \right]$$

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

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

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

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

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

$$\left[\begin{array}{c|c} \mathbf{V} & \mathbf{a} \\ \hline \mathbf{d} & \end{array} \right] \circ \left[\begin{array}{c|c} \mathbf{W} & \mathbf{a} \\ \hline \mathbf{d} & \end{array} \right] = \left[\begin{array}{c|c} \mathbf{Y} & \mathbf{a} \\ \hline \mathbf{d} & \end{array} \right]$$

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

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

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

Schwartz-Zippel Lemma over Fields

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

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

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

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}

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



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

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

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

$$\begin{aligned}\langle g \rangle &= \mathbb{G}, \langle \tilde{g} \rangle = \tilde{\mathbb{G}} \\ Enc(s) &= g^s \quad e : \mathbb{G} \times \mathbb{G} \rightarrow \tilde{\mathbb{G}} \\ e(g^a, g^b) &= \tilde{g}^{ab}\end{aligned}$$

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

$$\begin{aligned}\langle g \rangle &= \mathbb{G}, \langle \tilde{g} \rangle = \tilde{\mathbb{G}} \\ Enc(s) &= g^s \quad e : \mathbb{G} \times \mathbb{G} \rightarrow \tilde{\mathbb{G}} \\ e(g^a, g^b) &= \tilde{g}^{ab}\end{aligned}$$

Quadratic root detection public

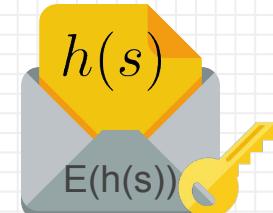
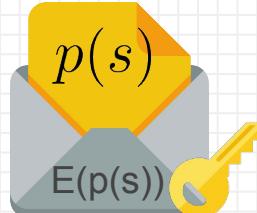
$$t(s)h(s) \stackrel{?}{=} p(s)$$

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

$$\begin{aligned}\text{Encode:} \quad E_{pk}(m) &= c \\ \text{Decode:} \quad D_{sk}(c) &= m\end{aligned}$$

Quadratic root detection needs **sk**

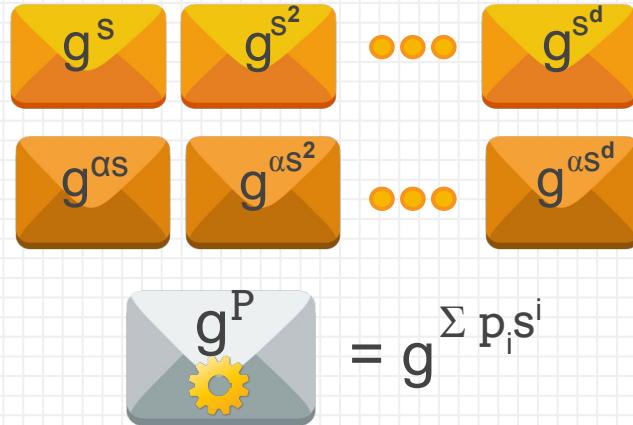
$$t(s)h(s) \stackrel{?}{=} p(s)$$



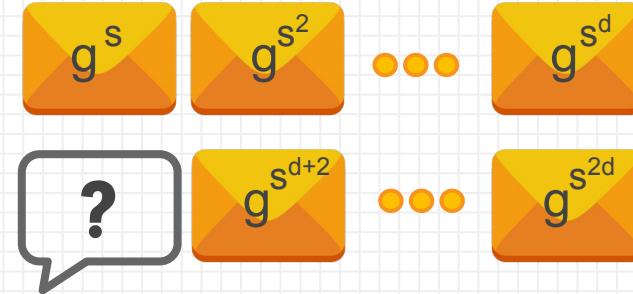
Assumptions on Discrete Log Encoding for Fields



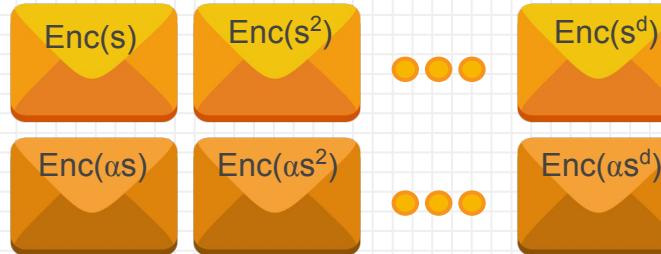
d-PKE



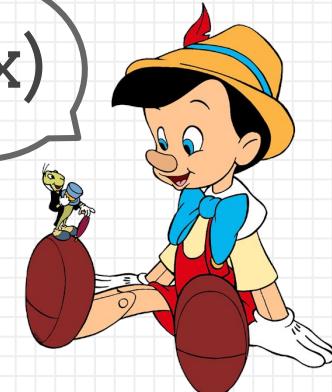
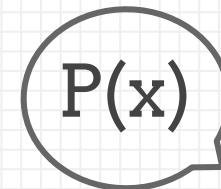
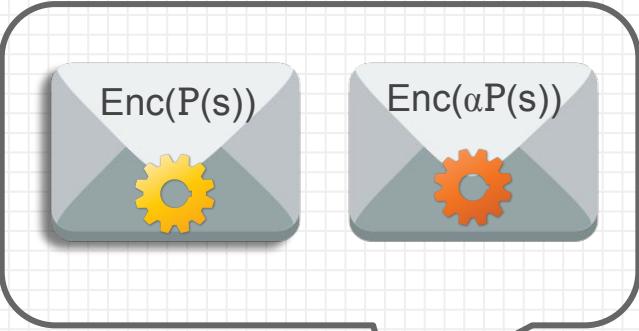
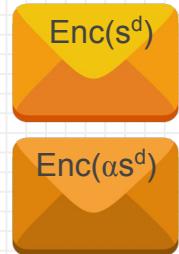
d-PDH



Preprocessing with α : Double the Proof



⋮
⋮



Technical Details

SNARK
Background

Framework
for Rinocchio

Challenges
New tools

Conclusion





NP Representation QRP

COMPUTATIONAL MODEL
FOR RING CIRCUITS

Proving a Solution for Equation $\mathbf{Ax}^2 + \mathbf{C} = 0$

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

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

$$\begin{matrix} m \\ d \end{matrix} \begin{pmatrix} V & | & a \end{pmatrix} \circ \begin{pmatrix} W & | & a \end{pmatrix} = \begin{pmatrix} Y & | & a \end{pmatrix}$$

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

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

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

Polynomial Equation with Coefficients in a Ring

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

Necessary property over Rings for **Ideals** $I_j = (x - r_j)$

Isomorphism for QRP soundness ifff ideals are co-prime:

$$\frac{R[x]}{(t(x))} \simeq \frac{R[x]}{I_1} \times \dots \times \frac{R[x]}{I_d} \simeq R \times \dots \times R$$
$$p(x) \quad \longmapsto \quad (p_1(x), \dots, p_d(x)) \quad \longmapsto (p(g_1), \dots, p(g_d))$$

Polynomial Equation with Coefficients in a Ring

Works for $R = \mathbb{F}$, as then
the ideals I_j are co-prime.

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

Necessary property over Rings for **Ideals** $I_j = (x - r_j)$

$$\frac{R[x]}{(t(x))} \simeq \frac{R[x]}{I_1} \times \dots \times \frac{R[x]}{I_d} \simeq R \times \dots \times R$$
$$p(x) \quad \mapsto \quad (p_1(x), \dots, p_d(x)) \quad \mapsto (p(g_1), \dots, p(g_d))$$

Exceptional Sets: to the rescue!

Def: Let R be a commutative ring. A set $A = \{g_1, \dots, g_n\} \subset R$ is **exceptional** iff:

$$\forall i \neq j, (g_i - g_j) \in R^*$$

Exceptional sets have **no further algebraic structure**.
Not even closure!

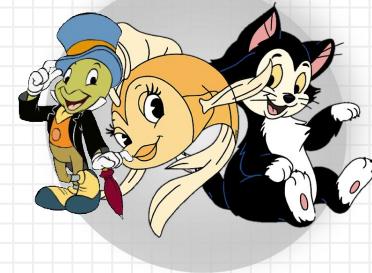


Exceptional Sets

Def: Let R be a commutative ring. A set $\mathbf{A} = \{g_1, \dots, g_n\} \subset R$ is **exceptional** iff:

$$\forall i \neq j, (g_i - g_j) \in R^*$$

Exceptional sets have **no further algebraic structure**.
Not even closure!



Given exceptional set \mathbf{A} , the **ideals** $I_j = (x - g_j)$ are **pairwise co-prime** (i.e. $\forall i \neq j, I_i + I_j = R[X]$).

- Proof: $-(x - g_i) + (x - g_j) = (g_i - g_j) \in R^*$
- Meaning: We can apply CRT in $R[X]$, for big enough $\mathbf{A} \subset R$.

Exceptional Sets

Def: Let R be a commutative ring. A set $\mathbf{A} = \{g_1, \dots, g_n\} \subset R$ is **exceptional** iff:

$$\forall i \neq j, (g_i - g_j) \in R^*$$

Exceptional sets have **no further algebraic structure**.
Not even closure!

Given exceptional set \mathbf{A} , the **ideals** $I_j = (x - g_j)$ are **pairwise co-prime** (i.e. $\forall i \neq j, I_i + I_j = R[X]$).



$$\begin{aligned}\frac{R[x]}{(t(x))} &\simeq \frac{R[x]}{I_1} \times \dots \times \frac{R[x]}{I_d} \simeq R \times \dots \times R \\ p(x) &\mapsto (p_1(x), \dots, p_d(x)) \mapsto (p(g_1), \dots, p(g_d))\end{aligned}$$

Exceptional Sets

Def: Let R be a commutative ring. A set $\mathbf{A} = \{g_1, \dots, g_n\} \subset R$ is **exceptional** iff:

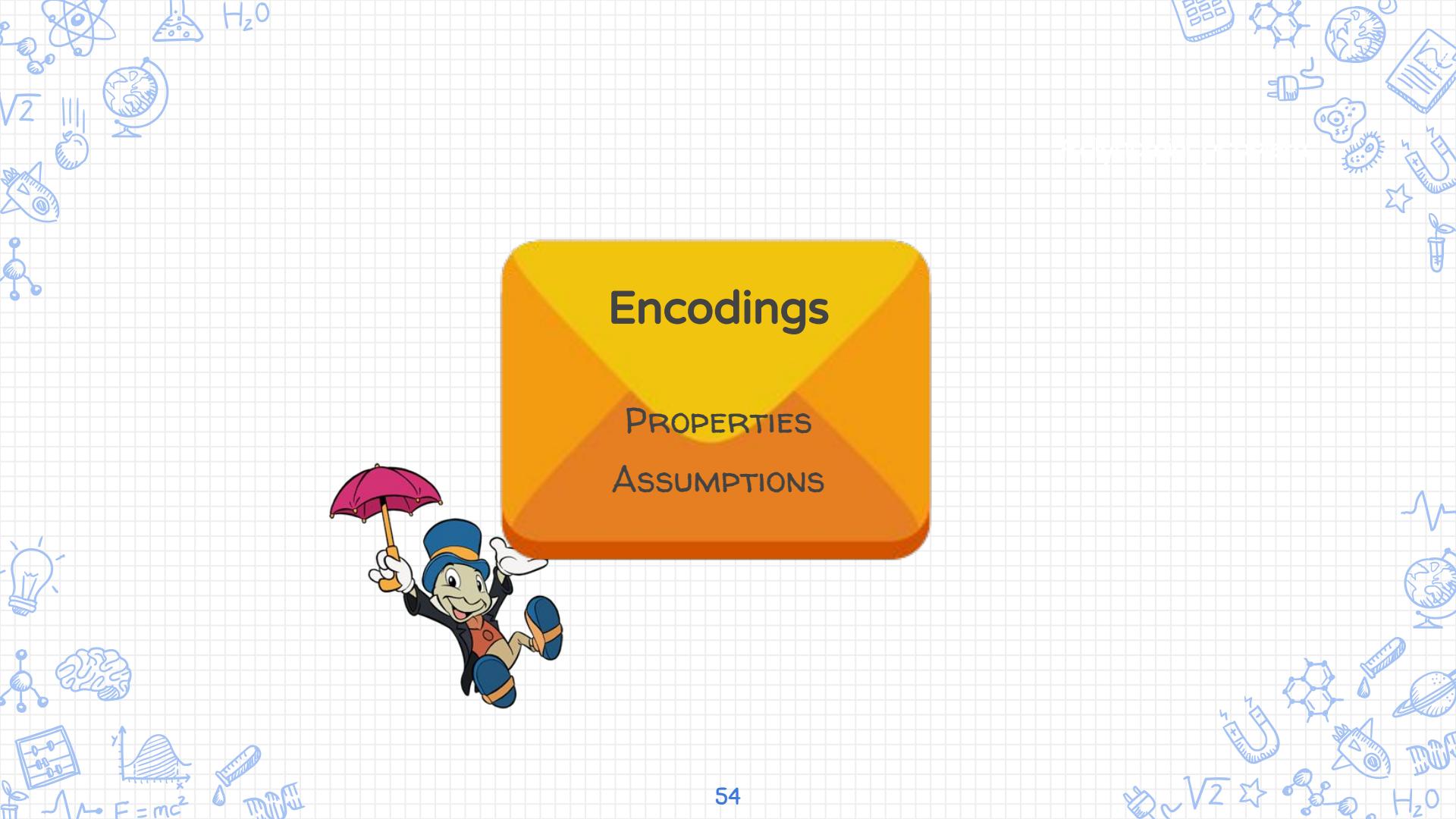
$$\forall i \neq j, (g_i - g_j) \in R^*$$

Exceptional sets have **no further algebraic structure**.
Not even closure!

Given exceptional set \mathbf{A} , the **ideals** $I_j = (x - g_j)$ are **pairwise co-prime** (i.e. $\forall i \neq j, I_i + I_j = R[X]$).



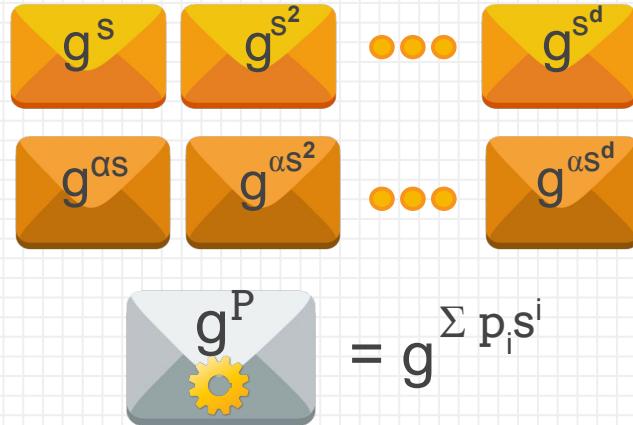
$$\begin{aligned}\frac{R[x]}{(t(x))} &\simeq \frac{R[x]}{I_1} \times \dots \times \frac{R[x]}{I_d} \simeq R \times \dots \times R \\ p(x) &\mapsto (p_1(x), \dots, p_d(x)) \mapsto (p(g_1), \dots, p(g_d))\end{aligned}$$



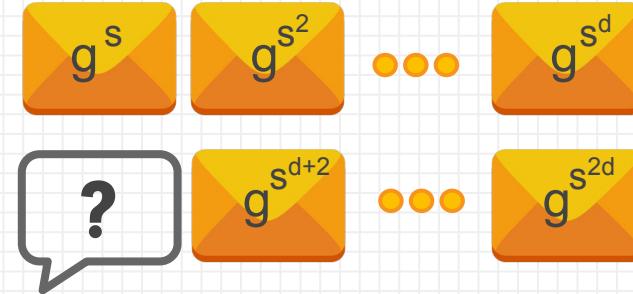
Assumptions on Discrete Log Encoding for Fields



d-PKE

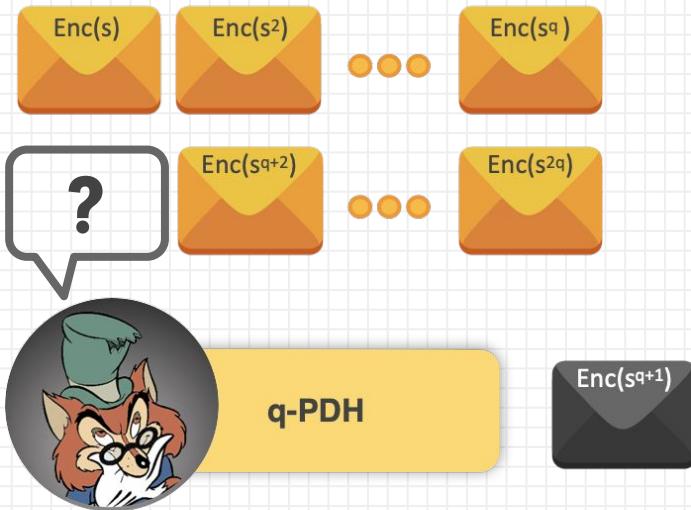


d-PDH

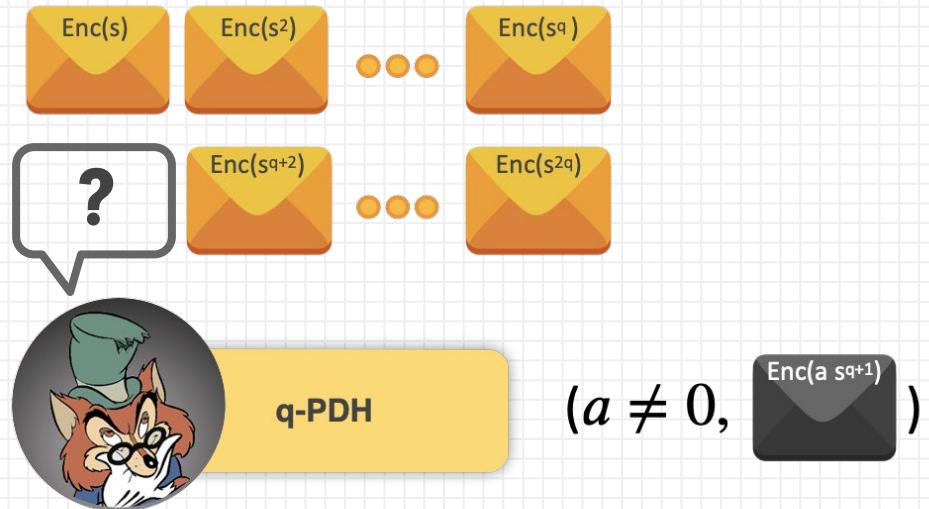


Ring Augmented PDH Assumption

$s \leftarrow \mathbb{F}^*$



$s \leftarrow A^*$, where A is an exceptional set such that $A^* \subset R^*$.



Encoding Instantiation for Ring \mathbb{Z}_{2^k}

Problem: There's no exceptional set A of size > 2 in \mathbb{Z}_{2^k} .

Sol: Move to $GR(2^k, d)$, the degree- d Galois extension of \mathbb{Z}_{2^k} .

$$a = a_0 + a_1X + \dots + a_{d-1}X^{d-1}; \quad a_i \in \mathbb{Z}_{2^k}$$



$$A = \{a_0 + a_1X + \dots + a_{d-1}X^{d-1} : a_i \in \{0,1\}\} \subset GR(2^k, d)$$

Encoding: d copies of a \mathbb{Z}_{2^k} -linearly homomorphic scheme, such as [JoyeLibert13].

$$E(a) = (Enc_{pk_1}(a_0), \dots, Enc_{pk_d}(a_{d-1})); \quad a_i \in \mathbb{Z}_{2^k}$$

Encoding Instantiation for Ring $\mathbb{R}_q = \mathbb{Z}_q[x]/R(x)$

Rings of the form $\mathcal{R}_q = \mathbb{Z}_q[X]/(h(X))$.



$$= \{0, \dots, p_1 - 1\} \subset \mathcal{R}_q; \quad q = \prod p_i \text{ s.t. } p_1 < p_2 < \dots$$

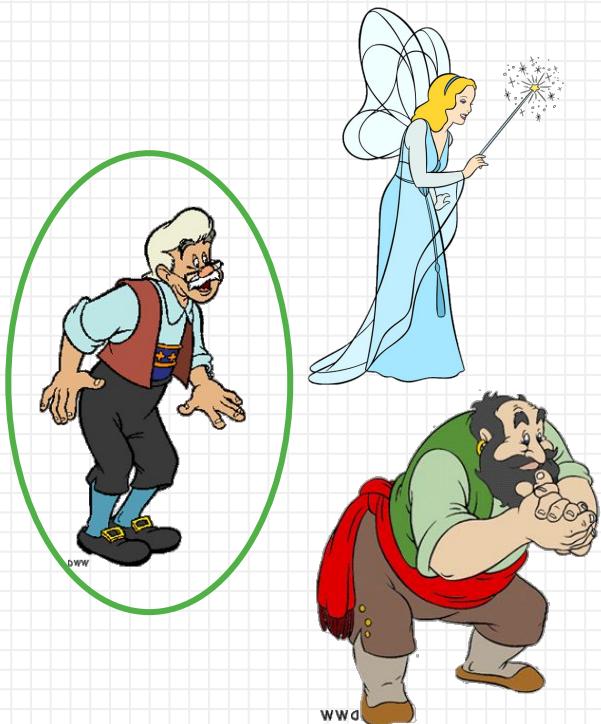
Compared with [FioreNitulescuPointcheval20]:

[FVP20] Boosting Verifiable Computation on Encrypted Data

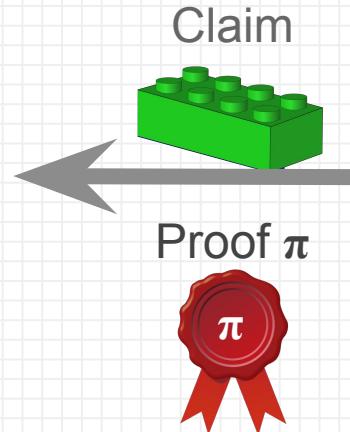
Dario Fiore, Anca Nitulescu, David Pointcheval

- ✗ Significantly better choices for RLWE parameters
- ✗ FVP20 only supports rings with q prime → very inefficient FHE
- ✗ We enable operations for plaintext packing, modulo switching
- ✗ We are only designated-verifier

Designated-Verifier Only

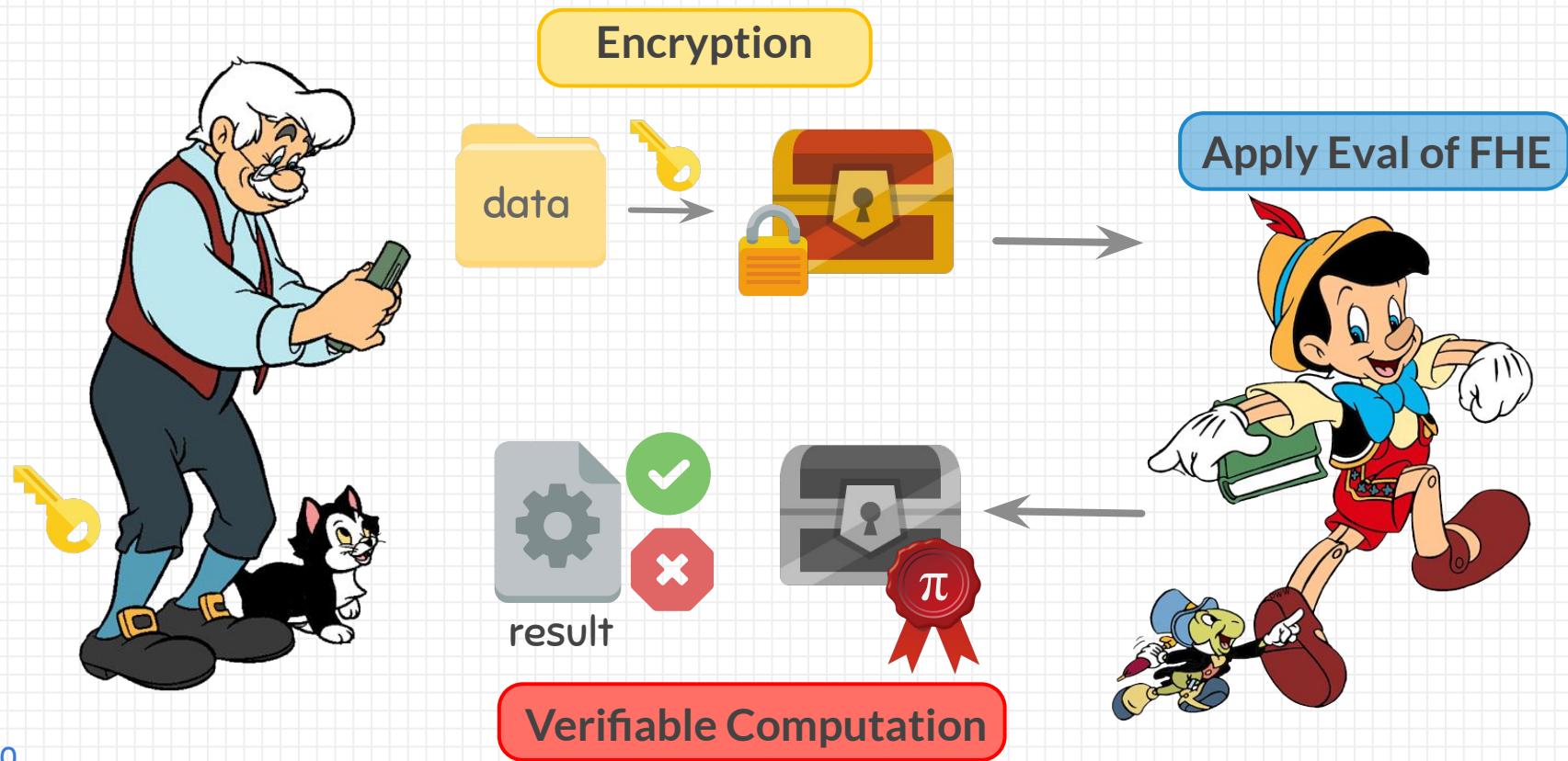


Verifiers



Prover

Application: Verifiable Computation on Encrypted Data



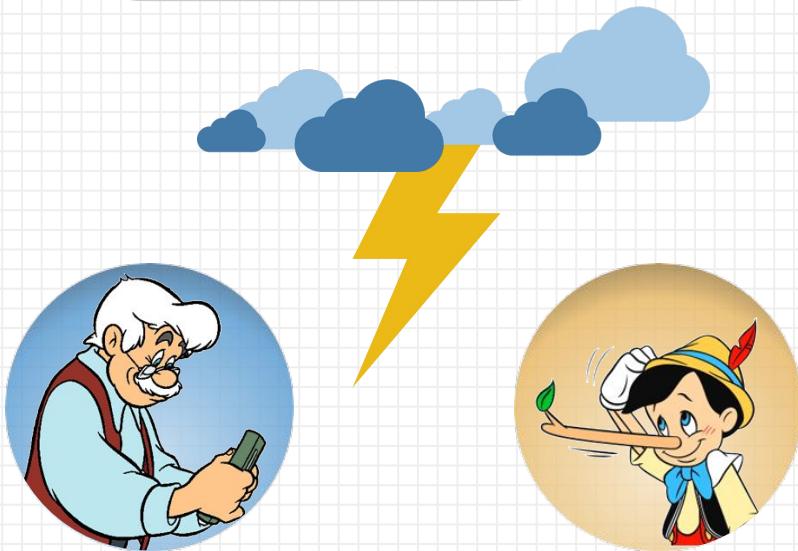
Conclusions

SNARK
Background

Framework
for Rinocchio

Challenges
New tools

Conclusion



Conclusions

Quadratic Programs and SNARKs over fields

- ✗ Lots of implementations, but they fall short in one aspect
- ✗ Emulating ring arithmetic on SNARKs is expensive and unfriendly to applications
- ✗ Today's cost: Compilation to circuits over fields, costly preprocessing

Rinocchio: SNARKs for Ring Arithmetics

- ✗ Circuit-SAT for arithmetic circuits over commutative rings: Quadratic Ring Programs (QRP)
- ✗ Better fits FHE schemes arithmetics $\mathbb{R}_q = \mathbb{Z}_q[x]/R(x)$ even for q not prime
- ✗ Supports sub-circuits for special operations in FHE: modulo switching
- ✗ Designated Verifier only
- ✗ Can be turned Zero-Knowledge using Context Hiding techniques

Conclusions

Quadratic Programs and SNARKs over fields

- ✗ Lots of implementations, but they fall short in one aspect
- ✗ Emulating ring arithmetic on SNARKs is expensive and unfriendly to applications
- ✗ Today's cost: Compilation to circuits over fields, costly preprocessing

Rinocchio: SNARKs for Ring Arithmetics

- ✗ Circuit-SAT for arithmetic circuits over commutative rings: Quadratic Ring Programs (QRP)
- ✗ Better fits FHE schemes arithmetics $\mathbb{R}_q = \mathbb{Z}_q[x]/R(x)$ even for q not prime
- ✗ Supports sub-circuits for special operations in FHE: modulo switching
- ✗ Designated Verifier only
- ✗ Can be turned Zero-Knowledge using Context Hiding techniques

Open Questions

- ✗ Non-commutative rings (matrices). Application: Machine Learning
- ✗ Other Encodings over rings → publicly verifiable

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](#)