The Application of Zero-Knowledge Proofs: Proof of Solvency

Youwei (Barry) Deng, Jeremy Clark

April 27, 2025

Concordia University

Table of Contents

What is ZKP

Goal: Prove a statement is true without revealing additional information, only the fact itself.

What is ZKP

Goal: Prove a statement is true without revealing additional information, only the fact itself.

- Compared to classic proofs
  - **Classic proofs** (pythagorean theorem, Euler's theorem, etc.): verifier passively reads the proof
  - **ZKP**: two new ingredients, interaction and randomness

What is ZKP

Goal: Prove a statement is true without revealing additional information, only the fact itself.

- Compared to classic proofs
  - **Classic proofs** (pythagorean theorem, Euler's theorem, etc.): verifier passively reads the proof
  - **ZKP**: two new ingredients, interaction and randomness

$$
\begin{array}{ccc}
\mathcal{P} & \xrightarrow{\quad m_1 \quad} & \mathcal{V} \\
 & \xleftarrow{\quad c_1 \xleftarrow{\$} \{0,1\}^t \quad} & \\
 & \xrightarrow{\quad m_2 \quad} & \\
 & \xleftarrow{\quad c_2 \xleftarrow{\$} \{0,1\}^t \quad} & \\
 & \cdots &
\end{array}
$$

interactive + probabilistic proof system

Zk-SNARKs

Succinctness: small proof size and short verifier time

Non-interactive: Fiat-Shamir transformation

Zk-SNARKs

Succinctness: small proof size and short verifier time

Non-interactive: Fiat-Shamir transformation

Building a zk-SNARK

commitment scheme + interactive oracle proof $\Rightarrow$ zk-SNARK

Zk-SNARKs

Succinctness: small proof size and short verifier time

Non-interactive: Fiat-Shamir transformation

Building a zk-SNARK

commitment scheme + interactive oracle proof $\Rightarrow$ zk-SNARK

- Commitment scheme: commit to a value and later reveal it
  - Binding: cannot open one commitment to two different values
  - Hiding: cannot learn the committed value from the commitment
- Random oracle: a black box that randomly uniformly outputs the same bits with the same input and different bits with different input

Important Commitment Schemes

- Polynomial commitments: univariate polynomial $f(X)$
- Multilinear commitments: multivariate polynomial $f(X_1, X_2, \ldots, X_n)$
- Vector commitments: vector $\overrightarrow{v} = (v_1, v_2, \ldots, v_n)$
- ...

Polynomial Commitment Scheme (PCS)

Given srs $= \{g, g^{\tau}, g^{\tau^2}, g^{\tau^3}, \ldots\}$, the PCS allows $\mathcal{P}$ to commit to a polynomial $f(X)$; denote the commitment to $f(X)$ by $\mathcal{C}_f$

$$\mathcal{C}_f = g^{f_0 + f_1\tau + f_2\tau^2 + f_3\tau^3 + f_4\tau^4 \cdots}$$

where f_i is the coefficient of X^i in $f(X)$.

Polynomial Commitment Scheme (PCS)

Given srs $= \{g, g^{\tau}, g^{\tau^2}, g^{\tau^3}, \dots\}$, the PCS allows $\mathcal{P}$ to commit to a polynomial $f(X)$; denote the commitment to $f(X)$ by $\mathcal{C}_f$

$$\mathcal{C}_f = g^{f_0 + f_1\tau + f_2\tau^2 + f_3\tau^3 + f_4\tau^4 \cdots}$$

where f_i is the coefficient of X^i in $f(X)$.

Example

Prove the evaluation of $f(a)$ is b

Polynomial Commitment Scheme (PCS)

Given srs $= \{g, g^{\tau}, g^{\tau^2}, g^{\tau^3}, \dots\}$, the PCS allows $\mathcal{P}$ to commit to a polynomial $f(X)$; denote the commitment to $f(X)$ by $\mathcal{C}_f$

$$\mathcal{C}_f = g^{f_0 + f_1\tau + f_2\tau^2 + f_3\tau^3 + f_4\tau^4 \cdots}$$

where f_i is the coefficient of X^i in $f(X)$.

Example

Prove the evaluation of $f(a)$ is b

A useful observation:

$$f(a) = b \iff q(X) \text{ exists s.t. } q(X) = \frac{f(X) - b}{X - a}$$

Application of PCS

$\mathcal{P}$		$\mathcal{V}$
$q(X) = \dfrac{f(X) - b}{X - a}$		
commit(f, q)		
	$\xrightarrow{\mathcal{C}_f, \mathcal{C}_q}$	
		check
		$e(\mathcal{C}_f / g^b, [1]_2) \stackrel{?}{=} e(\mathcal{C}_q, [X - a]_2)$

Application of PCS

$\mathcal{P}$		$\mathcal{V}$
$q(X) = \dfrac{f(X) - b}{X - a}$		
commit(f, q)		
	$\xrightarrow{\mathcal{C}_f, \mathcal{C}_q}$	
		check
		$e(\mathcal{C}_f / g^b, [1]_2) \stackrel{?}{=} e(\mathcal{C}_q, [X - a]_2)$

Schwartz-Zippel Lemma

Given a polynomial $f \in \mathbb{F}_{\leq d}[X]$. Let S be a finite subset of $\mathbb{F}$ and $X \in S$ be chosen uniformly at random. Then

$$\Pr[f(X) = 0] \leq \frac{d}{|S|}$$

where d is the degree of f.

Polynomial Interactive Oracle Proof (Poly-IOP)

Polynomial Constraints: a set of polynomial relations among polynomials $f_1(X), f_2(X), \ldots, f_n(X)$.

Polynomial Interactive Oracle Proof (Poly-IOP)

Example

Prove $f(X) = g(X)$ when $X = a$

Polynomial Interactive Oracle Proof (Poly-IOP)

Example

Prove $f(X) = g(X)$ when $X = a$

$\mathcal{P}$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\mathcal{V}$

$q(X) = (f(X) - g(X))/(X - a)$

$\textbf{commit}(f, g, q)$

$$\xrightarrow{\mathcal{C}_f, \mathcal{C}_g, \mathcal{C}_q}$$

$$\xleftarrow{\gamma \xleftarrow{\$} \mathbb{F}_p}$$

$$\xrightarrow{\alpha_1 = f(\gamma), \alpha_2 = g(\gamma), \beta = q(\gamma)}$$

check

(i) $\alpha_1 - \alpha_2 \stackrel{?}{=} \beta \cdot (\gamma - a)$

(ii) $\alpha_1 \stackrel{?}{=} f(\gamma), \alpha_2 \stackrel{?}{=} g(\gamma)$,
$\beta \stackrel{?}{=} q(\gamma)$

Table of Contents

What is PoS

Proof of Solvency

How can a centralized exchange prove that it has enough assets to cover all the liabilities of its customers?

- Reveal the assets directly?
- Trust third-party auditors?

What is PoS

Proof of Solvency

How can a centralized exchange prove that it has enough assets to cover all the liabilities of its customers?

- Reveal the assets directly?
- Trust third-party auditors?

Numerous incidents

- Mt. Gox, 2014
- QuadrigaCX, 2019
- FTX, 2022
- ...

What is PoS

Proof of Solvency

How can a centralized exchange prove that it has enough assets to cover all the liabilities of its customers?

- Reveal the assets directly?
- Trust third-party auditors?

Numerous incidents

- Mt. Gox, 2014
- QuadrigaCX, 2019
- FTX, 2022
- ...

Proof of Solvency (PoS)

- Proof of Liabilities (PoL)
- Proof of Assets (PoA)

Limitations

PoS is not a silver bullet

- It may rely on a trusted setup (depending on the proof system)
- It requires honest verifier and user check
- It is not helpful to prevent hacks, scams, etc

Limitations

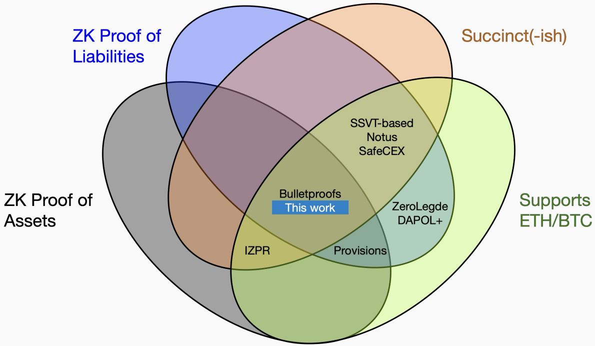
PoS is not a silver bullet

- It may rely on a trusted setup (depending on the proof system)
- It requires honest verifier and user check
- It is not helpful to prevent hacks, scams, etc

It raises the bar for exchanges to cheat

Comparison with Previous Works

- Most of the previous works focus on only one side
- This work aims for a full end-to-end solution



Proof of Liabilities

Recall the problem

- Each user's balance is included in the proof
- The summation of all balances (total liability) is correct
- Each balance is in a specified range (range proof)

Proof of Liabilities

Recall the problem

- Each user's balance is included in the proof
- The summation of all balances (total liability) is correct
- Each balance is in a specified range (range proof)

Intuition

- Interpolate a polynomial with balances

$$p(X): \quad \mathsf{bal}_1 \quad \mathsf{bal}_2 \quad \dots \quad \mathsf{bal}_{n-1} \quad \mathsf{bal}_n$$

Proof of Liabilities

Recall the problem

- Each user's balance is included in the proof
- The summation of all balances (total liability) is correct
- Each balance is in a specified range (range proof)

Intuition

- Interpolate a polynomial with balances
- Construct an accumulative polynomial

$$
\begin{array}{rccccc}
f_{\text{liab}}(X): & r_1 & r_2 & \dots & r_{n-1} & r_n \\
p(X): & \text{bal}_1 & \text{bal}_2 & \dots & \text{bal}_{n-1} & \text{bal}_n
\end{array}
$$

Proof of Liabilities

Recall the problem

- Each user's balance is included in the proof
- The summation of all balances (total liability) is correct
- Each balance is in a specified range (range proof)

Intuition

- Interpolate a polynomial with balances
- Construct an accumulative polynomial
- Open the evaluation point for each user

$$\begin{array}{rccccc} f_{\mathrm{liab}}(X): & r_1 & r_2 & \dots & r_{n-1} & r_n \\ p(X): & \mathrm{bal}_1 & \mathrm{bal}_2 & \dots & \mathrm{bal}_{n-1} & \mathrm{bal}_n \end{array}$$

Proof of Liabilities Example

User	Alice	Bob	Charlie	David
Balance	10	12	9	7

Table 1: User Balances

Proof of Liabilities Example

User	Alice	Bob	Charlie	David
Balance	10	12	9	7

Table 1: User Balances

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p(X)$	10	12	9	7

Table 2: Polynomials

Proof of Liabilities Example

User	Alice	Bob	Charlie	David
Balance	10	12	9	7

Table 1: User Balances

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p(X)$	10	12	9	7

Table 2: Polynomials

- $f(X) = p(X), X = n - 1$
- $f(X) = p(X) + f(X+1), \forall X \in [0, n-1)$

Proof of Liabilities Example

User	Alice	Bob	Charlie	David
Balance	10	12	9	7

Table 1: User Balances

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p(X)$	10	12	9	7

Table 2: Polynomials

- $f(X) = p(X), X = n-1 \Rightarrow \frac{f(X)-p(X)}{X-(n-1)} = q_1(X)$
- $f(X) = p(X) + f(X+1), \forall X \in [0, n-1)$
  $\Rightarrow \frac{f(X)-p(X)-f(X+1)}{X(X-1)(X-2)\cdots[X-(n-2)]} = q_2(X)$

Range Proof

- Prove a value $x \in [0, 2^k)$
- Prevent attacks by overflowing the liabilities and inserting negative values
- Binary decomposition[1]
  - $\bar{z} = \{z_1, z_2, \ldots, z_k\}, x = 2^0 \cdot z_1 + 2^1 \cdot z_2 + \cdots + 2^{k-1} \cdot z_k$
  - $x = z_1 + 2 \cdot (z_2 + 2 \cdot (z_3 + 2 \cdot (z_4 + \ldots)))$

	2^0	2^1	2^2	2^3	$\ldots$	2^{k-2}	2^{k-1}
$\bar{z}$	z_1	z_2	z_3	z_4	$\ldots$	z_{k-1}	z_k
λ	$z_1 + 2(z_2 + \ldots)$	$z_2 + 2(z_3 + \ldots)$	$z_3 + 2(z_4 + \ldots)$	$z_4 + 2(z_5 + \ldots)$	$\ldots$	$z_{k-1} + 2z_k$	z_k

[1] `https://hackmd.io/@dabo/B1U4kx8XI`

Range Proof

- Prove a value $x \in [0, 2^k)$
- Prevent attacks by overflowing the liabilities and inserting negative values
- Binary decomposition[1]

Example

	2^0	2^1	2^2	2^3
$x = 13$	$\lambda_0 = 1$	$\lambda_1 = 0$	$\lambda_2 = 1$	$\lambda_3 = 1$
$f_{\text{range}}(X)$	13	6	3	1
	$f_{\text{range}}(0)$	$f_{\text{range}}(1)$	$f_{\text{range}}(2)$	$f_{\text{range}}(3)$

- $f_{\text{range}}(X) \in \{0, 1\}, X = k - 1$
- $f_{\text{range}}(X) - 2 \cdot f_{\text{range}}(X + 1) \in \{0, 1\}, X \in [0, k)$

[1] https://hackmd.io/@dabo/B1U4kx8XI

Proof of Liability + Range Proof

X / Poly	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p(X)$	10	12	9	7
	5	6	4	3
	2	3	2	1
	1	1	1	0

Proof of Liability + Range Proof

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p(X)$	10	12	9	7
	5	6	4	3
	2	3	2	1
	1	1	1	0

The degree of $p(X)$ is proportional to the size of the range proof

- The largest set of the roots we can use is 2^{32}
- One million $\approx 2^{20}$
- We can only prove values in $[0, 2^{12})$

Proof of Liability + Range Proof

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p(X)$	10	12	9	7
	5	6	4	3
	2	3	2	1
	1	1	1	0

The degree of $p(X)$ is proportional to the size of the range proof

- The largest set of the roots we can use is 2^{32}
- One million $\approx 2^{20}$
- We can only prove values in $[0, 2^{12})$

Solution

- Break the range polynomial

Proof of Liability + Range Proof

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p_1(X)$	10	12	9	7
$p_2(X)$	5	6	4	3
$p_3(X)$	2	3	2	1
$p_4(X)$	1	1	1	0

- $f_{\text{liab}}(n-1) = p_1(n-1)$
- $f_{\text{liab}}(X) = f_{\text{liab}}(X+1) + p_1(X), X \in [0, n-1)$
- $p_k(X) \in \{0, 1\}$
- $p_i(X) - 2p_{i+1}(X) \in \{0, 1\}$

Proof of Liability + Range Proof

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p_1(X)$	10	12	9	7
$p_2(X)$	5	6	4	3
$p_3(X)$	2	3	2	1
$p_4(X)$	1	1	1	0

- $f_{\text{liab}}(n-1) = p_1(n-1)$
- $f_{\text{liab}}(X) = f_{\text{liab}}(X+1) + p_1(X), X \in [0, n-1)$
- $p_k(X) \in \{0, 1\}$
- $p_i(X) - 2p_{i+1}(X) \in \{0, 1\}$

Clash attack: count the same amount only once

Clash Attack

Poly \ X	0	1	2	3
$f_{\text{liab}}(X)$	38	28	16	7
$p_1(X)$	10	12	9	7
$p_2(X)$	5	6	4	3
$p_3(X)$	2	3	2	1
$p_4(X)$	1	1	1	0
$u(X)$	A	B	C	D

- $p_k(X) \in \{0, 1\}$
- $p_i(X) - 2p_{i+1}(X) \in \{0, 1\}$
- $f_{\text{liab}}(n-1) = p_1(n-1)$
- $f_{\text{liab}}(X) = f_{\text{liab}}(X+1) + p_1(X), X \in [0, n-1)$
- For each user t, $p_1(t) = \text{bal}_t$, $u(t) = \text{uid}_\text{t}$

Proof of Assets

Recall the problem

- The exchange proves the ownership of some wallet accounts
- The total asset is the sum of these accounts

Proof of Assets

Recall the problem

- The exchange proves the ownership of some wallet accounts
- The total asset is the sum of these accounts

Preliminaries

- Wallet address = Hash(pk)
- pk $= g^{\mathsf{sk}}$, where g is a generator of an elliptic curve group (for Bitcoin and Ethereum, the curve is `secp256k1`; for the PCS, the curve is `bls12-381`)
- Secured by the discrete logarithm assumption

Proof of Assets

Intuition

- Publish some public keys as an anonymity set $\{\mathsf{pk}_i\}$

	pk_1	pk_2	pk_3	$\ldots$	pk_{n-1}	pk_n
$b(X)$:	bal_1	bal_2	bal_3	$\ldots$	bal_{n-1}	bal_n

Proof of Assets

Intuition

- Publish some public keys as an anonymity set $\{\mathsf{pk}_i\}$
- Construct a selector polynomial $s(X)$ where each evaluation is in $\{0, 1\}$

$$\begin{array}{lcccccc} & \mathsf{pk}_1 & \mathsf{pk}_2 & \mathsf{pk}_3 & \ldots & \mathsf{pk}_{n-1} & \mathsf{pk}_n \\ b(X): & \mathsf{bal}_1 & \mathsf{bal}_2 & \mathsf{bal}_3 & \ldots & \mathsf{bal}_{n-1} & \mathsf{bal}_n \\ s(X): & s_1 & s_2 & s_3 & \ldots & s_{n-1} & s_n \end{array}$$

Proof of Assets

Intuition

- Publish some public keys as an anonymity set $\{\mathsf{pk}_i\}$
- Construct a selector polynomial $s(X)$ where each evaluation is in $\{0, 1\}$
- Construct an accumulative polynomial

$$
\begin{array}{rcccccc}
 & \mathsf{pk}_1 & \mathsf{pk}_2 & \mathsf{pk}_3 & \dots & \mathsf{pk}_{n-1} & \mathsf{pk}_n \\
b(X): & \mathsf{bal}_1 & \mathsf{bal}_2 & \mathsf{bal}_3 & \dots & \mathsf{bal}_{n-1} & \mathsf{bal}_n \\
s(X): & s_1 & s_2 & s_3 & \dots & s_{n-1} & s_n \\
f_{\mathsf{assets}}(X): & r_1 & r_2 & r_3 & \dots & r_{n-1} & r_n \\
 & s_1 \cdot \mathsf{bal}_1 + r_2 & s_2 \cdot \mathsf{bal}_2 + r_3 & s_3 \cdot \mathsf{bal}_3 + r_4 & \dots & s_{n-1} \cdot \mathsf{bal}_{n-1} + r_n & s_n \cdot \mathsf{bal}_n
\end{array}
$$

Proof of Assets

Intuition

- Publish some public keys as an anonymity set $\{\mathsf{pk}_i\}$
- Construct a selector polynomial $s(X)$ where each evaluation is in $\{0, 1\}$
- Construct an accumulative polynomial

$$
\begin{array}{rcccccc}
 & \mathsf{pk}_1 & \mathsf{pk}_2 & \mathsf{pk}_3 & \ldots & \mathsf{pk}_{n-1} & \mathsf{pk}_n \\
b(X): & \mathsf{bal}_1 & \mathsf{bal}_2 & \mathsf{bal}_3 & \ldots & \mathsf{bal}_{n-1} & \mathsf{bal}_n \\
s(X): & s_1 & s_2 & s_3 & \ldots & s_{n-1} & s_n \\
f_{\mathsf{assets}}(X): & r_1 & r_2 & r_3 & \ldots & r_{n-1} & r_n \\
 & s_1 \cdot \mathsf{bal}_1 + r_2 & s_2 \cdot \mathsf{bal}_2 + r_3 & s_3 \cdot \mathsf{bal}_3 + r_4 & \ldots & s_{n-1} \cdot \mathsf{bal}_{n-1} + r_n & s_n \cdot \mathsf{bal}_n
\end{array}
$$

How to construct such a selector polynomial in ZK?

Disjunctive Σ-Protocol (OR Proof)

OR Proof

Prove the claiming statement is R_1 or R_2 without telling which case it is.

$\mathcal{P}$		$\mathcal{V}$
$c_2 \xleftarrow{\$} \{0,1\}^t$		
	$\xrightarrow{m_1, m_2}$	
		$c \xleftarrow{\$} \{0,1\}^t$
	$\xleftarrow{c}$	
$c_1 = c \oplus c_2$		
	$\xrightarrow{r_1, r_2, c_1, c_2}$	
		check (i) $c \stackrel{?}{=} c_1 \oplus c_2$ (ii) $(m_1, c_1, r_1) \in R_1$ (iii) $(m_2, c_2, r_2) \in R_2$

Proof of Assets + OR Proof

A useful observation: the previous proof of assets has composite statements:

1. The selector s_i is 1 and prove the knowledge of the corresponding private key
   - $g_{\mathsf{bls}}^{s_i} = g_{\mathsf{bls}}^{1}$ (in `bls12-381`)
   - the knowledge of sk_i (in `secp256k1`)
2. the selector s_i is 0
   - $g_{\mathsf{bls}}^{s_i} = g_{\mathsf{bls}}^{0}$ (in `bls12-381`)

[2] See Section B of the paper.

Proof of Assets + OR Proof

A useful observation: the previous proof of assets has composite statements:

1. The selector s_i is 1 and prove the knowledge of the corresponding private key
   - $g_{\text{bls}}^{s_i} = g_{\text{bls}}^{1}$ (in `bls12-381`)
   - the knowledge of sk_i (in `secp256k1`)
2. the selector s_i is 0
   - $g_{\text{bls}}^{s_i} = g_{\text{bls}}^{0}$ (in `bls12-381`)

Can we open the commitment to the evaluation without revealing the evaluation itself?

[2] See Section B of the paper.

Proof of Assets + OR Proof

A useful observation: the previous proof of assets has composite statements:

1. The selector s_i is 1 and prove the knowledge of the corresponding private key
   - $g_{\text{bls}}^{s_i} = g_{\text{bls}}^{1}$ (in `bls12-381`)
   - the knowledge of sk_i (in `secp256k1`)
2. the selector s_i is 0
   - $g_{\text{bls}}^{s_i} = g_{\text{bls}}^{0}$ (in `bls12-381`)

PCS Extension

Open the committed evaluation g_{bls}^{x} [2].

[2] See Section B of the paper.

Proof of Assets + OR Proof

$\mathcal{P}$ $\qquad\qquad\qquad\qquad$ $\mathcal{V}$

$s_i = 1$

$c_2 \xleftarrow{\$} \{0,1\}^l$

$m_1^{(i)}, m_1^{(ii)}, m_2, r_2$

$\xrightarrow{m_1^{(i)}, m_1^{(ii)}, m_2}$

$\xleftarrow{c \xleftarrow{\$} \{0,1\}^l}$

$c_1 = c \oplus c_2$

$\xrightarrow{r_1^{(i)}, r_1^{(ii)}, r_2, c_1, c_2}$

$s_i = 0$

$c_1 \xleftarrow{\$} \{0,1\}^l$

$m_1^{(i)}, m_1^{(ii)}, m_2, r_1^{(i)}, r_1^{(ii)}$

$\xrightarrow{m_1^{(i)}, m_1^{(ii)}, m_2}$

$\xleftarrow{c \xleftarrow{\$} \{0,1\}^l}$

$c_2 = c \oplus c_1$

$\xrightarrow{r_1^{(i)}, r_1^{(ii)}, r_2, c_1, c_2}$

check $(m_1^{(i)}, r_1^{(i)}, c_1), (m_1^{(ii)}, r_1^{(ii)}, c_1),$
(m_2, r_2, c_2)

Full Protocol of Proof of Assets

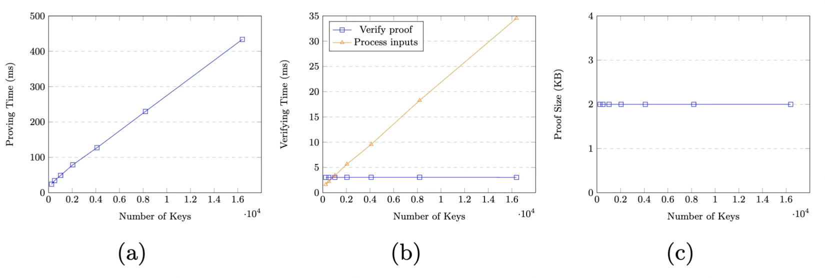
1. Bootstrapping (pre-computing)
    1.1 Interpolate selector polynomial $s(X)$
    1.2 Open the evaluations of $s(X)$
    1.3 Prove each evaluation is valid (the OR proof)
2. Interpolate the balance polynomial $b(X)$
3. Construct the accumulative polynomial $f_{\text{assets}}(X)$
4. Prove the polynomial constraints (accumulate assets) are correct

Proof of Solvency

- Run the Proof of Liabilities to output $f_{\mathsf{liab}}(X)$
- Run the Proof of Assets to output $f_{\mathsf{assets}}(X)$
- Open the evaluations of $f_{\mathsf{liab}}(0)$ and $f_{\mathsf{assets}}(0)$ and compute $\Delta = f_{\mathsf{assets}}(0) - f_{\mathsf{liab}}(0)$
- Prove $\Delta \geq 0$ using the range proof

Benchmark Performance

Proof of assets



Proof of liability

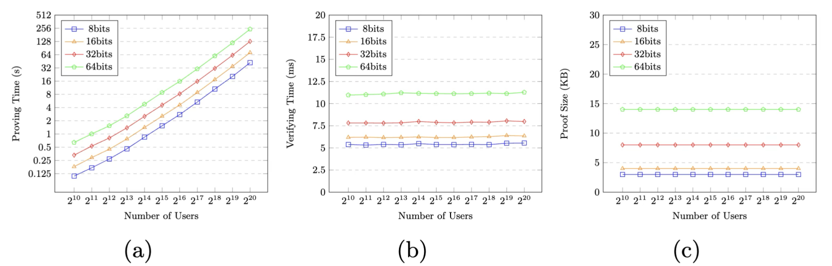


Table of Contents

Conclusion

- This work is an almost succinct proof of solvency
- It is practical to be deployed for the real-world exchanges, implemented in Rust (https://github.com/Shvier/proof-of-solvency)
- Future work
  - Multivariate polynomial for better performance
  - Incremental update (folding schemes)
  - Shorter range proof (lookup arguments)

State of the Art in ZKP

- Applications
  - In blockchain: zkRollup, zkBridge, zkEVM
  - In other fields: fight disinformation [3]
- Limitations
  - Cost
  - Performance
  - Complexity
- Further reading
  - ZK MOOC
  - Vitalik's Blog
  - A Graduate Course in Applied Cryptography

[3] https://rdi.berkeley.edu/zk-learning/assets/Lecture2-2023.pdf