Non-Interactive Zero-Knowledge Functional Proofs

Gongxian Zeng[1] Junzuo Lai[2] Zhengan Huang[1]
Linru Zhang[3] Xiangning Wang[3] Kwok-Yan Lam[3]
Huaxiong Wang[3] Jian Weng[2]

[1]Peng Cheng Laboratory, Shenzhen, China

[2]College of Information Science and Technology, Jinan University, Guangzhou, China

[3]Nanyang Technological University, Singapore

Agenda

1. Motivations & Contributions
2. Primitive of fNIZK
3. A generic construction of fNIZK
4. A concrete construction for set membership
5. References

NIZK

Let $\mathcal{L}$ be an NP language associated with an NP relation $\mathcal{R}$. A non-interactive zero-knowledge proof (NIZK) for $\mathcal{L}$ consists of a tuple of three efficient algorithms NIZK $=$ (Setup, Prove, Verify).

- Completeness
- Soundness: $x \notin \mathcal{L}$, the verifier would not accept π.
- Zero knowledge: except for the fact of the truth of the statement, the verifier cannot know any information about the witness from the accepting proof.

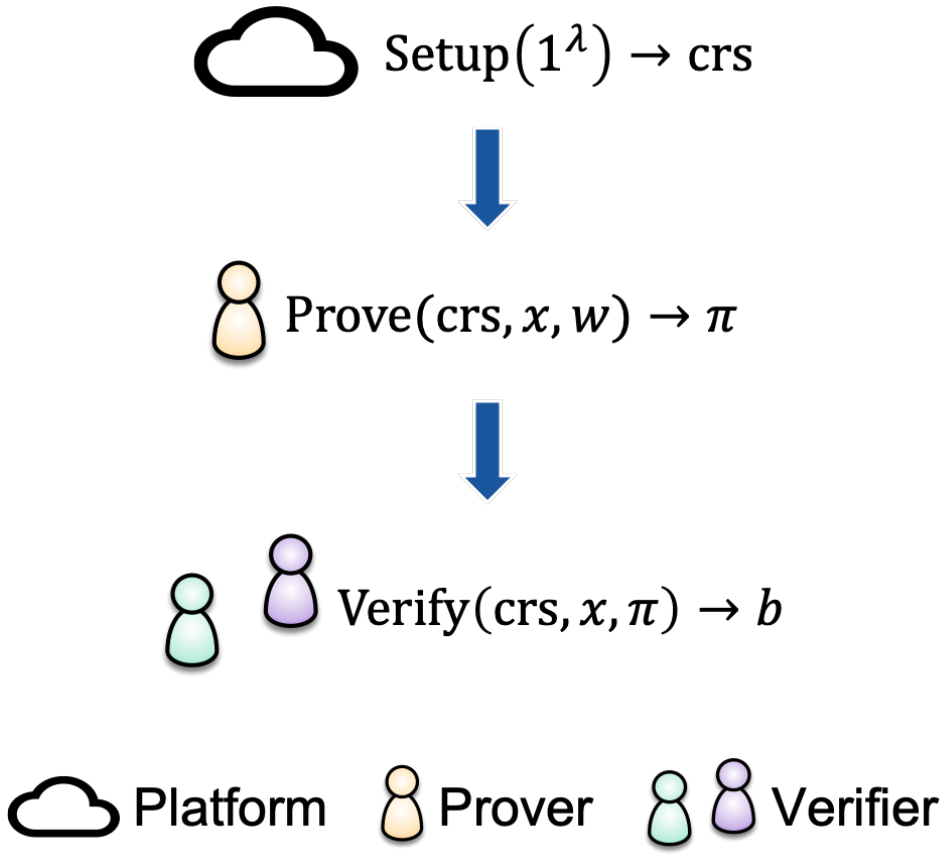


Figure 1: NIZK in the CRS model

Our goal

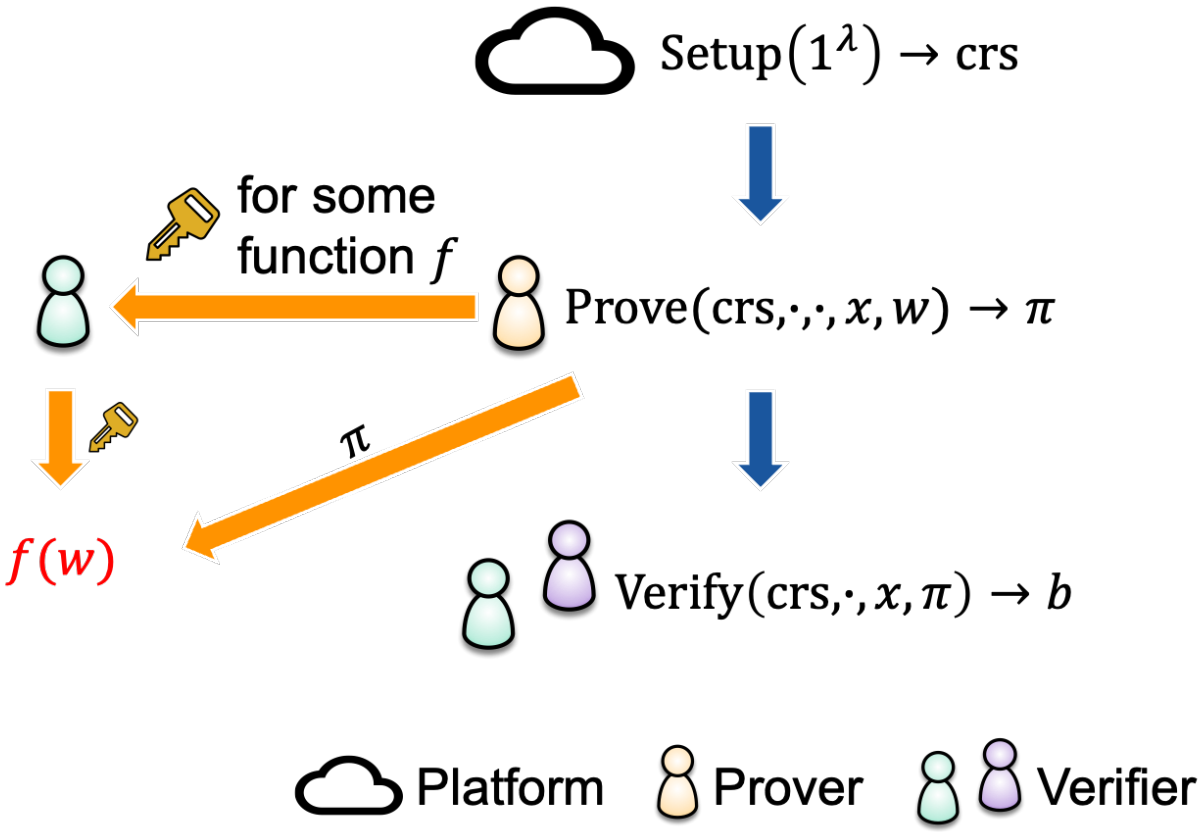


Figure 2: Non-interactive zero-knowledge functional proofs (fNIZKs)

⋆ In addition to verifying the truth of the statement, the verifier can also gain insights into certain **functions of the witness** using a secret key provided by the prover.

Applications of fNIZK

Applications: supervision, blockchain-based auction systems, anonymous attribute-based credential ...

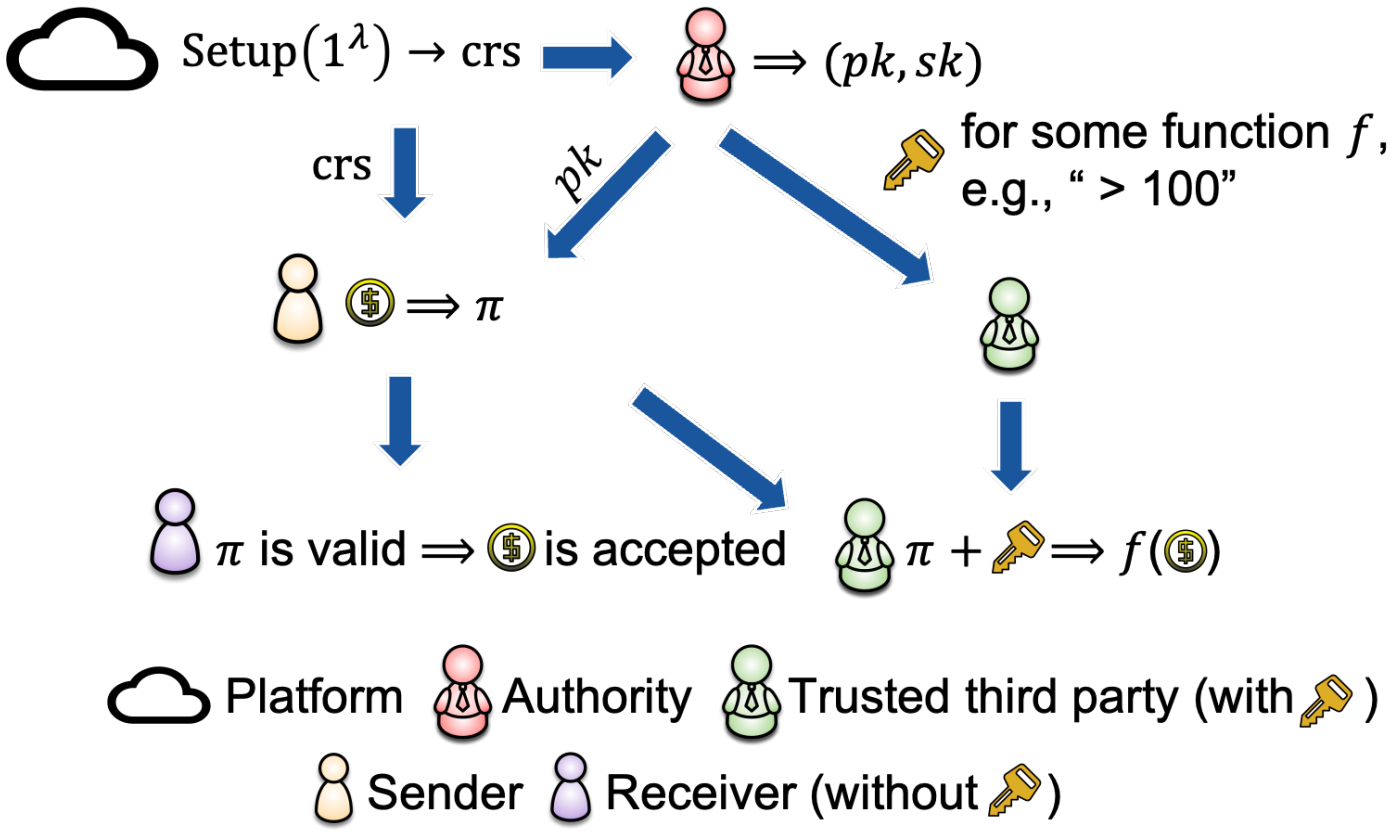


Figure 3: Supervision for anti-money laundering

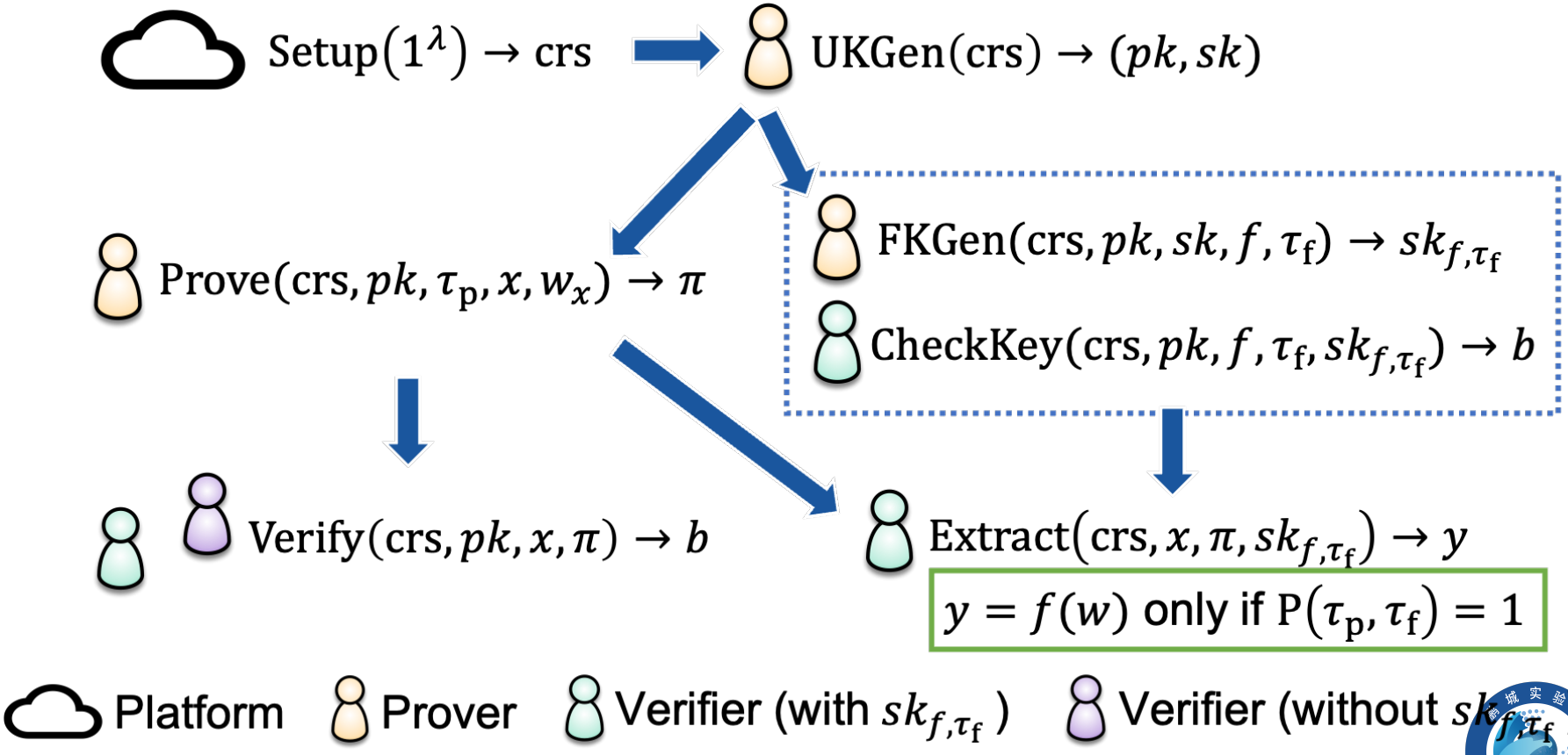
Contributions

We initiate the study of fNIZK. The specific contributions are outlined as follows:

1. We present a formal definition and security notions of non-interactive zero-knowledge functional proof (fNIZK).
2. We provide a generic construction of fNIZK for any NP relation $\mathcal{R}$, which enables the prover to share any function of the witness with a verifier.
3. For a widely-used relation about set membership proof (implying range proof), we construct a concrete and efficient fNIZK, called set membership functional proof (fSMP).

Syntax

A *non-interactive zero-knowledge functional proof (fNIZK proof)* for $\mathcal{L}_{\mathcal{R}}$, $\mathbb{F}$, $\mathcal{T}$ and P consists of a tuple of seven efficient algorithms $\mathsf{fNIZK} = (\mathsf{Setup}, \mathsf{UKGen}, \mathsf{FKGen}, \mathsf{CheckKey}, \mathsf{Prove}, \mathsf{Verify}, \mathsf{Extract})$.



Security I: Completeness

The security properties of fNIZK contains completeness, functional knowledge, adaptive soundness and zero knowledge.

Completeness. For any $(x, w) \in \mathcal{R}$, any $f \in \mathbb{F}$, any $\tau_{\mathtt{p}} \in \mathcal{T}$ and any $\tau_{\mathtt{f}} \in \mathcal{T} \cup \{*\}$,

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (pk, sk) \leftarrow \mathsf{UKGen}(\mathsf{crs}) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, pk, \tau_{\mathtt{p}}, x, w)\end{array} : \mathsf{Verify}(\mathsf{crs}, pk, x, \pi) = 1\right] \geq 1 - \mathsf{negl}(\lambda),$$

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (pk, sk) \leftarrow \mathsf{UKGen}(\mathsf{crs}) \\ sk_{f,\tau_{\mathtt{f}}} \leftarrow \mathsf{FKGen}(\mathsf{crs}, pk, sk, f, \tau_{\mathtt{f}})\end{array} : \mathsf{CheckKey}(\mathsf{crs}, pk, f, \tau_{\mathtt{f}}, sk_{f,\tau_{\mathtt{f}}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda),$$

$$\Pr\left[\begin{array}{l}\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (pk, sk) \leftarrow \mathsf{UKGen}(\mathsf{crs}) \\ sk_{f,\tau_{\mathtt{f}}} \leftarrow \mathsf{FKGen}(\mathsf{crs}, pk, sk, f, \tau_{\mathtt{f}}) \\ \pi \leftarrow \mathsf{Prove}(\mathsf{crs}, pk, \tau_{\mathtt{p}}, x, w)\end{array} : \mathsf{Extract}(\mathsf{crs}, x, \pi, sk_{f,\tau_{\mathtt{f}}}) = f(w) \,\middle|\, \mathsf{P}(\tau_{\mathtt{p}}, \tau_{\mathtt{f}}) = 1\right] \geq 1 - \mathsf{negl}(\lambda).$$

Security II: Functional knowledge and Adaptive soundness

Functional knowledge. For any PPT adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda),\ (pk, sk) \leftarrow \mathsf{UKGen}(\mathsf{crs}) \\ (\pi, x, f, \tau_{\mathtt{f}}, sk_{f,\tau_{\mathtt{f}}}) \leftarrow \mathcal{A}(\mathsf{crs}, pk, sk) \\ \quad \text{s.t. } (x \in \mathcal{L}_{\mathcal{R}}) \wedge (f \in \mathbb{F}) \wedge \tau_{\mathtt{f}} \in (\mathcal{T} \cup \{*\}) \\ \qquad \wedge (\mathsf{Verify}(\mathsf{crs}, pk, x, \pi) = 1) \\ \qquad \wedge (\mathsf{CheckKey}(\mathsf{crs}, pk, f, \tau_{\mathtt{f}}, sk_{f,\tau_{\mathtt{f}}}) = 1) \\ \qquad \wedge (\mathsf{P}(\mathsf{Ext}_\tau(\pi), \tau_{\mathtt{f}}) = 1) \\ y \leftarrow \mathsf{Extract}(\mathsf{crs}, x, \pi, sk_{f,\tau_{\mathtt{f}}}) \end{array} : \begin{array}{c} \exists\ w \in \mathcal{W},\ \text{s.t.} \\ ((x, w) \in \mathcal{R}) \\ \wedge (y = f(w)) \end{array}\right] \geq 1 - \mathsf{negl}(\lambda).$$

Adaptive soundness. For any computationally unbounded adversary $\mathcal{A}$,

$$\Pr\left[\begin{array}{l} \mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda) \\ (pk, x, \pi) \leftarrow \mathcal{A}(\mathsf{crs}) \end{array} : \begin{array}{c} x \notin \mathcal{L}_{\mathcal{R}} \\ \wedge\ \mathsf{Verify}(\mathsf{crs}, pk, x, \pi) = 1 \end{array}\right] \leq \mathsf{negl}(\lambda).$$

Security III: Zero knowledge

Zero knowledge. $\forall$ PPT adversary $\mathcal{A}$, $\exists$ Sim:

$$\left|\Pr[\mathsf{ExpReal}^{\mathsf{zk}}_{\mathsf{fNIZK},\mathcal{A},n}(\lambda)=1]-\Pr[\mathsf{ExpIdeal}^{\mathsf{zk}}_{\mathsf{fNIZK},\mathcal{A},\mathsf{Sim},n}(\lambda)=1]\right| \le \mathsf{negl}(\lambda).$$

$\mathsf{ExpReal}^{\mathsf{zk}}_{\mathsf{fNIZK},\mathcal{A},n}(\lambda)$:

$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$, $W := \emptyset$, $Q := \emptyset$
$((pk_i, sk_i) \leftarrow \mathsf{UKGen}(\mathsf{crs}))_{i\in[n]}$
$(U_{\mathtt{cor}}, st_1^{\mathcal{A}}) \leftarrow \mathcal{A}_1(\mathsf{crs}, (pk_i)_{i\in[n]})$
  s.t. $U_{\mathtt{cor}} \subset [n]$
$(i^*, \tau_{\mathtt{p}}, x, w, w', st_2^{\mathcal{A}}) \leftarrow \mathcal{A}_2^{\mathcal{O}^{\mathsf{FKGen}}(\cdot)}((sk_i)_{i\in U_{\mathtt{cor}}}, st_1^{\mathcal{A}})$
  s.t. $(i^* \notin U_{\mathtt{cor}}) \wedge ((x,w)\in\mathcal{R})$
    $\wedge(\forall(i^*, f', \tau'_{\mathtt{f}}) \in Q$ satisfying $\mathsf{P}(\tau_{\mathtt{p}}, \tau'_{\mathtt{f}}) = 1$,
      $f'(w) = f'(w'))$
$W := \{i^*, \tau_{\mathtt{p}}, w, w'\}$
$\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, pk_{i^*}, \tau_{\mathtt{p}}, x, w)$
Return $b \leftarrow \mathcal{A}_3^{\mathcal{O}^{\mathsf{FKGen}}(\cdot)}(\pi, st_2^{\mathcal{A}})$

$\mathcal{O}^{\mathsf{FKGen}}(i', f', \tau'_{\mathtt{f}})$:
If $W \neq \emptyset$:
  Parse $W = \{i^*, \tau_{\mathtt{p}}, w, w'\}$
  If $(i' = i^*) \wedge (\mathsf{P}(\tau_{\mathtt{p}}, \tau'_{\mathtt{f}}) = 1) \wedge (f'(w) \neq f'(w'))$:
    Return $\perp$
$Q := Q \cup \{(i', f', \tau'_{\mathtt{f}})\}$
Return $sk_{i',f',\tau'_{\mathtt{f}}} \leftarrow \mathsf{FKGen}(\mathsf{crs}, pk_{i'}, sk_{i'}, f', \tau'_{\mathtt{f}})$

$\mathsf{ExpIdeal}^{\mathsf{zk}}_{\mathsf{fNIZK},\mathcal{A},\mathsf{Sim},n}(\lambda)$:

$(\mathsf{crs}, st^{\mathsf{Sim}}) \leftarrow \mathsf{Sim}_1(1^\lambda)$, ...
... (same to the right)
...
...
...
...
...
...
...
...
...
...
$\pi \leftarrow \mathsf{Sim}_2(\mathsf{crs}, pk_{i^*}, \tau_{\mathtt{p}}, x, w', st^{\mathsf{Sim}})$
...

$\mathcal{O}^{\mathsf{FKGen}}(i', f', \tau'_{\mathtt{f}})$:
...
...
...
...
...
...
...

Figure 4: Games for defining zero knowledge property for fNIZK

NIZK Π deduced by fNIZK

Specifically, for a fNIZK scheme fNIZK $=$ (Setup, UKGen, FKGen, CheckKey, Prove, Verify, Extract), consider a non-interactive proof scheme $\Pi = (\Pi.\mathsf{Setup}, \Pi.\mathsf{Prove}, \Pi.\mathsf{Verify})$ as in Fig. 5.

$\Pi.\mathsf{Setup}(1^\lambda)$:	$\Pi.\mathsf{Prove}(\mathsf{crs}^{\mathsf{zk}}, x, w)$:	$\Pi.\mathsf{Verify}(\mathsf{crs}^{\mathsf{zk}}, x, \pi)$:
$\mathsf{crs} \leftarrow \mathsf{Setup}(1^\lambda)$	$\tau_{\mathsf{p}} \leftarrow \mathcal{T}$	$b \leftarrow \mathsf{Verify}(\mathsf{crs}, pk, x, \pi)$
$(pk, sk) \leftarrow \mathsf{UKGen}(\mathsf{crs})$	$\pi \leftarrow \mathsf{Prove}(\mathsf{crs}, pk, \tau_{\mathsf{p}}, x, w)$	Return b
Return $\mathsf{crs}^{\mathsf{zk}} = (\mathsf{crs}, pk)$	Return π	

Figure 5: NIZK Π deduced by fNIZK

Theorem 1

If fNIZK *is a fNIZK scheme for an* NP *language* $\mathcal{L}_{\mathcal{R}}$*, a function family* $\mathbb{F}$*, a label space* $\mathcal{T}$ *and a predicate function* P*, then* Π *is a NIZK scheme for* $\mathcal{L}_{\mathcal{R}}$*.*

Building blocks I

We want to construct a fNIZK for a general function family $\mathbb{F}$ such that

$$\mathsf{Extract}(\mathsf{crs}, x, \pi, sk_{f,\tau_{\mathtt{f}}}) = f(w) \text{ when } \mathsf{P}(\tau_{\mathtt{p}}, \tau_{\mathtt{f}}) = 1.$$

(1) Function family $(\mathbb{F} \Rightarrow \widehat{\mathbb{F}})$. We define a function family $\widehat{\mathbb{F}}$ as follows: a function $\widehat{f}$ (with domain $\mathcal{W} \times \mathcal{T}$) belongs to $\widehat{\mathbb{F}}$, if and only if there exists $(f, \tau_{\mathtt{f}}) \in \mathbb{F} \times (\mathcal{T} \cup \{*\})$ satisfying

$$\widehat{f}(w, \tau_{\mathtt{p}}) = \begin{cases} f(w) & \text{if } \mathsf{P}(\tau_{\mathtt{p}}, \tau_{\mathtt{f}}) = 1 \\ \perp & \text{if } \mathsf{P}(\tau_{\mathtt{p}}, \tau_{\mathtt{f}}) = 0 \end{cases}$$

(2) FE for $\widehat{\mathbb{F}}$. Let $\mathsf{FE} = (\mathsf{FE.Setup}, \mathsf{FE.KGen}, \mathsf{FE.Enc}, \mathsf{FE.Dec})$ be a functional encryption scheme for $\widehat{\mathbb{F}}$ on message space $\mathcal{M} = \mathcal{W} \times \mathcal{T}$.

Building blocks II

(3) NIZK for $\mathcal{R}_{\mathsf{ct}}$ and $\mathcal{R}_{\mathsf{k}}$. Consider the following two NP relations

$$\mathcal{R}_{\mathsf{ct}} = \{((\tau_{\mathtt{p}}, x, mpk, c), (w, r_{\mathsf{enc}})) : (x, w) \in \mathcal{R} \wedge \mathsf{FE.Enc}(mpk, (w, \tau_{\mathtt{p}}); r_{\mathsf{enc}}) = c\},$$

$$\mathcal{R}_{\mathsf{k}} = \{((mpk, \widehat{f}_{f,\tau_{\mathtt{f}}}, sk_{\widehat{f}_{f,\tau_{\mathtt{f}}}}), (msk, r_{\mathsf{kg}})) : \mathsf{FE.KGen}(mpk, msk, \widehat{f}_{f,\tau_{\mathtt{f}}}; r_{\mathsf{kg}}) = sk_{\widehat{f}_{f,\tau_{\mathtt{f}}}}\},$$

where

- $\mathcal{R}_{\mathsf{ct}}$: $(x, w) \in \mathcal{R}$ and c is a well-formed ciphertext for w.
- $\mathcal{R}_{\mathsf{k}}$: $sk_{\widehat{f}_{f,\tau_{\mathtt{f}}}}$ is a well-formed secret key for function $\widehat{f}_{f,\tau_{\mathtt{f}}}$.

As stated in [FLS99], we can construct NIZKs for any NP language. Therefore, we can construct two NIZK schemes, $\mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}} = (\mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}.\mathsf{Setup}, \mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}.\mathsf{Prove}, \mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}.\mathsf{Verify})$ and $\mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}} = (\mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}.\mathsf{Setup}, \mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}.\mathsf{Prove}, \mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}.\mathsf{Verify})$, for $\mathcal{L}_{\mathcal{R}_{\mathsf{ct}}}$ and $\mathcal{L}_{\mathcal{R}_{\mathsf{k}}}$, respectively.

Construction

The generic construction of fNIZK proof fNIZK = (Setup, UKGen, FKGen, CheckKey, Prove, Verify, Extract)

$\underline{\mathsf{Setup}(1^\lambda)}$:
$\mathsf{crs}_{\mathcal{R}_{\mathsf{ct}}} \leftarrow \mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}.\mathsf{Setup}(1^\lambda)$
$\mathsf{crs}_{\mathcal{R}_{\mathsf{k}}} \leftarrow \mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}.\mathsf{Setup}(1^\lambda)$
Return $\mathsf{crs} := (\mathsf{crs}_{\mathcal{R}_{\mathsf{ct}}}, \mathsf{crs}_{\mathcal{R}_{\mathsf{k}}})$

$\underline{\mathsf{Prove}(\mathsf{crs}, pk, \tau_{\mathsf{p}}, x, w)}$:
$r_{\mathsf{enc}} \leftarrow \mathcal{RS}_{\mathsf{FE.Enc}}$, $c \leftarrow \mathsf{FE.Enc}(pk, (w, \tau_{\mathsf{p}}); r_{\mathsf{enc}})$
$\Pi_{\mathcal{R}_{\mathsf{ct}}} \leftarrow \mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}.\mathsf{Prove}(\mathsf{crs}_{\mathcal{R}_{\mathsf{ct}}}, (\tau_{\mathsf{p}}, x, pk, c), (w, r_{\mathsf{enc}}))$
Return $\pi := (\tau_{\mathsf{p}}, \Pi_{\mathcal{R}_{\mathsf{ct}}}, c)$

$\underline{\mathsf{Verify}(\mathsf{crs}, pk, x, \pi)}$:
Parse $\pi = (\tau_{\mathsf{p}}, \Pi_{\mathcal{R}_{\mathsf{ct}}}, c)$
Return $b \leftarrow \mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}.\mathsf{Verify}(\mathsf{crs}_{\mathcal{R}_{\mathsf{ct}}}, (\tau_{\mathsf{p}}, x, pk, c), \Pi_{\mathcal{R}_{\mathsf{ct}}})$

$\underline{\mathsf{Extract}(\mathsf{crs}, x, \pi, sk_{f,\tau_{\mathsf{f}}})}$:
Parse $\pi = (\tau_{\mathsf{p}}, \Pi_{\mathcal{R}_{\mathsf{ct}}}, c)$, $sk_{f,\tau_{\mathsf{f}}} = (sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}}, \Pi_{\mathcal{R}_{\mathsf{k}}})$
Return $y \leftarrow \mathsf{FE.Dec}(c, sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}})$

$\underline{\mathsf{UKGen}(\mathsf{crs})}$:
$(mpk, msk) \leftarrow \mathsf{FE.Setup}(1^\lambda)$
Return $(pk = mpk, sk = msk)$

$\underline{\mathsf{FKGen}(\mathsf{crs}, pk, sk, f, \tau_{\mathsf{f}})}$:
$r_{\mathsf{kg}} \leftarrow \mathcal{RS}_{\mathsf{FE.KGen}}$
$sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}} \leftarrow \mathsf{FE.KGen}(pk, sk, \widehat{f}_{f,\tau_{\mathsf{f}}}; r_{\mathsf{kg}})$
$\Pi_{\mathcal{R}_{\mathsf{k}}} \leftarrow \mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}.\mathsf{Prove}(\mathsf{crs}_{\mathcal{R}_{\mathsf{k}}}, (pk, \widehat{f}_{f,\tau_{\mathsf{f}}}, sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}}), (sk, r_{\mathsf{kg}}))$
Return $sk_{f,\tau_{\mathsf{f}}} := (sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}}, \Pi_{\mathcal{R}_{\mathsf{k}}})$

$\underline{\mathsf{CheckKey}(\mathsf{crs}, pk, f, \tau_{\mathsf{f}}, sk_{f,\tau_{\mathsf{f}}})}$:
Parse $sk_{f,\tau_{\mathsf{f}}} = (sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}}, \Pi_{\mathcal{R}_{\mathsf{k}}})$
$b \leftarrow \mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}.\mathsf{Verify}(\mathsf{crs}_{\mathcal{R}_{\mathsf{k}}}, (pk, \widehat{f}_{f,\tau_{\mathsf{f}}}, sk_{\widehat{f}_{f,\tau_{\mathsf{f}}}}), \Pi_{\mathcal{R}_{\mathsf{k}}})$
Return b

Figure 6: Construction of fNIZK from FE, $\mathsf{NIZK}_{\mathcal{R}_{\mathsf{ct}}}$ and $\mathsf{NIZK}_{\mathcal{R}_{\mathsf{k}}}$

Security analysis

Theorem 2

(Informal) *The construction* fNIZK *in Fig. 6 satisfies completeness, functional knowledge, adaptive soundness and zero knowledge.*

Please refer to the proof for Theorem 2 in our paper [ZLH+23].

SMP

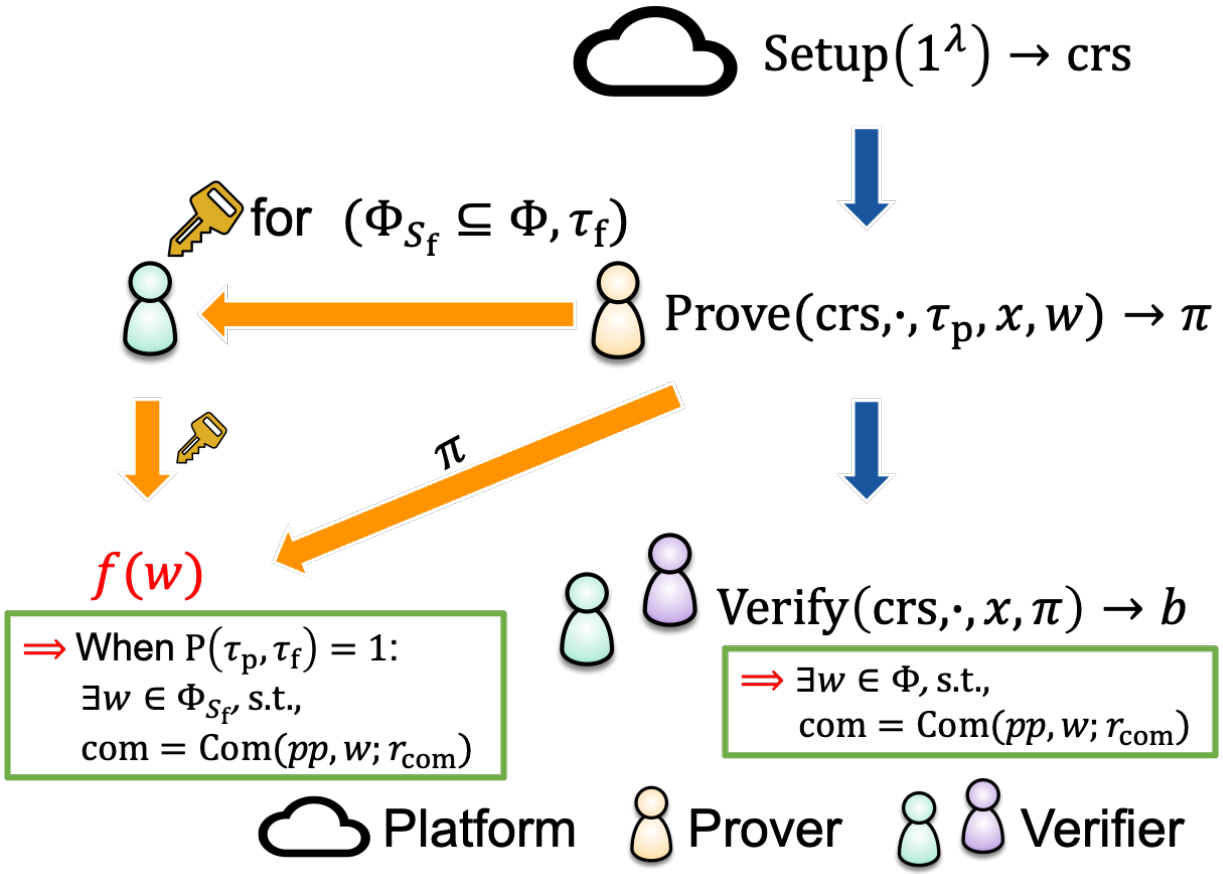
Relation for SMP. Following the definition of [CCS08], the relation about set membership is

$$\mathcal{R}_{\mathsf{sm}} = \{((\mathsf{com}, \Phi), (w, r_{\mathsf{com}})) : \mathsf{com} = \mathsf{Com}(pp, w; r_{\mathsf{com}}) \wedge w \in \Phi\},$$

where pp is the public parameter of the commitment scheme in the relation, com is the commitment to the message w, r_{com} denotes the internal randomness, and Φ represents a set.

Why SMP? Set membership proofs (SMPs) [CCS08] are widely utilized as building blocks in various cryptographic schemes such as anonymous credentials [CL01, TG20], Zcash [SCG+14], and e-cash [CHL05]. Thus, we construct a fNIZK scheme for the relation about set membership proof, we call it set membership functional proof (fSMP).

fSMP

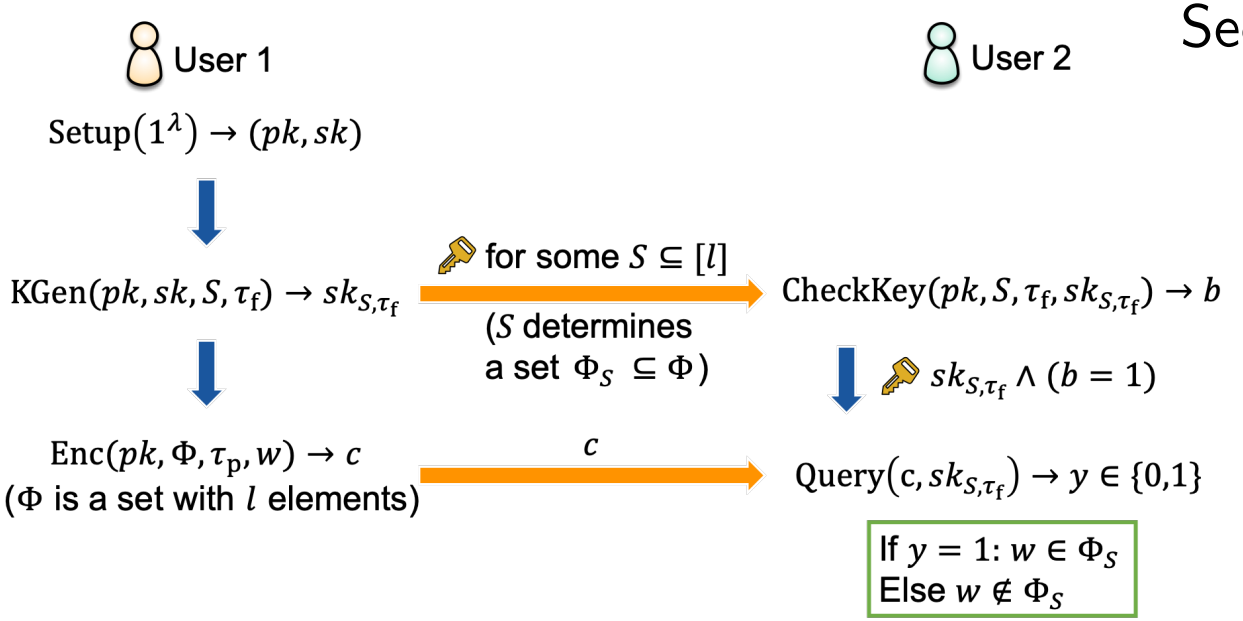


In fSMP, prover outputs a proof associated with label τ_p to demonstrate that the committed value $w \in \Phi$, and a verifier with a secret functional key for (Φ_{S_f}, τ_f) can additionally check whether $w \in \Phi_{S_f}$ or not from the proof, when $P(\tau_p, \tau_f) = 1$.

Building block: SME

A ***set membership encryption (SME)*** scheme SME (with set of size $l = \text{poly}(\lambda)$) for message set $\mathcal{W}$, label space $\mathcal{T}$ and predicate P contains five algorithms

$$\text{SME} = (\text{Setup}, \text{KGen}, \text{CheckKey}, \text{Enc}, \text{Query}).$$



Security properties:

- Correctness;
- IND security;
- Supporting Sigma protocols, if $\exists$ an Σ-protocol for $\mathcal{R}_c$.

$$\mathcal{R}_c = \{((\tau_p, c, pk, \Phi), (w, r_{\text{enc}})) : c = \text{Enc}(pk, \Phi, \tau_p, w; r_{\text{enc}})\}.$$

Building block: Commitment

Commitment. We require that the commitment scheme

$$\mathsf{Commit} = (\mathsf{Commit.Setup}, \mathsf{Commit.Com}, \mathsf{Commit.Dec})$$

also supports Sigma protocols. In other words, there exists an efficient Sigma protocol to prove $\mathsf{com} = \mathsf{Com}(pp, w; r_{\mathsf{com}})$ with statement com and witness (w, r_{com}). One example that satisfies our requirements is Pedersen commitment [Ped91], which we can prove by Okamoto's Sigma protocol [Oka95].

$$\widetilde{\mathcal{R}_{\mathsf{sm}}} = \{((\tau_{\mathsf{p}}, \mathsf{com}, c, pk, \Phi),(w, r_{\mathsf{com}}, r_{\mathsf{enc}})) : \mathsf{com} = \mathsf{Commit.Com}(pp, w; r_{\mathsf{com}}) \\ \wedge\ c = \mathsf{SME.Enc}(pk, \Phi, \tau_{\mathsf{p}}, w; r_{\mathsf{enc}})\}$$

A NIZK for $\widetilde{\mathcal{R}_{\mathsf{sm}}}$. Since both SME and commitment scheme support Sigma protocols, then a Sigma protocol for $\widetilde{\mathcal{R}_{\mathsf{sm}}}$ can be constructed by the composition of Sigma protocols [BS23]. Thus, a NIZK can be obtained by applying Fiat-Shamir transform [FS86].

Function family for fSMP

We define the function family $\mathbb{F}$ as follows.
Each function $f \in \mathbb{F}$ indicates a set $S_f \subset [l]$, such that for any $x = (\mathsf{com}, \Phi = \{w_1, \cdots, w_l\})$ and any $w_x = (w, r_{\mathsf{com}})$,

$$f(w) = \begin{cases} 1 & \text{if } w \in \Phi_{S_f} \\ 0 & \text{if } w \in \Phi \setminus \Phi_{S_f} \\ \perp & \text{otherwise} \end{cases}$$

Construction of fSMP from SME

$\underline{\mathsf{Setup}(1^\lambda)}$:
$\mathsf{crs}_{\mathsf{nizk}} \leftarrow \mathsf{NIZK.Setup}(1^\lambda)$
$pp \leftarrow \mathsf{Commit.Setup}(1^\lambda)$
Return $\mathsf{crs} := (\mathsf{crs}_{\mathsf{nizk}}, pp)$

$\underline{\mathsf{UKGen}(\mathsf{crs})}$:
$(pk, sk) \leftarrow \mathsf{SME.Setup}(1^\lambda)$
Return (pk, sk)

$\underline{\mathsf{FKGen}(\mathsf{crs}, pk, sk, f, \tau_{\mathtt{f}})}$:
$sk_{S_f, \tau_{\mathtt{f}}} \leftarrow \mathsf{SME.KGen}(pk, sk, S_f, \tau_{\mathtt{f}})$ // $S_f \subset [l]$
Return $sk_{f, \tau_{\mathtt{f}}} := sk_{S_f, \tau_{\mathtt{f}}}$

$\underline{\mathsf{CheckKey}(\mathsf{crs}, pk, f, \tau_{\mathtt{f}}, sk_{f, \tau_{\mathtt{f}}})}$:
$b \leftarrow \mathsf{SME.CheckKey}(pk, S_f, \tau_{\mathtt{f}}, sk_{f, \tau_{\mathtt{f}}})$
Return b

$\underline{\mathsf{Prove}(\mathsf{crs}, pk, \tau_{\mathtt{p}}, x, w_x)}$:
Parse $x = (\mathsf{com}, \Phi)$, $w_x = (w, r_{\mathsf{com}})$ // $|\Phi| = l$
$r_{\mathsf{enc}} \leftarrow \mathcal{RS}_{\mathsf{SME.Enc}}$
$c \leftarrow \mathsf{SME.Enc}(pk, \Phi, \tau_{\mathtt{p}}, w; r_{\mathsf{enc}})$
$\pi_{\widetilde{\mathcal{R}_{\mathsf{sm}}}} \leftarrow \mathsf{NIZK.Prove}(\mathsf{crs}_{\mathsf{nizk}}, (\tau_{\mathtt{p}}, \mathsf{com}, c, pk, \Phi), (w, r_{\mathsf{com}}, r_{\mathsf{enc}}))$
Return $\pi := (\tau_{\mathtt{p}}, \pi_{\widetilde{\mathcal{R}_{\mathsf{sm}}}}, c)$

$\underline{\mathsf{Verify}(\mathsf{crs}, pk, x, \pi)}$:
Parse $x = (\mathsf{com}, \Phi)$, $\pi = (\tau_{\mathtt{p}}, \pi_{\widetilde{\mathcal{R}_{\mathsf{sm}}}}, c)$
$b \leftarrow \mathsf{NIZK.Verify}(\mathsf{crs}_{\mathsf{nizk}}, (\tau_{\mathtt{p}}, \mathsf{com}, c, pk, \Phi), \pi_{\widetilde{\mathcal{R}_{\mathsf{sm}}}})$
Return b

$\underline{\mathsf{Extract}(\mathsf{crs}, x, \pi, sk_{f, \tau_{\mathtt{f}}})}$:
Parse $x = (\mathsf{com}, \Phi)$, $\pi = (\tau_{\mathtt{p}}, \pi_{\widetilde{\mathcal{R}_{\mathsf{sm}}}}, c)$
Return $y \leftarrow \mathsf{SME.Query}(c, sk_{f, \tau_{\mathtt{f}}})$

Figure 7: Construction of fSMP from SME

Security analysis

Theorem 3

(Informal) *If the underlying* SME *is IND secure and supports Sigma protocols, then* fSMP *satisfies completeness, functional knowledge, adaptive soundness and zero knowledge.*

Please refer to the proof for Theorem 3 in our paper [ZLH+23].

Main idea: SME from dual IPE

Dual IPE. For vectors $(\mathbf{x}_1, \mathbf{x}_2), (\mathbf{y}_1, \mathbf{y}_2) \in (\mathbb{Z}_p)^{l_1} \times (\mathbb{Z}_p)^{l_2}$, we define a function $\mathtt{DuIP} : (\mathbb{Z}_p)^{l_1} \times (\mathbb{Z}_p)^{l_2} \times (\mathbb{Z}_p)^{l_1} \times (\mathbb{Z}_p)^{l_2} \to \{0, 1\}$ as follows:

$$\mathtt{DuIP}((\mathbf{x}_1, \mathbf{x}_2), (\mathbf{y}_1, \mathbf{y}_2)) = \begin{cases} 0 & \text{if } \langle \mathbf{x}_1, \mathbf{y}_1 \rangle = \langle \mathbf{x}_2, \mathbf{y}_2 \rangle = 0 \\ 1 & \text{otherwise} \end{cases}$$

It is required to be payload-hiding and partial attribute-hiding (only attribute-hiding for $\mathbf{x}_1$ and $\mathbf{y}_1$).

Then, we can achieve SME by encoding w and Φ into two vectors, such that if $w \in \Phi$, the inner product of the two vectors equals 0.

Finally, a concrete construction of dual IPE is presented in our paper [ZLH+23]. Therefore, we can construct an efficient fSMP.

1. Motivations & Contributions
2. Primitive of fNIZK
3. A generic construction of fNIZK
4. A concrete construction for set membership
5. References

References I

[BS23] Dan Boneh and Victor Shoup.
A graduate course in applied cryptography.
Draft 0.6, 2023.

[CCS08] Jan Camenisch, Rafik Chaabouni, and Abhi Shelat.
Efficient protocols for set membership and range proofs.
In *ASIACRYPT 2008*, pages 234–252. Springer, 2008.

[CHL05] Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya.
Compact e-cash.
In *EUROCRYPT 2005*, pages 302–321. Springer, 2005.

[CL01] Jan Camenisch and Anna Lysyanskaya.
An efficient system for non-transferable anonymous credentials with optional anonymity revocation.
In *EUROCRYPT 2001*, pages 93–118. Springer, 2001.

[FLS99] Uriel Feige, Dror Lapidot, and Adi Shamir.
Multiple noninteractive zero knowledge proofs under general assumptions.
SIAM J. Comput., 29(1):1–28, 1999.

[FS86] Amos Fiat and Adi Shamir.
How to prove yourself: Practical solutions to identification and signature problems.
In *CRYPTO 1986*, pages 186–194. Springer, 1986.

[Oka95] Tatsuaki Okamoto.
An efficient divisible electronic cash scheme.
In *CRYPTO 1995*, pages 438–451. Springer, 1995.

[Ped91] Torben Pryds Pedersen.
Non-interactive and information-theoretic secure verifiable secret sharing.
In *CRYPTO 1991*, pages 129–140. Springer, 1991.

References II

[SCG+14] Eli Ben Sasson, Alessandro Chiesa, Christina Garman, Matthew Green, Ian Miers, Eran Tromer, and Madars Virza.
Zerocash: Decentralized anonymous payments from bitcoin.
In *S&P 2014*, pages 459–474. IEEE, 2014.

[TG20] Syh-Yuan Tan and Thomas Groß.
Monipolyan expressive q-SDH-based anonymous attribute-based credential system.
In *ASIACRYPT 2020*, pages 498–526. Springer, 2020.

[ZLH+23] Gongxian Zeng, Junzuo Lai, Zhengan Huang, Linru Zhang, Xiangning Wang, Kwok-Yan Lam, Huaxiong Wang, and Jian Weng.
Non-interactive zero-knowledge functional proofs.
Cryptology ePrint Archive, 2023.


Thank you