# Identity-Based Identification Schemes

Guomin Yang

Centre for Computer and Information Security Research
School of Computing and Information Technology
University of Wollongong

# Outline

# Standard Identification (SI) Scheme

| **Prover (pk, sk)** | | **Verifier (pk)** |
|---|---|---|
| | $\xrightarrow{\text{P1}}$ | |
| | $\xleftarrow{\text{V1}}$ | |
| | $\xrightarrow{\text{P2}}$ | |
| | $\xleftarrow{\text{V2}}$ | |
| | $\ldots$ | |
| | $\xrightarrow{\text{Pn}}$ | Accept/Reject |

# Standard Identification (SI) Scheme

| **Prover (pk, sk)** | | **Verifier (pk)** |
|---|---|---|
| | P1 $\longrightarrow$ | |
| | $\longleftarrow$ V1 | |
| | P2 $\longrightarrow$ | |
| | $\longleftarrow$ V2 | |
| | ... | |
| | Pn $\longrightarrow$ | Accept/Reject |

1. Passive Attack (imp-pa)
   1. Stage 1: the attacker obtains communication transcripts between the prover and an honest verifier
   2. Stage 2: the attacker tries to impersonate the prover

# Standard Identification (SI) Scheme

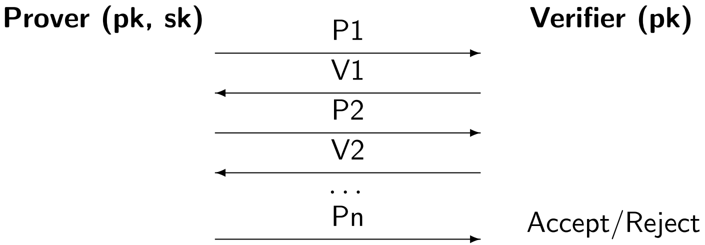
**Prover (pk, sk)** | | **Verifier (pk)**

$$\xrightarrow{\text{P1}}$$
$$\xleftarrow{\text{V1}}$$
$$\xrightarrow{\text{P2}}$$
$$\xleftarrow{\text{V2}}$$
$$\dots$$
$$\xrightarrow{\text{Pn}}$$

Accept/Reject

1. Passive Attack (imp-pa)
   1. Stage 1: the attacker obtains communication transcripts between the prover and an honest verifier
   2. Stage 2: the attacker tries to impersonate the prover
2. Active Attack (imp-aa)
   1. Stage 1: the attacker communicates with the prover as a verifier, one session at a time
   2. Stage 2: the attacker tries to impersonate the prover

# Standard Identification (SI) Scheme




1. Passive Attack (imp-pa)
   1. Stage 1: the attacker obtains communication transcripts between the prover and an honest verifier
   2. Stage 2: the attacker tries to impersonate the prover
2. Active Attack (imp-aa)
   1. Stage 1: the attacker communicates with the prover as a verifier, one session at a time
   2. Stage 2: the attacker tries to impersonate the prover
3. Concurrent Attack (imp-ca) similar to Active Attack, but the attacker can have concurrent sessions with the prover in Stage 1

# Identity Based Identification (IBI) Schemes

**PKG (mpk, msk)** **User (ID)**

$usk_{ID} = \mathrm{UKGen}(msk, ID)$

$\xrightarrow{\quad usk_{ID} \quad}$

**Prover ($usk_{ID}$)** **Verifier (mpk,ID)**

$\xrightarrow{\quad \mathrm{P1} \quad}$

$\xleftarrow{\quad \mathrm{V1} \quad}$

$\xrightarrow{\quad \mathrm{P2} \quad}$

$\xleftarrow{\quad \mathrm{V2} \quad}$

$\ldots$

$\xrightarrow{\quad \mathrm{Pn} \quad}$ Accept/Reject

# Canonical IBI

**PKG (mpk, msk)** **User (ID)**

$usk_{ID} = \mathrm{UKGen}(msk, ID)$

$\xrightarrow{\quad usk_{ID} \quad}$

*The identification protocol has three moves*

**Prover** ($usk_{ID}$) **Verifier (mpk,ID)**

$\xrightarrow{\quad \mathrm{Cmt} \quad}$

$\xleftarrow{\quad \mathrm{Ch} \quad}$

$\xrightarrow{\quad \mathrm{Rsp} \quad}$ Accept/Reject

# Canonical IBI to IBS

Algorithm $Sign(usk_{ID}, M)$

1. $(\mathsf{Cmt}, St) \leftarrow \mathsf{P}(usk_{ID})$
2. $\mathsf{Ch} = H(ID, \mathsf{Cmt}, M)$
3. $\mathsf{Rsp} \leftarrow \mathsf{P}(St, \mathsf{Ch})$
4. Return $\sigma = (\mathsf{Cmt}, \mathsf{Rsp})$

Algorithm $Ver(ID, M, \sigma)$

1. Parse $\sigma$ as $(\mathsf{Cmt}, \mathsf{Rsp})$
2. $\mathsf{Ch} = H(ID, \mathsf{Cmt}, M)$
3. Return $\mathsf{V}(ID, \mathsf{Cmt} \| \mathsf{Ch} \| \mathsf{Rsp})$

**Theorem (BNN'04)**

*If the canonical IBI scheme is secure under passive attacks, then the IBS scheme is existentially unforgeable under adaptive chosen message attacks in the random oracle model.*

# Bellare-Namprempre-Neven (BNN) IBI Framework

A Binary Relation $\mathbf{R} = \{(x_1, y_1), (x_2, y_1), \cdots, (x_m, y_n)\}$

Notations:

- $\langle \mathbf{R} \rangle$: description of $\mathbf{R}$
- $\mathbf{R}^{-1}(y) = \{x : (x, y) \in \mathbf{R}\}$

Trapdoor Samplable Relation:

- Samplable: easy to sample uniformly distributed $(x, y) \in \mathbf{R}$
- Regular: for any $y \neq y'$, $|\mathbf{R}^{-1}(y)| = |\mathbf{R}^{-1}(y')|$
- Invertible: $\exists$ a trapdoor $t$ such that $\mathsf{Inv}(t, y)$ outputs $x \in \mathbf{R}^{-1}(y)$

# SI to IBI Transformation

## Definition

A standard identification scheme SI is said to be **convertible** if its key space $\{(pk, sk)\}$ can be described by a trapdoor samplable relation $\mathbf{R}$.

# SI to IBI Transformation

## Definition

A standard identification scheme SI is said to be **convertible** if its key space $\{(pk, sk)\}$ can be described by a trapdoor samplable relation $\mathbf{R}$.

The Transformation:

Algorithm $\mathsf{MKGen}(1^k)$

1. Generate a relation $\mathbf{R}$ with trapdoor $t$
2. $mpk = \langle \mathbf{R} \rangle, msk = t$
3. Return $(mpk, msk)$

Algorithm $\mathsf{UKGen}(msk, I)$

1. Set $y = H(I)$
2. Compute $x \leftarrow \mathsf{Inv}(t, y)$; $usk[I] = x$
3. Return $usk[I]$

Prover: run the SI prover algorithm with secret key $usk[I]$
Verifier: run the SI verifier algorithm with public key $H(I)$

## Theorem

*The IBI is id-imp-atk secure if the SI is imp-atk secure for atk* $\in \{pa,aa,ca\}$.

# Example: the Guillou–Quisquater SI scheme (Crypto'88)

Algorithm $\mathrm{Kg}(1^k)$
$(N, e, d) \leftarrow \mathrm{K}_{\mathrm{rsa}}(1^k)$
$x \leftarrow \mathbb{Z}_N^*$
$X = x^e \bmod N$
$pk = ((N, e), X)$
$sk = ((N, e), x)$
Return $(pk, sk)$

| P | | V |
|---|---|---|
| $y \leftarrow \mathbb{Z}_N^*$ | | |
| $Y \leftarrow y^e \bmod N$ | $\xrightarrow{Y}$ | |
| | $\xleftarrow{c}$ | $c \leftarrow \mathbb{Z}_{2^{m(k)}}$ |
| $z = x^c y \bmod N$ | $\xrightarrow{z}$ | |
| | | If $z^e \equiv X^c Y \bmod N$ then acc else rej |

GQ is imp-pa secure under the RSA assumption.

# Example: the Guillou–Quisquater SI scheme (Crypto'88)

Algorithm $\mathsf{Kg}(1^k)$
$(N, e, d) \leftarrow \mathsf{K}_{\mathsf{rsa}}(1^k)$
$x \leftarrow \mathbb{Z}_N^*$
$X = x^e \bmod N$
$pk = ((N, e), X)$
$sk = ((N, e), x)$
Return $(pk, sk)$

$$
\begin{array}{lcl}
\text{P} & & \text{V} \\
y \leftarrow \mathbb{Z}_N^* & & \\
Y \leftarrow y^e \bmod N & \xrightarrow{\quad Y \quad} & \\
 & \xleftarrow{\quad c \quad} & c \leftarrow \mathbb{Z}_{2^{m(k)}} \\
z = x^c y \bmod N & \xrightarrow{\quad z \quad} & \\
 & & \text{If } z^e \equiv X^c Y \bmod N \\
 & & \text{then acc else rej}
\end{array}
$$

GQ is imp-pa secure under the RSA assumption.

$$\mathbf{R} = \{(x, X) \in \mathbb{Z}_N^* \times \mathbb{Z}_N^* : X = x^e \bmod N\}$$

$$\langle \mathbf{R} \rangle = (N, e), \ \text{Trapdoor: } (N, d)$$

Regular: **R** is a one-to-one mapping

# Example: the transformed IBI scheme from GQ-SI

MKGen$(1^k)$

$(N, e, d) \leftarrow \mathsf{K}_{\mathsf{rsa}}(1^k)$

$msk \leftarrow (N, d)$

$mpk \leftarrow (N, e)$

UKGen$(msk, I)$

$X \leftarrow H(I)$

$x = X^d \bmod N$

Return $usk[I] = x$

| P$(usk[I])$ | | V$(I)$ |
|---|---|---|
| $y \leftarrow \mathbb{Z}_N^*$ | | |
| $Y \leftarrow y^e \bmod N$ | | |
| | $\xrightarrow{Y}$ | |
| | $\xleftarrow{c}$ | $c \leftarrow \mathbb{Z}_{2^{m(k)}}$ |
| $z = usk[I]^c y \bmod N$ | $\xrightarrow{z}$ | |
| | | If $z^e \equiv H(I)^c Y \bmod N$ then acc else rej |

# IBI Framework 2 (Passive Security)

A Binary Relation $\mathbf{R} = \{(x_1, y_1), (x_2, y_1), \cdots, (x_m, y_n)\}$.
**Trapdoor Weak-one-more Relation (TWR)**:

- Invertible: $\exists$ a trapdoor $t$ such that $\mathsf{Inv}(t, y)$ outputs $x \in \mathbf{R}^{-1}(y)$
- Weak-one-more secure: given

$$\{y_1, y_2, \cdots, y_k\} \text{ and } \{x_1, x_2, \cdots, x_{k-1}\}$$

such that $(x_j, y_j) \in \mathbf{R}$ for $1 \le j \le k-1$, difficult to find $x_k$.

# TWR Instantiations

1. RSA $(N, e)$

$$\mathbf{R}^{RSA} = \{(x, y) \in \mathbb{Z}_N^* : x^e = y \bmod N\}$$

with $t = (N, d)$

# TWR Instantiations

1. RSA $(N, e)$

$$\mathbf{R}^{RSA} = \{(x, y) \in \mathbb{Z}_N^* : x^e = y \bmod N\}$$

with $t = (N, d)$

2. CDH in a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, q, P, S, e)$

$$\mathbf{R}^{CDH} = \{(x, y) \in \mathbb{G}_1^2 : e(P, x) = e(S, y)\}$$

with $t = \log_P S$

# TWR Instantiations

1. RSA $(N, e)$

$$\mathbf{R}^{RSA} = \{(x, y) \in \mathbb{Z}_N^* : x^e = y \bmod N\}$$

   with $t = (N, d)$

2. CDH in a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, q, P, S, e)$

$$\mathbf{R}^{CDH} = \{(x, y) \in \mathbb{G}_1^2 : e(P, x) = e(S, y)\}$$

   with $t = \log_P S$

3. Digital signature scheme $SIG$ (UF-KMA)

$$\mathbf{R}^{SIG} = \{(\sigma, m) : \mathcal{V}(pk, m, \sigma) = 1\}$$

   with $t = sk$

Remark: the third one is not samplable

# HVZK-PoK System

**Prover (x)** **Verifier (y)**

$$\xrightarrow{\text{Cmt}}$$
$$\xleftarrow{\text{Ch}}$$
$$\xrightarrow{\text{Rsp}}$$

Accept/Reject

1. **Honest Verifier Zero Knowledge**. There exists a simulation algorithm that can simulate a protocol transcript without knowing the prover's secret key $x$.
2. **Special Soundness**. Given two transcripts $(\text{Cmt}, \text{Ch}_1, \text{Rsp}_1)$ and $(\text{Cmt}, \text{Ch}_2, \text{Rsp}_2)$ where $\text{Ch}_1 \neq \text{Ch}_2$, we can extract the prover's secret key $x$.

# TWR + HVZK-PoK = IBI with imp-pa security

Algorithm MKGen($1^k$)

1. Generate a TWR relation $\mathbf{R}$ with trapdoor $t$
2. $mpk = \langle \mathbf{R} \rangle, msk = t$
3. Return $(mpk, msk)$

Algorithm UKGen$(msk, I)$

1. Set $y = H(I)$
2. Compute $x \leftarrow \mathsf{Inv}(t, y)$; $usk[I] = x$
3. Return $usk[I]$

Prover: run the HVZK-PoK prover algorithm with secret key $usk[I]$
Verifier: run the HVZK-PoK verifier algorithm with public key $H(I)$

# TWR + HVZK-PoK example

TWR: a bilinear group $(\mathbb{G}_1, \mathbb{G}_2, q, P, e)$ and $Q \in \mathbb{G}_1$

$$\mathbf{R} = \{(x, y) : x \in \mathbb{G}_1;\ y \in \mathbb{Z}_q;\ e(yP + Q, x) = e(P, P)\}$$

with trapdoor $t = \log_P Q$.

- Invertible: Given $y \in \mathbb{Z}_q$, compute $x = \frac{1}{y+t}P$.
- Weak-one-more secure: under the assumption that the K-CAA problem is hard
  " Given $(P, Q, h_1, \cdots, h_K, \frac{1}{h_1+t}P, \cdots, \frac{1}{h_K+t}P)$, compute $(h, \frac{1}{h+t}P)$ such that $h \notin \{h_1, h_2, ..., h_K\}$."
- This relation is NOT samplable!

# TWR + HVZK-PoK = IBI with imp-pa security

An HVZK-PoK for the TWR

1. Cmt $:= r(yP + Q)$ where $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$
2. Ch $:= c$ where $c \stackrel{R}{\leftarrow} \mathbb{Z}_q$
3. Rsp $:= rcP + x$

Verification:

$$e(\mathrm{Rsp}, yP + Q) = e(P, P)e(P, \mathrm{Cmt})^c.$$

# IBI Framework 2 (Active and Concurrent Security)

A Binary Relation $\mathbf{R} = \{(x_1, y_1), (x_2, y_1), \cdots, (x_m, y_n)\}$.
**Trapdoor Strong-one-more Relation (TWR)**:

- Invertible: $\exists$ a trapdoor $t$ such that $\mathsf{Inv}(t, y)$ outputs $x \in \mathbf{R}^{-1}(y)$
- Strong-one-more secure: given

$$\{y_1, y_2, \cdots, y_k\} \text{ and } \{x_1, x_2, \cdots, x_k\}$$

  such that $(x_j, y_j) \in \mathbf{R}$ for $1 \le j \le k$, difficult to find $x'_i \neq x_i \wedge (x'_i, y_i) \in \mathbf{R}$ for any $i$.

# TSR Instantiations

1. RSA: $(N, e, g)$ where $g \xleftarrow{R} \mathbb{Z}_N^*$.

$$\mathbf{R}^{RSA} = \{((x_1, x_2), Y) \in (\mathbb{Z}_e \times \mathbb{Z}_N^*) \times \mathbb{Z}_N^* : g^{-x_1} x_2^{-e} \equiv Y \bmod N\}$$

with $t = (N, d)$.

2. Digital signature scheme $SIG$ (SUF-KMA)

$$\mathbf{R}^{SIG} = \{(\sigma, m) : \mathcal{V}(pk, m, \sigma) = 1\}$$

with $t = sk$.

# Witness Indistinguishable Proof

Let $(x, y) \in \mathbf{R}$ and $(x', y) \in \mathbf{R}$ for a many-to-one TSR $\mathbf{R}$.

**Prover (x)** | **Verifier (y)**

Cmt $\longrightarrow$

Ch $\longleftarrow$

Rsp $\longrightarrow$

Accept/Reject

**Prover (x')** | **Verifier (y)**

Cmt' $\longrightarrow$

Ch' $\longleftarrow$

Rsp' $\longrightarrow$

Accept/Reject

Witness Indistinguishability (WI): $(Cmt, Ch, Rsp)$ and $(Cmt', Ch', Rsp')$ are computationally indistinguishable.

Fact: WI is preserved under concurrent protocol execution (Feige-Shamir STOC'90)

# TSR + WI-PoK = IBI with imp-aa/ca security

Construction: same as TWR + HVZK-PoK

Proof idea: If $\mathcal{A}$ can break the concurrent security of the IBI scheme, $\mathcal{B}$ can solve the Strong-one-more problem.

- $\mathcal{B}$ simulates Stage 1 using $x$.
- If $\mathcal{A}$ can impersonate the prover in Stage 2, $\mathcal{B}$ can extract $\hat{x}$ from $\mathcal{A}$ such that $(\hat{x}, y) \in \mathbf{R}$ because of the Proof ok Knowledge property.
- Since Stage 1 is witness indistinguishable, with probability at least $1/2$, $\hat{x} \neq x$.

# TSR + WI-PoK example (Okamoto-IBI Crypto'92)

RSA based TSR:

$$\mathbf{R}^{RSA} = \{((x_1, x_2), Y) \in (\mathbb{Z}_e \times \mathbb{Z}_N^*) \times \mathbb{Z}_N^* : g^{-x_1} x_2^{-e} \equiv Y \bmod N\}$$

WI-PoK:

$$\begin{array}{lcl}
\mathsf{P}(x_1, x_2) & & \mathsf{V}(Y) \\
y_1 \xleftarrow{R} \mathbb{Z}_e;\ y_2 \xleftarrow{R} \mathbb{Z}_N^* & & \\
\mathsf{Cmt} \leftarrow g^{y_1} y_2^e \bmod N & & \\
 & \xrightarrow{\quad \mathsf{Cmt} \quad} & c \xleftarrow{R} \mathbb{Z}_{2^{\lambda(k)}} \\
 & \xleftarrow{\quad c \quad} & \\
z_1 \leftarrow y_1 + cx_1 \bmod e & & \\
z_2 \leftarrow g^{\lfloor (y_1 + cx_1)/e \rfloor} y_2 x_2^c \bmod N & \xrightarrow{\quad \mathsf{Rsp} = (z_1, z_2) \quad} & \\
 & & \mathsf{Cmt} \stackrel{?}{=} g^{z_1} z_2^e Y^c \bmod N
\end{array}$$