A Tightly Secure Identity-based Signature Scheme from Isogenies

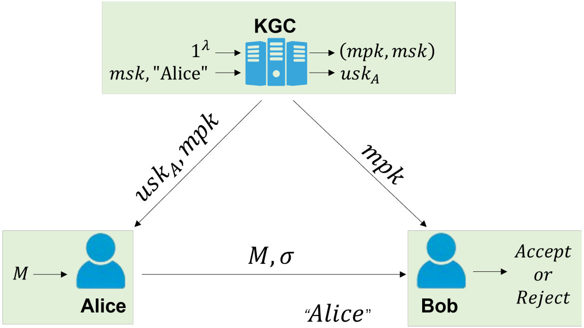
Jiawei Chen[1] Hyungrok Jo[1] Shingo Sato[1] Junji Shikata[1]

[1]Yokohama National University, Japan

PQCrypto2023 Session I

Identity-based Signature (IBS)

- IBS aims to simplify the public-key infrastructure (PKI) requirement when mapping the public key to user's identities.



$\boldsymbol{mpk}$: master public key.
$\boldsymbol{M}$: message.
$\boldsymbol{usk_A}$: user's secret key.
$\boldsymbol{msk}$: master secret key.
$\boldsymbol{\sigma}$: signature.
"**Alice**": Alice's ID.

Figure 1: The framework of IBS

Isogeny-based cryptography

- One of post-quantum cryptography
- Hard problem: Given two supersingular elliptic curves over finite fields E, E', compute isogeny $\phi : E \to E'$.
- Two main isogeny-based key exchange protocols:
  - Supersingular Isogeny Diffie Hellman(SIDH) ✗

Isogeny-based cryptography

- One of post-quantum cryptography
- Hard problem: Given two supersingular elliptic curves over finite fields E, E', compute isogeny $\phi : E \to E'$.
- Two main isogeny-based key exchange protocols:
    - Supersingular Isogeny Diffie Hellman(SIDH) ✗
    - Commutative Supersingular Isogeny Diffie Hellman(CSIDH) ✓

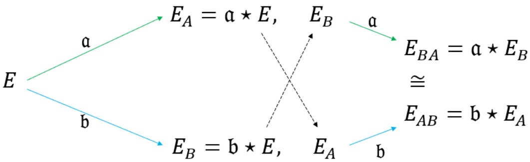


Figure 2: CSIDH key exchange protocol

IBS from isogenies

Digital Signature	IBS
SeaSign	Peng et al.[1]
CSI-FiSh	Shaw and Dutta(SD)[2]
Lossy CSI-FiSh[3]	Our Scheme

Table 1: Brief history of IBS from isogenies

- There exist some flaws in IBS of Peng et al.
- Security proof of SD uses rewind technology, hence reduction is not tight.

[1] Cong Peng et al. "CsiIBS: A post-quantum identity-based signature scheme based on isogenies". In: *Journal of Information Security and Applications* 54 (2020), p. 102504.

[2] Surbhi Shaw and Ratna Dutta. "Identification Scheme and Forward-Secure Signature in Identity-Based Setting from Isogenies". In: *ProvSec*. Vol. 13059. LNCS. Springer, 2021, pp. 309–326.

[3] Ali El Kaafarani, Shuichi Katsumata, and Federico Pintore. "Lossy CSI-FiSh: Efficient Signature Scheme with Tight Reduction to Decisional CSIDH-512". In: *Public Key Cryptography (2)*. Vol. 12111. LNCS. Springer, 2020, pp. 157–186.

Provable security reduction

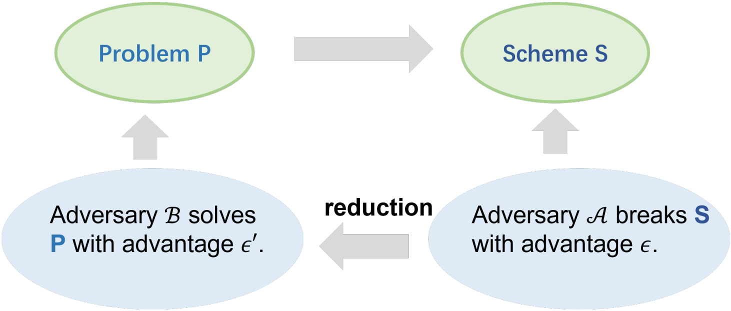


Figure 3: Security reduction

Security loss: $L = \frac{\epsilon}{\epsilon'}$
Tight reduction: $L = O(1)$

Motivation and Contribution

Motivation

- The security reduction of SD is not tight.
- Digital signature of Pan and Wagner[4]+certificate transform $\Longrightarrow$ limited efficient tightly secure IBS[5] (PW).

[4] Jiaxin Pan and Benedikt Wagner. "Lattice-Based Signatures with Tight Adaptive Corruptions and More". In: *Public Key Cryptography (2)*. Vol. 13178. LNCS. Springer, 2022, pp. 347–378.

[5] Youngkyung Lee et al. "Tight security for the generic construction of identity-based signature (in the multi-instance setting)". In: *Theoretical Computer Science* 847 (2020), pp. 122–133.

Motivation and Contribution

Motivation

- The security reduction of SD is not tight.
- Digital signature of Pan and Wagner[4]+certificate transform $\Longrightarrow$ limited efficient tightly secure IBS[5] (PW).

Contribution

- We present an IBS scheme from the lossy CSI-FiSh and prove its tight security reduction.
- Smaller USK-size and Signature-size than PW when one of the parameters S_1 is chosen properly (e.g. $\leq 2^8 - 1$).

[4] Jiaxin Pan and Benedikt Wagner. "Lattice-Based Signatures with Tight Adaptive Corruptions and More". In: *Public Key Cryptography (2)*. Vol. 13178. LNCS. Springer, 2022, pp. 347–378.

[5] Youngkyung Lee et al. "Tight security for the generic construction of identity-based signature (in the multi-instance setting)". In: *Theoretical Computer Science* 847 (2020), pp. 122–133.

Syntax

An IBS scheme consists of four polynomial-time algorithms $(Setup, Ext, Sgn, Vrf)$ where

- $Setup(1^\lambda) \rightarrow (mpk, msk)$
- $Ext(mpk, msk, id) \rightarrow usk_{id}$
- $Sgn(mpk, usk_{id}, m, id) \rightarrow \sigma$
- $Vrf(mpk, id, m, \sigma) \rightarrow Accept/Reject$

Syntax

An IBS scheme consists of four polynomial-time algorithms $(\mathit{Setup}, \mathit{Ext}, \mathit{Sgn}, \mathit{Vrf})$ where

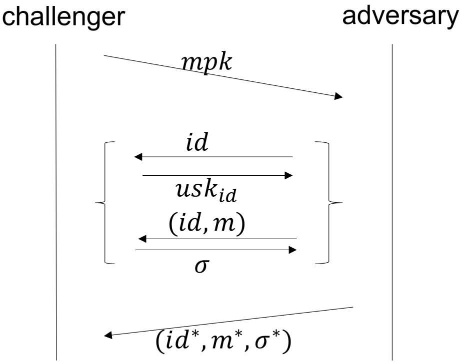
- $\mathit{Setup}(1^\lambda) \to (\mathit{mpk}, \mathit{msk})$
- $\mathit{Ext}(\mathit{mpk}, \mathit{msk}, \mathit{id}) \to \mathit{usk}_{\mathit{id}}$
- $\mathit{Sgn}(\mathit{mpk}, \mathit{usk}_{\mathit{id}}, m, \mathit{id}) \to \sigma$
- $\mathit{Vrf}(\mathit{mpk}, \mathit{id}, m, \sigma) \to \mathit{Accept}/\mathit{Reject}$

Requirement of correctness

$$\mathit{Vrf}(\mathit{mpk}, \mathit{id}, m, \mathit{Sgn}(\mathit{mpk}, \mathit{usk}_{\mathit{id}}, m, \mathit{id})) = \mathit{Accept}$$

Security Definition

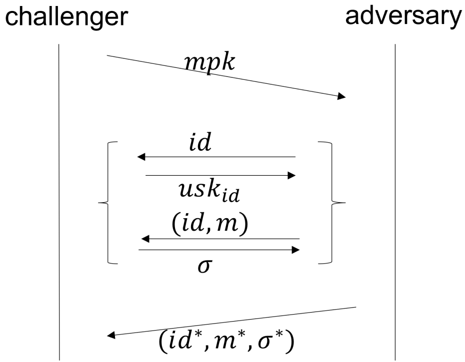
EUF-CMA: security game



$$\boldsymbol{Vrf}(mpk, id^*, m^*, \sigma^*) \stackrel{?}{=} Accept$$

Security Definition

EUF-CMA: security game



$$\boldsymbol{Vrf}(mpk, id^*, m^*, \sigma^*) \stackrel{?}{=} Accept$$

EUF-CMA-MK:

- The adversary cannot query *id* if (id, m) has been queried.

CSIDH setting

A group G acts freely and transitively on a set $\mathcal{X}$

$$\star : G \times \mathcal{X} \to \mathcal{X}$$

- (Identity) $e \star x = x$.
- (Compatibility) $g_1 \star (g_2 \star x) = (g_1 g_2) \star x$.
- $g \mapsto g \star \mathcal{X}$ is bijective.

CSIDH setting

A group G acts freely and transitively on a set $\mathcal{X}$

$$\star : G \times \mathcal{X} \to \mathcal{X}$$

- (Identity) $e \star x = x$.
- (Compatibility) $g_1 \star (g_2 \star x) = (g_1 g_2) \star x$.
- $g \mapsto g \star \mathcal{X}$ is bijective.

- G =the ideal class group $Cl(\mathcal{O})$ of a quadratic order $\mathcal{O} \subset \mathbb{Q}(\sqrt{-p})$
- $\mathcal{X}$ =the set of supersingular elliptic curves $E/\mathbb{F}_p$ such that $End_p(E) \cong \mathcal{O}$

(Lossy) CSI-FiSh assumption: $Cl(\mathcal{O}) =< \mathfrak{g} >$ (only holds at CSIDH-512.)

Lossy CSI-FiSh identification

$$pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}\}$$
$$\mathcal{R} := \{((E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), a) | E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2\}$$

Lossy CSI-FiSh identification

$$pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}\}$$
$$\mathcal{R} := \{((E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), a) | E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2\}$$

Prover		Verifier
$(sk = a)$		$(pk = \{(E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}))$
$r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}$	$\xrightarrow{F_1, F_2}$	
	$\xleftarrow{ch}$	$ch \xleftarrow{\$} \{0, 1\}$
$(ch = 0), resp = r, (ch = 1), resp = r - a$	$\xrightarrow{resp}$	$(ch = 0), F_i = \mathfrak{g}^r \star E_i^{(0)}$
		$(ch = 1), F_i = \mathfrak{g}^{r-a} \star E_i^{(1)}$

Figure 4: Base lossy CSI-FiSh identification

Enlarge the challenge space

- Repeat T times.
  - Choose $r_1, \cdots, r_T \xleftarrow{\$} \mathbb{Z}_N$ and compute $\{F_{i,j} = \mathfrak{g}^{r_j} \star E_i^{(0)}\}_{i=1,2; j=1,\cdots,T}$
  - Signing time becomes T times larger

Enlarge the challenge space

- Repeat T times.
  - Choose $r_1, \cdots, r_T \xleftarrow{\$} \mathbb{Z}_N$ and compute $\{F_{i,j} = \mathfrak{g}^{r_j} \star E_i^{(0)}\}_{i=1,2;j=1,\cdots,T}$
  - Signing time becomes T times larger
- Use S public keys.

Prover		Verifier
$(sk = \{a_j \vert j = 1, \cdots, S\})$		$(pk = \{E_1^{(0)}, E_2^{(0)}, E_1^{(j)}, E_2^{(j)} \vert j = 1, \cdots, S\})$
$r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}$	$\xrightarrow{F_1, F_2}$	
	$\xleftarrow{ch}$	$ch \xleftarrow{\$} \{0, 1, \cdots, S\}$
$(ch = 0), resp = r, (ch = j), resp = r - a_j$	$\xrightarrow{resp}$	$(ch = 0), F_i = \mathfrak{g}^r \star E_i^{(0)}$
		$(ch = j), F_i = \mathfrak{g}^{r-a_j} \star E_i^{(j)}$

Enlarge the challenge space

- Repeat T times.
  - Choose $r_1, \cdots, r_T \xleftarrow{\$} \mathbb{Z}_N$ and compute $\{F_{i,j} = \mathfrak{g}^{r_j} \star E_i^{(0)}\}_{i=1,2;j=1,\cdots,T}$
  - Signing time becomes T times larger
- Use S public keys.

Prover		Verifier
$(sk = \{a_j \mid j = 1, \cdots, S\})$		$(pk = \{E_1^{(0)}, E_2^{(0)}, E_1^{(j)}, E_2^{(j)} \mid j = 1, \cdots, S\})$
$r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}$	$\xrightarrow{F_1, F_2}$	
	$\xleftarrow{ch}$	$ch \xleftarrow{\$} \{0, 1, \cdots, S\}$
$(ch = 0), resp = r, (ch = j), resp = r - a_j$	$\xrightarrow{resp}$	$(ch = 0), F_i = \mathfrak{g}^r \star E_i^{(0)}$ $(ch = j), F_i = \mathfrak{g}^{r - a_j} \star E_i^{(j)}$

- To achieve λ security level, $T \cdot log(S + 1) \geq \lambda$.

Our proposal from the base Lossy CSI-FiSh

Algorithm $Setup(1^\lambda) \rightarrow mpk, msk$

1: $pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}, H, H'\}$
2: $a, b, c \xleftarrow{\$} \mathbb{Z}_N$
3: compute $E_1^{(0)} = \mathfrak{g}^b \star E_0, E_2^{(0)} = \mathfrak{g}^c \star E_0$
4: compute $E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2$
5: **return** $mpk = (pp, E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), msk = a, b, c$

$H, H' : \{0,1\}^* \rightarrow \{0,1\}$: random oracles

Our proposal from the base Lossy CSI-FiSh

Algorithm $Setup(1^\lambda) \rightarrow mpk, msk$

1: $pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}, H, H'\}$
2: $a, b, c \xleftarrow{\$} \mathbb{Z}_N$
3: compute $E_1^{(0)} = \mathfrak{g}^b \star E_0, E_2^{(0)} = \mathfrak{g}^c \star E_0$
4: compute $E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2$
5: **return** $mpk = (pp, E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), msk = a, b, c$

Algorithm $Ext(mpk, msk, id) \rightarrow usk$

1: $r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}, i = 1, 2$
2: $ch \leftarrow H(F_1, F_2, id)$
3: $(ch = 0), resp = r; (ch = 1), resp = r - a$
4: **return** $usk = (F_1, F_2, resp)$

$H, H' : \{0,1\}^* \rightarrow \{0,1\}$: random oracles

Our proposal from the base Lossy CSI-FiSh

Algorithm $Setup(1^\lambda) \to mpk, msk$

1: $pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}, H, H'\}$
2: $a, b, c \xleftarrow{\$} \mathbb{Z}_N$
3: compute $E_1^{(0)} = \mathfrak{g}^b \star E_0, E_2^{(0)} = \mathfrak{g}^c \star E_0$
4: compute $E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2$
5: **return** $mpk = (pp, E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), msk = a, b, c$

Algorithm $Ext(mpk, msk, id) \to usk$

1: $r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}, i = 1, 2$
2: $ch \leftarrow H(F_1, F_2, id)$
3: $(ch = 0), resp = r; (ch = 1), resp = r - a$
4: **return** $usk = (F_1, F_2, resp)$

$H, H' : \{0,1\}^* \to \{0,1\}$: random oracles

Our proposal from the base Lossy CSI-FiSh

Algorithm $Setup(1^\lambda) \to mpk, msk$

1: $pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}, H, H'\}$
2: $a, b, c \xleftarrow{\$} \mathbb{Z}_N$
3: compute $E_1^{(0)} = \mathfrak{g}^b \star E_0, E_2^{(0)} = \mathfrak{g}^c \star E_0$
4: compute $E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2$
5: **return** $mpk = (pp, E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), msk = a, b, c$

Algorithm $Ext(mpk, msk, id) \to usk$

1: $r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}, i = 1, 2$
2: $ch \leftarrow H(F_1, F_2, id)$
3: $(ch = 0), resp = r; (ch = 1), resp = r - a$
4: **return** $usk = (F_1, F_2, resp)$

$H, H' : \{0,1\}^* \to \{0,1\}$: random oracles

Algorithm $Sgn(mpk, usk, m, id) \to \sigma$

1: parse usk as $(F_1, F_2, resp)$
2: compute $ch = H(F_1, F_2, id)$
3: $r' \xleftarrow{\$} \mathbb{Z}_N, F_i' = \mathfrak{g}^{r'} \star E_i^{(ch)}, i = 0, 1$
4: $ch' \leftarrow H'(F_1', F_2', m, id)$
5: $(ch' = 0), resp' = r'; (ch' = 1), resp' = r' - resp$
6: **return** $\sigma = (F_1, F_2, ch', resp')$

Our proposal from the base Lossy CSI-FiSh

Algorithm $Setup(1^\lambda) \to mpk, msk$

1: $pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}, H, H'\}$
2: $a, b, c \xleftarrow{\$} \mathbb{Z}_N$
3: compute $E_1^{(0)} = \mathfrak{g}^b \star E_0, E_2^{(0)} = \mathfrak{g}^c \star E_0$
4: compute $E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2$
5: **return** $mpk = (pp, E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), msk = a, b, c$

Algorithm $Ext(mpk, msk, id) \to usk$

1: $r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}, i = 1, 2$
2: $ch \leftarrow H(F_1, F_2, id)$
3: $(ch = 0), resp = r; (ch = 1), resp = r - a$
4: **return** $usk = (F_1, F_2, resp)$

$H, H' : \{0,1\}^* \to \{0,1\}$: random oracles

Algorithm $Sgn(mpk, usk, m, id) \to \sigma$

1: parse usk as $(F_1, F_2, resp)$
2: compute $ch = H(F_1, F_2, id)$
3: $r' \xleftarrow{\$} \mathbb{Z}_N, F_i' = \mathfrak{g}^{r'} \star E_i^{(ch)}, i = 0, 1$
4: $ch' \leftarrow H'(F_1', F_2', m, id)$
5: $(ch' = 0), resp' = r'; (ch' = 1), resp' = r' - resp$
6: **return** $\sigma = (F_1, F_2, ch', resp')$

Our proposal from the base Lossy CSI-FiSh

Algorithm $Setup(1^\lambda) \to mpk, msk$

$pp = \{p, \mathfrak{g}, N = \#Cl(\mathcal{O}), E_0 \in \mathcal{X}, H, H'\}$

$a, b, c \xleftarrow{\$} \mathbb{Z}_N$

compute $E_1^{(0)} = \mathfrak{g}^b \star E_0, E_2^{(0)} = \mathfrak{g}^c \star E_0$

compute $E_i^{(1)} = \mathfrak{g}^a \star E_i^{(0)}, i = 1, 2$

return $mpk = (pp, E_1^{(0)}, E_2^{(0)}, E_1^{(1)}, E_2^{(1)}), msk = a, b, c$

Algorithm $Ext(mpk, msk, id) \to usk$

$r \xleftarrow{\$} \mathbb{Z}_N, F_i = \mathfrak{g}^r \star E_i^{(0)}, i = 1, 2$

$ch \leftarrow H(F_1, F_2, id)$

$(ch = 0), resp = r; (ch = 1), resp = r - a$

return $usk = (F_1, F_2, resp)$

$H, H' : \{0,1\}^* \to \{0,1\}$: random oracles

Algorithm $Sgn(mpk, usk, m, id) \to \sigma$

parse usk as $(F_1, F_2, resp)$

compute $ch = H(F_1, F_2, id)$

$r' \xleftarrow{\$} \mathbb{Z}_N, F_i' = \mathfrak{g}^{r'} \star E_i^{(ch)}, i = 0, 1$

$ch' \leftarrow H'(F_1', F_2', m, id)$

$(ch' = 0), resp' = r'; (ch' = 1), resp' = r' - resp$

return $\sigma = (F_1, F_2, ch', resp')$

Algorithm $Vrf(mpk, id, m, \sigma) \to Accept/Reject$

parse σ as $(F_1, F_2, ch', resp')$

compute F_1', F_2' from $ch', resp'$

compute $ch = H(F_1, F_2, id)$

if $F_i' = \mathfrak{g}^{resp'} \star E_i^{(ch)}$, $(ch' = 0)$, or $F_i' = \mathfrak{g}^{resp'} \star F_i$, $(ch' = 1)$, **return** $Accept$;

otherwise **return** $Reject$.

Enlarge the hash space

- Use S_0 mpk and repeat T_1 times to enlarge the hash space of H
- Use T_2, S_1 to enlarge the hash space of H'.

Enlarge the hash space

- Use S_0 mpk and repeat T_1 times to enlarge the hash space of H
- Use T_2, S_1 to enlarge the hash space of H'.
- To achieve λ security level

$$T_1 \cdot log(S_0 + 1) \geq \lambda$$
$$T_1 T_2 \cdot log(S_1 + 1) \geq \lambda$$

Comparison

	security bound	security model
SD	$\sqrt{q \cdot \epsilon} + negl$	
PW	$2S_0\epsilon + negl$	EUF-CMA
Our scheme	$S_0\epsilon + negl$	EUF-CMA-MK[6]

Figure 5: Comparison with SD and PW

q: the maximum number of query to the random oracle.
ϵ: the maximum probability of breaking the underlying computational problem.
S_0: parameter of the corresponding computational assumptions.

[6] The adversary can not query *id* if (id, m) has been queried.

Comparison

(T_1, T_2, S_0, S_1)	PW		Our Scheme	
	USK	Signature	USK	Signature
(16,3,255,7)	74.0KB	66.9KB	3.7KB	8.7KB
(16,2,255,15)	74.0KB	66.9KB	8.0KB	16.4KB
(8,2,65535,255)	18.9MB	16.8MB	69.9KB	131.1KB
(8,1,65535,65535)	18.9MB	16.8MB	18.0MB	33.6MB

Figure 6: Comparison with PW under the 128-bit security level

USK: user's secret key.
T_1, T_2: the numbers of parallel executions of the underlying (lossy) identification scheme.
S_0, S_1: parameters of the corresponding computational assumptions.

Conclusion and Future Work

Conclusion

- A tightly secure IBS scheme based on the lossy CSI-FiSh.
- When the parameter S_1 is chosen properly, the key-size and signature-size of our IBS are smaller than PW.

[7] Luca De Feo et al. "SCALLOP: Scaling the CSI-FiSh". In: *Public Key Cryptography (1)*. Vol. 13940. LNCS. Springer, 2023, pp. 345–375.

[8] Luca De Feo et al. "SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *ASIACRYPT (1)*. Vol. 12491. LNCS. Springer, 2020, pp. 64–93.

Conclusion and Future Work

Conclusion

- A tightly secure IBS scheme based on the lossy CSI-FiSh.
- When the parameter S_1 is chosen properly, the key-size and signature-size of our IBS are smaller than PW.

Future Work

- SCALLOP[7] $\Longrightarrow$ larger $Cl(\mathcal{O}) \Longrightarrow$ expected quantum security.
- Construct IBS from other Isogney-based signature schemes, such as SQISign[8].

[7] Luca De Feo et al. "SCALLOP: Scaling the CSI-FiSh". In: *Public Key Cryptography (1)*. Vol. 13940. LNCS. Springer, 2023, pp. 345–375.

[8] Luca De Feo et al. "SQISign: Compact Post-quantum Signatures from Quaternions and Isogenies". In: *ASIACRYPT (1)*. Vol. 12491. LNCS. Springer, 2020, pp. 64–93.

Thank You

chen-jiawei-hm@ynu.jp