Threshold Batched Identity-Based Encryption from Pairings in the Plain Model

Junqing Gong, Brent Waters, Hoeteck Wee, and **David Wu**

May 2026

Batch Decryption

[CGPP24, SAA24, BFOQ25, AFP25, …]

*Take a collection of **ciphertexts** under the same public key*

*and an **arbitrary** set S*

Batch Decryption

[CGPP24, SAA24, BFOQ25, AFP25, …]

*Take a collection of **ciphertexts** under the same public key*

*and an **arbitrary** set* $S = \{\mathrm{ct}_2, \mathrm{ct}_4, \mathrm{ct}_5\}$,

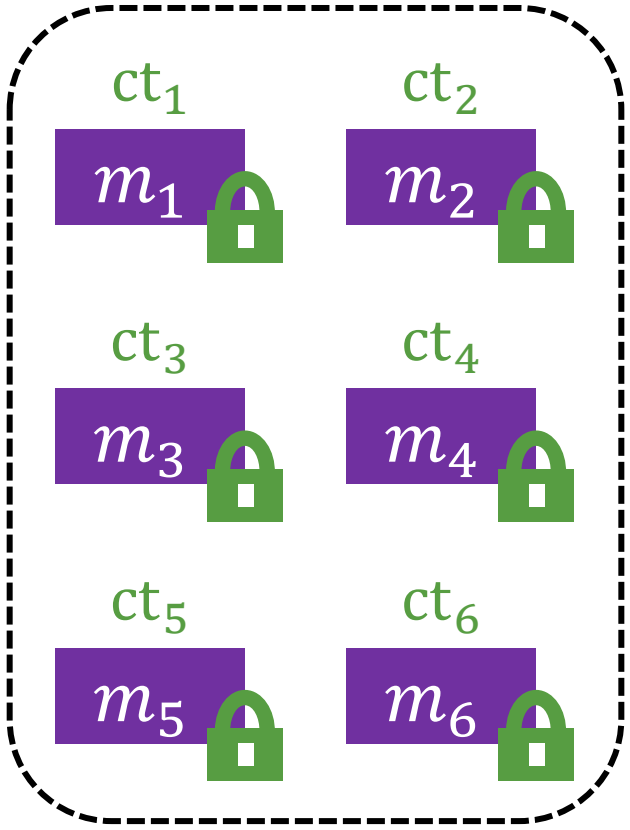
Goal: output a decryption key sk_S such that

$+$ m_i $\longrightarrow$ m_i if $\mathrm{ct}_i \in S$

m_i $\longrightarrow$ $\perp$ if $\mathrm{ct}_i \notin S$

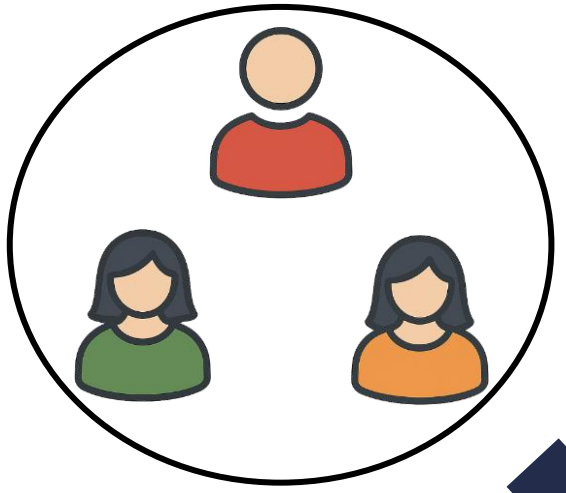
$$|\mathrm{sk}_S| = o(|S|)$$

Application: Succinct “Commit-then-Open”



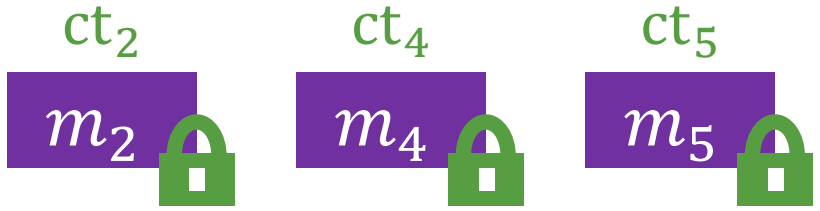
Parties submit encrypted messages to a decryption committee

Decryption committee
(with shares of master key)



Issue decryption key
for selected subset

Choose subset of messages to process



Chosen **without** knowledge of contents
(as long as some decrypters are honest)

Can be used to reveal selected messages

Example scenario:
encrypted mempools
(cf. [CGPP24, SAA24, BFOQ25, AFP25])

Batched Identity-Based Encryption (IBE)

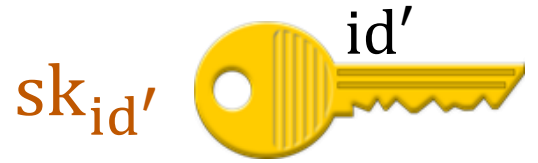
[AFP25, CGPW25, SAS24]

Batched IBE: convenient abstraction to build batch decryption

---

Vanilla IBE: ciphertexts are associated with an identity id

secret keys are associated with an identity id'



Correctness: decryption succeeds if $\mathrm{id} = \mathrm{id}'$

Security: ct computationally hides m if $\mathrm{id} \neq \mathrm{id}'$

Batched Identity-Based Encryption (IBE)

[AFP25, CGPW25, SAS24]

Batched IBE: convenient abstraction to build batch decryption

---

Batched IBE: ciphertexts are associated with an identity id

secret keys are associated with set of identities S

Correctness: decryption succeeds if $\mathrm{id} \in S$

Security: ct computationally hides m if $\mathrm{id} \notin S$

Succinctness: $|\mathrm{sk}_S| = \mathrm{poly}(\lambda)$; size of secret key does not depend on the size of the set

"dual of broadcast encryption:" both capture set membership policies

Batched IBE to Batch Decryption

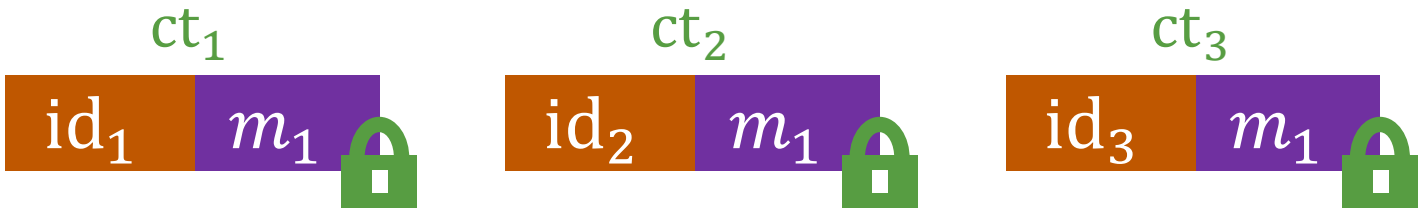
Batched IBE: convenient abstraction to build batch decryption

---

Batched IBE: ciphertexts are associated with an identity id

secret keys are associated with set of identities S

When encrypting, choose a random identity $\mathrm{id} \leftarrow \{0,1\}^\lambda$



With overwhelming probability, all identities are distinct

Secret key for a collection of ciphertexts is a secret key for the set of associated identities

Constructions of Batched IBE

Can be built from attribute-based encryption with succinct keys [BLT25]

ABE with succinct keys relies on lattices (uses homomorphic evaluation machinery)

Small universe constructions [BFOQ25, BCFGOPQW25]

identity space is $\mathrm{poly}(\lambda)$

problematic for batch decryption since random identities might collide

Tag-based constructions [CGPP24, AFP25, CGPW25]

supports large identity space (e.g., $\{0,1\}^\lambda$)

assumes that ciphertexts and secret keys have an additional tag τ

Can decrypt if:
$\mathrm{tg} = \mathrm{tg}'$ **and** $\mathrm{id} \in S$

bounded-collusion restriction: adversary only gets one key per tag

This talk

Our Results: Tag-Based Batched IBE

Prior work [AFP25, CGPW25]:

$|\mathrm{mpk}| = 2|\mathbb{G}_1| + B|\mathbb{G}_2|$

$|\mathrm{ct}| = 3|\mathbb{G}_1| + |m|$

$|\mathrm{sk}| = |\mathbb{G}_2|$

$B = \max$ batch size

Work over asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with prime order p

Security in the generic bilinear group model (+ random oracle)

Approach follows a signature-based witness encryption ("witness encryption for BLS verification")

This work: new framework for constructing batched IBE

$|\mathrm{mpk}| = 5|\mathbb{G}_1| + B|\mathbb{G}_2| + |\mathbb{G}_T|$

$|\mathrm{ct}| = 3|\mathbb{G}_1| + |m|$

$|\mathrm{sk}| = 2|\mathbb{G}_2| + |\mathbb{Z}_p|$

Same ciphertext size whereas public key and secret key longer by constant number of group elements

Security based on a q-type assumption in the **plain model**

Follows a classic algebraic approach (e.g., Boneh-Boyen)

Our Results: Tag-Based Batched IBE

Prior work [AFP25, CGPW25]:

$|\mathrm{mpk}| = 2|\mathbb{G}_1| + B|\mathbb{G}_2|$

$|\mathrm{ct}| = 3|\mathbb{G}_1| + |m|$

$|\mathrm{sk}| = |\mathbb{G}_2|$

$B = \max$ batch size

Work over asymmetric bilinear group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ with prime order p

Security in the generic bilinear group model (+ random oracle)

Approach follows a signature-based witness encryption ("witness encryption for BLS verification")

This work: new framework for constructing batched IBE

$|\mathrm{mpk}| = 5|\mathbb{G}_1| + B|\mathbb{G}_2| + |\mathbb{G}_T|$

$|\mathrm{ct}| = 3|\mathbb{G}_1| + |m|$

$|\mathrm{sk}| = 2|\mathbb{G}_2| + |\mathbb{Z}_p|$

Plain model construction

$|\mathrm{mpk}| = 4|\mathbb{G}_1| + B|\mathbb{G}_2|$

$|\mathrm{ct}| = 2|\mathbb{G}_1| + |m|$

shorter ciphertext

$|\mathrm{sk}| = |\mathbb{G}_2|$

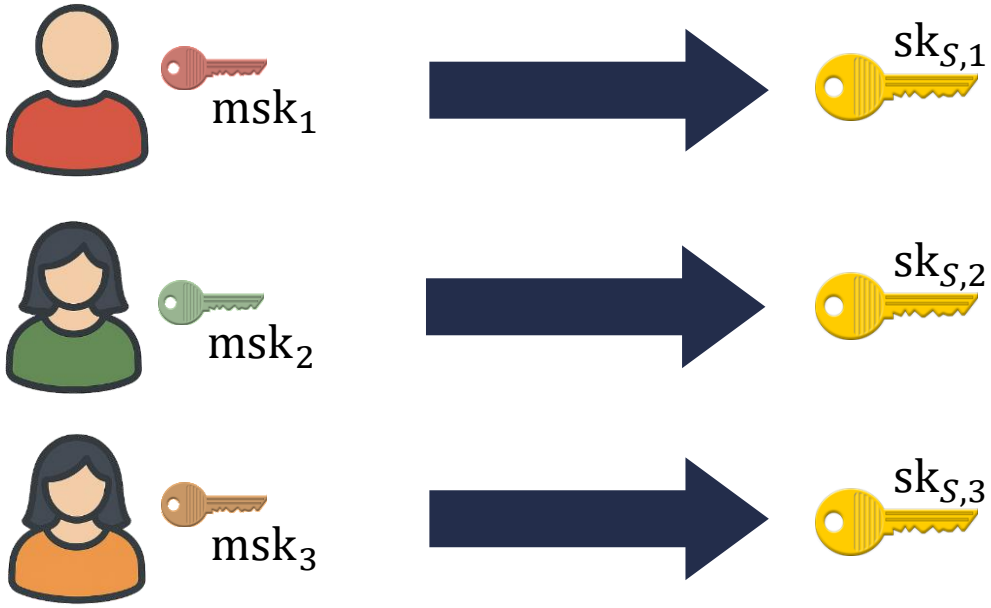
Generic bilinear group model + random oracle model

Same construction framework

Our Results: Tag-Based Batched IBE

Centralized setting

Threshold setting



master secret key is secret-shared

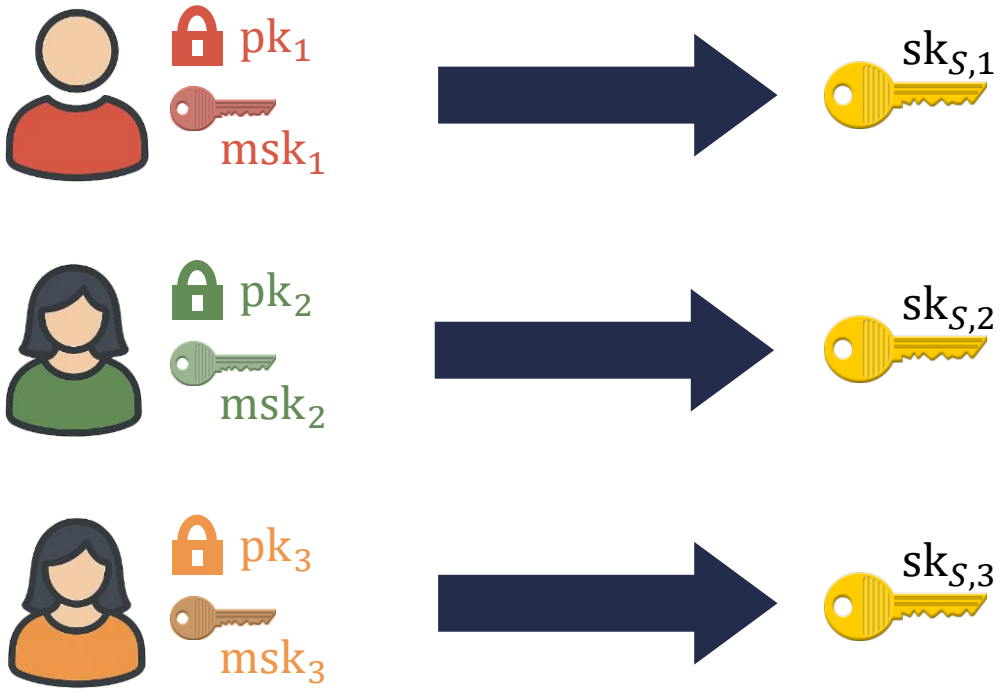
each share-holder can independently issue a decryption key share for a set S

all schemes (both prior and ours) are thresholdizable

Drawback: still require **trusted** dealer (or **interactive** protocol) to generate shares

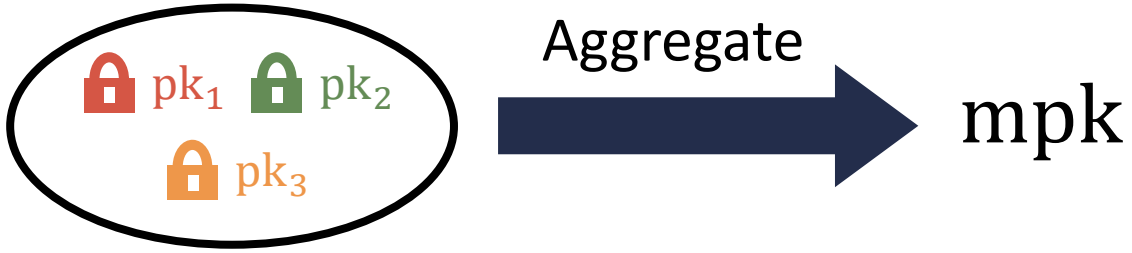
Our Results: Threshold Batched IBE with Silent Setup

Threshold setting with silent setup



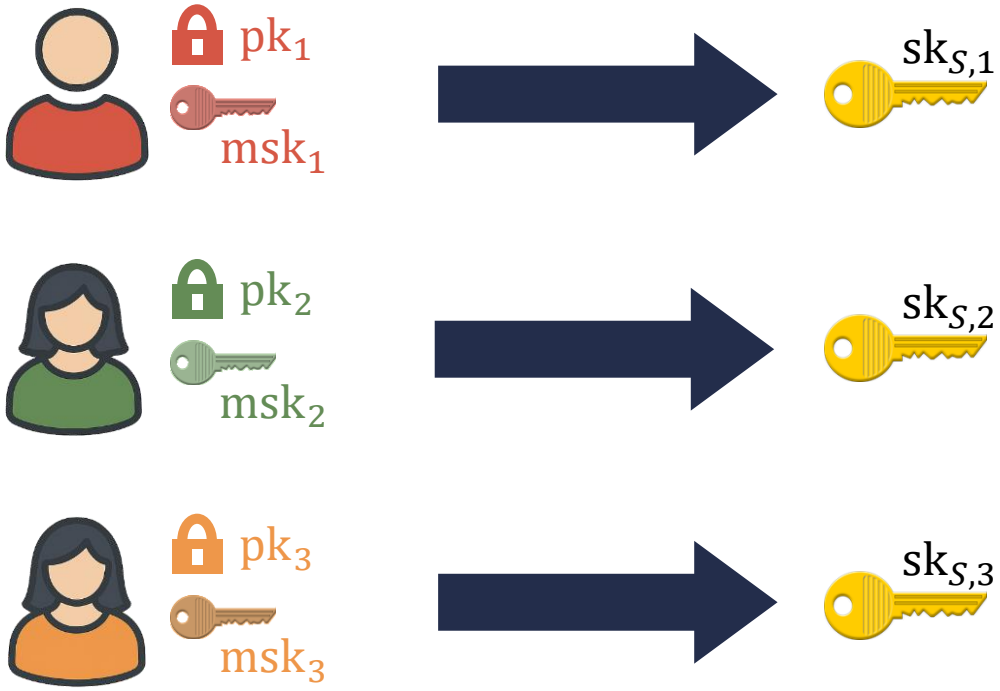
Each user's secret key msk_i functions as their share

Each user picks their own public key pk_i and secret key msk_i



Our Results: Threshold Batched IBE with Silent Setup

Threshold setting with silent setup



Each user picks their own public key pk_i and secret key msk_i

This work:

- Security relies on q-type assumption in plain model
- Parameter sizes:
  - $|\mathrm{ct}| = 2|\mathbb{G}_1| + 2|\mathbb{G}_2| + |m|$
  - $|\mathrm{sk}_{S,i}| = |\mathbb{Z}_p| + 2|\mathbb{G}_2|$
  - $|\mathrm{pp}| = O(LB)$ group elements

Prior work: BEAST-MEV scheme [BCFGOPQW25]

- Security in the generic bilinear group model
- Polynomial-size identity space
- Parameter sizes:
  - $|\mathrm{ct}| \approx O(\lambda + \log B)$ group elements
  - $|\mathrm{sk}_{S,i}|$ = constant number of group elements
  - $|\mathrm{pp}| = O(\lambda L + B)$ group elements

L: committee size
B: batch size

Construction Template

This talk: focus solely on **centralized** scheme (without thresholds)

Will work in asymmetric prime-order pairing group $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ of order p

Notation:

$$[x]_1 \coloneqq g_1^x \in \mathbb{G}_1$$
$$[y]_2 \coloneqq g_2^y \in \mathbb{G}_2$$
$$[x]_1 \cdot [y]_2 \coloneqq [xy]_T = e(g_1^x, g_2^y) = e(g_1, g_2)^{xy}$$

Construction Template

Starting point: a **0-key** secure scheme (secure against an adversary that does not make key-generation queries)

$\text{msk}: \alpha \leftarrow \mathbb{Z}_p$

$\text{mpk}: [\alpha]_T,\ [\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$ $\qquad \tau \leftarrow \mathbb{Z}_p$

$\text{Encrypt}(\text{mpk}, \text{id} \in \mathbb{Z}_p, [m]_T \in \mathbb{G}_T)$:

$s \leftarrow \mathbb{Z}_p$ $\qquad \text{ct}:\ [s]_1,\ [s(\tau - \text{id})]_1,\ [s\alpha]_T + [m]_T$

$\text{KeyGen}(\text{msk}, S \subseteq \mathbb{Z}_p)$:

$\text{sk}_S:\ [\alpha + F_S(\tau)]_2$ $\qquad F_S(\tau) := \prod_{\text{id} \in S} (\tau - \text{id})$

encode the set S as the root s of a polynomial

B: batch size

Construction Template: A 0-Key Scheme

Starting point: a **0-key** secure scheme (secure against an adversary that does not make key-generation queries)

msk: $\alpha \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ [\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

ct: $[s]_1,\ [s(\tau - \mathrm{id})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[\alpha + F_S(\tau)]_2$

Key observation:

- Roots of F_S are elements of S
- Linear polynomial $(\tau - \mathrm{id})$ is a factor of F_S **if and only** if $\mathrm{id} \in S$

Write $F_S(\tau) = (\tau - \mathrm{id}) \cdot F_{S\setminus\{\mathrm{id}\}}(\tau)$

compute using powers-of-τ in mpk

Then: $[s]_1 \cdot [\alpha + F_S(\tau)]_2 - [s(\tau - \mathrm{id})]_1 \cdot [F_{S\setminus\{\mathrm{id}\}}(\tau)]_2 = [s\alpha]_T$

B: batch size

Construction Template: A 0-Key Scheme

Starting point: a **0-key** secure scheme (secure against an adversary that does not make key-generation queries)

msk: $\alpha \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ [\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

ct: $[s]_1,\ [s(\tau - \mathrm{id})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[\alpha + F_S(\tau)]_2$

Problem: ciphertexts are **malleable**

$$[s(\tau - \mathrm{id})]_1 + [s]_1 \cdot (\mathrm{id} - \mathrm{id}') = [s(\tau - \mathrm{id}')]_1$$

Transformed an encryption for id to an encryption for id'

B: batch size

From 0-Key Security to 1-Key Security

Starting point: a **0-key** secure scheme (secure against an adversary that does not make key-generation queries)

$\text{msk}: \alpha \leftarrow \mathbb{Z}_p$

$\text{mpk}: [\alpha]_T,\ [\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

$\text{ct}:\ [s]_1,\ [s(\tau - \text{id})]_1,\ [s\alpha]_T + [m]_T$

$\text{sk}_S:\ [\alpha + F_S(\tau)]_2$

From 0-key to 1-key security: introduce randomizing scalar w to key and ciphertext

$\text{ct}:\ [s]_1,\ [sw(\tau - \text{id})]_1,\ [s\alpha]_T + [m]_T$

will also publish $[w\tau]_1$ *in mpk*

$\text{sk}_S:\ [\alpha + w \cdot F_S(\tau)]_2$

Previous mauling attack now requires adversary to be able to compute $sw(\text{id} - \text{id}')$

Decryption relation:

$$[s]_1 \cdot [\alpha + w \cdot F_S(\tau)]_2 - [sw(\tau - \text{id})]_1 \cdot \left[F_{S\setminus\{\text{id}\}}(\tau)\right]_2 = [s\alpha]_T$$

B: batch size

From 1-Key Security to Many-Key Security

A 1-key secure construction

msk: $\alpha \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ [\tau]_1,\ [w\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

ct: $[s]_1,\ [sw(\tau - \mathrm{id})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[\alpha + w \cdot F_S(\tau)]_2$

Insecure if adversary gets multiple keys (can exploit linearity)

Can lift to many-key security by introducing a tag (batch label)

Recall: in tag-based setting, adversary only gets **one key** per tag

Idea: each choice of tag induces its own copy of a 1-key-secure scheme

Tag plays similar role as an "identity" in a traditional IBE scheme

Two possible approaches: Boneh-Franklin approach or Boneh-Boyen approach

B: batch size

Batched IBE (with Tags)

A 1-key secure construction

msk: $\alpha \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ [\tau]_1,\ [w\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

ct: $[s]_1,\ [sw(\tau - \mathrm{id})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[\alpha + w \cdot F_S(\tau)]_2$

Recall the structure Boneh-Boyen IBE scheme:

msk: $\alpha, u, h \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ \underbrace{[u]_1,\ [h]_1}_{\text{“hash key”}}$

“hash of identity tg*”*

ct: $[s]_1,\ [s(u + h \cdot \mathrm{tg})]_1,\ [s\alpha]_T + [m]_T$

$\mathrm{sk}_{\mathrm{tg}}$: $[r]_2,\ [\alpha + r(u + h \cdot \mathrm{tg})]_2$

Decryption: $[s]_1 \cdot [\alpha + r(u + h \cdot \mathrm{tg})]_2 - [s(u + h \cdot \mathrm{tg})]_1 \cdot [r]_2 = [s\alpha]_T$

B: batch size

Batched IBE (with Tags)

A 1-key secure construction

msk: $\alpha \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ [\tau]_1,\ [w\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

ct: $[s]_1,\ [sw(\tau - \mathrm{id})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[\alpha + w \cdot F_S(\tau)]_2$

Tag-based construction:

ct: $[s]_1,\ [sw(\tau - \mathrm{id})]_1,\ [s(u + h \cdot \mathrm{tg})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[r]_2,\ [\alpha + r(u + h \cdot \mathrm{tg}) + w \cdot F_S(\tau)]_2$

Hash of the tag prevents mixing and matching *across* different tags

Security reduces to a q-type assumption in the plain model

Can support k keys per tag with cost that scales additively with $k \cdot \mathrm{poly}(\lambda)$

B: batch size

Variants and Generalizations

Our batched IBE scheme

msk: $\alpha, u, h \leftarrow \mathbb{Z}_p$

mpk: $[\alpha]_T,\ [u]_1,\ [h]_1,\ [\tau]_1,\ [w\tau]_1,\ [\tau]_2,\ [\tau^2]_2, \ldots, [\tau^B]_2$

ct: $[s]_1,\ [sw(\tau - \mathrm{id})]_1,\ [s(u + h \cdot \mathrm{tg})]_1,\ [s\alpha]_T + [m]_T$

sk_S: $[r]_2,\ [\alpha + r(u + h \cdot \mathrm{tg}) + w \cdot F_S(\tau)]_2$

Use Boneh-Franklin approach to embed tag rather than Boneh-Boyen

- Security in the generic group model + random oracle model
- Shorter ciphertexts (2 group elements + message) and secret key (1 group element)

Integrate with Waters-Wu construction of threshold IBE with silent setup

- Yields construction of threshold batched IBE with silent setup

B: batch size

Open Problems

Recent work [BNRT26]: threshold batch decryption from pairings without tag-based restrictions

Threshold batched IBE with silent setup (without tag-based restrictions)

Pairing-based threshold batched IBE with short public parameters (sublinear in batch size)

Post-quantum constructions with good concrete efficiency (currently relies on lattice-based ABE)

Thanks!

https://eprint.iacr.org/2025/2103.pdf