HIBE with Tight Multi-challenge Security

Roman Langrehr  ETH Zurich (Switzerland), Part of the work done at KIT (Karlsruhe, Germany)
Jiaxin Pan  NTNU (Trondheim, Norway)
Outline

(H)IBE

Tight multi-challenge security

Related works

The difficulty

Our solution

Future work
Identity-based encryption

- Alice needs to obtain only the master public key
- Encryption with identities (e.g. e-mail address)
Hierarchical Identity-based encryption

- Hierarchy of key generators

Alice

Trusted Third Party

Bob

mpk

usk

usk_{Bob}
Key delegation

Identities have the form \((id_1, \ldots, id_p)\).

- Each user can generate keys for its children
The adversary must not ask user secret keys for prefixes of challenge identities (id*).
Security game (IND-HID-CPA)

- The adversary must not ask user secret keys for prefixes of challenge identities (id*).
- IND-HID-CCA is easy once you have IND-HID-CPA.
Tight security

Scheme (e.g. HIBE) \rightarrow \text{Reduction} \rightarrow \text{Assumption (e.g. Diffie-Hellman)}

Can be broken with probability $\epsilon$ using resources $\rho$.

Can be broken with probability $\epsilon/\ell$ using resources $\rho$.

Larger security loss requires larger security parameter.

Security loss $\ell$ can depend on:

- scheme parameters (e.g. maximum hierarchy depth $L$)
- $\lambda$: the security parameter
- the attacker's resources (e.g. # user secret key queries $Q_k$ or # challenge ciphertext queries $Q_c$)

Tight security: \\
\begin{align*}
\text{allowed} \\
\text{not allowed}
\end{align*}
Tight security

- Scheme (e.g. HIBE) → Reduction → Assumption (e.g. Diffie-Hellman)

  Can be broken with probability $\varepsilon$ using resources $\rho$.

  Can be broken with probability $\varepsilon/\ell$ using resources $\rho$.

Larger security loss requires larger security parameter.

Security loss $\ell$ can depend on:

- scheme parameters (e.g. maximum hierarchy depth $L$)
- $\lambda$: the security parameter
- the attacker's resources (e.g. # user secret key queries $Q_k$ or # challenge ciphertext queries $Q_c$)
Tight security

Scheme (e.g. HIBE) \(\xrightarrow{\text{Reduction}}\) Assumption (e.g. Diffie-Hellman)

Can be broken with probability \(\varepsilon\) using resources \(\rho\).

Can be broken with probability \(\varepsilon/\ell\) using resources \(\rho\).

Larger security loss requires larger security parameter.

Security loss \(\ell\) can depend on:

- scheme parameters (e.g. maximum hierarchy depth \(L\))
- \(\lambda\): the security parameter
- the attacker’s resources (e.g. \# user secret key queries \(Q_k\) or \# challenge ciphertext queries \(Q_c\))
Tight security

Scheme (e.g. HIBE) \quad \xrightarrow{\text{Reduction}} \quad \text{Assumption (e.g. Diffie-Hellman)}

Can be broken with probability $\varepsilon$ using resources $\rho$.
Can be broken with probability $\varepsilon/\ell$ using resources $\rho$.

Larger security loss requires larger security parameter.
Security loss $\ell$ can depend on:

- scheme parameters (e.g. maximum hierarchy depth $L$)
- $\lambda$: the security parameter
- the attacker’s resources (e.g. \# user secret key queries $Q_k$ or \# challenge ciphertext queries $Q_c$)

Tight security: \begin{align*}
\text{allowed} \\
\text{not allowed}
\end{align*}
Multi-challenge security

Challenger

\[ b \leftarrow \{0, 1\} \]

mpk

Adversary

\[ \text{id} \]

\[ \text{id}^*, \text{m}_0, \text{m}_1 \]

\[ \text{C}^* \leftarrow \text{Enc}(\text{mpk}, \text{id}^*, \text{m}_b) \]

\[ b' \]

\[ b \equiv b' \]
Multi-challenge security

Challenger

mpk

id

b $\xleftarrow{\$} \{0,1\}$

id*, m0, m1

C* $\xleftarrow{\$} \text{Enc}(mpk, id^*, m_b)$

b $\xleftarrow{\ ?} b'$

Adversary

usk[id]

b' $\xrightarrow{\ ?} b$

Single-challenge security

Multi-challenge security
Multi-challenge security

Challenger

\[ b \overset{\$}{\leftarrow} \{0, 1\} \]

Adversary

\[ mpk \]

\[ \text{id} \]

\[ \text{usk}[\text{id}] \]

\[ \text{id}^*, m_0, m_1 \]

\[ C^* \overset{\$}{\leftarrow} \text{Enc}(mpk, \text{id}^*, m_b) \]

\[ b' \]

\[ b \overset{?}{=} b' \]

Single-challenge security

generic: \( O(Q_c) \) loss

Multi-challenge security

Tight multi-instance security: Easy to achieve by rerandomizing the master public key.
Multi-challenge security

Challenger

\[ b \overset{\$}{\leftarrow} \{0, 1\} \]

Adversary

\[ C^* \overset{\$}{\leftarrow} \text{Enc}(\text{mpk, id}^*, m_b) \]

\[ b' \]

\[ b = b' \]

Single-challenge security

generic: \( \mathcal{O}(Q_c) \) loss

Multi-challenge security

Tight multi-instance security: Easy to achieve by rerandomizing the master public key.
History: HIBE

HIBEs in prime-order pairing groups:

<table>
<thead>
<tr>
<th>Reference</th>
<th>Complexity</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Wat09], [CW13], [BKP14]</td>
<td>$O(Q_k)$ (single-challenge)</td>
<td></td>
</tr>
<tr>
<td>[Lew12], [GCTC16]</td>
<td>$O(Q_kL)$ (single-challenge)</td>
<td></td>
</tr>
<tr>
<td>[LP19]</td>
<td>$O(nL^2)$ resp. $O(nL)$ (single-challenge)</td>
<td></td>
</tr>
<tr>
<td>This work</td>
<td>$O(nL^2)$ (multi-challenge)</td>
<td></td>
</tr>
</tbody>
</table>

- $Q_k$: number of user secret key queries
- $L$: maximum hierarchy depth
- $n$: Bit-length of the identities
History: Tight IBE

Tight IBEs in prime-order pairing groups:

| [CW13], [BKP14] | $O(n)$ (single-challenge) |
| [AHY15], [GCD$^+$16], [GDCC16], [HJP18] | $O(n)$ (multi-challenge) |

- $n$: Bit-length of the identities
History: Tight IBE

Tight IBEs in prime-order pairing groups:

| [CW13], [BKP14] | $O(n)$ (single-challenge) |
| [AHY15], [GCD$^+$16], [GDCC16], [HJP18] | $O(n)$ (multi-challenge) |

- $n$: Bit-length of the identities

Tight single-challenge HIBE + Tight multi-challenge IBE $\xrightarrow{?}$ Tight multi-challenge HIBE
IND-HID-CPA security for (H)IBE

The challenge:

- The reduction must answer user secret key queries for $id_1, \ldots, id_{Q_k}$.
- The reduction must take advantage of the adversaries decryption capabilities for $id_1^*, \ldots, id_{Q_c}^*$.
- The adversary adaptively chooses $id_1, \ldots, id_{Q_k}$ and $id_1^*, \ldots, id_{Q_c}^*$.
Partitioning

- Different parts use "slightly different" secret key.
- A usk key from one part is not helpful for decrypting a ciphertext from a different part.
Partitioning

- Different parts use "slightly different" secret key.
- A usk key from one part is not helpful for decrypting a ciphertext from a different part.

![Diagram showing initial, intermediate, and final stages with points representing queried user secret key and challenge ciphertext.]
Query-by-query Partitioning

• Typically used by non-tight (H)IBE schemes
• $O(Q_k)$ security loss
Query-by-query Partitioning

- Typically used by non-tight (H)IBE schemes
- $O(Q_k)$ security loss
Query-by-query Partitioning

- Typically used by non-tight (H)IBE schemes
- $O(Q_k)$ security loss
Query-by-query Partitioning

- Typically used by non-tight (H)IBE schemes
- $O(Q_k)$ security loss
Query-by-query Partitioning

• Typically used by non-tight (H)IBE schemes
• $\mathcal{O}(Q_k)$ security loss
Query-by-query Partitioning

- Typically used by non-tight (H)IBE schemes
- $\mathcal{O}(Q_k)$ security loss
Query-by-query Partitioning

• Typically used by non-tight (H)IBE schemes
• $O(Q_k)$ security loss
Query-by-query Partitioning

• Typically used by non-tight (H)IBE schemes
• $O(Q_k)$ security loss
Bit-by-bit Partitioning

- Typically used by tight (H)IBE schemes.
- One part per identity
- $\mathcal{O}(n)$ security loss
Bit-by-bit Partitioning

- Typically used by tight (H)IBE schemes.
- One part per identity
- $O(n)$ security loss

\[
id_1 = 0 \quad \text{vs} \quad id_1 = 1
\]
Bit-by-bit Partitioning

• Typically used by tight (H)IBE schemes.
• One part per identity
• $O(n)$ security loss
Bit-by-bit Partitioning

• Typically used by tight (H)IBE schemes.
• One part per identity
• \( \mathcal{O}(n) \) security loss
Bit-by-bit Partitioning

- Typically used by tight (H)IBE schemes.
- One part per identity
- $O(n)$ security loss
Partitioning techniques

1. Embedding a challenge of the underlying assumption...
   - ...in a part of the msk that appears only in user secret keys with \( id_i = b \).
   - ...“reacts” with the randomness of the usk resp. ciphertext.
Partitioning techniques

1. Embedding a challenge of the underlying assumption...
   - ...in a part of the msk that appears only in user secret keys with $id_i = b$.
   - ...“reacts” with the randomness of the usk resp. ciphertext.

2. Choose randomness of a subspace [GHKW16]
   - hides part of the msk from usk queries.
Usage in the single-challenge setting

Tight IBE:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Challenge queries</th>
<th>usk queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CW13],[BKP14]</td>
<td>(information-theoretic)</td>
<td>Embedding a challenge</td>
</tr>
</tbody>
</table>
Usage in the single-challenge setting

Tight IBE:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Challenge queries</th>
<th>usk queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CW13],[BKP14]</td>
<td>(information-theoretic)</td>
<td>Embedding a challenge</td>
</tr>
</tbody>
</table>

Tight HIBE:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Challenge queries</th>
<th>usk queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>[LP19]</td>
<td>(information-theoretic)</td>
<td>Subspace</td>
</tr>
</tbody>
</table>
## Usage in the multi-challenge setting

### Tight IBE:

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Challenge queries</th>
<th>usk queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>[GDCC16], [HJP18]</td>
<td>Subspace</td>
<td>Embedding a challenge</td>
</tr>
</tbody>
</table>
[BKP14] (single-challenge IBE)

Use [GHKW16] for the challenge queries

[GDCC16], [HJP18] (multi-challenge IBE)

Use [GHKW16] for the usk queries

[LP19] (single-challenge HIBE)
[BKP14]
(single-challenge IBE)

Use [GHKW16] for
the challenge queries

[GDCC16], [HJP18]
(multi-challenge IBE)

New tight
multi-challenge HIBE?

Use [GHKW16] for the usk queries

[LP19]
(single-challenge HIBE)
[BKP14]
(single-challenge IBE)

Use [GHKW16] for the challenge queries

[GDCC16], [HJP18]
(multi-challenge IBE)

Use [GHKW16] for the usk queries

[LP19]
(single-challenge HIBE)

Doesn’t work

New tight multi-challenge HIBE?
Simplified version of BKP-like schemes

- Master secret key:
  For every bit position $i \in \{1, \ldots, n \cdot L\}$ and bit $b \in \{0, 1\}$:

$$X_{i,b}$$
Simplified version of BKP-like schemes

- Master secret key:
  For every bit position $i \in \{1, \ldots, n \cdot L\}$ and bit $b \in \{0, 1\}$:

- User secret key for id: $\sum_{i \mid |id|} X_{i,|id|}$
Simplified version of BKP-like schemes

- Master secret key:
  For every bit position $i \in \{1, \ldots, n \cdot L\}$ and bit $b \in \{0, 1\}$:

- User secret key for id:
  $\sum_i |id| X_{i, id[i]}$

- Challenge ciphertext for id*:
  $\sum_i |id^*| X_{i, id^*[i]}$
The difficulty

Use the [GHKW16] subspace technique for the user secret keys [LP19].

In a suitable (hidden) basis:

<table>
<thead>
<tr>
<th>ct randomness</th>
<th>msk</th>
<th>usk randomness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Half of the entropy is hidden. ✓
The difficulty

Use the [GHKW16] subspace technique for the challenge ciphertexts [GDCC16, HJP18].

In a suitable (hidden) basis:

\[
\begin{array}{ccc}
\text{ct randomness} & \text{msk} & \text{usk randomness} \\
\ast & 0 & \\
\end{array}
\]

Half of the entropy is hidden. ✓
The difficulty

Use the [GHKW16] subspace technique for both usks and cts.

In a suitable (hidden) basis:

<table>
<thead>
<tr>
<th>ct randomness</th>
<th>msk</th>
<th>usk randomness</th>
</tr>
</thead>
<tbody>
<tr>
<td>*</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0</td>
</tr>
</tbody>
</table>

Only one quarter of the entropy is hidden. ✗
Our solution

New technique to randomize multiple challenge ciphertexts...

• ...based on the “Embedding a challenge” approach.
• ...achieves the same efficiency.
• ...compatible with [LP19]
Our solution

Previous work (only IBE)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Challenge queries</th>
<th>usk queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>[GDCC16], [HJP18]</td>
<td>Subspace</td>
<td>Embedding a challenge</td>
</tr>
</tbody>
</table>

This work (also HIBE)

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Challenge queries</th>
<th>usk queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>This work</td>
<td>Embedding a challenge</td>
<td>Subspace</td>
</tr>
</tbody>
</table>
Our solution

MDDH challenge

In a suitable (hidden) basis:

<table>
<thead>
<tr>
<th>ct randomness</th>
<th>msk</th>
<th>usk randomness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Roman Langrehr, Jiaxin Pan 2020-06-01
Our solution – More details

MDDH challenge: \[ f \in \text{Span}(D) \] or \[ f \text{ is uniformly random} \]

In a suitable (hidden) basis:

<table>
<thead>
<tr>
<th>ct randomness</th>
<th>msk</th>
<th>usk randomness</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>*</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Our solution – More details

MDDH challenge:

\[ f \in \text{Span}(D) \text{ or } f \text{ is uniformly random} \]

In a suitable (hidden) basis:

\[
\begin{array}{c|c|c}
\text{ct randomness} & \text{msk} & \text{usk randomness} \\
\hline
\text{ct randomness} & \text{msk} & \text{usk randomness} \\
\end{array}
\]

\[
\begin{pmatrix}
D D^{-1} \\
\end{pmatrix}^T
\]

Roman Langrehr, Jiaxin Pan 2020-06-01
Our solution – More details

MDDH challenge:
\[ f \in \text{Span}(D) \text{ or } f \text{ is uniformly random} \]

In a suitable (hidden) basis:

\[ \begin{array}{c|c|c}
\text{ct randomness} & \text{msk} & \text{usk randomness} \\
\hline
\end{array} \]

\[ \begin{pmatrix} D \cdot D^{-1} \end{pmatrix}^T \]
Our solution – More details

MDDH challenge: $f \in \text{Span}(D)$ or $f$ is uniformly random

But sometimes we have to embed the same challenge in multiple ciphertexts!
Our solution – More details

MDDH challenge:

In a suitable (hidden) basis:

ct randomness | msk | usk randomness

$D, \ F \ \in \ Span(D)$ or $f$ is uniformly random

Roman Langrehr, Jiaxin Pan 2020-06-01
### Comparison of HIBE schemes (in prime-order pairing groups)

| Scheme  | $|mpk|$ | $|usk|$ | $|C|$ | Loss | MC | Assumption |
|---------|--------|--------|------|------|----|------------|
| [Wat05] | $\mathcal{O}(nL)$ | $\mathcal{O}(nL)$ | $\mathcal{O}(p)$ | $\mathcal{O}(nQ_k^L)$ | $\times$ | DBDH |
| [Wat09] | $\mathcal{O}(L)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(Q_k)$ | $\times$ | 2-LIN |
| [Lew12] | $\mathcal{O}(1)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(Q_k)$ | $\times$ | 2-LIN |
| [CW13]  | $\mathcal{O}(L)$ | $\mathcal{O}(L)$ | $\mathcal{O}(1)$ | $\mathcal{O}(Q_k)$ | $\times$ | SXDH |
| [BKP14] | $\mathcal{O}(L)$ | $\mathcal{O}(L)$ | $\mathcal{O}(1)$ | $\mathcal{O}(Q_k)$ | $\times$ | SXDH |
| [GCTC16]| $\mathcal{O}(1)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(Q_k)$ | $\times$ | SXDH |
| [LP19]  | $\mathcal{O}(nL^2)$ | $\mathcal{O}(nL^2)$ | $\mathcal{O}(1)$ | $\mathcal{O}(nL^2)$ | $\times$ | SXDH |
| [LP19]$^H$ | $\mathcal{O}(\gamma L)$ | $\mathcal{O}(\gamma L)$ | $\mathcal{O}(1)$ | $\mathcal{O}(\gamma L)$ | $\times$ | SXDH |
| [LP19]$_2$ | $\mathcal{O}(nL^2)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(nL)$ | $\times$ | SXDH |
| [LP19]$_2^H$ | $\mathcal{O}(\gamma L)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(\gamma)$ | $\times$ | SXDH |
| Ours$_1$ | $\mathcal{O}(nL^2)$ | $\mathcal{O}(nL^2)$ | $\mathcal{O}(1)$ | $\mathcal{O}(nL^2)$ | $\checkmark$ | SXDH |
| Ours$_1^H$ | $\mathcal{O}(\gamma L)$ | $\mathcal{O}(\gamma L)$ | $\mathcal{O}(1)$ | $\mathcal{O}(\gamma L)$ | $\checkmark$ | SXDH |
| Ours$_2$ | $\mathcal{O}(nL^2)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(nL^2)$ | $\checkmark$ | SXDH |
| Ours$_2^H$ | $\mathcal{O}(\gamma L)$ | $\mathcal{O}(p)$ | $\mathcal{O}(p)$ | $\mathcal{O}(\gamma L)$ | $\checkmark$ | SXDH |

- $L$: maximum hierarchy depth
- $p$: actual hierarchy depth
- $n$: bit-length of identities
- $\gamma$: bit-length of hashes
- $Q_k$: number of user secret key queries
Future work: Beyond bit-by-bit partitioning

[AHY15] achieved a trade-off between mpk and usk/C size for IBE: Parameter $c \in [0, 1]$

| Scheme  | $|\text{mpk}|$       | $|\text{usk}|$     | $|C|$   | Loss  |
|---------|----------------------|---------------------|--------|-------|
| [AHY15] | $O(n^{1-c})$         | $O(n^c)$            | $O(n^c)$ | $O(n)$ |
Future work: Beyond bit-by-bit partitioning

[AHY15] achieved a trade-off between mpk and usk/C size for IBE:
Parameter $c \in [0, 1]$

<table>
<thead>
<tr>
<th>Scheme</th>
<th>mpk</th>
<th>usk</th>
<th>C</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>[AHY15]</td>
<td>$\mathcal{O}(n^{1-c})$</td>
<td>$\mathcal{O}(n^c)$</td>
<td>$\mathcal{O}(n^c)$</td>
<td>$\mathcal{O}(n)$</td>
</tr>
</tbody>
</table>

[CGW17] achieved constant size mpk (and tighter security loss) in composite-order pairing groups (4 factors):

<table>
<thead>
<tr>
<th>Scheme</th>
<th>mpk</th>
<th>usk</th>
<th>C</th>
<th>Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>[CGW17]</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>$\mathcal{O}(\log(Q_k))$</td>
</tr>
</tbody>
</table>
Nuttapong Attrapadung, Goichiro Hanaoka, and Shota Yamada.  
A framework for identity-based encryption with almost tight security.  
In Tetsu Iwata and Jung Hee Cheon, editors, ASIACRYPT 2015, Part I, volume  
doi:10.1007/978-3-662-48797-6_22.

Olivier Blazy, Eike Kiltz, and Jiaxin Pan.  
(Hierarchical) identity-based encryption from affine message authentication.  
In Juan A. Garay and Rosario Gennaro, editors, CRYPTO 2014, Part I, volume  
References II

Jie Chen, Junqing Gong, and Jian Weng.
Tightly secure IBE under constant-size master public key.

Jie Chen and Hoeteck Wee.
Fully, (almost) tightly secure IBE and dual system groups.
doi:10.1007/978-3-642-40084-1_25.
Junqing Gong, Jie Chen, Xiaolei Dong, Zhenfu Cao, and Shaohua Tang. 
Extended nested dual system groups, revisited.

Junqing Gong, Zhenfu Cao, Shaohua Tang, and Jie Chen.
Extended dual system group and shorter unbounded hierarchical identity based encryption.
References IV


References V

Dennis Hofheinz, Dingding Jia, and Jiaxin Pan.
Identity-based encryption tightly secure under chosen-ciphertext attacks.
doi:10.1007/978-3-030-03329-3_7.

Allison B. Lewko.
Tools for simulating features of composite order bilinear groups in the prime order setting.
doi:10.1007/978-3-642-29011-4_20.
Roman Langrehr and Jiaxin Pan.
Tightly secure hierarchical identity-based encryption.
doi:10.1007/978-3-030-17253-4_15.

Brent R. Waters.
Efficient identity-based encryption without random oracles.
Brent Waters.
Dual system encryption: Realizing fully secure IBE and HIBE under simple assumptions.
doi:10.1007/978-3-642-03356-8_36.
Pictures

Alice, Bob, Trusted Party: freepik.com
Encrypted Mail: Icon made by SimpleIcon from www.flaticon.com