Faster Lattice-based KEMs via Fujisaki-Okamoto Transform in the Multi-User Setting via Prefix-Hashing

J. Duman\textsuperscript{1}, K. Hövelmanns\textsuperscript{2}, E. Kiltz\textsuperscript{1}, V. Lyubashevsky\textsuperscript{3}, G. Seiler\textsuperscript{4}

Ruhr-University Bochum\textsuperscript{1}
Eindhoven University of Technology\textsuperscript{2}
IBM Zürich\textsuperscript{34}
ETH Zürich\textsuperscript{4}

9. Juny 2021
Common safeguard against multi-user attacks: hash also public-keys, notably done by Kyber and Saber.

We formally show:
1. This indeed improves multi-user security.
2. Too wasteful: hashing a short prefix of $pk$, gives the same security guarantees.

**Intro**

Fujisaki-Okamoto (FO) Transform

Transform

PKE | Fujisaki-Okamoto (FO) Transform | KEM

| IND-CPA secure | IND-CCA secure |

- Standard method of almost all NIST PQC Candidates: start with IND-CPA secure PKE and apply variant of FO [FO99, FO13, HHK17]

- Encaps$_{pk}(r) = (\text{Enc}_p(r; G(r)), H(r))$
  
  ciphertext  key

$\text{Encaps}_{pk}(r)$

$\text{Enc}_p(r; G(r))$

$H(r)$
We formally show:

1. This indeed improves multi-user security.
2. Too wasteful: hashing a short prefix of $pk$, gives the same security guarantees.

**Intro**

Fujisaki-Okamoto (FO) Transform

IND-CPA secure

PKE

IND-CCA secure

KEM

**Transform**

PKE → KEM

**Standard method of almost all NIST PQC Candidates:** start with IND-CPA secure PKE and apply variant of FO [FO99, FO13, HHK17]

**Encaps_{pk}(; r) = (Enc_{pk}(r; G(pk, r), H(pk, r))**

**ciphertext**

**key**

**Common safeguard against multi-user attacks:** hash also public-keys, notably done by Kyber and Saber
Intro

- Standard method of almost all NIST PQC Candidates: start with IND-CPA secure PKE and apply variant of FO [FO99, FO13, HHK17]

  \[ \text{Encaps}_{pk}(; r) = (\text{Enc}_{pk}(r; \text{G}(pk, r)), \text{H}(pk, r)) \]

- Common safeguard against *multi-user attacks*: hash also *public-keys*, notably done by Kyber and Saber

- We formally show:
  1. this indeed *improves multi-user security*.
  2. Too wasteful: hashing a *short prefix of* \( pk \), gives the same security guarantees
Intro

Fujisaki-Okamoto (FO) Transform

n-IND-CPA secure

n-IND-CCA secure

Prefix hashing: improve the FO by hashing of a short prefix of the public-key $=:$ id instead of large public-key

Encaps_{pk}(; r) = (Enc_{pk}(r; G(id, r)), H(id, r))

$\Rightarrow$ important for Lattice-based KEMs since the public-keys are large (e.g. 1KB, instead of 32 Bytes as in ECC) and hashing is most expensive part
Prefix hashing: improve the FO by hashing of a short prefix of the public-key $\text{id}$ instead of large public-key

$\text{Encaps}_{pk}(; r) = (\text{Enc}_{pk}(r; G(\text{id}, r)), H(\text{id}, r))$

$\Rightarrow$ important for Lattice-based KEMs since the public-keys are large (e.g. 1KB, instead of 32 Bytes as in ECC) and hashing is most expensive part

$\Rightarrow$ yielding 2x-3x speed-up over (round 3) key-generation and encapsulation for Kyber and up to 40% improvement of the same in Saber
Intro

\[ \text{n-IND-CPA secure} \quad \text{Fujisaki-Okamoto (FO) Transform} \quad \text{n-IND-CCA secure} \]

\[ \text{PKE} \quad \text{KEM} \]

- Prefix hashing: improve the FO by hashing of a *short prefix of the public-key* \( \equiv : \text{id} \) instead of large public-key
  \[ \text{Encaps}_{pk}(; \; r) = (\text{Enc}_{pk}(r; \; G(\text{id}, \; r)), \; \text{H}(\text{id}, \; r)) \]

- \Rightarrow important for *Lattice-based* KEMs since the public-keys are large (e.g. 1KB, instead of 32 Bytes as in ECC) and hashing is most expensive part

- yielding 2x-3x speed-up over (round 3) key-generation and encapsulation for Kyber and up to 40% improvement of the same in Saber

- without weakening multi-user security
Single-User IND-CPA

- Adversary wants to learn some information on plaintext $m$. 

\[ m \quad \xrightarrow{\text{Enc}_{pk}(m)} \quad pk \]
I \Rightarrow \text{Adversary wants to learn some information on the plaintexts } m_1, \ldots, m_n \text{.}

\text{Multi-User IND-CPA (n-IND-CPA)}

\begin{align*}
\text{pk}_1 & \quad \text{pk}_1 \\
m_1, \ldots, m_n & \quad c_1 \\
\text{pk}_2 & \quad \text{pk}_2 \\
& \quad c_2 \\
& \quad \vdots \\
\text{pk}_n & \quad \text{pk}_n \\
& \quad c_n
\end{align*}
Multi-User IND-CPA (n-IND-CPA)

\[ m_1, \ldots, m_n \]
Multi-User IND-CPA (n-IND-CPA)

\[ m_1, \ldots, m_n \]

\[ \Rightarrow \text{Adversary wants to learn some information on the } n \text{ plaintexts } m_1, \ldots, m_n \]
IND-CCA implies n-IND-CCA

By a hybrid argument we know that IND-CCA security implies n-IND-CCA security [BBM00] 😊
IND-CCA implies n-IND-CCA

- By a hybrid argument we know that IND-CCA security implies n-IND-CCA security [BBM00]
- Looses a factor of $n$, where $n = \#\text{Users}$
- $\implies$ PKE needs to be instantiated with worse parameters
IND-CCA implies n-IND-CCA

- By a hybrid argument we know that IND-CCA security implies n-IND-CCA security [BBM00]
- Loses a factor of $n$, where $n = \#\text{Users}$
- $\implies$ PKE needs to be instantiated with worse parameters
- We show: a direct proof of n-IND-CCA yields direct reduction to n-IND-CPA security of PKE
- Beats the hybrid argument if:
  - $\text{Adv}_{\text{PKE}}^{\text{n-IND-CPA}} \ll n \cdot \text{Adv}_{\text{PKE}}^{\text{IND-CPA}}$
Advantage: more efficient than additionally hashing

Disadvantage: worse multi-user security

\[ \text{Encaps}_{pk}(; r) = (\text{Enc}_{pk}(r; G(r)), H(r)) \]

ciphertext  key
FO without *pk* hashing

- \( \text{Encaps}_{\text{pk}}(; r) = (\text{Enc}_{\text{pk}}(r; G(r)), \text{H}(r)) \)
  - ciphertext
  - key

- Advantage: more efficient than additionally hashing *pk*
- Disadvantage: worse multi-user security
FO with Public-Key Hashing

\[ \text{Encaps}_{pk}(; r) = (\text{Enc}_{pk}(r; G(pk, r)), H(pk, r)) \]

ciphertext

key
FO with Public-Key Hashing

- $\text{Encaps}_{pk}(; r) = (\text{Enc}_{pk}(r; G(pk, r)), H(pk, r))$
  - ciphertext
  - key

- (Essentially) used by Kyber and Saber to protect against multi-user attacks

- Advantage: improves multi-user security

- Disadvantage: wasteful if e.g. $|pk| \approx 1KB$
FO with Prefix Hashing

- $\text{Encaps}_{pk}(r) = (\text{Enc}_{pk}(r; G(id, r)), H(id, r))$
  - ciphertext
  - key

- $id := \text{ID}(pk) = \text{short prefix of the public-key, e.g.} 32 \text{ Bytes}$
FO with Prefix Hashing

- $\text{Encaps}_{pk}(; r) = (\text{Enc}_{pk}(r; G(id, r)), H(id, r))$
  - ciphertext
  - key

- $\text{id} := \text{ID}(pk) =$ short prefix of the public-key, e.g. 32 Bytes

- Best of both worlds: improves multi-user security and (almost) as efficient as without any $pk$ hashing
Correctness Errors

- [HHK17] $\delta$-Correctness of PKE informally: probability of decryption error for a random key
- This work: $\delta(n)$-Correctness of PKE
  “worst $\delta$-correctness from $n$ random keys”
Correctness Errors

- [HHK17] $\delta$-Correctness of PKE informally: probability of decryption error for a random key
- This work: $\delta(n)$-Correctness of PKE “worst $\delta$-correctness from $n$ random keys”
- Trivial bounds: $\delta \leq \delta(n) \leq n \cdot \delta$
- Worst-case: $\delta(n) = n \cdot \delta$
- Best-case: $\delta(n) = \delta$
Correctness Errors

- [HHK17] $\delta$-Correctness of PKE informally: probability of decryption error for a random key
- This work: $\delta(n)$-Correctness of PKE “worst $\delta$-correctness from $n$ random keys”
- Trivial bounds: $\delta \leq \delta(n) \leq n \cdot \delta$
- Worst-case: $\delta(n) = n \cdot \delta$
- Best-case: $\delta(n) = \delta$
- For Kyber and Saber: $\delta < \delta(n) < n \cdot \delta$. 
## Results (Simplified)

<table>
<thead>
<tr>
<th>FO variant</th>
<th>( \text{Adv}^{n-\text{IND-CCA}}_{\text{KEM}} ) (ROM)</th>
<th>( \text{Adv}^{n-\text{IND-CCA}}_{\text{KEM}} ) (QROM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( pk ) hashing</td>
<td>( \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} \delta(n) )</td>
<td>( \sqrt{q_{\text{RO}} \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^{2} \delta(n) )</td>
</tr>
<tr>
<td>prefix hashing</td>
<td>( \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} \delta(n) + \frac{n^2}{2^\ell} )</td>
<td>( \sqrt{q_{\text{RO}} \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^{2} \delta(n) + \frac{n^2}{2^\ell} )</td>
</tr>
<tr>
<td>no hashing</td>
<td>( \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} n\delta(1) )</td>
<td>( n \cdot \sqrt{q_{\text{RO}} \text{Adv}^{\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^{2} n\delta(1) )</td>
</tr>
</tbody>
</table>
## Results (Simplified)

<table>
<thead>
<tr>
<th>FO variant</th>
<th>$\text{Adv}^{n-\text{IND-CCA}}_{\text{KEM}}$ (ROM)</th>
<th>$\text{Adv}^{n-\text{IND-CCA}}_{\text{KEM}}$ (QROM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>pk hashing</td>
<td>$\text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} \delta(n)$</td>
<td>$\sqrt{q_{\text{RO}}} \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}}^2 \delta(n)$</td>
</tr>
<tr>
<td>prefix hashing</td>
<td>$\text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} \delta(n) + \frac{n^2}{2^\ell}$</td>
<td>$\sqrt{q_{\text{RO}} \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2 \delta(n) + \frac{n^2}{2^\ell}$</td>
</tr>
<tr>
<td>no hashing</td>
<td>$\text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} n \delta(1)$</td>
<td>$n \cdot \sqrt{q_{\text{RO}} \text{Adv}^{\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2 n \delta(1)$</td>
</tr>
</tbody>
</table>

$\frac{n^2}{2^\ell} =$ probability of a collision in the prefixes, e.g. $2^\ell \approx 2^{256}$ and $n = 2^{30}$
## Results (Simplified)

<table>
<thead>
<tr>
<th>FO variant</th>
<th>$\text{Adv}^{n-\text{IND-CCA}}_{\text{KEM}}$ (ROM)</th>
<th>$\text{Adv}^{n-\text{IND-CCA}}_{\text{KEM}}$ (QROM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pk$ hashing</td>
<td>$\text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} \delta(n)$</td>
<td>$\sqrt{q_{\text{RO}} \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2 \delta(n)$</td>
</tr>
<tr>
<td>prefix hashing</td>
<td>$\text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} \delta(n) + \frac{n^2}{2^\ell}$</td>
<td>$\sqrt{q_{\text{RO}} \text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2 \delta(n) + \frac{n^2}{2^\ell}$</td>
</tr>
<tr>
<td>no hashing</td>
<td>$\text{Adv}^{n-\text{IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}} n\delta(1)$</td>
<td>$n \cdot \sqrt{q_{\text{RO}} \text{Adv}^{\text{IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2 n\delta(1)$</td>
</tr>
</tbody>
</table>

- $\frac{n^2}{2^\ell}$ = probability of a collision in the prefixes, e.g. $2^\ell \approx 2^{256}$ and $n = 2^{30}$
- $pk$-hashing security $\approx$ prefix hashing security
### Results (Simplified)

<table>
<thead>
<tr>
<th>FO variant</th>
<th>$\text{Adv}^{n\text{-IND-CCA}}_{\text{KEM}}$ (ROM)</th>
<th>$\text{Adv}^{n\text{-IND-CCA}}_{\text{KEM}}$ (QROM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pk$ hashing</td>
<td>$\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}}\delta(n)$</td>
<td>$\sqrt{q_{\text{RO}}\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2\delta(n)$</td>
</tr>
<tr>
<td>prefix hashing</td>
<td>$\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}}\delta(n) + \frac{n^2}{2^\ell}$</td>
<td>$\sqrt{q_{\text{RO}}\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2\delta(n) + \frac{n^2}{2^\ell}$</td>
</tr>
<tr>
<td>no hashing</td>
<td>$\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PKE}} + q</em>{\text{RO}}n\delta(1)$</td>
<td>$n \cdot \sqrt{q_{\text{RO}}\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PKE}}} + q</em>{\text{RO}}^2n\delta(1)$</td>
</tr>
</tbody>
</table>

- $\frac{n^2}{2^\ell} = \text{probability of a collision in the prefixes, e.g. } 2^\ell \approx 2^{256} \text{ and } n = 2^{30}$
- $pk$-hashing security $\approx$ prefix hashing security
- $\{pk, \text{prefix}\}$-hashing security $> \text{no hashing security}$
Results (Simplified)

<table>
<thead>
<tr>
<th>FO variant</th>
<th>$\text{Adv}^n_{\text{KEM}}$ (ROM)</th>
<th>$\text{Adv}^n_{\text{KEM}}$ (QROM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pk$ hashing</td>
<td>$\text{Adv}^n_{\text{PKE}} + q_{RO}\delta(n)$</td>
<td>$\sqrt{q_{RO}\text{Adv}^n_{\text{PKE}}} + q_{RO}^2\delta(n)$</td>
</tr>
<tr>
<td>prefix hashing</td>
<td>$\text{Adv}^n_{\text{PKE}} + q_{RO}\delta(n) + \frac{n^2}{2^\ell}$</td>
<td>$\sqrt{q_{RO}\text{Adv}^n_{\text{PKE}}} + q_{RO}^2\delta(n) + \frac{n^2}{2^\ell}$</td>
</tr>
<tr>
<td>no hashing</td>
<td>$\text{Adv}^n_{\text{PKE}} + q_{RO}n\delta(1)$</td>
<td>$n \cdot \sqrt{q_{RO}\text{Adv}^n_{\text{PKE}}} + q_{RO}^2n\delta(1)$</td>
</tr>
</tbody>
</table>

- $\frac{n^2}{2^\ell} = \text{probability of a collision in the prefixes, e.g. } 2^\ell \approx 2^{256} \text{ and } n = 2^{30}$
- $pk$-hashing security $\approx$ prefix hashing security
- $\{pk, prefix\}$-hashing security $> \text{no hashing security}$
- prefix hashing efficiency $\gg pk$-hashing efficiency
## Results (Simplified)

<table>
<thead>
<tr>
<th>FO variant</th>
<th>$\text{Adv}^{n\text{-IND-CCA}}_{\text{KEM}}$ (ROM)</th>
<th>$\text{Adv}^{n\text{-IND-CCA}}_{\text{KEM}}$ (QROM)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$pk$ hashing</td>
<td>$\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PK}} + q</em>{\text{RO}} \delta(n)$</td>
<td>$\sqrt{q_{\text{RO}} \text{Adv}^{n\text{-IND-CPA}}<em>{\text{PK}}} + q</em>{\text{RO}}^2 \delta(n)$</td>
</tr>
<tr>
<td>prefix hashing</td>
<td>$\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PK}} + q</em>{\text{RO}} \delta(n) + \frac{n^2}{2\ell}$</td>
<td>$\sqrt{q_{\text{RO}} \text{Adv}^{n\text{-IND-CPA}}<em>{\text{PK}}} + q</em>{\text{RO}}^2 \delta(n) + \frac{n^2}{2\ell}$</td>
</tr>
<tr>
<td>no hashing</td>
<td>$\text{Adv}^{n\text{-IND-CPA}}<em>{\text{PK}} + q</em>{\text{RO}} n \delta(1)$</td>
<td>$n \cdot \sqrt{q_{\text{RO}} \text{Adv}^{n\text{-IND-CPA}}<em>{\text{PK}}} + q</em>{\text{RO}}^2 n \delta(1)$</td>
</tr>
</tbody>
</table>

- $\frac{n^2}{2\ell} = \text{probability of a collision in the prefixes, e.g. } 2^\ell \approx 2^{256} \text{ and } n = 2^{30}$
- $pk$-hashing security $\approx$ prefix hashing security
- $\{pk, \text{prefix}\}$-hashing security $>\!\!\!> no$ hashing security
- prefix hashing efficiency $\gg pk$-hashing efficiency
- $\Rightarrow$ use prefix hashing
# Application to Kyber and Saber

- FO with Prefix Hashing yields significant speed up to Kyber and Saber
- Speedup of Kyber is larger, due to the efficiency of the underlying IND-CPA-secure PKE

<table>
<thead>
<tr>
<th>NIST Level</th>
<th>Kyber</th>
<th>Saber</th>
<th>Speed-up</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Original</td>
<td>This Work</td>
<td>Speed-up</td>
<td>Original</td>
</tr>
<tr>
<td>1</td>
<td>KE</td>
<td>KED</td>
<td>45%</td>
<td>KE</td>
</tr>
<tr>
<td></td>
<td>23562</td>
<td>12883</td>
<td>54%</td>
<td>42169</td>
</tr>
<tr>
<td></td>
<td>37144</td>
<td>16981</td>
<td>0%</td>
<td>57831</td>
</tr>
<tr>
<td></td>
<td>28595</td>
<td>28529</td>
<td></td>
<td>57780</td>
</tr>
<tr>
<td>3</td>
<td>KE</td>
<td>KED</td>
<td>38%</td>
<td>KE</td>
</tr>
<tr>
<td></td>
<td>40487</td>
<td>25272</td>
<td>50%</td>
<td>74577</td>
</tr>
<tr>
<td></td>
<td>55726</td>
<td>27624</td>
<td>0%</td>
<td>95958</td>
</tr>
<tr>
<td></td>
<td>43553</td>
<td>43442</td>
<td></td>
<td>95388</td>
</tr>
<tr>
<td>5</td>
<td>KE</td>
<td>KED</td>
<td>30%</td>
<td>KE</td>
</tr>
<tr>
<td></td>
<td>55770</td>
<td>38815</td>
<td>47%</td>
<td>116178</td>
</tr>
<tr>
<td></td>
<td>77011</td>
<td>40692</td>
<td>0%</td>
<td>142034</td>
</tr>
<tr>
<td></td>
<td>61470</td>
<td>61473</td>
<td></td>
<td>142957</td>
</tr>
</tbody>
</table>
Conclusion

- For multi-user security, hashing the prefix of a public-key seems to be the right thing to do in the context of the FO
- Prefix hashing (more than) satisfies the NIST Security requirements
- Significant speedup for Kyber and Saber key-generation and encapsulation using prefix hashing, up to (56 − 66 %) and (30 − 39 %)
- Open Question: any other disadvantages for prefix hashing?
Conclusion

- For multi-user security, hashing the prefix of a public-key seems to be the right thing to do in the context of the FO
- Prefix hashing (more than) satisfies the NIST Security requirements
- Significant speedup for Kyber and Saber key-generation and encapsulation using prefix hashing, up to (56 – 66 %) and (30 – 39 %)
- Open Question: any other disadvantages for prefix hashing?
- Thank you for your attention
Mihir Bellare, Alexandra Boldyreva, and Silvio Micali.
Public-key encryption in a multi-user setting: Security proofs and improvements.

Eiichiro Fujisaki and Tatsuaki Okamoto.
How to enhance the security of public-key encryption at minimum cost.

Eiichiro Fujisaki and Tatsuaki Okamoto.
Secure integration of asymmetric and symmetric encryption schemes.

Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz.
A modular analysis of the Fujisaki-Okamoto transformation.