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

J. Duman[1], K. Hövelmanns[2], E. Kiltz[1], V. Lyubashevsky[3], G. Seiler[4]

Ruhr-University Bochum[1]
Eindhoven University of Technology[2]
IBM Zürich[34]
ETH Zürich[4]

9. Juny 2021

Intro

$$\text{PKE} \xrightarrow[]{\text{Fujisaki-Okamoto (FO) Transform}} \text{KEM}$$

IND-CPA secure PKE → IND-CCA secure KEM

- Standard method of almost all NIST PQC Candidates: start with IND-CPA secure PKE and apply variant of FO [FO99, FO13, HHK17]
- $\mathsf{Encaps}_{pk}(;r) = (\underbrace{\mathsf{Enc}_{pk}(r;\mathsf{G}(r))}_{\text{ciphertext}}, \underbrace{\mathsf{H}(r)}_{key})$

Intro

$$\underset{\text{PKE}}{\text{IND-CPA secure}} \xrightarrow{\text{Fujisaki-Okamoto (FO) Transform}} \underset{\text{KEM}}{\text{IND-CCA secure}}$$

- Standard method of almost all NIST PQC Candidates: start with IND-CPA secure PKE and apply variant of FO [FO99, FO13, HHK17]
- $\mathsf{Encaps}_{pk}(;r) = (\underbrace{\mathsf{Enc}_{pk}(r; \mathsf{G}(pk, r))}_{\text{ciphertext}}, \underbrace{\mathsf{H}(pk, r)}_{key})$
- Common safeguard against *multi-user attacks:* hash also *public-keys*, notably done by Kyber and Saber

Intro

$$\underset{\text{PKE}}{\overset{\text{n-IND-CPA secure}}{}} \xrightarrow{\text{Fujisaki-Okamoto (FO) Transform}} \underset{\text{KEM}}{\overset{\text{n-IND-CCA secure}}{}}$$

- Standard method of almost all NIST PQC Candidates: start with IND-CPA secure PKE and apply variant of FO [FO99, FO13, HHK17]
- $\mathsf{Encaps}_{pk}(;r) = (\underbrace{\mathsf{Enc}_{pk}(r; \mathsf{G}(pk, r))}_{\text{ciphertext}}, \underbrace{\mathsf{H}(pk, r)}_{key})$
- 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

$$\underset{\text{PKE}}{\text{n-IND-CPA secure}} \xrightarrow{\substack{\text{Fujisaki-Okamoto} \\ \text{(FO) Transform}}} \underset{\text{KEM}}{\text{n-IND-CCA secure}}$$

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

Intro

$$\underset{\text{PKE}}{\text{n-IND-CPA secure}} \xrightarrow{\text{Fujisaki-Okamoto (FO) Transform}} \underset{\text{KEM}}{\text{n-IND-CCA secure}}$$

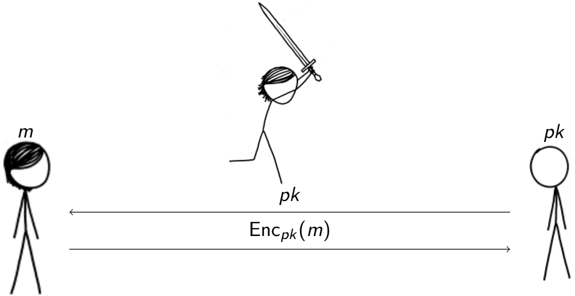
- Prefix hashing: improve the FO by hashing of a *short prefix* of the public-key $=:$ id instead of large public-key
  $\mathsf{Encaps}_{pk}(;r) = (\mathsf{Enc}_{pk}(r; \mathsf{G}(\mathsf{id}, r)), \mathsf{H}(\mathsf{id}, r))$
- $\Rightarrow$ important for *Lattice-based* KEMs since the public-keys 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

Intro

$$\underset{\text{PKE}}{\text{n-IND-CPA secure}} \xrightarrow{\text{Fujisaki-Okamoto (FO) Transform}} \underset{\text{KEM}}{\text{n-IND-CCA secure}}$$

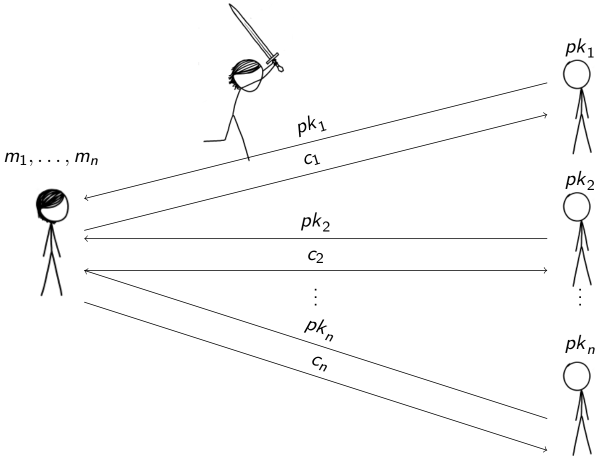
- Prefix hashing: improve the FO by hashing of a *short prefix* of the public-key $=: \mathsf{id}$ instead of large public-key
  $\mathsf{Encaps}_{pk}(;r) = (\mathsf{Enc}_{pk}(r; \mathsf{G}(\mathsf{id}, r)), \mathsf{H}(\mathsf{id}, r))$
- $\Rightarrow$ important for *Lattice-based* KEMs since the public-keys 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

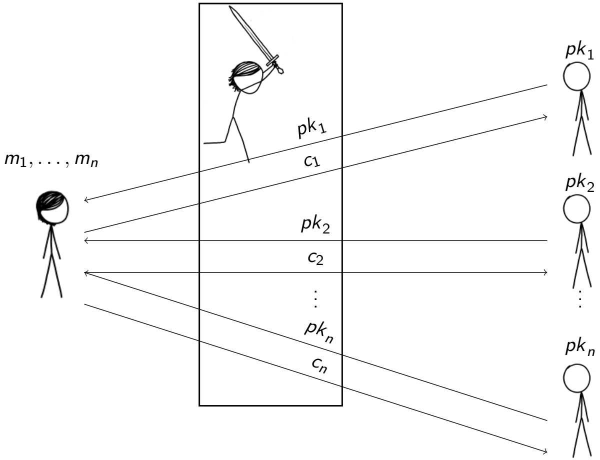


- Adversary wants to learn some information on plaintext m

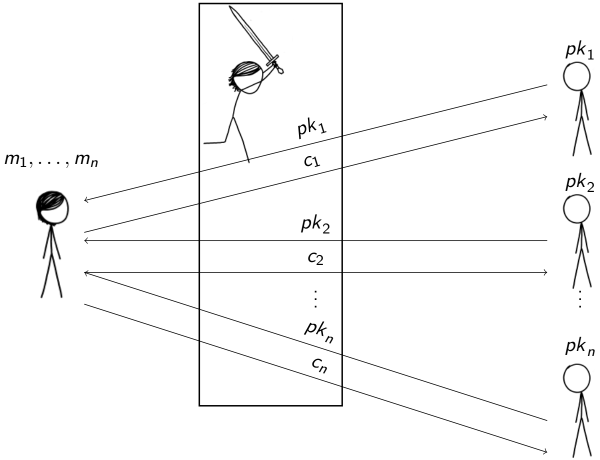
Multi-User IND-CPA (n-IND-CPA)



Multi-User IND-CPA (n-IND-CPA)



Multi-User IND-CPA (n-IND-CPA)



- $\Rightarrow$ Adversary wants to learn some information on the n 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] ☺
- Looses a factor of n, where $n = \#\mathsf{Users}$
- $\implies$ 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:
  - $\mathsf{Adv}_{\mathsf{PKE}}^{\text{n-IND-CPA}} \ll n \cdot \mathsf{Adv}_{\mathsf{PKE}}^{\text{IND-CPA}}$

FO without *pk* hashing

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

FO without pk hashing

- $\text{Encaps}_{pk}(;r) = (\underbrace{\text{Enc}_{pk}(r;\text{G}(r))}_{\text{ciphertext}}, \underbrace{\text{H}(r)}_{\text{key}})$
- Advantage: more efficient than additionally hashing pk
- Disadvantage: worse multi-user security

FO with Public-Key Hashing

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

FO with Public-Key Hashing

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

- $\mathsf{Encaps}_{pk}(;r) = (\underbrace{\mathsf{Enc}_{pk}(r; \mathsf{G}(\mathsf{id}, r))}_{\text{ciphertext}}, \underbrace{\mathsf{H}(\mathsf{id}, r)}_{\text{key}})$
- $\mathsf{id} := \mathsf{ID}(pk)$ = short prefix of the public-key, e.g. 32 Bytes

FO with Prefix Hashing

- $\mathsf{Encaps}_{pk}(;r) = (\underbrace{\mathsf{Enc}_{pk}(r;\mathsf{G}(\mathsf{id},r))}_{\text{ciphertext}}, \underbrace{\mathsf{H}(\mathsf{id},r)}_{\text{key}})$
- $\mathsf{id} := \mathsf{ID}(pk)$ = 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] δ-Correctness of PKE informally:
  probability of decryption error for a random key
- This work: $\delta(n)$-Correctness of PKE
  "worst δ-correctness from n random keys"

Correctness Errors

- [HHK17] δ-Correctness of PKE informally: probability of decryption error for a random key
- This work: $\delta(n)$-Correctness of PKE "worst δ-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] δ-Correctness of PKE informally:
  probability of decryption error for a random key
- This work: $\delta(n)$-Correctness of PKE
  "worst δ-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)

FO variant	$\mathsf{Adv}_{\mathsf{KEM}}^{\text{n-IND-CCA}}$ (ROM)	$\mathsf{Adv}_{\mathsf{KEM}}^{\text{n-IND-CCA}}$ (QROM)
pk hashing	$\mathsf{Adv}_{\mathsf{PKE}}^{\text{n-IND-CPA}} + q_{\mathsf{RO}}\delta(n)$	$\sqrt{q_{\mathsf{RO}}\mathsf{Adv}_{\mathsf{PKE}}^{\text{n-IND-CPA}}} + q_{\mathsf{RO}}^2\delta(n)$
prefix hashing	$\mathsf{Adv}_{\mathsf{PKE}}^{\text{n-IND-CPA}} + q_{\mathsf{RO}}\delta(n) + \frac{n^2}{2^\ell}$	$\sqrt{q_{\mathsf{RO}}\mathsf{Adv}_{\mathsf{PKE}}^{\text{n-IND-CPA}}} + q_{\mathsf{RO}}^2\delta(n) + \frac{n^2}{2^\ell}$
no hashing	$\mathsf{Adv}_{\mathsf{PKE}}^{\text{n-IND-CPA}} + q_{\mathsf{RO}}n\delta(1)$	$n \cdot \sqrt{q_{\mathsf{RO}}\mathsf{Adv}_{\mathsf{PKE}}^{\text{IND-CPA}}} + q_{\mathsf{RO}}^2 n\delta(1)$

Results (Simplified)

FO variant	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{n\text{-}IND\text{-}CCA}}$ (ROM)	$\mathsf{Adv}_{\mathsf{KEM}}^{\mathsf{n\text{-}IND\text{-}CCA}}$ (QROM)
pk hashing	$\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{n\text{-}IND\text{-}CPA}} + q_{\mathsf{RO}}\delta(n)$	$\sqrt{q_{\mathsf{RO}}\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{n\text{-}IND\text{-}CPA}}} + q_{\mathsf{RO}}^2\delta(n)$
prefix hashing	$\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{n\text{-}IND\text{-}CPA}} + q_{\mathsf{RO}}\delta(n) + \frac{n^2}{2^\ell}$	$\sqrt{q_{\mathsf{RO}}\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{n\text{-}IND\text{-}CPA}}} + q_{\mathsf{RO}}^2\delta(n) + \frac{n^2}{2^\ell}$
no hashing	$\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{n\text{-}IND\text{-}CPA}} + q_{\mathsf{RO}}n\delta(1)$	$n \cdot \sqrt{q_{\mathsf{RO}}\mathsf{Adv}_{\mathsf{PKE}}^{\mathsf{IND\text{-}CPA}}} + q_{\mathsf{RO}}^2 n\delta(1)$

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

Results (Simplified)

FO variant	$\mathrm{Adv}^{\text{n-IND-CCA}}_{\mathrm{KEM}}$ (ROM)	$\mathrm{Adv}^{\text{n-IND-CCA}}_{\mathrm{KEM}}$ (QROM)
pk hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\mathrm{PKE}} + q_{\mathrm{RO}}\delta(n)$	$\sqrt{q_{\mathrm{RO}}\mathrm{Adv}^{\text{n-IND-CPA}}_{\mathrm{PKE}}} + q_{\mathrm{RO}}^2\delta(n)$
prefix hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\mathrm{PKE}} + q_{\mathrm{RO}}\delta(n) + \frac{n^2}{2^\ell}$	$\sqrt{q_{\mathrm{RO}}\mathrm{Adv}^{\text{n-IND-CPA}}_{\mathrm{PKE}}} + q_{\mathrm{RO}}^2\delta(n) + \frac{n^2}{2^\ell}$
no hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\mathrm{PKE}} + q_{\mathrm{RO}}n\delta(1)$	$n \cdot \sqrt{q_{\mathrm{RO}}\mathrm{Adv}^{\text{IND-CPA}}_{\mathrm{PKE}}} + q_{\mathrm{RO}}^2 n\delta(1)$

- $\frac{n^2}{2^\ell}$ = 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)

FO variant	$\mathrm{Adv}_{\mathrm{KEM}}^{\text{n-IND-CCA}}$ (ROM)	$\mathrm{Adv}_{\mathrm{KEM}}^{\text{n-IND-CCA}}$ (QROM)
pk hashing	$\mathrm{Adv}_{\mathrm{PKE}}^{\text{n-IND-CPA}} + q_{\mathrm{RO}}\delta(n)$	$\sqrt{q_{\mathrm{RO}}\mathrm{Adv}_{\mathrm{PKE}}^{\text{n-IND-CPA}}} + q_{\mathrm{RO}}^2\delta(n)$
prefix hashing	$\mathrm{Adv}_{\mathrm{PKE}}^{\text{n-IND-CPA}} + q_{\mathrm{RO}}\delta(n) + \frac{n^2}{2^\ell}$	$\sqrt{q_{\mathrm{RO}}\mathrm{Adv}_{\mathrm{PKE}}^{\text{n-IND-CPA}}} + q_{\mathrm{RO}}^2\delta(n) + \frac{n^2}{2^\ell}$
no hashing	$\mathrm{Adv}_{\mathrm{PKE}}^{\text{n-IND-CPA}} + q_{\mathrm{RO}}n\delta(1)$	$n \cdot \sqrt{q_{\mathrm{RO}}\mathrm{Adv}_{\mathrm{PKE}}^{\text{IND-CPA}}} + q_{\mathrm{RO}}^2 n\delta(1)$

- $\frac{n^2}{2^\ell}$ = 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
- $\{pk, \text{prefix}\}$-hashing security $>$ no hashing security

Results (Simplified)

FO variant	$\mathrm{Adv}^{\text{n-IND-CCA}}_{\text{KEM}}$ (ROM)	$\mathrm{Adv}^{\text{n-IND-CCA}}_{\text{KEM}}$ (QROM)
pk hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}} + q_{\text{RO}}\delta(n)$	$\sqrt{q_{\text{RO}}\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}}} + q^2_{\text{RO}}\delta(n)$
prefix hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}} + q_{\text{RO}}\delta(n) + \frac{n^2}{2^\ell}$	$\sqrt{q_{\text{RO}}\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}}} + q^2_{\text{RO}}\delta(n) + \frac{n^2}{2^\ell}$
no hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}} + q_{\text{RO}}n\delta(1)$	$n \cdot \sqrt{q_{\text{RO}}\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}}} + q^2_{\text{RO}}n\delta(1)$

- $\frac{n^2}{2^\ell}$ = 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
- $\{pk, \text{prefix}\}$-hashing security $>$ no hashing security
- prefix hashing efficiency $\gg$ *pk*-hashing efficiency

Results (Simplified)

FO variant	$\mathrm{Adv}^{\text{n-IND-CCA}}_{\text{KEM}}$ (ROM)	$\mathrm{Adv}^{\text{n-IND-CCA}}_{\text{KEM}}$ (QROM)
pk hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}} + q_{\text{RO}}\delta(n)$	$\sqrt{q_{\text{RO}}\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}}} + q_{\text{RO}}^2\delta(n)$
prefix hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}} + q_{\text{RO}}\delta(n) + \frac{n^2}{2^\ell}$	$\sqrt{q_{\text{RO}}\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}}} + q_{\text{RO}}^2\delta(n) + \frac{n^2}{2^\ell}$
no hashing	$\mathrm{Adv}^{\text{n-IND-CPA}}_{\text{PKE}} + q_{\text{RO}}n\delta(1)$	$n \cdot \sqrt{q_{\text{RO}}\mathrm{Adv}^{\text{IND-CPA}}_{\text{PKE}}} + q_{\text{RO}}^2 n\delta(1)$

- $\frac{n^2}{2^\ell}$ = 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
- $\{pk, \text{prefix}\}$-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

NIST Level		Kyber			Saber		
		Original	This Work	Speed-up	Original	This Work	Speed-up
1	**K**	23562	12883	45%	42169	36220	14%
	E	37144	16981	54%	57831	39232	32%
	D	28595	28529	0%	57780	57806	0%
3	**K**	40487	25272	**38%**	74577	64180	**14%**
	E	55726	27624	**50%**	95958	69304	**28%**
	D	43553	43442	0%	95388	95301	0%
5	**K**	55770	38815	30%	116178	102101	12%
	E	77011	40692	47%	142034	109203	23%
	D	61470	61473	0%	142957	143090	0%

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.
In Bart Preneel, editor, *EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 259–274. Springer, Heidelberg, May 2000.

Eiichiro Fujisaki and Tatsuaki Okamoto.
How to enhance the security of public-key encryption at minimum cost.
In Hideki Imai and Yuliang Zheng, editors, *PKC'99*, volume 1560 of *LNCS*, pages 53–68. Springer, Heidelberg, March 1999.

Eiichiro Fujisaki and Tatsuaki Okamoto.
Secure integration of asymmetric and symmetric encryption schemes.
Journal of Cryptology, 26(1):80–101, January 2013.

Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz.
A modular analysis of the Fujisaki-Okamoto transformation.
In Yael Kalai and Leonid Reyzin, editors, *TCC 2017, Part I*, volume 10677 of *LNCS*, pages 341–371. Springer, Heidelberg, November 2017.