Generic Authenticated Key Exchange in the Quantum Random Oracle Model

Kathrin Hövelmanns[1] Eike Kiltz[1]
Sven Schäge[1] Dominique Unruh[2]

[1]Horst Görtz Institute for IT Security, Ruhr University Bochum, Germany

[2]Institute of Computer Science, University of Tartu, Estonia

PKC 2020

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Desired:	Active security (CCA)
Easier to achieve:	Passive security (OW/CPA)
	E.g., from lattice assumptions

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Desired:	Active security (CCA)
Easier to achieve:	Passive security (OW/CPA) E.g., from lattice assumptions

Can we turn passive into active, generically?

Frequently used solution: Fujisaki-Okamoto and its variants

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Desired:	Active security (CCA)
Easier to achieve:	Passive security (OW/CPA)
	E.g., from lattice assumptions

Can we turn passive into active, generically?

Frequently used solution: Fujisaki-Okamoto and its variants

Major technical problem: Probability of decryption failure

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Context: NIST 'competition'

Goal: Quantum-secure public-key encryption, key exchange, signatures

Pre-quantum: DH key exchange + authentification

Post-quantum:

- DH key exchange: Broken
- Quantum Signatures: Quite costly → Can we do without them?

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Actually two protocols, achieving different levels of security

Protocol 1: Relatively weak: Revealed state breaks security

Protocol 2: Strengthen protocol 1 by adding session-specific DH layer

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Actually two protocols, achieving different levels of security

Protocol 1: Relatively weak: Revealed state breaks security

Protocol 2: Strengthen protocol 1 by adding session-specific DH layer

→ Not suitable for post-quantum ☹

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Builds on BCGNP08 design principle:

Add session-specific layer via any (passively secure) KEM

Session-specific layer + add. trick

→ Resistance against exposure of secret data

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12

Builds on BCGNP08 design principle:

Add session-specific layer via any (passively secure) KEM

Session-specific layer + add. trick

→ Resistance against exposure of secret data

... but the underlying scheme is assumed to be perfectly correct

→ Possibly not suitable for post-quantum ☹

Applying FO-then-FSXY12 also results in quite a lot of hashing

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12 ... but possibly unfit for post-quantum security

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12 ... but possibly unfit for post-quantum security

Our goal: A simplified transformation that

- is secure against quantum adversaries,
- even for non-perfectly correct schemes, and
- gets rid of unnecessary hashing steps

Prior work on AKE without signatures

AKE from KEMs: Already proposed in BCGNP08 and FSXY12 ... but possibly unfit for post-quantum security

Our goal: A simplified transformation that

- is secure against quantum adversaries,
- even for non-perfectly correct schemes, and
- gets rid of unnecessary hashing steps

Our proposal: 'AKE version' of Fujisaki-Okamoto

Turns passively secure PKE into post-quantum secure AKE

Kyber-Key exchange: Kyber-PKE + this work

Outline

1. The Fujisaki-Okamoto transformation
2. Two-move authenticated key exchange (AKE)
3. Our protocol: Fujisaki-Okamoto AKE
4. Open questions

Overview:
The Fujisaki-Okamoto transformation

Limitations of the original work

Decryption failure?

Reminder: Property of many lattice-based encryption schemes

HHK17: Even negligible probability might affect security!

The importance of decryption failures

Intuition: Negligible probability → negligible issue

The importance of decryption failures

Intuition: Negligible probability → negligible issue... but:

Active attacker can query decapsulation oracle on any ciphertext

Failure depending on sk → leaks information

The importance of decryption failures

Intuition: Negligible probability → negligible issue... but:

- Active attacker can query decapsulation oracle on any ciphertext
  - Failure depending on sk → leaks information
- Reflected in D'AVV18 attack

The importance of decryption failures

Intuition: Negligible probability → negligible issue... but:

Active attacker can query decapsulation oracle on any ciphertext

Failure depending on sk → leaks information

Reflected in D'AVV18 attack

Possible solutions:

1. Only build schemes with perfect correctness
   - Costly ☹
   - What about the NIST proposals? ☹
2. Give proofs that deal with non-perfect correctness

Limitations of the original work

Decryption failure?

Reminder: Property of many lattice-based encryption schemes

Limitations of the original work

Decryption failure?

Reminder: Property of many lattice-based encryption schemes

Original proof in random oracle model

Limitations of the original work

Decryption failure?

Reminder: Property of many lattice-based encryption schemes

Original proof in random oracle model → What if A is quantum?

Short excursion: The Quantum Random Oracle Model

Random Oracle Model (ROM)

Proof heuristic: Replace hash fct. with perfectly random fct. H

Random Oracle Model (ROM)

Proof heuristic: Replace hash fct. with perfectly random fct. H

Common proof strategy:

A can distinguish $\mathsf{H}(x^*)$ from random

$\Rightarrow$ Reduction learns preimage x^* (and x^* solves P)

Example: Learning message m^* $\Rightarrow$ Inverting a ciphertext

Random Oracle Model (ROM)

Proof heuristic: Replace hash fct. with perfectly random fct. H

Common proof strategy:

A can distinguish $\mathsf{H}(x^*)$ from random

$\Rightarrow$ Reduction learns preimage x^* (and x^* solves P)

Example: Learning message m^* $\Rightarrow$ Inverting a ciphertext

Question: What if A is quantum?

Quantum Random Oracle Model (QROM)

Scenario: Quantum adversary interacting with non-quantum network ⇒

- "Online" primitives (decryption, signing, ...) stay classical
- "Offline" primitives (like hash functions) computable in superposition

Quantum Random Oracle Model (QROM)

Scenario: Quantum adversary interacting with non-quantum network ⇒

- "Online" primitives (decryption, signing, ...) stay classical
- "Offline" primitives (like hash functions) computable in superposition

What's new: A might evaluate hash function on some superposition

$$\sum_{x \in X} \alpha_x |x\rangle$$

Quantum Random Oracle Model (QROM)

Scenario: Quantum adversary interacting with non-quantum network ⇒

- "Online" primitives (decryption, signing, ...) stay classical
- "Offline" primitives (like hash functions) computable in superposition

What's new: A might evaluate hash function on some superposition

$$\sum_{x \in X} \alpha_x |x\rangle$$

Possibility of A pulling 'quantum tricks' → More complicated proofs ☹

Quantum Random Oracle Model (QROM)

Scenario: Quantum adversary interacting with non-quantum network ⇒

- "Online" primitives (decryption, signing, ...) stay classical
- "Offline" primitives (like hash functions) computable in superposition

What's new: A might evaluate hash function on some superposition

$$\sum_{x \in X} \alpha_x |x\rangle$$

Possibility of A pulling 'quantum tricks' → More complicated proofs ☹

Example: How do we extract a particular preimage?

Extracting preimages with 'Oneway to Hiding'

"Random-until-QUERY": :

$\Pr[A \text{ distinguishes } \mathsf{H}(x^*) \text{ from } \$] \le \epsilon$

$\epsilon := \Pr[A \text{ queries } \mathsf{H} \text{ on } x^*]$

Extracting preimages with 'Oneway to Hiding'

"Random-until-QUERY" in the quantum world ('Oneway to Hiding'):

$\Pr[A \text{ distinguishes } \mathsf{H}(x^*) \text{ from } \$] \leq \epsilon$

~~$\epsilon := \Pr[A \text{ queries } \mathsf{H} \text{ on } x^*]$~~

Extracting preimages with 'Oneway to Hiding'

"Random-until-QUERY" in the quantum world ('Oneway to Hiding'):

$\Pr[A \text{ distinguishes } \mathsf{H}(x^*) \text{ from } \$] \leq 2q \cdot \sqrt{\epsilon}$

~~$\epsilon := \Pr[A \text{ queries H on } x^*]$~~

$\epsilon := \Pr[\text{Measuring a random query to H gives us } x^*]$

and q := # queries to H

Extracting preimages with 'Oneway to Hiding'

"Random-until-QUERY" in the quantum world ('Oneway to Hiding'):

$\Pr[A \text{ distinguishes } \mathrm{H}(x^*) \text{ from } \$] \leq 2q \cdot \sqrt{\epsilon}$

~~$\epsilon := \Pr[A \text{ queries H on } x^*]$~~

$\epsilon := \Pr[\text{Measuring a random query to H gives us } x^*]$

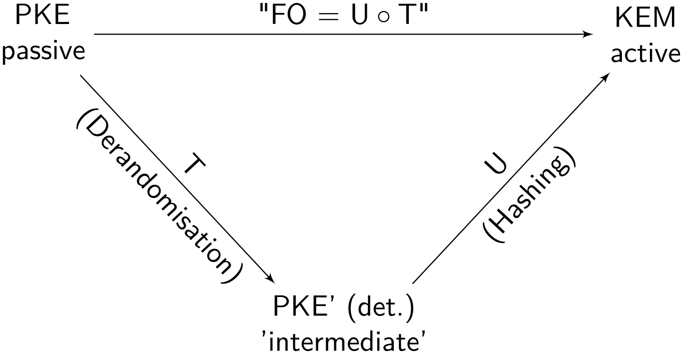
and q := # queries to H

Recent improvements :

Variant	Bound
Semi-classical [AHU18]	$2\sqrt{q\epsilon}$
Double-sided [BH+19]	$2\sqrt{\epsilon}$

The FO transformation in the QROM

Overview: Common ground of all current FO proofs



PKE
passive

"FO = U ∘ T"

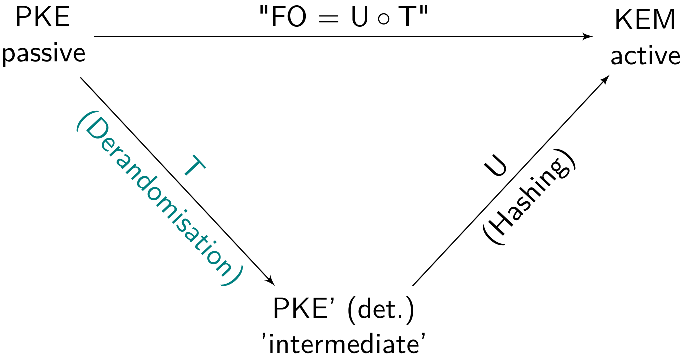
KEM
active

T
(Derandomisation)

U
(Hashing)

PKE' (det.)
'intermediate'

Overview: Common ground of all current FO proofs



PKE
passive

"FO = U ∘ T"

KEM
active

T
(Derandomisation)

U
(Hashing)

PKE' (det.)
'intermediate'

Transformation T

Encrypt-with-Hash construction: $\mathsf{PKE}' := \mathsf{T}[\mathsf{PKE}, \mathsf{G}]$

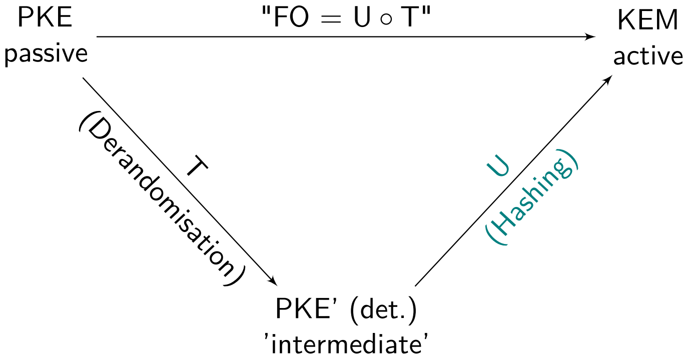
- Encryption: $\mathsf{Enc}'(m) := \mathsf{Enc}(m; \mathsf{G}(m))$
  → deterministic!

Use $\mathsf{G}(m)$ as Enc's randomness

Overview: Common ground of all current FO proofs

$$\begin{tikzcd}
\begin{matrix}\text{PKE}\\ \text{passive}\end{matrix} \arrow[rr, "\text{\"FO} = \text{U} \circ \text{T\"}"] \arrow[dr, "\text{T}", "\text{(Derandomisation)}"'] & & \begin{matrix}\text{KEM}\\ \text{active}\end{matrix} \\
& \begin{matrix}\text{PKE' (det.)}\\ \text{'intermediate'}\end{matrix} \arrow[ur, "\text{U}", "\text{(Hashing)}"'] &
\end{tikzcd}$$

Overview: Common ground of all current FO proofs



PKE
passive

"FO = U ∘ T"

KEM
active

T
(Derandomisation)

U
(Hashing)

PKE' (det.)
'intermediate'

Transformation U

$\mathsf{KEM} := \mathsf{U}[\mathsf{PKE}', \mathsf{H}]$

- Encapsulation:
  1. Choose uniformly random plaintext m
  2. Use Enc' to encrypt m to ciphertext c
  3. $k := \mathsf{H}(m, c)$

Transformation U

$\mathsf{KEM} := \mathsf{U}[\mathsf{PKE}', \mathsf{H}]$

- Encapsulation:
  1. Choose uniformly random plaintext m
  2. Use Enc$'$ to encrypt m to ciphertext c
  3. $k := \mathsf{H}(m, c)$

- Decapsulation:

```
1. Use Dec$'$ to decrypt $c$ to plaintext $m'$
2. If $c$ decrypts to $\perp$
3.     return $\perp$
4. return $k' := \mathsf{H}(m', c)$
```

Transformation U

$\mathsf{KEM} := \mathsf{U}[\mathsf{PKE}', \mathsf{H}]$

- Encapsulation:

```
1. Choose uniformly random plaintext $m$
2. Use Enc′ to encrypt $m$ to ciphertext $c$
3. $k := \mathsf{H}(m, c)$
```

- Decapsulation:

```
1. Use Dec′ to decrypt $c$ to plaintext $m'$
2. If $c$ decrypts to $\perp$
3.     return $\perp$
4. return $k' := \mathsf{H}(m', c)$
```

Actually, there are many different variants of U.

Transformation U

$\mathsf{KEM} := \mathsf{U}[\mathsf{PKE}', \mathsf{H}]$

- Encapsulation:
  1. Choose uniformly random plaintext m
  2. Use Enc′ to encrypt m to ciphertext c
  3. $k := \mathsf{H}(m, c)$ or $\mathsf{H}(m)$

- Decapsulation:
  1. Use Dec′ to decrypt c to plaintext m'
  2. If c decrypts to $\perp$
  3.   return $\perp$
  4. return $k' := \mathsf{H}(m', c)$ or $\mathsf{H}(m')$

Actually, there are many different variants of U.

Transformation U

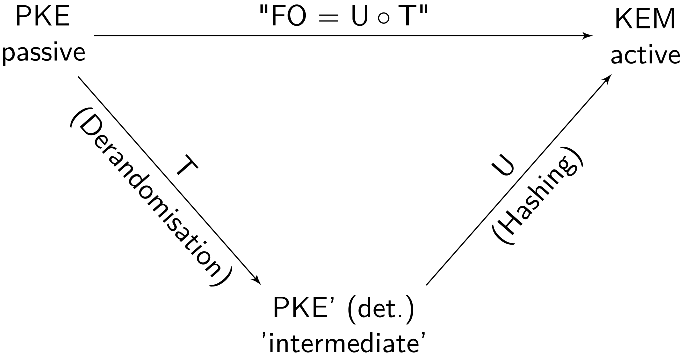
$\mathsf{KEM} := \mathsf{U}[\mathsf{PKE}', \mathsf{H}]$

- Encapsulation:
  1. Choose uniformly random plaintext m
  2. Use Enc′ to encrypt m to ciphertext c
  3. $k := \mathsf{H}(m, c)$ or $\mathsf{H}(m)$

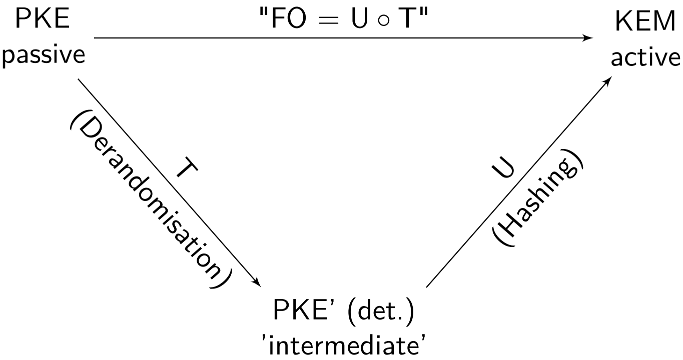
- Decapsulation:
  1. Use Dec′ to decrypt c to plaintext m'
  2. If c decrypts to $\perp$
  3. return $\perp$ or pseudorandom value ("implicit rejection")
  4. return $k' := \mathsf{H}(m', c)$ or $\mathsf{H}(m')$

Actually, there are many different variants of U.

Overview: Common ground of all current FO proofs



Overview: Common ground of all current FO proofs



PKE

passive

"FO = U ∘ T"

KEM

active

T

(Derandomisation)

U

(Hashing)

PKE' (det.)

'intermediate'

At least one step encounters quantum extraction problem

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)
This work	CPA	$\sqrt{q \cdot \mathsf{CPA}}$

q := # random oracle queries

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)
This work	CPA	$\sqrt{q \cdot \mathsf{CPA}}$
BHHHP19	OW (det.)	$\sqrt{\cancel{q \cdot}\, \mathsf{OW}}$

q := # random oracle queries

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)
This work	CPA	$\sqrt{q \cdot \mathrm{CPA}}$
BHHHP19	OW (det.)	$\sqrt{\cancel{q \cdot}\, \mathrm{OW}}$

q := # random oracle queries

PKE already deterministic → sufficient to apply second step (U)

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)
This work	CPA	$\sqrt{q \cdot \mathsf{CPA}}$
BHHHP19	OW (det.)	$\sqrt{\cancel{q \cdot}\ \mathsf{OW}}$
KSSSS20	OW (det.)	$q \cdot \mathsf{OW}$
	CPA	$q^2 \cdot \mathsf{CPA}$

q := # random oracle queries

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)
This work	CPA	$\sqrt{q \cdot \mathrm{CPA}}$
BHHHP19	OW (det.)	$\sqrt{\cancel{q\cdot}\ \mathrm{OW}}$
KSSSS20	OW (det.)	$q \cdot \mathrm{OW}$
	CPA	$q^2 \cdot \mathrm{CPA}$

q := # random oracle queries

'Rootless' bound:

Achieved by new extraction technique ('Measure-rewind-measure')

Simplified overview: Subsequent CCA bounds

Goal: Tightly relate FO-KEM security to that of the underlying scheme

	Underlying notion	CCA Bound (simplified)
Wishful thinking	CPA	CPA (achieved in ROM)
This work	CPA	$\sqrt{q \cdot \mathrm{CPA}}$
BHHHP19	OW (det.)	$\sqrt{\cancel{q\cdot}\ \mathrm{OW}}$
KSSSS20	OW (det.)	$q \cdot \mathrm{OW}$
	CPA	$q^2 \cdot \mathrm{CPA}$

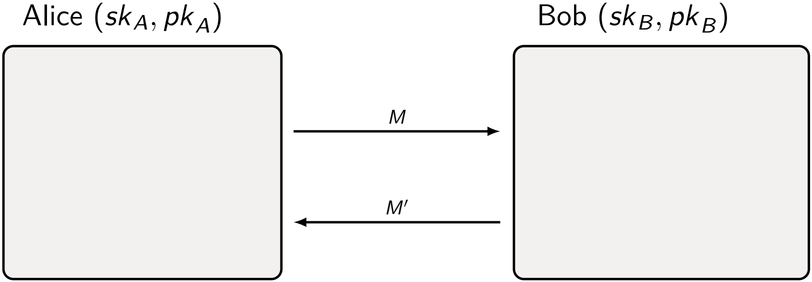
q := # random oracle queries

Cave: Results for different variants (like on the U-Slide), with additional requirements

More details at https://simons.berkeley.edu/talks/cca-encryption-qrom-i

Authenticated key exchange

Our setting: 2-move protocols



Alice (sk_A, pk_A)

Bob (sk_B, pk_B)

M

M'

Goal: $K = K'$ (w.o.p.), and $K \approx_c \$$

Attacking 2-move protocols

In practice, there are many ways to attack:

Attacking 2-move protocols

In practice, there are many ways to attack:

Learning session keys of already established sessions

Attacking 2-move protocols

In practice, there are many ways to attack:

Learning session keys of already established sessions

Corrupting a user → Learning sk_A or sk_B (or even both!)

Attacking 2-move protocols

In practice, there are many ways to attack:

Learning session keys of already established sessions

Corrupting a user → Learning sk_A or sk_B (or even both!)

Learning the session's state or the randomness that was used

Attacking 2-move protocols

In practice, there are many ways to attack:

Learning session keys of already established sessions

Corrupting a user → Learning sk_A or sk_B (or even both!)

Learning the session's state or the randomness that was used

'Tampering': Modifying the exchanged messages

Many different security models that come with subtle differences

Our security model

Two (game-based) models for two-move AKE:

1.) Key indistinguishability against active attacks

Our security model

Two (game-based) models for two-move AKE:

1.) Key indistinguishability against active attacks

Captures state-of-the-art attack capabilities:

- Key compromise impersonation attacks (KCI)
- Maximal exposure attacks (MEX)
- Reflection attacks
- Weak perfect forward secrecy (wPFS)

Our security model

Two (game-based) models for two-move AKE:

1.) Key indistinguishability against active attacks

Captures state-of-the-art attack capabilities:

- Key compromise impersonation attacks (KCI)
- Maximal exposure attacks (MEX)
- Reflection attacks
- Weak perfect forward secrecy (wPFS)

2.) Slightly weaker variant of the model above:

Disallow state reveal for the test session if adversary 'tampers'

- Only affects
  - initiator session
  - time interval between sending and receiving

Our security model

Two (game-based) models for two-move AKE:

1.) Key indistinguishability against active attacks

Captures state-of-the-art attack capabilities:

- Key compromise impersonation attacks (KCI)
- Maximal exposure attacks (MEX)
- Reflection attacks
- Weak perfect forward secrecy (wPFS)

2.) Slightly weaker variant of the model above:

Disallow state reveal for the test session if adversary 'tampers'

- Only affects
    - initiator session
    - time interval between sending and receiving
- In practice: restricted by initiator's waiting time

Our security model

Two (game-based) models for two-move AKE:

1.) Key indistinguishability against active attacks

Captures state-of-the-art attack capabilities:

- Key compromise impersonation attacks (KCI)
- Maximal exposure attacks (MEX)
- Reflection attacks
- Weak perfect forward secrecy (wPFS)

2.) Slightly weaker variant of the model above:

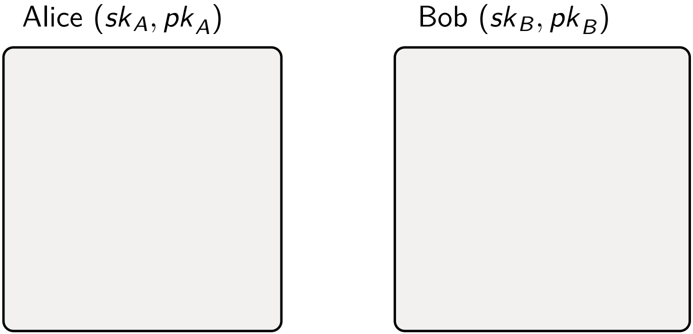
Disallow state reveal for the test session if adversary 'tampers'

- Only affects
  - initiator session
  - time interval between sending and receiving
- In practice: restricted by initiator's waiting time

Essentially same notion as the one used in FSXY12

Our protocol: Fujisaki-Okamoto key exchange

Our protocol

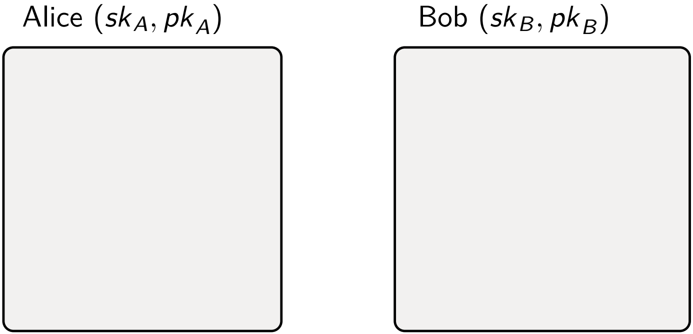


Alice (sk_A, pk_A)

Bob (sk_B, pk_B)

Goal: Authentication and key indistinguishability

Our protocol



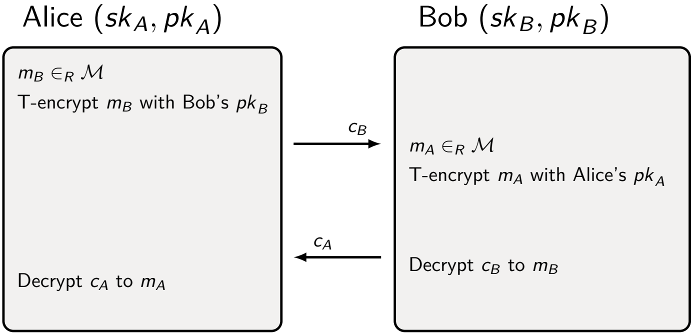
Alice (sk_A, pk_A)

Bob (sk_B, pk_B)

Goal: Authentication and key indistinguishability

Strategy: 'Multi-user FO':

Our protocol



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

c_B

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

c_A

Decrypt c_B to m_B

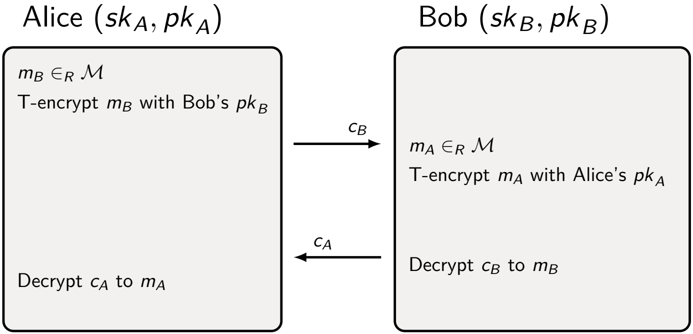
Decrypt c_A to m_A

Goal: Authentication and key indistinguishability

Strategy: 'Multi-user FO':

Exchange FO-ciphertexts = ciphertexts according to T-Transform

Our protocol



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

Decrypt c_A to m_A

c_B

c_A

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

Decrypt c_B to m_B

Goal: Authentication and key indistinguishability

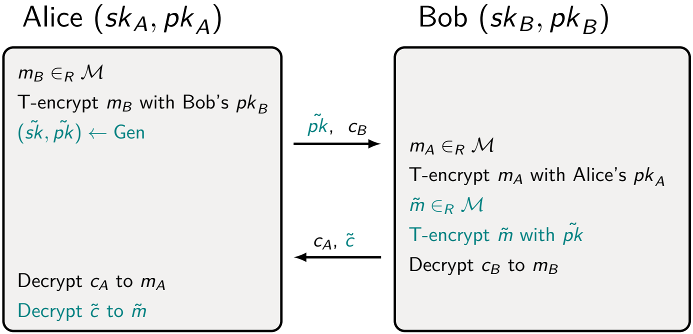
Strategy: 'Multi-user FO':

Exchange FO-ciphertexts = ciphertexts according to T-Transform

Key computation: Multi-user variant of U-Transform

Hash whole transcript: $K := \mathrm{H}(pk_A, pk_B, m_A, m_B, c_A, c_B)$

Our protocol



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow$ Gen

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

$\tilde{pk}$, c_B

c_A, $\tilde{c}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

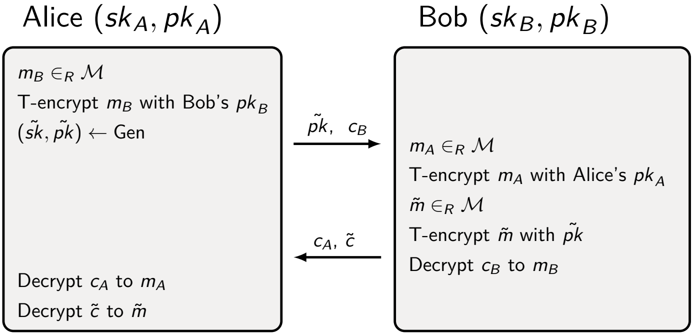
$\tilde{m} \in_R \mathcal{M}$

T-encrypt $\tilde{m}$ with $\tilde{pk}$

Decrypt c_B to m_B

Freshness: Add session-specific ('ephemeral') FO communication

Our protocol



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow$ Gen

$\xrightarrow{\tilde{pk},\ c_B}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

$\tilde{m} \in_R \mathcal{M}$

T-encrypt $\tilde{m}$ with $\tilde{pk}$

Decrypt c_B to m_B

$\xleftarrow{c_A,\ \tilde{c}}$

Decrypt c_A to m_A

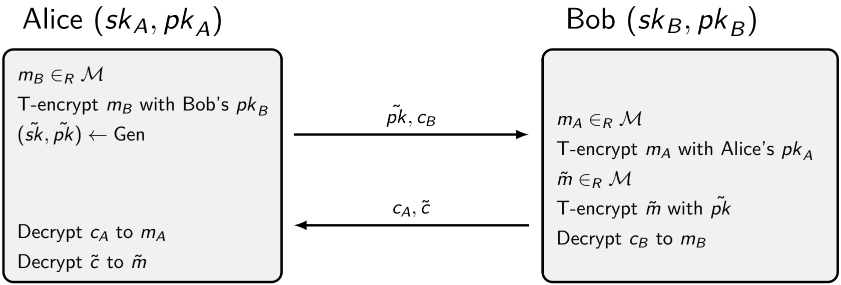
Decrypt $\tilde{c}$ to $\tilde{m}$

Freshness: Add session-specific ('ephemeral') FO communication

Include 'ephemeral transcript' in hash:

$$K := \mathrm{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$$

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{Gen}$

$\xrightarrow{\tilde{pk}, c_B}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

$\tilde{m} \in_R \mathcal{M}$

T-encrypt $\tilde{m}$ with $\tilde{pk}$

Decrypt c_B to m_B

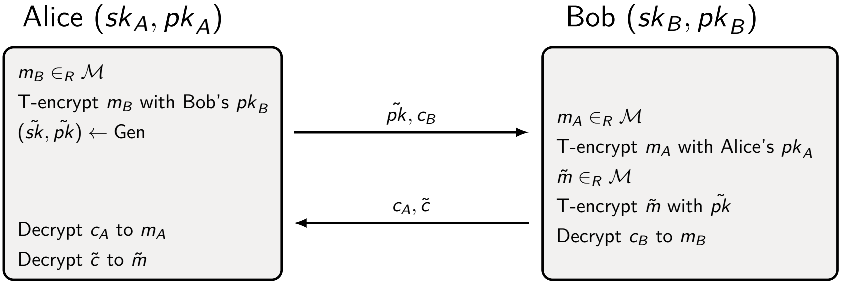
$\xleftarrow{c_A, \tilde{c}}$

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

$$K := \mathrm{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$$

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow \text{Gen}$

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

$\tilde{pk}, c_B$

$c_A, \tilde{c}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

$\tilde{m} \in_R \mathcal{M}$

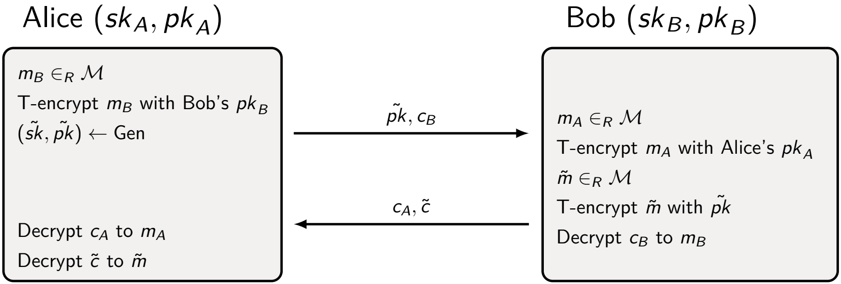
T-encrypt $\tilde{m}$ with $\tilde{pk}$

Decrypt c_B to m_B

$$K := \mathrm{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$$

Observation: Nontrivial strategy $\rightarrow$ E only obtains 2 out of $(m_i, m_j, \tilde{m})$

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{Gen}$

$\xrightarrow{\tilde{pk}, c_B}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

$\tilde{m} \in_R \mathcal{M}$

T-encrypt $\tilde{m}$ with $\tilde{pk}$

Decrypt c_B to m_B

$\xleftarrow{c_A, \tilde{c}}$

Decrypt c_A to m_A

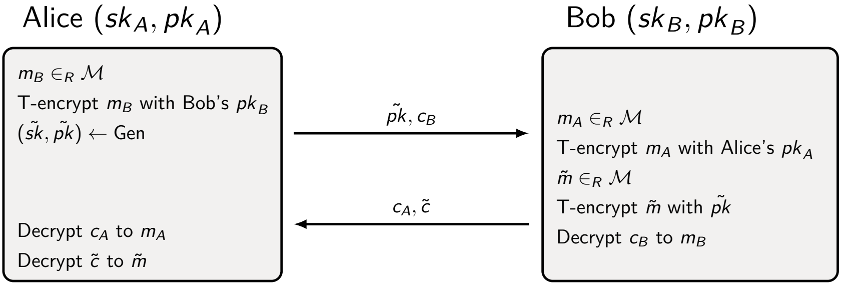
Decrypt $\tilde{c}$ to $\tilde{m}$

$K := \mathsf{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$

Observation: Nontrivial strategy → 😈 only obtains 2 out of $(m_i, m_j, \tilde{m})$

With observation, AKE proof ≈ multi-user version of our KEM proof

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow$ Gen

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

$\tilde{pk}, c_B$

$c_A, \tilde{c}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

$\tilde{m} \in_R \mathcal{M}$

T-encrypt $\tilde{m}$ with $\tilde{pk}$

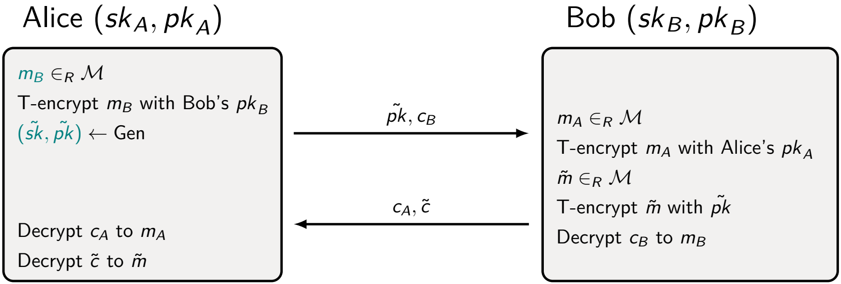
Decrypt c_B to m_B

$$K := \mathrm{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$$

Observation: Nontrivial strategy $\rightarrow$ $\lightning$ only obtains 2 out of $(m_i, m_j, \tilde{m})$

Exception: Aforementioned 'state reveal attack':

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow$ Gen

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

$\tilde{pk}, c_B$

$c_A, \tilde{c}$

Bob (sk_B, pk_B)

$m_A \in_R \mathcal{M}$

T-encrypt m_A with Alice's pk_A

$\tilde{m} \in_R \mathcal{M}$

T-encrypt $\tilde{m}$ with $\tilde{pk}$

Decrypt c_B to m_B

$K := \mathrm{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$

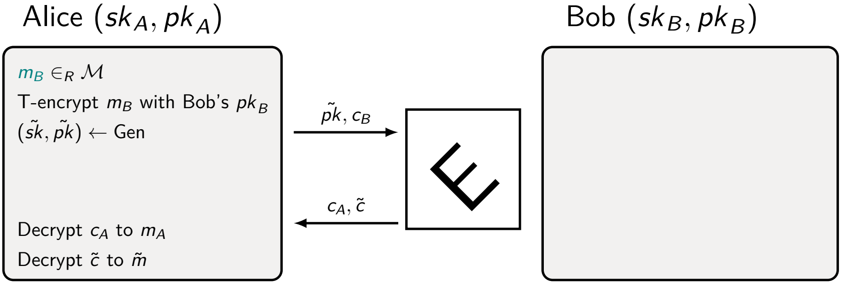
Observation: Nontrivial strategy $\rightarrow \rotatebox[origin=c]{45}{E}$ only obtains 2 out of $(m_i, m_j, \tilde{m})$

Exception: Aforementioned 'state reveal attack':

Alice's state: independent of sk_A

Bob's response (and m_A, $\tilde{m}$): independent of sk_B

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow \mathsf{Gen}$

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

Bob (sk_B, pk_B)

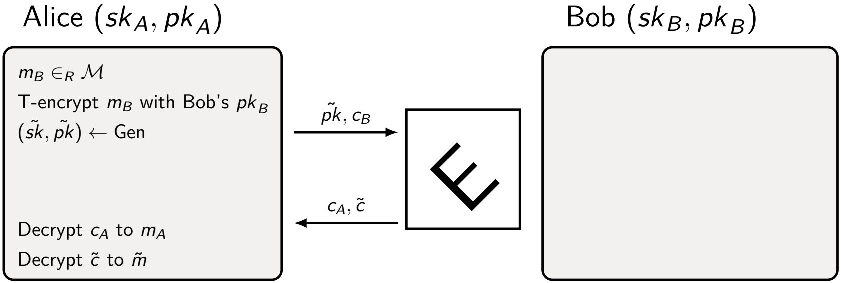
$$K := \mathsf{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$$

Observation: Nontrivial strategy → $\mathcal{E}$ only obtains 2 out of $(m_i, m_j, \tilde{m})$

Exception: Aforementioned 'state reveal attack':

Reveal the state to learn m_B and pretend to be Bob to control m_A, $\tilde{m}$

Security of our protocol (Intuition)



Alice (sk_A, pk_A)

$m_B \in_R \mathcal{M}$

T-encrypt m_B with Bob's pk_B

$(\tilde{sk}, \tilde{pk}) \leftarrow$ Gen

Decrypt c_A to m_A

Decrypt $\tilde{c}$ to $\tilde{m}$

Bob (sk_B, pk_B)

$K := \mathrm{H}(pk_A, pk_B, \tilde{pk}, m_A, m_B, \tilde{m}, c_A, c_B, \tilde{c})$

Observation: Nontrivial strategy → E only obtains 2 out of $(m_i, m_j, \tilde{m})$

Exception: Aforementioned 'state reveal attack':

To succeed, E has to reveal Alice's session state before time-out

Open questions

Open questions

Active security requires 'worst-case' correctness

→ Can we soften this requirement, generically?

Passive-to-active transformations starting from KEMs?

→ Possible applications in AKE and when defining "hybrid" modes

KSSSS20: New quantum extraction technique → Tighter bounds

Can we apply MRM to our proof structure?

→ Tighter bounds for PKE and AKE → Efficiency

References

BCGNP08: Efficient One-round Key Exchange in the Standard Model, eprint: 2008/007

FSXY12: Strongly secure authenticated key exchange from factoring, codes, and lattices, eprint: 2012/211

HHK17: A Modular Analysis of the Fujisaki-Okamoto Transformation, eprint: 2017/604

AHU18: Quantum security proofs using semi-classical oracles, eprint: 2018/904

D'AVV18: On the impact of decryption failures on the security of LWE/LWR based schemes, eprint: 2018/1089

BHHHP19: Tighter proofs of CCA security in the quantum random oracle model, eprint: 2019/590

KSSSS20: Measure-Rewind-Measure: Tighter Quantum Random Oracle Model Proofs for One-Way to Hiding and CCA Security (Eurocrypt 2020)