# (Un)breakable curses - re-encryption in the Fujisaki-Okamoto transform

Kathrin Hövelmanns[1], Andreas Hülsing[1,2], Christian Majenz[3], **Fabrizio Sisinni**[3]

[1] Eindhoven University of Technology, [2] SandboxAQ, [3] Technical University of Denmark

Funded by
the European Union

**DTU Compute**
Department of Applied Mathematics and Computer Science

## Motivation

- Most NIST pqc proposals utilize the Fujisaki-Okamoto (FO) transformation to enhance their security.
- One of the steps in the FO transformation, called **re-encryption**, solves the problem of ciphertext malleability.
- At the same time, the re-encryption step is vulnerable to side-channel attacks.

R. Ueno, K. Xagawa, Y. Tanaka, A. Ito, J. Takahashi, and N. Homma. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs

# Our contribution

- We perform a comprehensive study the alternative used by NTRU and McEliece in place of re-encryption.

# Our contribution

- We perform a comprehensive study the alternative used by NTRU and McEliece in place of re-encryption.

- We formalize a computational notion, provided also by re-encryption, and show how to use it to obtain Chosen Ciphertext Attack (CCA) security.

## Our contribution

- We perform a comprehensive study the alternative used by NTRU and McEliece in place of re-encryption.
- We formalize a computational notion, provided also by re-encryption, and show how to use it to obtain Chosen Ciphertext Attack (CCA) security.
- We prove a novel QROM security result for KEMs with explicit rejection mechanism based on deterministic PKEs.

## Our contribution

- We perform a comprehensive study the alternative used by NTRU and McEliece in place of re-encryption.
- We formalize a computational notion, provided also by re-encryption, and show how to use it to obtain Chosen Ciphertext Attack (CCA) security.
- We prove a novel QROM security result for KEMs with explicit rejection mechanism based on deterministic PKEs.
- We show that all the alternatives to re-encryption have the same side-channel vulnerability in case of derandomized PKE schemes.

# Outline

- The Fujisaki-Okamoto transformation
  - The FO transform
  - Modular analysis of FO transform
- A Generalization of Re-Encryption
  - Computational Rigidity
  - Range-checking Oracles vs Range-checking Algorithms
- New modular analysis of the FO transform
  - From deterministic to rigid PKE
  - From PKE to KEM
  - From randomized to deterministic PKE

# The Fujisaki-Okamoto transformation

# The FO transform

To a PKE $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and two random oracles $G$ and $H$, we associate a KEM as

$$FO[\Pi, G, H] = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps}^{\perp}),$$

where Encaps and Decaps$^{\perp}$ are defined as follows

**Encaps**$(pk)$

$m \leftarrow_{\$} \mathcal{M}$
$c \leftarrow \mathsf{Enc}(pk, m; G(m))$
$K := H(m)$
**return** $(K, c)$

**Decaps**$^{\perp}(sk, c)$

$m' := \mathsf{Dec}(sk, c)$
$c' := \mathsf{Enc}(pk, m'; G(m'))$
**if** $m' = \perp$ **or** $c' \neq c$
  **return** $\perp$
**else**
  **return** $K := H(m')$

E. Fujisaki, T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Scheme
Alexander W. Dent. A Designer's Guide to KEMs

# The FO transform

To a PKE $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and two random oracles $G$ and $H$, we associate a KEM as

$$FO[\Pi, G, H] = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps}^{\perp}),$$

where Encaps and Decaps$^{\perp}$ are defined as follows

| $\underline{\mathsf{Encaps}(pk)}$ | $\underline{\mathsf{Decaps}^{\perp}(sk, c)}$ |
|---|---|
| $m \leftarrow_{\$} \mathcal{M}$ | $m' := \mathsf{Dec}(sk, c)$ |
| $c \leftarrow \mathsf{Enc}(pk, m; G(m))$ | $c' := \mathsf{Enc}(pk, m'; G(m'))$ |
| $K := H(m)$ | **if** $m' = \perp$ **or** $c' \neq c$ |
| **return** $(K, c)$ |     **return** $\perp$ |
| | **else** |
| |     **return** $K := H(m')$ |

E. Fujisaki, T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Scheme
Alexander W. Dent. A Designer's Guide to KEMs

# The FO transform

To a PKE $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$ and two random oracles $G$ and $H$, we associate a KEM as

$$FO[\Pi, G, H] = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps}^{\perp}),$$

where Encaps and $\mathsf{Decaps}^{\perp}$ are defined as follows

<u>$\mathsf{Encaps}(pk)$</u>

$m \leftarrow_{\$} \mathcal{M}$
$c \leftarrow \mathsf{Enc}(pk, m; G(m))$
$K := H(m)$
**return** $(K, c)$

<u>$\mathsf{Decaps}^{\perp}(sk, c)$</u>

$m' := \mathsf{Dec}(sk, c)$
$c' := \mathsf{Enc}(pk, m'; G(m'))$
**if** $m' = \perp$ **or** $c' \neq c$
  **return** $\perp$
**else**
  **return** $K := H(m')$

---

E. Fujisaki, T. Okamoto. Secure Integration of Asymmetric and Symmetric Encryption Scheme
Alexander W. Dent. A Designer's Guide to KEMs

## Modular analysis of FO transform

$$
\begin{tikzcd}
\Pi \arrow[r, "{T[\Pi, G]}"] \arrow[rr, bend right, "FO_m"'] & \Pi' \arrow[r, "{U_m[\Pi', H]}"] & \mathsf{KEM}_m
\end{tikzcd}
$$

---

D. Hofheinz, K. Hövelmanns, E. Kiltz, A Modular Analysis of the Fujisaki-Okamoto Transformation

# Modular analysis of FO transform

$$\begin{tikzcd} \Pi \arrow[rr, "{T[\Pi, G]}"] \arrow[rrrr, bend right, "FO_m"'] & & \Pi' \arrow[rr, "{U_m[\Pi', H]}"] & & \mathsf{KEM}_m \end{tikzcd}$$

**The $T$ transform.** Given a PKE scheme $\Pi$ and a random oracle $G$, it derandomizes encryption, performs the re-encryption step, and outputs a deterministic PKE.

---

D. Hofheinz, K. Hövelmanns, E. Kiltz, A Modular Analysis of the Fujisaki-Okamoto Transformation

# Modular analysis of FO transform

$$
\begin{tikzcd}
\Pi \arrow[r, "{T[\Pi, G]}"] \arrow[rr, bend right, "FO_m"'] & \Pi' \arrow[r, "{U_m[\Pi', H]}"] & \mathsf{KEM}_m
\end{tikzcd}
$$

**The $T$ transform.** Given a PKE scheme $\Pi$ and a random oracle $G$, it derandomizes encryption, performs the re-encryption step, and outputs a deterministic PKE.

**The $U$ transform.** Given a deterministic PKE scheme $\Pi'$ and a random oracle $H$, it outputs an IND-CCA KEM with explicit or implicit rejection mechanism.

D. Hofheinz, K. Hövelmanns, E. Kiltz, A Modular Analysis of the Fujisaki-Okamoto Transformation

# Rigidity

What guarantee does re-encryption provide?

D. J. Bernstein, E. Persichetti, Towards KEM Unification

# Rigidity

What guarantee does re-encryption provide?

**Rigidity**

Given a deterministic PKE $\Pi = (\mathsf{KG}, \mathsf{Enc}, \mathsf{Dec})$, we say that $\Pi$ is **rigid** if for every key pair $(pk, sk)$ and every ciphertext $c$ it holds

$$\mathsf{Dec}(sk, c) = \perp \quad \vee \quad \mathsf{Enc}(pk, \mathsf{Dec}(sk, c)) = c.$$

D. J. Bernstein, E. Persichetti, Towards KEM Unification

# A Generalization of Re-Encryption

## A Generalization of Re-Encryption
# Computational Rigidity

We say that a ciphertext $c$ is **non-rigid** if $\exists (pk, sk) \leftarrow \mathsf{KG}()$ such that

$$\mathsf{Enc}(pk, \mathsf{Dec}(sk, c)) \neq c.$$

# Computational Rigidity

We say that a ciphertext $c$ is **non-rigid** if $\exists (pk, sk) \leftarrow \mathsf{KG}()$ such that

$$\mathsf{Enc}(pk, \mathsf{Dec}(sk, c)) \neq c.$$

Given a PKE $\Pi$ and an adversary $A$, we define the **Find Non Rigid Ciphertext (FNRC) game** as follows

$\underline{\mathsf{FNRC}_{\Pi}(A)\text{:}}$

01 $(pk, sk) \leftarrow \mathsf{KG}()$

02 $\mathscr{L}_{\mathcal{C}} \leftarrow A^{\mathcal{O}}(pk)$

03 **return** $[\![\mathscr{L}_{\mathcal{C}}$ contains a non-rigid ciphertext $]\!]$

## A Generalization of Re-Encryption
# How to get a non-rigid ciphertext for deterministic PKE

Assume that $c$ is a non-rigid ciphertext.

$$c \xrightarrow{\mathsf{Dec}(sk, \cdot)} m' \xrightarrow{\mathsf{Enc}(pk, \cdot)} c'$$

We have two possibilities:

# How to get a non-rigid ciphertext for deterministic PKE

Assume that $c$ is a non-rigid ciphertext.

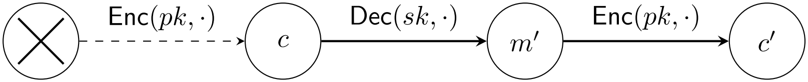
$$m \xrightarrow{\mathsf{Enc}(pk,\cdot)} c \xrightarrow{\mathsf{Dec}(sk,\cdot)} m' \xrightarrow{\mathsf{Enc}(pk,\cdot)} c'$$

We have two possibilities:

**1.** The ciphertext $c$ is the encryption of a message $m \implies m$ triggers a decryption failure.

# How to get a non-rigid ciphertext for deterministic PKE

Assume that $c$ is a non-rigid ciphertext.



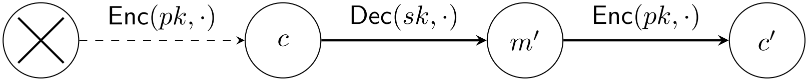

We have two possibilities:

**1.** The ciphertext $c$ is the encryption of a message $m \implies m$ triggers a decryption failure.

**2.** Ciphertext $c$ cannot be obtained through encryption.

# How to get a non-rigid ciphertext for deterministic PKE

Assume that $c$ is a non-rigid ciphertext.




We have two possibilities:

**1.** The ciphertext $c$ is the encryption of a message $m \implies m$ triggers a decryption failure.

**2.** Ciphertext $c$ cannot be obtained through encryption.

**Disclaimer.** Since the former case can be addressed using known techniques, we will focus on the latter.

## Range-checking Oracles/Algorithms

We need a way to check whether a ciphertext is the encryption of a message or not.

# Range-checking Oracles/Algorithms

We need a way to check whether a ciphertext is the encryption of a message or not.

Given a PKE scheme, we define its **Range-Checking Oracle** (RCO) as the oracle that takes as input a ciphertext and answers the question:

"Is this ciphertext the encryption of a message?

## A Generalization of Re-Encryption
# Range-checking Oracles/Algorithms

We need a way to check whether a ciphertext is the encryption of a message or not.

Given a PKE scheme, we define its **Range-Checking Oracle** (RCO) as the oracle that takes as input a ciphertext and answers the question:

"Is this ciphertext the encryption of a message?

We call an implementation of such an oracle **range-checking algorithm**.

## A Generalization of Re-Encryption
# Range-checking Oracles/Algorithms

We need a way to check whether a ciphertext is the encryption of a message or not.

Given a PKE scheme, we define its **Range-Checking Oracle** (RCO) as the oracle that takes as input a ciphertext and answers the question:

"Is this ciphertext the encryption of a message?

We call an implementation of such an oracle **range-checking algorithm**.

For example, re-encryption is a range-checking algorithm.

## Range-checking Oracles/Algorithms

We need a way to check whether a ciphertext is the encryption of a message or not.

Given a PKE scheme, we define its **Range-Checking Oracle** (RCO) as the oracle that takes as input a ciphertext and answers the question:

"Is this ciphertext the encryption of a message?

We call an implementation of such an oracle **range-checking algorithm**.

For example, re-encryption is a range-checking algorithm.

We formalize the intuition that an implementation might not be perfect introducing a computational notion.

# NTRU and McEliece range-checking algorithms

Examples of range-checking algorithms other than re-encryption, both using different predicates $P_x$

$\text{Range}_{\text{McEliece}}(sk, c)$**:**

$m' := \text{Dec}(sk, c)$
**if** $P_{priv}(c, m') = false$
  **return** $0$
**else return** $1$

$\text{Range}_{\text{NTRU}}(sk, c)$**:**

**if** $P_{pub}(c) = false$
  **return** $0$
$(m', r') := \text{Dec}(sk, c)$
**if** $P_{priv}(m', r') = false$
  **return** $0$
**else return** $1$

Daniel J. Bernstein, Understanding binary-Goppa decoding
NTRU. Algorithm Specifications And Supporting Documentation

# New modular analysis of the FO transform

## Overview of our results

$$\begin{tikzcd}
\text{PKE OW} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \begin{array}{c}\text{PKE}_{\mathsf{der}}\\ \text{OW, det.}\end{array} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \begin{array}{c}\text{PKE}_{\mathsf{rig}}\text{ OW,}\\ \text{comp.rigid}\end{array} \arrow[r, "\mathsf{U}_m^{\perp}"] & \begin{array}{c}\text{KEM}_m^{\perp}\\ \text{IND-CCA}\end{array}
\end{tikzcd}$$

The figure shows a slight simplification of our results for KEMs with explicit rejection.

# Rigidity step

$$\begin{tikzcd}
\mathsf{PKE} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \mathsf{PKE}_{\mathsf{der}} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \mathsf{PKE}_{\mathsf{rig}} \arrow[r, "\mathsf{U}_m^{\perp}"] & \mathsf{KEM}_m^{\perp}
\end{tikzcd}$$

We define $\mathsf{Rig}[\Pi, \mathsf{Range}] := (\mathsf{KG}_{\mathsf{rig}}, \mathsf{Enc}, \mathsf{Dec}_{\mathsf{rig}})$, where

| $\underline{\mathsf{KG}_{\mathsf{rig}}():}$ | $\underline{\mathsf{Dec}_{\mathsf{rig}}(sk', c):}$ |
|---|---|
| $(pk, sk) \leftarrow \mathsf{KG}()$ | $m' := \mathsf{Dec}(sk, c)$ |
| $sk' := (sk, pk)$ | **if** $m' = \perp \vee \mathsf{Range}(sk', c) = 0$ |
| **return** $(pk, sk')$ |     **return** $\perp$ |
| | **return** $m'$ |

# New modular analysis of the FO transform
## Rigidity step

$$\begin{tikzcd}
\mathsf{PKE} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \mathsf{PKE}_{\mathsf{der}} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \mathsf{PKE}_{\mathsf{rig}} \arrow[r, "\mathsf{U}_m^{\perp}"] & \mathsf{KEM}_m^{\perp}
\end{tikzcd}$$

**Properties of** Rig

Given a deterministic PKE $\Pi$ and a range-checking algorithm Range, we have

- if $\Pi$ is correct and Range is a good approximation $\implies$ $\mathsf{Rig}[\Pi, \mathsf{Range}]$ is correct and computationally rigid.
- if $\Pi$ is OW secure $\implies$ $\mathsf{Rig}[\Pi, \mathsf{Range}]$ is OW secure.

# From PKE to KEM

$$\begin{tikzcd}
\mathsf{PKE} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \mathsf{PKE}_{\mathsf{der}} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \mathsf{PKE}_{\mathsf{rig}} \arrow[r, "\mathsf{U}_m^{\perp}"] & \mathsf{KEM}_m^{\perp}
\end{tikzcd}$$

We define $\mathsf{U}_m^{\perp}[\Pi, H] = (\mathsf{KG}, \mathsf{Encaps}, \mathsf{Decaps}_m^{\perp})$, where

$\underline{\mathsf{Encaps}(pk)}$

$m \leftarrow_\$ \mathcal{M}$
$c \leftarrow \mathsf{Enc}(pk, m)$
$K := H(m)$
**return** $(K, c)$

$\underline{\mathsf{Decaps}_m^{\perp}(sk, c)}$

$m' := \mathsf{Dec}(sk, c)$
**if** $m' = \perp$
  **return** $\perp$
**return** $K := H(m')$

## New modular analysis of the FO transform

# From PKE to KEM

$$\begin{tikzcd}
\mathsf{PKE} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \mathsf{PKE}_{\mathsf{der}} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \mathsf{PKE}_{\mathsf{rig}} \arrow[r, "\mathsf{U}_m^{\perp}"] & \mathsf{KEM}_m^{\perp}
\end{tikzcd}$$

**Properties of $\mathsf{U}_m^{\perp}$**

Given a deterministic, computationally rigid PKE $\Pi$ and a random oracle $H$, we have

- if $\Pi$ is OW secure $\xRightarrow{\text{ROM}}$ $\mathsf{U}_m^{\perp}[\Pi, H]$ is IND-CCA secure.
- if $\Pi$ is OW-VCA secure $\xRightarrow{\text{QROM}}$ $\mathsf{U}_m^{\perp}[\Pi, H]$ is IND-CCA secure.

# New modular analysis of the FO transform
## Derandomization step

$$\begin{tikzcd}
\mathsf{PKE} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \mathsf{PKE}_{\mathsf{der}} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \mathsf{PKE}_{\mathsf{rig}} \arrow[r, "\mathsf{U}_m^{\perp}"] & \mathsf{KEM}_m^{\perp}
\end{tikzcd}$$

We define $\mathsf{Der}[\Pi, G] := (\mathsf{KG}, \mathsf{Enc}_{\mathsf{der}}, \mathsf{Dec})$, where

$$\mathsf{Enc}_{\mathsf{der}}(pk, m) := \mathsf{Enc}(pk, m; G(m)).$$

## New modular analysis of the FO transform
# The curse is unavoidable

$$\begin{tikzcd} \mathsf{PKE} \arrow[r, "\mathsf{Der}"] \arrow[rrr, bend right, "\mathsf{FO}_m^{\perp,\mathsf{rg}}"'] & \mathsf{PKE}_{\mathsf{der}} \arrow[r, "\mathsf{Rig}"] \arrow[rr, bend right, "\mathsf{rgU}_m^{\perp}"'] & \mathsf{PKE}_{\mathsf{rig}} \arrow[r, "\mathsf{U}_m^{\perp}"] & \mathsf{KEM}_m^{\perp} \end{tikzcd}$$

**The curse is unavoidable**

If the Der transformation is applied, to define a good range-checking algorithm, we must query the random oracle used during the derandomization step.
In this case, the attack described by Ueno et al. is still a threat.

---

R. Ueno, K. Xagawa, Y. Tanaka, A. Ito, J. Takahashi, and N. Homma. Curse of re-encryption: A generic power/EM analysis on post-quantum KEMs

## New modular analysis of the FO transform
# Summary

1. We formalize the notion of computational rigidity.
2. We analyze alternatives to re-encryption to achieve rigidity.
3. We introduce the notion of range-checking oracle/algorithm as a generalization of the re-encryption step.
4. We prove how these new notions can be used to enforce CCA security both in the ROM and in the QROM.
5. We prove that, for derandomize PKE schemes using a random oracle, all alternatives to re-encryption suffer from the same side-channel weakness.