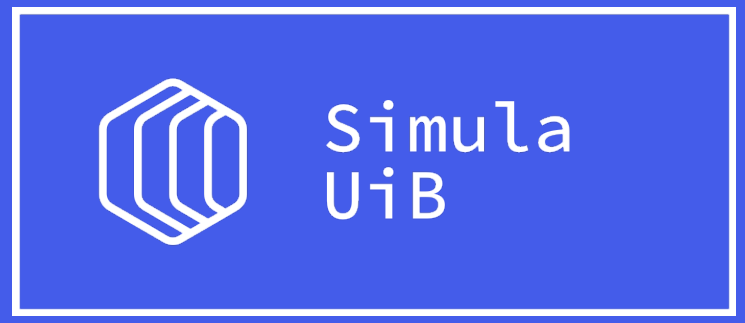


My Research Interests

Hans Heum

Two main interests:

Two main interests:

1. Concrete security in systems with many users.

Two main interests:

1. Concrete security in systems with many users.
2. Concrete security of Post-Quantum Cryptography.

Provable Security

Provable Security

- CCA-security = Security against **Chosen Ciphertext Attacks**, the strongest form of provable security.

Provable Security

- CCA-security = Security against **Chosen Ciphertext Attacks**, the strongest form of provable security.
- Play **games**, where an adversary **A** needs to guess a bit.

Provable Security

- CCA-security = Security against **Chosen Ciphertext Attacks**, the strongest form of provable security.
- Play **games**, where an adversary **A** needs to guess a bit.
- **A** is a probabilistic Turing machine that can run any algorithm it likes…

Provable Security

- CCA-security = Security against **Chosen Ciphertext Attacks**, the strongest form of provable security.
- Play **games**, where an adversary **A** needs to guess a bit.
- **A** is a probabilistic Turing machine that can run any algorithm it likes…
- and its **advantage** is how much **better** it can do than just guessing randomly.

Provable Security

- CCA-security = Security against **Chosen Ciphertext Attacks**, the strongest form of provable security.
- Play **games**, where an adversary **A** needs to guess a bit.
- **A** is a probabilistic Turing machine that can run any algorithm it likes…
- and its **advantage** is how much **better** it can do than just guessing randomly.

$\text{Game Guess}_{\mathcal{SE}}$

procedure Initialize

$K \xleftarrow{\$} \mathcal{K}$; $b \xleftarrow{\$} \{0,1\}$

procedure LR(M_0, M_1)

return $C \xleftarrow{\$} \mathcal{E}_K(M_b)$

procedure Finalize(b')

return $(b == b')$

Provable Security

- CCA-security = Security against **Chosen Ciphertext Attacks**, the strongest form of provable security.
- Play **games**, where an adversary **A** needs to guess a bit.
- **A** is a probabilistic Turing machine that can run any algorithm it likes…
- and its **advantage** is how much **better** it can do than just guessing randomly.

Note: for all cryptosystems other than the One-Time Pad, there are adversaries with advantage larger than zero.

$\text{Game Guess}_{\mathcal{SE}}$

procedure Initialize

$K \xleftarrow{\$} \mathcal{K}$; $b \xleftarrow{\$} \{0,1\}$

procedure LR(M_0, M_1)

return $C \xleftarrow{\$} \mathcal{E}_K(M_b)$

procedure Finalize(b')

return $(b == b')$

What do I mean by «Concrete Security»?

What do I mean by «Concrete Security»?

- Old paradigm: **Asymptotic security**

What do I mean by «Concrete Security»?

- Old paradigm: **Asymptotic security**

$f(\lambda) = \mathsf{negl}(\lambda)$ if, for every polynomial $\mathsf{poly}(\lambda)$, there exists $n > N$ such that $f(\lambda) < 1/\mathsf{poly}(\lambda)$.

A system is said to be *secure* if, for any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

What do I mean by «Concrete Security»?

- Old paradigm: **Asymptotic security**

  $f(\lambda) = \mathsf{negl}(\lambda)$ if, for every polynomial $\mathsf{poly}(\lambda)$, there exists $n > N$ such that $f(\lambda) < 1/\mathsf{poly}(\lambda)$.

  A system is said to be *secure* if, for any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

- Modern paradigm: **Concrete security**

What do I mean by «Concrete Security»?

- Old paradigm: **Asymptotic security**

  $f(\lambda) = \mathsf{negl}(\lambda)$ if, for every polynomial $\mathsf{poly}(\lambda)$, there exists $n > N$ such that $f(\lambda) < 1/\mathsf{poly}(\lambda)$.

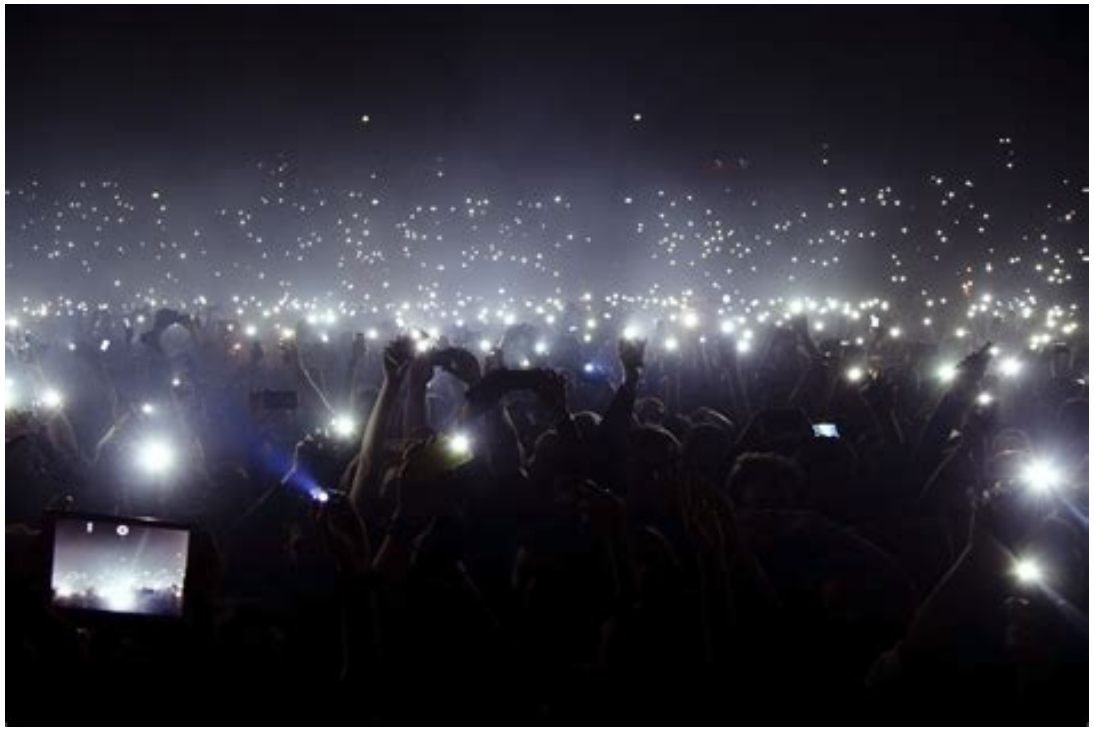
  A system is said to be *secure* if, for any adversary $\mathcal{A}$, $\mathsf{Adv}_{\mathcal{A}}(\lambda) = \mathsf{negl}(\lambda)$.

- Modern paradigm: **Concrete security**
  - We now care about the exact advantage, so that we can calculate exact security levels.

Multi-User Security

(Research interest 1)

Multi-Key Security Degradation



Multi-Key Security Degradation

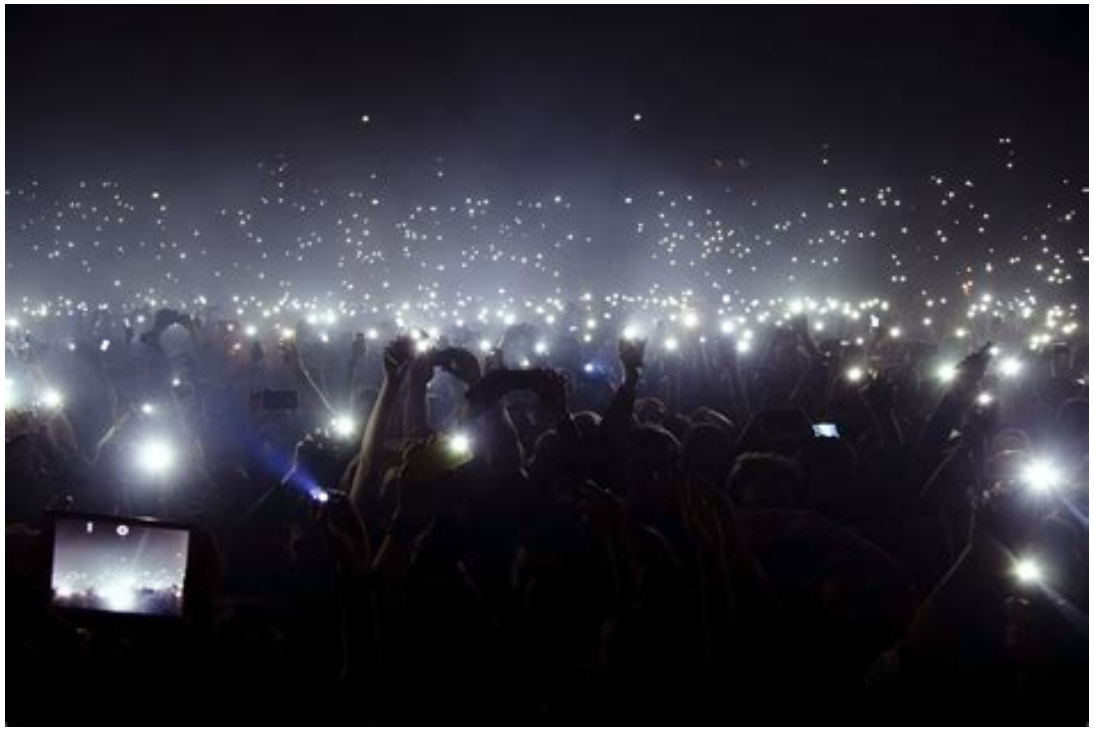
- We now let the adversary attack any one of n systems.

Multi-Key Security Degradation

- We now let the adversary attack any one of n systems.
- Question 1: What is the probability that **A** will crack **at least one** of the systems?

Multi-Key Security Degradation

- We now let the adversary attack any one of n systems.
- Question 1: What is the probability that **A** will crack **at least one** of the systems?
- Question 2: If she cracks one of them, how does that affect the security of the rest?



Multi-Key Security Degradation

Proven fact: Reductions of proofs of security from a system with n keys to a system with 1 key are lossy in general, losing a factor of n.

Multi-Key Security Degradation

Proven fact: Reductions of proofs of security from a system with n keys to a system with 1 key are lossy in general, losing a factor of n.

$$\mathsf{Adv}^{\mathrm{n}}_{\mathcal{A}}(\lambda) \geq n \cdot \mathsf{Adv}^{1}_{\mathcal{A}}(\lambda)$$

Multi-Key Security Degradation

- Example: 1 billion users, 64 bit keys

Multi-Key Security Degradation

- Example: 1 billion users, 64 bit keys

Advantage against one when simply trying all keys: $\mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) = 2^{-\lambda} = 2^{-64}$

Multi-Key Security Degradation

- Example: 1 billion users, 64 bit keys

Advantage against one when simply trying all keys: $\mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) = 2^{-\lambda} = 2^{-64}$

1 billion users: $n \approx 2^{30}$

Multi-Key Security Degradation

- Example: 1 billion users, 64 bit keys

Advantage against one when simply trying all keys: $\mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) = 2^{-\lambda} = 2^{-64}$

1 billion users: $n \approx 2^{30}$

$$\begin{aligned} \Rightarrow \mathsf{Adv}^{\mathrm{n}}_{\mathcal{A}}(\lambda) &= n \cdot \mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) \\ &= 2^{30} \cdot 2^{-64} \\ &= 2^{-34} \end{aligned}$$

Multi-Key Security Degradation

- Example: 1 billion users, 64 bit keys

Advantage against one when simply trying all keys: $\mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) = 2^{-\lambda} = 2^{-64}$

1 billion users: $n \approx 2^{30}$

$$\begin{aligned} \Rightarrow \mathsf{Adv}^{\mathrm{n}}_{\mathcal{A}}(\lambda) &= n \cdot \mathsf{Adv}^{1}_{\mathcal{A}}(\lambda) \\ &= 2^{30} \cdot 2^{-64} \\ &= 2^{-34} \end{aligned}$$

- Really only have 34 bits of security!

What I’m currently looking at

What I’m currently looking at

- A **hybrid** system: KEM + DEM = PKE

What I’m currently looking at

- A **hybrid** system: KEM + DEM = PKE
- Each of these have a linear loss of security in the multi-user setting

What I’m currently looking at

- A **hybrid** system: KEM + DEM = PKE
- Each of these have a linear loss of security in the multi-user setting
- … but what about their **composition?**

What I’m currently looking at

- A **hybrid** system: KEM + DEM = PKE
- Each of these have a linear loss of security in the multi-user setting
- … but what about their **composition?**

Theorem 1. *For every PKE adversary $\mathcal{A}$, there is a KEM adversary $\mathcal{B}$ and a DEM adversary $\mathcal{C}$, such that*

$$\mathrm{Adv}^{\text{LRIND-CCA},n}_{\text{PKE}}(\mathcal{A}) \leq 2 \cdot \mathrm{Adv}^{\text{IND-CCA},n}_{\text{KEM}}(\mathcal{B}) + \mathrm{Adv}^{\text{OT-LRIND-CCA},n}_{\text{DEM}}(\mathcal{C}). \quad (5)$$

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Game $\mathrm{G}_1^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(pk_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$K' \leftarrow\!\!\$\ \mathcal{K}$	**return** $\perp$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K'$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{D.enc}(K', m_{b_j})$	$K \leftarrow \mathbf{K}_i[c_1]$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	**else**	
return $b_j = b'_j$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{B}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$d_1, \ldots, d_n \leftarrow\!\!\$\ \{0,1\}$	$(K, c_1) \leftarrow \mathrm{Oenc}(i, j)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	$\mathsf{sk}_i \leftarrow \mathrm{Cor}(i)$
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K$	$K \leftarrow \mathbf{K}_i[c_1]$	**return** sk_i
$(j, d'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{d_j})$	**else**	
$b'_j \leftarrow\!\!! (d_j = d'_j)$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{Odec}(i, c_1)$	
return (j, b'_j)		**if** $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Game $\mathrm{G}_1^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(pk_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$K' \leftarrow\!\!\$\ \mathcal{K}$	**return** $\perp$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K'$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{D.enc}(K', m_{b_j})$	$K \leftarrow \mathbf{K}_i[c_1]$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	**else**	
return $b_j = b'_j$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{B}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$d_1, \ldots, d_n \leftarrow\!\!\$\ \{0,1\}$	$(K, c_1) \leftarrow \mathrm{Oenc}(i, j)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	$\mathsf{sk}_i \leftarrow \mathrm{Cor}(i)$
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K$	$K \leftarrow \mathbf{K}_i[c_1]$	**return** sk_i
$(j, d'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{d_j})$	**else**	
$b'_j \leftarrow\!\!! (d_j = d'_j)$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{Odec}(i, c_1)$	
return (j, b'_j)		**if** $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Game $\mathrm{G}_1^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(pk_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$K' \leftarrow\!\!\$\ \mathcal{K}$	**return** $\perp$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K'$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{D.enc}(K', m_{b_j})$	$K \leftarrow \mathbf{K}_i[c_1]$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	**else**	
return $b_j = b'_j$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{C}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{Oenc}(j, m_0, m_1)$	$i' \leftarrow \mathbf{K}_i[c_1]$	
$i' = 0$	$\mathbf{K}_i[c_1] \leftarrow i'$	$m \leftarrow \mathrm{Odec}(i', c_2)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$i' \leftarrow i' + 1$	**else**	
return (j, b'_j)	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{B}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$d_1, \ldots, d_n \leftarrow\!\!\$\ \{0,1\}$	$(K, c_1) \leftarrow \mathrm{Oenc}(i, j)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	$\mathsf{sk}_i \leftarrow \mathrm{Cor}(i)$
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K$	$K \leftarrow \mathbf{K}_i[c_1]$	**return** sk_i
$(j, d'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{d_j})$	**else**	
$b'_j \leftarrow\!\!! (d_j = d'_j)$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{Odec}(i, c_1)$	
return (j, b'_j)		**if** $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

$$
\begin{aligned}
\mathrm{Adv}^{\text{LRIND-CCA},n}_{\mathrm{PKE}}(\mathcal{A}) &\coloneqq 2 \cdot \Pr[\text{LRIND-CCA}^n_{\mathrm{PKE}}(\mathcal{A}) = 1] - 1 \\
&= 2 \cdot \Pr[\mathrm{G}_0^n(\mathcal{A}) = 1] - 1 \\
&= 2\left(\Pr[\mathrm{G}_0^n(\mathcal{A}) = 1] - \Pr[\mathrm{G}_1^n(\mathcal{A}) = 1] + \Pr[\mathrm{G}_1^n(\mathcal{A}) = 1]\right) - 1 \\
&= 2\left(\Pr[\mathrm{G}_0^n(\mathcal{A}) = 1] - \Pr[\mathrm{G}_1^n(\mathcal{A}) = 1]\right) + 2 \cdot \Pr[\mathrm{G}_1^n(\mathcal{A}) = 1] - 1 \\
&\leq 2 \cdot \mathrm{Adv}^{\text{IND-CCA},n}_{\mathrm{KEM}}(\mathcal{B}) + \mathrm{Adv}^{\text{OT-LRIND-CCA},n}_{\mathrm{DEM}}(\mathcal{C}).
\end{aligned}
$$

□

Game $\mathrm{G}_1^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(pk_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$K' \leftarrow\!\!\$\ \mathcal{K}$	**return** $\perp$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K'$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{D.enc}(K', m_{b_j})$	$K \leftarrow \mathbf{K}_i[c_1]$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	**else**	
return $b_j = b'_j$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{C}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{Oenc}(j, m_0, m_1)$	$i' \leftarrow \mathbf{K}_i[c_1]$	
$i' = 0$	$\mathbf{K}_i[c_1] \leftarrow i'$	$m \leftarrow \mathrm{Odec}(i', c_2)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$i' \leftarrow i' + 1$	**else**	
return (j, b'_j)	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{B}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$d_1, \ldots, d_n \leftarrow\!\!\$\ \{0,1\}$	$(K, c_1) \leftarrow \mathrm{Oenc}(i, j)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	$\mathsf{sk}_i \leftarrow \mathrm{Cor}(i)$
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K$	$K \leftarrow \mathbf{K}_i[c_1]$	**return** sk_i
$(j, d'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{d_j})$	**else**	
$b'_j \leftarrow\!\!! (d_j = d'_j)$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{Odec}(i, c_1)$	
return (j, b'_j)		**if** $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

$$
\begin{aligned}
\mathrm{Adv}^{\text{LRIND-CCA},n}_{\mathrm{PKE}}(\mathcal{A}) &\coloneqq 2 \cdot \Pr[\text{LRIND-CCA}^n_{\mathrm{PKE}}(\mathcal{A}) = 1] - 1 \\
&= 2 \cdot \Pr[\mathrm{G}^n_0(\mathcal{A}) = 1] - 1 \\
&= 2 \left(\Pr[\mathrm{G}^n_0(\mathcal{A}) = 1] - \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1] + \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1]\right) - 1 \\
&= 2 \left(\Pr[\mathrm{G}^n_0(\mathcal{A}) = 1] - \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1]\right) + 2 \cdot \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1] - 1 \\
&\leq 2 \cdot \mathrm{Adv}^{\text{IND-CCA},n}_{\mathrm{KEM}}(\mathcal{B}) + \mathrm{Adv}^{\text{OT-LRIND-CCA},n}_{\mathrm{DEM}}(\mathcal{C}).
\end{aligned}
$$

□

Game $\mathrm{G}_1^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(pk_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$K' \leftarrow\!\!\$\ \mathcal{K}$	**return** $\perp$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K'$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{D.enc}(K', m_{b_j})$	$K \leftarrow \mathbf{K}_i[c_1]$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	**else**	
return $b_j = b'_j$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{C}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{Oenc}(j, m_0, m_1)$	$i' \leftarrow \mathbf{K}_i[c_1]$	
$i' = 0$	$\mathbf{K}_i[c_1] \leftarrow i'$	$m \leftarrow \mathrm{Odec}(i', c_2)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$i' \leftarrow i' + 1$	**else**	
return (j, b'_j)	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

… It gets complicated.

How (parts of) the proof looks

Game $\mathrm{G}_0^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{b_j})$	**return** $\perp$	**return** sk_i
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	**return** $\langle c_1, c_2 \rangle$	**if** $K = \perp$	
return $b_j = b'_j$		**return** $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{B}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$d_1, \ldots, d_n \leftarrow\!\!\$\ \{0,1\}$	$(K, c_1) \leftarrow \mathrm{Oenc}(i, j)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	$\mathsf{sk}_i \leftarrow \mathrm{Cor}(i)$
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K$	$K \leftarrow \mathbf{K}_i[c_1]$	**return** sk_i
$(j, d'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$c_2 \leftarrow \mathrm{D.enc}(K, m_{d_j})$	**else**	
$b'_j \leftarrow\!\!!(d_j = d'_j)$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{Odec}(i, c_1)$	
return (j, b'_j)		**if** $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

$$
\begin{aligned}
\mathrm{Adv}^{\text{LRIND-CCA},n}_{\mathrm{PKE}}(\mathcal{A}) &\coloneqq 2 \cdot \Pr[\text{LRIND-CCA}^n_{\mathrm{PKE}}(\mathcal{A}) = 1] - 1 \\
&= 2 \cdot \Pr[\mathrm{G}^n_0(\mathcal{A}) = 1] - 1 \\
&= 2\left(\Pr[\mathrm{G}^n_0(\mathcal{A}) = 1] - \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1] + \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1]\right) - 1 \\
&= 2\left(\Pr[\mathrm{G}^n_0(\mathcal{A}) = 1] - \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1]\right) + 2 \cdot \Pr[\mathrm{G}^n_1(\mathcal{A}) = 1] - 1 \\
&\leq 2 \cdot \mathrm{Adv}^{\text{IND-CCA},n}_{\mathrm{KEM}}(\mathcal{B}) + \mathrm{Adv}^{\text{OT-LRIND-CCA},n}_{\mathrm{DEM}}(\mathcal{C}).
\end{aligned}
$$

□

Game $\mathrm{G}_1^n(\mathcal{A})$	Oracle $\mathrm{Oenc}(i, j, m_0, m_1)$	Oracle $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	Oracle $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(pk_i)$	**if** $\langle c_1, c_2 \rangle \in \mathbf{C}_i$	append i to **cor**
$b_1, \ldots, b_n \leftarrow\!\!\$\ \{0,1\}$	$K' \leftarrow\!\!\$\ \mathcal{K}$	**return** $\perp$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$\mathbf{K}_i[c_1] \leftarrow K'$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	
$\mathbf{C}_1, \ldots, \mathbf{C}_n, \mathbf{J}_1, \ldots, \mathbf{J}_n, \mathbf{cor} \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{D.enc}(K', m_{b_j})$	$K \leftarrow \mathbf{K}_i[c_1]$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	append $\langle c_1, c_2 \rangle$ to $\mathbf{C}_i$, j to $\mathbf{J}_i$	**else**	
return $b_j = b'_j$	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

Adversary $\mathcal{C}$	if $\mathcal{A}$ calls $\mathrm{Oenc}(i, j, m_0, m_1)$	if $\mathcal{A}$ calls $\mathrm{Odec}(i, \langle c_1, c_2 \rangle)$	if $\mathcal{A}$ calls $\mathrm{Cor}(i)$
$(\mathsf{pk}_1, \mathsf{sk}_1), \ldots, (\mathsf{pk}_n, \mathsf{sk}_n) \leftarrow\!\!\$\ \mathrm{K.gen}$	$(K, c_1) \leftarrow\!\!\$\ \mathrm{K.enc}(\mathsf{pk}_i)$	**if** $\mathbf{K}_i[c_1] \neq \emptyset$	**return** sk_i
$\mathbf{K}_1, \ldots, \mathbf{K}_n \leftarrow \emptyset$	$c_2 \leftarrow \mathrm{Oenc}(j, m_0, m_1)$	$i' \leftarrow \mathbf{K}_i[c_1]$	
$i' = 0$	$\mathbf{K}_i[c_1] \leftarrow i'$	$m \leftarrow \mathrm{Odec}(i', c_2)$	
$(j, b'_j) \leftarrow\!\!\$\ \mathcal{A}(\mathsf{pk}_1, \ldots, \mathsf{pk}_n)$	$i' \leftarrow i' + 1$	**else**	
return (j, b'_j)	**return** $\langle c_1, c_2 \rangle$	$K \leftarrow \mathrm{K.dec}(\mathsf{sk}_i, c_1)$	
		if $K = \perp$	
		return $\perp$	
		$m \leftarrow \mathrm{D.dec}(K, c_2)$	
		return m	

… It gets complicated.
(Also it's wrong.)

Current status

Current status

- It now looks like introducing key corruptions leads to an additional factor of n security loss…

Current status

- It now looks like introducing key corruptions leads to an additional factor of n security loss…
- though we still hope to find a way to circumvent this.

Current status

- It now looks like introducing key corruptions leads to an additional factor of n security loss…
- though we still hope to find a way to circumvent this.
- Provable security is important!

Current status

- It now looks like introducing key corruptions leads to an additional factor of n security loss…
- though we still hope to find a way to circumvent this.
- Provable security is important!
- and the devil’s in the details.

Quantum Cryptanalysis

(Research interest 2)

«In order to test the security of Post-Quantum Cryptography, you need people that continuously try to find quantum algorithms that break it. However, you would need someone with both a good grasp of Quantum Computing, ideally with a background in physics, as well as an intimate knowledge of what's happening in the field of cryptography, and right now these fields are mostly separate. […] I really wish there were more people out there working on this.»

– Marco Tomamichel, Quantum information theorist,
in conversation at QIP 2020.

«In order to test the security of Post-Quantum Cryptography, you need people that continuously try to find quantum algorithms that break it. However, you would need someone with both a good grasp of Quantum Computing, ideally with a background in physics, as well as an intimate knowledge of what's happening in the field of cryptography, and right now these fields are mostly separate. […] I really wish there were more people out there working on this.»

– Marco Tomamichel, Quantum information theorist,
in conversation at QIP 2020.

… that's me :)

Thank you.