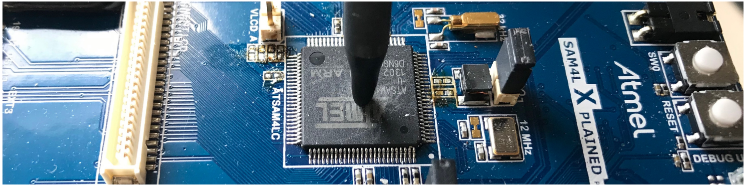


Systematic Study of Decryption and Re-Encryption Leakage: the Case of Kyber

Melissa Azouaoui, Olivier Bronchain, Clément Hoffmann,
Yulia Kuzovkova, Tobias Schneider, François-Xavier Standaert

12 April 2022

UCLouvain NXP

Content

Why Post-Quantum Cryptography (PQC) & SCA ?

---

[1] https://www.ssi.gouv.fr/publication/anssi-views-on-the-post-quantum-cryptography-transition/

Why Post-Quantum Cryptography (PQC) & SCA ?

- Will be soon standardized:
  - NIST Standardization effort.
  - ANSSI targets around 2030 for PQ standalone solutions.[1]

---

[1] https://www.ssi.gouv.fr/publication/anssi-views-on-the-post-quantum-cryptography-transition/

Why Post-Quantum Cryptography (PQC) & SCA ?

- Will be soon standardized:
  - NIST Standardization effort.
  - ANSSI targets around 2030 for PQ standalone solutions.[1]
- SCA is a threat to most embedded systems with cryptography:
  - Symmetric cryptography: block-ciphers.
  - Asymmetric cryptography: RSA & ECC.

---

[1] https://www.ssi.gouv.fr/publication/anssi-views-on-the-post-quantum-cryptography-transition/

Why Post-Quantum Cryptography (PQC) & SCA ?

- Will be soon standardized:
  - NIST Standardization effort.
  - ANSSI targets around 2030 for PQ standalone solutions.[1]
- SCA is a threat to most embedded systems with cryptography:
  - Symmetric cryptography: block-ciphers.
  - Asymmetric cryptography: RSA & ECC.
- Powerful side-channel attacks against PQ KEM's:
  - Many single-trace attacks.

---

[1] https://www.ssi.gouv.fr/publication/anssi-views-on-the-post-quantum-cryptography-transition/

Why Post-Quantum Cryptography (PQC) & SCA ?

- Will be soon standardized:
  - NIST Standardization effort.
  - ANSSI targets around 2030 for PQ standalone solutions.[1]
- SCA is a threat to most embedded systems with cryptography:
  - Symmetric cryptography: block-ciphers.
  - Asymmetric cryptography: RSA & ECC.
- Powerful side-channel attacks against PQ KEM's:
  - Many single-trace attacks.
- PQC is expensive on Cortex-M4:
  - ≈ 800 `kCycles` for unprotected Saber.
  - $\approx 13,000$ `kCycles` for 4-share Saber.

---

[1] https://www.ssi.gouv.fr/publication/anssi-views-on-the-post-quantum-cryptography-transition/

What is a Key Encapsulation Mechanism (KEM) ?

Goal:

How:

Security property:

Alice Bob

What is a Key Encapsulation Mechanism (KEM) ?

Goal:

- Alice transfers a symmetric key (m) to Bob.

How:

Security property:

Alice Bob

$m \leftarrow \{0,1\}^n$

What is a Key Encapsulation Mechanism (KEM) ?

Goal:

- Alice transfers a symmetric key (m) to Bob.

How:

1. Alice "*Encapsulates*" the secret m with pk.

Security property:

Alice Bob

$m \leftarrow \{0,1\}^n$

$c \leftarrow Encap_{pk}(m)$

What is a Key Encapsulation Mechanism (KEM) ?

Goal:

- Alice transfers a symmetric key (m) to Bob.

How:

1. Alice "*Encapsulates*" the secret m with pk.
2. Alice sends $c = Encap_{pk}(m)$.

Security property:

Alice Bob

$m \leftarrow \{0,1\}^n$

$c \leftarrow Encap_{pk}(m)$

$$\xrightarrow{c}$$

What is a Key Encapsulation Mechanism (KEM) ?

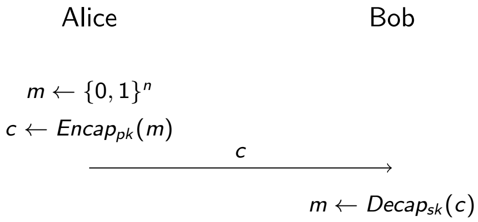
Goal:

- Alice transfers a symmetric key (m) to Bob.

How:

1. Alice "*Encapsulates*" the secret m with pk.
2. Alice sends $c = Encap_{pk}(m)$.
3. Bob *Decapsulates* $m = Decap_{sk}(c)$.

Security property:



Alice

Bob

$m \leftarrow \{0,1\}^n$

$c \leftarrow Encap_{pk}(m)$

$\xrightarrow{c}$

$m \leftarrow Decap_{sk}(c)$

What is a Key Encapsulation Mechanism (KEM) ?

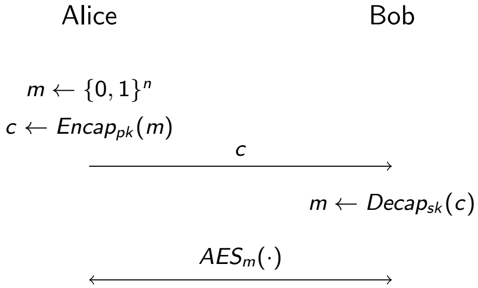
Goal:

- Alice transfers a symmetric key (m) to Bob.

How:

1. Alice *"Encapsulates"* the secret m with pk.
2. Alice sends $c = Encap_{pk}(m)$.
3. Bob *Decapsulates* $m = Decap_{sk}(c)$.
4. Alice and Bob are sharing a secret m.

Security property:



Alice

Bob

$m \leftarrow \{0,1\}^n$

$c \leftarrow Encap_{pk}(m)$

$\xrightarrow{c}$

$m \leftarrow Decap_{sk}(c)$

$\xleftrightarrow{AES_m(\cdot)}$

What is a Key Encapsulation Mechanism (KEM) ?

Goal:

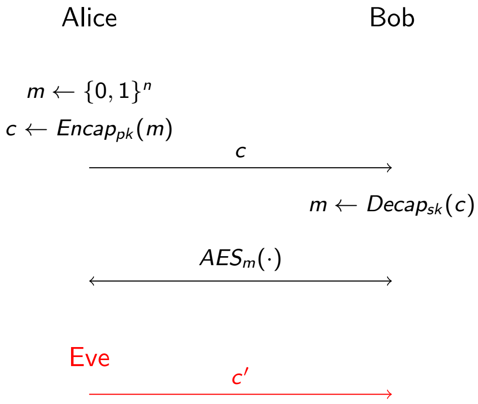
- Alice transfers a symmetric key (m) to Bob.

How:

1. Alice *"Encapsulates"* the secret m with pk.
2. Alice sends $c = Encap_{pk}(m)$.
3. Bob *Decapsulates* $m = Decap_{sk}(c)$.
4. Alice and Bob are sharing a secret m.

Security property:

- CCA-secure: Sending invalid c' does not reveal information on sk.



Alice

Bob

$m \leftarrow \{0,1\}^n$

$c \leftarrow Encap_{pk}(m)$

$\xrightarrow{c}$

$m \leftarrow Decap_{sk}(c)$

$\xleftrightarrow{AES_m(\cdot)}$

Eve

$\xrightarrow{c'}$

What is a Key Encapsulation Mechanism (KEM) ?

Goal:

- Alice transfers a symmetric key (m) to Bob.

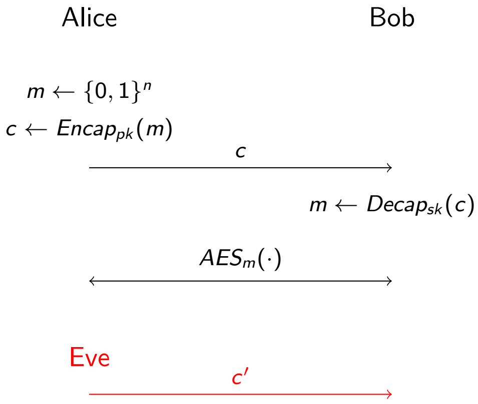
How:

1. Alice ”*Encapsulates*” the secret m with pk.
2. Alice sends $c = Encap_{pk}(m)$.
3. Bob *Decapsulates* $m = Decap_{sk}(c)$.
4. Alice and Bob are sharing a secret m.

Security property:

- CCA-secure: Sending invalid c' does not reveal information on sk.

$\rightarrow$ We focus on the Decapsulation.



Alice

Bob

$m \leftarrow \{0,1\}^n$

$c \leftarrow Encap_{pk}(m)$

$\xrightarrow{c}$

$m \leftarrow Decap_{sk}(c)$

$\xleftrightarrow{AES_m(\cdot)}$

Eve

$\xrightarrow{c'}$

Example of (simplified) CPA lattice-based PKE.

Why a toy example of CPA-secure public key scheme?:

$m' = \quad (\qquad\qquad\qquad\qquad)$

Our simplified CPAPKE.Dec$_{sk}(c)$:

Example of (simplified) CPA lattice-based PKE.

Why a toy example of CPA-secure public key scheme?:

- Building block for CCA-secure KEMs.
- Will be used to illustrate various attacks.

$m' = \quad (\qquad\qquad\qquad\qquad)$

Our simplified CPAPKE.Dec$_{sk}(c)$:

Example of (simplified) CPA lattice-based PKE.

Why a toy example of CPA-secure public key scheme?:

- Building block for CCA-secure KEMs.
- Will be used to illustrate various attacks.

$$m' = \left(\qquad \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}} \right)$$

Our simplified CPAPKE.Dec$_{sk}(c)$:

- Secret key $\boldsymbol{sk}$ is a vector.

Example of (simplified) CPA lattice-based PKE.

Why a toy example of CPA-secure public key scheme?:

- Building block for CCA-secure KEMs.
- Will be used to illustrate various attacks.

$$m' = \left(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}} \right)$$

Our simplified CPAPKE.Dec$_{sk}(c)$:

- Secret key $\boldsymbol{sk}$ is a vector.
- Ciphertext $\boldsymbol{c}$ is a vector.

Example of (simplified) CPA lattice-based PKE.

Why a toy example of CPA-secure public key scheme?:

- Building block for CCA-secure KEMs.
- Will be used to illustrate various attacks.

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= \{0, 1\}$$

Our simplified CPAPKE.Dec$_{sk}(c)$:

- Secret key $\boldsymbol{sk}$ is a vector.
- Ciphertext $\boldsymbol{c}$ is a vector.
- The exchanged secret m' is a bit.

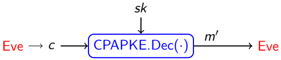
Build a CCA KEM from CPA PKE

CCAKEM.Dec

Build a CCA KEM from CPA PKE

With a CPA PKE:

- In CCA context, Eve generates invalid c and observes m'.

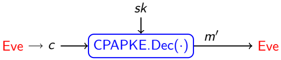


CCAKEM.Dec

Build a CCA KEM from CPA PKE

With a CPA PKE:

- In CCA context, Eve generates invalid c and observes m'.
- → Insecure since only CPA-secure.



CCAKEM.Dec

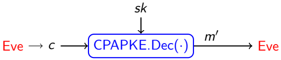
Build a CCA KEM from CPA PKE: FO-transform

Fujisaki-Okamoto (FO) transform:

- Leverage PKE CPA-secure scheme.

With a CPA PKE:

- In CCA context, Eve generates invalid c and observes m'.
- $\rightarrow$ Insecure since only CPA-secure.



CCAKEM.Dec

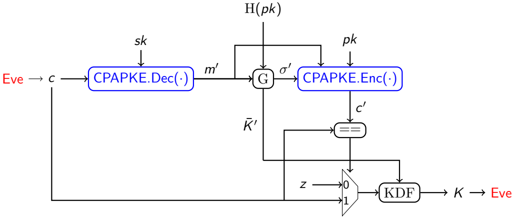
Build a CCA KEM from CPA PKE: FO-transform

Fujisaki-Okamoto (FO) transform:

- Leverage PKE CPA-secure scheme.
- Re-encrypt m' with sk to obtain c'.
- Secret is returned only if $c == c'$.

With a CPA PKE:

- In CCA context, Eve generates invalid c and observes m'.
- → Insecure since only CPA-secure.



CCAKEM.Dec

Build a CCA KEM from CPA PKE: FO-transform

Fujisaki-Okamoto (FO) transform:

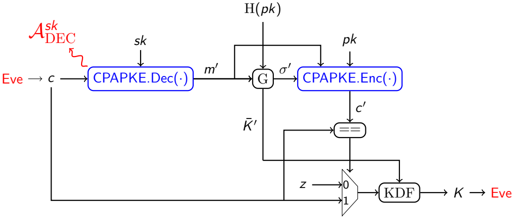
- Leverage PKE CPA-secure scheme.
- Re-encrypt m' with sk to obtain c'.
- Secret is returned only if $c == c'$.

Side-channel attacks:

1. $\mathcal{A}^{sk}_{\mathrm{DEC}}$: classical DPA.

With a CPA PKE:

- In CCA context, Eve generates invalid c and observes m'.
- → Insecure since only CPA-secure.



CCAKEM.Dec

Build a CCA KEM from CPA PKE: FO-transform

Fujisaki-Okamoto (FO) transform:

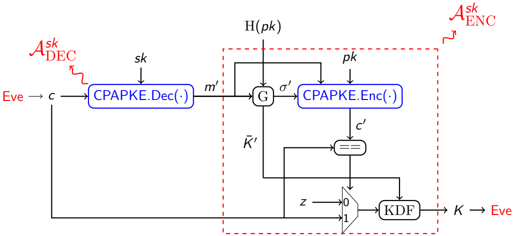
- Leverage PKE CPA-secure scheme.
- Re-encrypt m' with sk to obtain c'.
- Secret is returned only if $c == c'$.

Side-channel attacks:

1. $\mathcal{A}^{sk}_{\mathrm{DEC}}$: classical DPA.
2. $\mathcal{A}^{sk}_{\mathrm{ENC}}$: exploits leakeage in re-encryption.

With a CPA PKE:

- In CCA context, Eve generates invalid c and observes m'.
- → Insecure since only CPA-secure.



CCAKEM.Dec

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{DEC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

Standard DPA against *sk*:

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{DEC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

Standard DPA against *sk*:

1. Random (valid) ciphertext $\boldsymbol{c}$.

What are the potential side-channel attacks ? $\mathcal{A}_{\mathrm{DEC}}^{sk}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

CPAPKE.Dec has to compute:

$c_0 \cdot sk_0$

$c_1 \cdot sk_1$

$c_2 \cdot sk_2$

$c_3 \cdot sk_3$

Standard DPA against sk:

1. Random (valid) ciphertext $\boldsymbol{c}$.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{DEC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

CPAPKE.Dec has to compute:

$c_0 \cdot sk_0$

$c_1 \cdot sk_1$

$c_2 \cdot sk_2$

$c_3 \cdot sk_3$

Standard DPA against sk:

1. Random (valid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on $u_i \cdot sk_i$.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{DEC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

CPAPKE.Dec has to compute:

$c_0 \cdot sk_0 \rightsquigarrow$ Info. on sk_0

$c_1 \cdot sk_1 \rightsquigarrow$ Info. on sk_1

$c_2 \cdot sk_2 \rightsquigarrow$ Info. on sk_2

$c_3 \cdot sk_3 \rightsquigarrow$ Info. on sk_3

Standard DPA against sk:

1. Random (valid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on $u_i \cdot sk_i$.
3. Update guess on sk_i.

What are the potential side-channel attacks ? $\mathcal{A}_{\mathrm{DEC}}^{sk}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

CPAPKE.Dec has to compute:

$c_0 \cdot sk_0 \rightsquigarrow$ Info. on sk_0

$c_1 \cdot sk_1 \rightsquigarrow$ Info. on sk_1

$c_2 \cdot sk_2 \rightsquigarrow$ Info. on sk_2

$c_3 \cdot sk_3 \rightsquigarrow$ Info. on sk_3

Standard DPA against sk:

1. Random (valid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on $u_i \cdot sk_i$.
3. Update guess on sk_i.
4. Repeat to improve guess on sk_i.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{DEC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & c_1 & c_2 & c_3 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

CPAPKE.Dec has to compute:

$c_0 \cdot sk_0 \rightsquigarrow$ Info. on sk_0

$c_1 \cdot sk_1 \rightsquigarrow$ Info. on sk_1

$c_2 \cdot sk_2 \rightsquigarrow$ Info. on sk_2

$c_3 \cdot sk_3 \rightsquigarrow$ Info. on sk_3

Standard DPA against sk:

1. Random (valid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on $u_i \cdot sk_i$.
3. Update guess on sk_i.
4. Repeat to improve guess on sk_i.

→ One pair (c, L) improves guess on all sk_i.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\text{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

Recent SPA against *sk*:

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

Recent SPA against *sk*:

1. Sparse (invalid) ciphertext $\boldsymbol{c}$.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= MSB(c_0 \cdot sk_0) = \{0, 1\}$$

Recent SPA against sk:

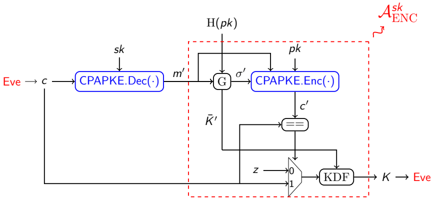
1. Sparse (invalid) ciphertext $\boldsymbol{c}$.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\text{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= MSB(c_0 \cdot sk_0) = \{0, 1\}$$

CCAKEM.Dec re-encrypt the bit m':



Recent SPA against sk:

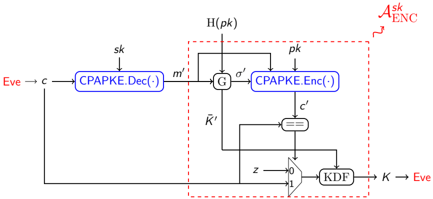
1. Sparse (invalid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on m':
   - Many operations depend of a single bit in m'.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= MSB(c_0 \cdot sk_0) = \{0, 1\}$$

CCAKEM.Dec re-encrypt the bit m':



Recent SPA against sk:

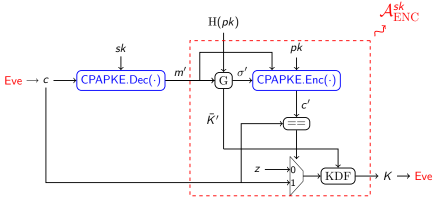
1. Sparse (invalid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on m':
   - Many operations depend of a single bit in m'.
3. Update guess of sk_0.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= MSB(c_0 \cdot sk_0) = \{0,1\}$$

CCAKEM.Dec re-encrypt the bit m':



Recent SPA against sk:

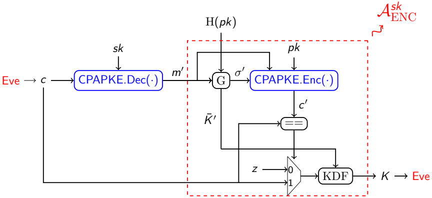
1. Sparse (invalid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on m':
   - Many operations depend of a single bit in m'.
3. Update guess of sk_0.
4. Repeat to improve guess on sk_0.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\mathrm{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= MSB(c_0 \cdot sk_0) = \{0, 1\}$$

CCAKEM.Dec re-encrypt the bit m':



Recent SPA against sk:

1. Sparse (invalid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on m':
   - Many operations depend of a single bit in m'.
3. Update guess of sk_0.
4. Repeat to improve guess on sk_0.
5. Repeat for another sk_i.

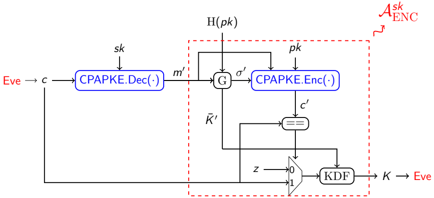
→ One pair (c, L) improves guess on one sk_i.

What are the potential side-channel attacks ? $\mathcal{A}^{sk}_{\text{ENC}}$

$$m' = MSB(\underbrace{\begin{bmatrix} c_0 & 0 & 0 & 0 \end{bmatrix}}_{\text{ciphertext}} \underbrace{\begin{bmatrix} sk_0 \\ sk_1 \\ sk_2 \\ sk_3 \end{bmatrix}}_{\text{secret key}})$$

$$= MSB(c_0 \cdot sk_0) = \{0, 1\}$$

CCAKEM.Dec re-encrypt the bit m':



Recent SPA against sk:

1. Sparse (invalid) ciphertext $\boldsymbol{c}$.
2. Collect leakage L on m':
    - Many operations depend of a single bit in m'.
3. Update guess of sk_0.
4. Repeat to improve guess on sk_0.
5. Repeat for another sk_i.

→ One pair (c, L) improves guess on one sk_i.

→ CCA attack on CPA-secure PKE thanks to leakage.

Summary so far

Summary so far

Attacks:

- $\mathcal{A}^{sk}_{\mathrm{DEC}}$: Standard DPA recovering all sk_i in parallel.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$: CCA attack exploiting leakage to observe output of CPA-secure PKE.

Summary so far

Attacks:

- $\mathcal{A}^{sk}_{\mathrm{DEC}}$: Standard DPA recovering all sk_i in parallel.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$: CCA attack exploiting leakage to observe output of CPA-secure PKE.

Questions ?

- How does FO-transform impact the cost of SCA-secure implementations:
  - CPAPKE.Enc is more costly to protect.
  - CPAPKE.Enc generates many leakages on a single bit.

Summary so far

Attacks:

- $\mathcal{A}^{sk}_{\mathrm{DEC}}$: Standard DPA recovering all sk_i in parallel.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$: CCA attack exploiting leakage to observe output of CPA-secure PKE.

Questions ?

- How does FO-transform impact the cost of SCA-secure implementations:
    - CPAPKE.Enc is more costly to protect.
    - CPAPKE.Enc generates many leakages on a single bit.
- Interesting to use different protection for CPAPKE.Enc and CPAPKE.Dec ?

Summary so far

Attacks:

- $\mathcal{A}^{sk}_{\mathrm{DEC}}$: Standard DPA recovering all sk_i in parallel.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$: CCA attack exploiting leakage to observe output of CPA-secure PKE.

Questions ?

- How does FO-transform impact the cost of SCA-secure implementations:
  - CPAPKE.Enc is more costly to protect.
  - CPAPKE.Enc generates many leakages on a single bit.
- Interesting to use different protection for CPAPKE.Enc and CPAPKE.Dec ?
- What is the room to alternative to the FO-transform ?

Methodology

Methodology

How do we proceed:

- Model SCA attacks with information theoretic metrics.
- Model cost with paper & pencil approximations.
- Compare the impact of attacks on security of designs.

Methodology

How do we proceed:

- Model SCA attacks with information theoretic metrics.
- Model cost with paper & pencil approximations.
- Compare the impact of attacks on security of designs.

Pro & cons:

+ Easy and fast way to explore the design space.
- Model / approximatereal attacks.

Methodology

How do we proceed:

- Model SCA attacks with information theoretic metrics.
- Model cost with paper & pencil approximations.
- Compare the impact of attacks on security of designs.

Pro & cons:

+ Easy and fast way to explore the design space.
- Model / approximatereal attacks.

→ We provide trends and not exact numbers.

Content

Introduction

Modeling Security

Modeling Performance

Trends in Perf. vs. Security

Take Home Message

How to model attacks ?

Attack complexity

$$N \approx —$$

A few parameters:

N Data complexity of the attack.

2

How to model attacks ?

Attack complexity

$$N \approx \frac{}{\lambda}$$

Physi. param.

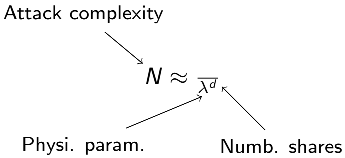
A few parameters:

- N Data complexity of the attack.
- λ Platform dependent parameter (inverse of noise).

---

[2]

How to model attacks ?



Attack complexity

$$N \approx \frac{}{\lambda^d}$$

Physi. param.

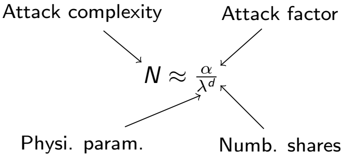
Numb. shares

A few parameters:

- N Data complexity of the attack.
- λ Platform dependent parameter (inverse of noise).
- d Number of shares used in the implementations.

2

How to model attacks ?



Attack complexity

Attack factor

$N \approx \frac{\alpha}{\lambda^d}$

Physi. param.

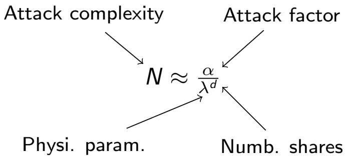
Numb. shares

A few parameters:

N	Data complexity of the attack.
λ	Platform dependent parameter (inverse of noise).
d	Number of shares used in the implementations.
α	Constant factor related to the attack methodology.

2

How to model attacks ?



A few parameters:

N	Data complexity of the attack.
λ	Platform dependent parameter (inverse of noise).
d	Number of shares used in the implementations.
α	Constant factor related to the attack methodology.

→ For each attacks, we will evaluate α[2].

---

[2] See full paper for more detailed attack modeling.

Modeling $\mathcal{A}^{sk}_{\mathrm{ENC}}$ and $\mathcal{A}^{sk}_{\mathrm{DEC}}$

Attacks against CPAPKE.Dec ($\mathcal{A}^{sk}_{\mathrm{DEC}}$)

Attacks against CPAPKE.Enc ($\mathcal{A}^{sk}_{\mathrm{ENC}}$)

Modeling $\mathcal{A}^{sk}_{\mathrm{ENC}}$ and $\mathcal{A}^{sk}_{\mathrm{DEC}}$

Attacks against CPAPKE.Dec ($\mathcal{A}^{sk}_{\mathrm{DEC}}$)

$\mathcal{A}^{sk}_{\mathrm{DEC}}$ can:

- Attack all the *sk* coefficients in parallel.
- Exploit few leakages in CPAPKE.Dec.

Attacks against CPAPKE.Enc ($\mathcal{A}^{sk}_{\mathrm{ENC}}$)

Modeling $\mathcal{A}^{sk}_{\mathrm{ENC}}$ and $\mathcal{A}^{sk}_{\mathrm{DEC}}$

Attacks against CPAPKE.Dec ($\mathcal{A}^{sk}_{\mathrm{DEC}}$)

$\mathcal{A}^{sk}_{\mathrm{DEC}}$ can:

- Attack all the *sk* coefficients in parallel.
- Exploit few leakages in CPAPKE.Dec.

For Kyber768:

$$\alpha_{Dec} \approx 2$$

Attacks against CPAPKE.Enc ($\mathcal{A}^{sk}_{\mathrm{ENC}}$)

Modeling $\mathcal{A}^{sk}_{\mathrm{ENC}}$ and $\mathcal{A}^{sk}_{\mathrm{DEC}}$

Attacks against CPAPKE.Dec ($\mathcal{A}^{sk}_{\mathrm{DEC}}$)

$\mathcal{A}^{sk}_{\mathrm{DEC}}$ can:

- Attack all the *sk* coefficients in parallel.
- Exploit few leakages in CPAPKE.Dec.

For Kyber768:

$$\alpha_{Dec} \approx 2$$

Attacks against CPAPKE.Enc ($\mathcal{A}^{sk}_{\mathrm{ENC}}$)

$\mathcal{A}^{sk}_{\mathrm{ENC}}$ can:

- Recover all different sk_i sequentially.
- Exploit all leakages in CPAPKE.Enc.

Modeling $\mathcal{A}^{sk}_{\mathrm{ENC}}$ and $\mathcal{A}^{sk}_{\mathrm{DEC}}$

Attacks against CPAPKE.Dec ($\mathcal{A}^{sk}_{\mathrm{DEC}}$)

$\mathcal{A}^{sk}_{\mathrm{DEC}}$ can:

- Attack all the *sk* coefficients in parallel.
- Exploit few leakages in CPAPKE.Dec.

For Kyber768:

$$\alpha_{Dec} \approx 2$$

Attacks against CPAPKE.Enc ($\mathcal{A}^{sk}_{\mathrm{ENC}}$)

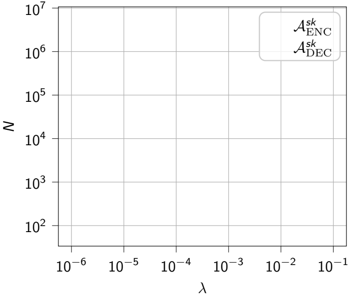
$\mathcal{A}^{sk}_{\mathrm{ENC}}$ can:

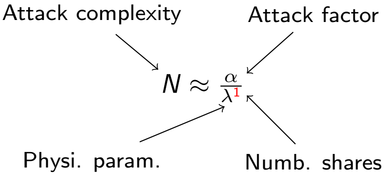
- Recover all different sk_i sequentially.
- Exploit all leakages in CPAPKE.Enc.

For Kyber768:

$$\alpha_{Enc} \approx 1/50$$

Comparing attacks for unprotected implem. ($d = 1$)




Attack complexity

Attack factor

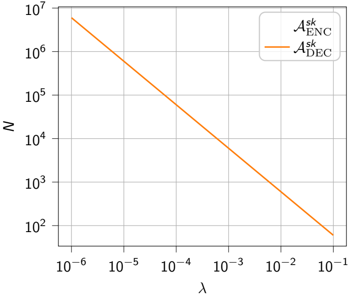
$$N \approx \frac{\alpha}{\lambda^{1}}$$

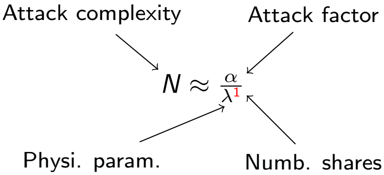
Physi. param.

Numb. shares

Comparing attack complexities N:

Comparing attacks for unprotected implem. ($d = 1$)





Attack complexity

Attack factor

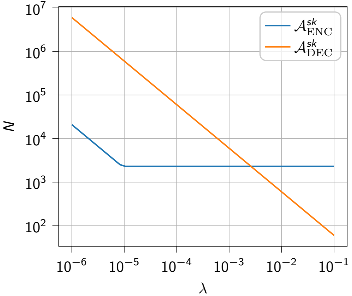
$$N \approx \frac{\alpha}{\lambda^{1}}$$

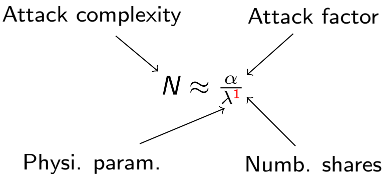
Physi. param.

Numb. shares

Comparing attack complexities N:

Comparing attacks for unprotected implem. ($d = 1$)





Attack complexity

Attack factor

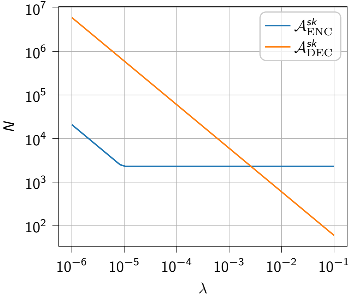
$$N \approx \frac{\alpha}{\lambda^{1}}$$

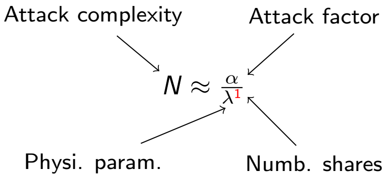
Physi. param.

Numb. shares

Comparing attack complexities N:

Comparing attacks for unprotected implem. ($d = 1$)




Attack complexity

Attack factor

$$N \approx \frac{\alpha}{\lambda^{1}}$$

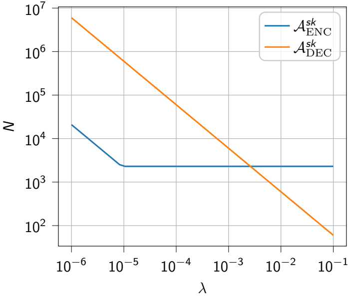
Physi. param.

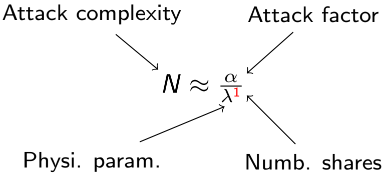
Numb. shares

Comparing attack complexities N:

- Noise increase (smaller λ) means harder attack.

Comparing attacks for unprotected implem. ($d = 1$)





Attack complexity

Attack factor

$$N \approx \frac{\alpha}{\lambda^{1}}$$

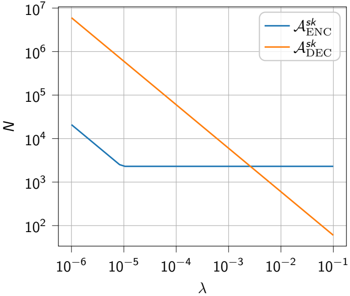
Physi. param.

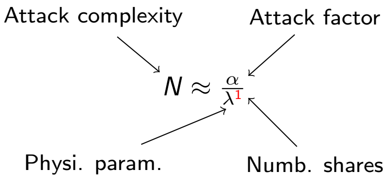
Numb. shares

Comparing attack complexities N:

- Noise increase (smaller λ) means harder attack.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$ saturates for large λ.

Comparing attacks for unprotected implem. ($d = 1$)




Attack complexity

Attack factor

$$N \approx \frac{\alpha}{\lambda^{1}}$$

Physi. param.

Numb. shares

Comparing attack complexities N:

- Noise increase (smaller λ) means harder attack.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$ saturates for large λ.
- $\mathcal{A}^{sk}_{\mathrm{ENC}}$ more efficient than $\mathcal{A}^{sk}_{\mathrm{DEC}}$ by a factor ≈ 100.

Content

Modeling CPAPKE.Dec and CPAPKE.Enc costs (1)

Cost of CPAPKE.Dec

Cost of CPAPKE.Enc

Modeling CPAPKE.Dec and CPAPKE.Enc costs (1)

Cost of CPAPKE.Dec

Masking involves:

- Arithmetic masking for lattice operations.
- Boolean masking for polynomial compressions.

Cost of CPAPKE.Enc

Modeling CPAPKE.Dec and CPAPKE.Enc costs (1)

Cost of CPAPKE.Dec

Masking involves:

- Arithmetic masking for lattice operations.
- Boolean masking for polynomial compressions.

→ Arithmetic to Boolean conversions (hence quadratic overheads):

$$\zeta_{Enc} = \beta_{Enc} \cdot d_{Enc}^2$$

Cost of CPAPKE.Enc

Modeling CPAPKE.Dec and CPAPKE.Enc costs (1)

Cost of CPAPKE.Dec

Masking involves:

- Arithmetic masking for lattice operations.
- Boolean masking for polynomial compressions.

→ Arithmetic to Boolean conversions (hence quadratic overheads):

$$\zeta_{Enc} = \beta_{Enc} \cdot d_{Enc}^2$$

Cost of CPAPKE.Enc

Masking involves:

- Arithmetic masking for lattice arithmetic.
- Boolean masking for polynomial comparison.
- Masked hash functions

Modeling CPAPKE.Dec and CPAPKE.Enc costs (1)

Cost of CPAPKE.Dec

Masking involves:

- Arithmetic masking for lattice operations.
- Boolean masking for polynomial compressions.

→ Arithmetic to Boolean conversions (hence quadratic overheads):

$$\zeta_{Enc} = \beta_{Enc} \cdot d_{Enc}^2$$

Cost of CPAPKE.Enc

Masking involves:

- Arithmetic masking for lattice arithmetic.
- Boolean masking for polynomial comparison.
- Masked hash functions

→ Various masking conversions required (hence quadratic overheads):

$$\zeta_{Dec} = \beta_{Dec} \cdot d_{Dec}^2$$

Modeling CPAPKE.Dec and CPAPKE.Enc costs (2)

$$\frac{\beta_{Enc}}{\beta_{Dec}}$$

---

[3] Bos et al. “Masking Kyber: First- and Higher-Order Implementations”. In: *TCHES 2021* ().

[4] Bronchain and Cassiers. “Bitslicing Arithmetic/Boolean Masking Conversions for Fun and Profit with Application to Lattice-Based KEMs”. In: *eprint 2022/158* ().

Modeling CPAPKE.Dec and CPAPKE.Enc costs (2)

Software implementation of Kyber768 from[3]:

Operation	**Number of shares**					
	2	*3*	*4*	*5*	*6*	*7*
`crypto_kem_dec`	3 178	57 141	97 294	174 220	258 437	350 529
`indcpa_dec`	200	4 203	7 047	13 542	20 323	27 230
`indcpa_enc`	2 024	18 879	32 594	53 298	75 692	104 191
`comparison`	693	32 293	54 725	102 922	156 075	210 518

$$\frac{\beta_{Enc}}{\beta_{Dec}}$$

---

[3] Bos et al. “Masking Kyber: First- and Higher-Order Implementations”. In: *TCHES 2021* ().

[4] Bronchain and Cassiers. “Bitslicing Arithmetic/Boolean Masking Conversions for Fun and Profit with Application to Lattice-Based KEMs”. In: *eprint 2022/158* ().

Modeling CPAPKE.Dec and CPAPKE.Enc costs (2)

Software implementation of Kyber768 from[3]:

Operation	**Number of shares**					
	2	*3*	*4*	*5*	*6*	*7*
`crypto_kem_dec`	3 178	57 141	97 294	174 220	258 437	350 529
`indcpa_dec`	200	4 203	7 047	13 542	20 323	27 230
`indcpa_enc`	2 024	18 879	32 594	53 298	75 692	104 191
`comparison`	693	32 293	54 725	102 922	156 075	210 518

$$\frac{\beta_{Enc}}{\beta_{Dec}} \approx \frac{(104,191+210,518)}{(27,230)}$$

$$\approx 11.63$$

---

[3] Bos et al. "Masking Kyber: First- and Higher-Order Implementations". In: *TCHES 2021* ().

[4] Bronchain and Cassiers. "Bitslicing Arithmetic/Boolean Masking Conversions for Fun and Profit with Application to Lattice-Based KEMs". In: *eprint 2022/158* ().

Modeling CPAPKE.Dec and CPAPKE.Enc costs (2)

Software implementation of Kyber768 from[3]:

Operation	**Number of shares**					
	2	*3*	*4*	*5*	*6*	*7*
`crypto_kem_dec`	3 178	57 141	97 294	174 220	258 437	350 529
`indcpa_dec`	200	4 203	7 047	13 542	20 323	27 230
`indcpa_enc`	2 024	18 879	32 594	53 298	75 692	104 191
`comparison`	693	32 293	54 725	102 922	156 075	210 518

$$\frac{\beta_{Enc}}{\beta_{Dec}} \approx \frac{(104,191 + 210,518)}{(27,230)}$$
$$\approx 11.63$$

Caution: Numbers can change between implementations:

- $\beta_{Enc}/\beta_{Dec} \approx 40$ with numbers from[4]

---

[3] Bos et al. "Masking Kyber: First- and Higher-Order Implementations". In: *TCHES 2021* ().

[4] Bronchain and Cassiers. "Bitslicing Arithmetic/Boolean Masking Conversions for Fun and Profit with Application to Lattice-Based KEMs". In: *eprint 2022/158* ().

Content

Introduction

Modeling Security

Modeling Performance

Trends in Perf. vs. Security

Take Home Message

What is the impact of attacks on costs ?

1. How many of shares to secure Enc & Dec:

What is the impact of attacks on costs ?

1. How many of shares to secure Enc & Dec:

We fix:

γ : target security.
λ : platform dependent parameter.
α : attack parameter.

What is the impact of attacks on costs ?

1. How many of shares to secure Enc & Dec:

We fix:

γ : target security.

λ : platform dependent parameter.

α : attack parameter.

We derive the number of shares d_{Enc} and d_{Dec}:

$$\gamma \geq \frac{\alpha}{\lambda^{d}}$$

What is the impact of attacks on costs ?

1. How many of shares to secure Enc & Dec:

We fix:

- γ : target security.
- λ : platform dependent parameter.
- α : attack parameter.

We derive the number of shares d_{Enc} and d_{Dec}:

$$\gamma \geq \frac{\alpha}{\lambda^{d}}$$

2. Compare the costs to secure Enc & Dec:

What is the impact of attacks on costs ?

1. How many of shares to secure Enc & Dec:

We fix:

γ : target security.

λ : platform dependent parameter.

α : attack parameter.

We derive the number of shares d_{Enc} and d_{Dec}:

$$\gamma \geq \frac{\alpha}{\lambda^{d}}$$

2. Compare the costs to secure Enc & Dec:

- For a fixed set of parameters $(\gamma, \lambda, \alpha)$.
- What is the time spent in securing CPAPKE.Enc & CPAPKE.Dec

What is the impact of attacks on costs ?

1. How many of shares to secure Enc & Dec:

We fix:

- γ : target security.
- λ : platform dependent parameter.
- α : attack parameter.

We derive the number of shares d_{Enc} and d_{Dec}:

$$\gamma \geq \frac{\alpha}{\lambda^{d}}$$

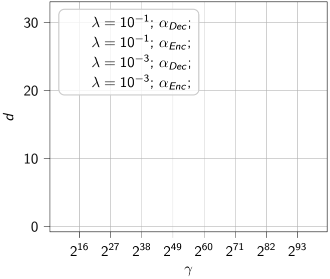
2. Compare the costs to secure Enc & Dec:

- For a fixed set of parameters $(\gamma, \lambda, \alpha)$.
- What is the time spent in securing CPAPKE.Enc & CPAPKE.Dec

We use the ratio:

$$\frac{\zeta_{Enc}}{\zeta_{Dec}} = \frac{\beta_{Enc} \cdot d_{Enc}^2}{\beta_{Enc} \cdot d_{Enc}^2}$$

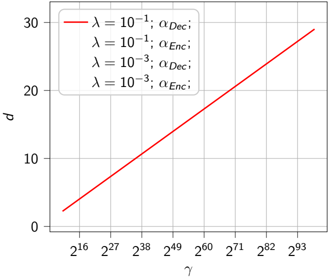
How many of shares to secure Enc & Dec:



More shares for:

$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

How many of shares to secure Enc & Dec:

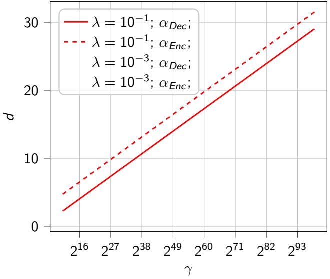


$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

More shares for:

- More security γ.

How many of shares to secure Enc & Dec:



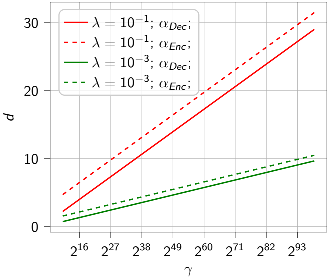
$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

More shares for:

- More security γ.
- More efficient attacks α.

→ constant absolute difference between d_{Enc} and d_{Dec}.

How many of shares to secure Enc & Dec:



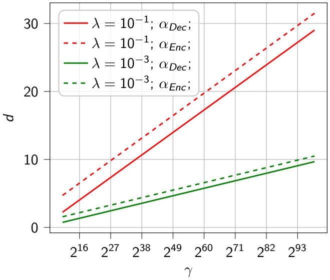
$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

More shares for:

- More security γ.
- More efficient attacks α.
- Less noise λ.

→ constant absolute difference between d_{Enc} and d_{Dec}.

How many of shares to secure Enc & Dec:



$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

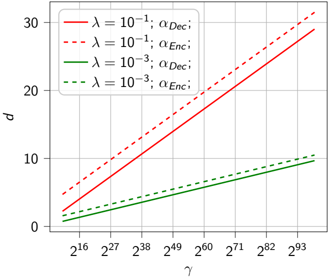
More shares for:

- More security γ.
- More efficient attacks α.
- Less noise λ.

→ constant absolute difference between d_{Enc} and d_{Dec}.

Relative difference between d_{Enc} and d_{Dec}:

How many of shares to secure Enc & Dec:



$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

More shares for:

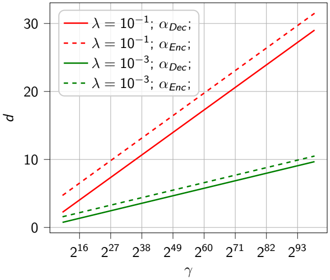
- More security γ.
- More efficient attacks α.
- Less noise λ.

$\rightarrow$ constant absolute difference between d_{Enc} and d_{Dec}.

Relative difference between d_{Enc} and d_{Dec}:

- Small γ: Large d's relative difference.

How many of shares to secure Enc & Dec:



$$d \geq \frac{\log(\alpha) - \log(\gamma)}{\log(\lambda)}$$

More shares for:

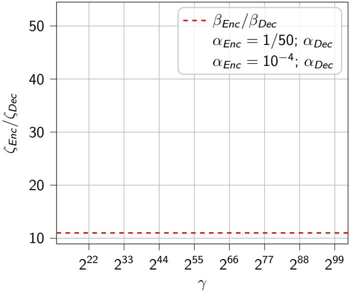
- More security γ.
- More efficient attacks α.
- Less noise λ.

→ constant absolute difference between d_{Enc} and d_{Dec}.

Relative difference between d_{Enc} and d_{Dec}:

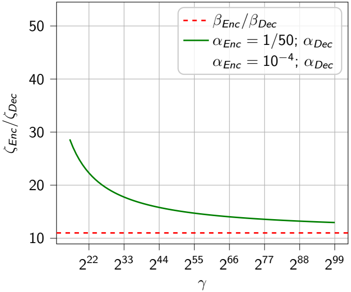
- Small γ: Large d's relative difference.
- Large γ: Small d's relative difference.

Compare the costs to secure Enc & Dec



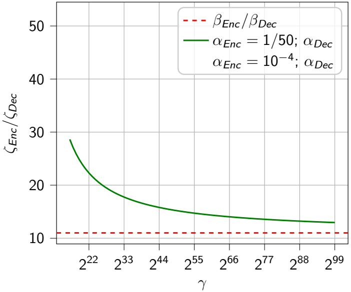
Observations:

Compare the costs to secure Enc & Dec



Observations:

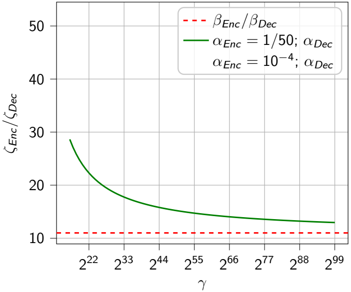
Compare the costs to secure Enc & Dec



Observations:

- Small γ: Large d's relative difference.
  - Enc dominates largely the cost due to larger d_{Enc}.
  - Incentive to get rid of FO-transform.

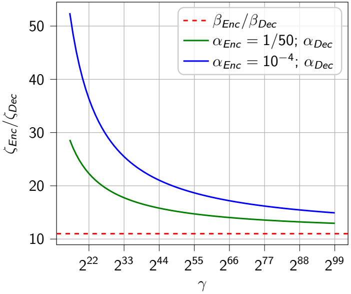
Compare the costs to secure Enc & Dec



Observations:

- Small γ: Large d's relative difference.
  - Enc dominates largely the cost due to larger d_{Enc}.
  - Incentive to get rid of FO-transform.
- Large γ: small d's relative difference.
  - Enc dominates less the cost.
  - Alternatives should be more efficient than $\frac{\beta_{Enc}}{\beta_{Dec}}$.

Compare the costs to secure Enc & Dec



Observations:

- Small γ: Large d's relative difference.
    - Enc dominates largely the cost due to larger d_{Enc}.
    - Incentive to get rid of FO-transform.
- Large γ: small d's relative difference.
    - Enc dominates less the cost.
    - Alternatives should be more efficient than $\frac{\beta_{Enc}}{\beta_{Dec}}$.

→ Same holds for more efficient $\mathcal{A}^{sk}_{\mathrm{ENC}}$.

Content

Introduction

Modeling Security

Modeling Performance

Trends in Perf. vs. Security

Take Home Message

Take home message

Future for SCA and PQ KEMs:

Take home message

Future for SCA and PQ KEMs:

- FO-transform leads to easy-to-mount attacks exploiting re-encryption.

Take home message

Future for SCA and PQ KEMs:

- FO-transform leads to easy-to-mount attacks exploiting re-encryption.
- Re-encryption dominates the cycle count because:
  - More computations.
  - More shares to compensate attacks.

Take home message

Future for SCA and PQ KEMs:

- FO-transform leads to easy-to-mount attacks exploiting re-encryption.
- Re-encryption dominates the cycle count because:
  - More computations.
  - More shares to compensate attacks.
- Its proportion decreases with target security.

Take home message

Future for SCA and PQ KEMs:

- FO-transform leads to easy-to-mount attacks exploiting re-encryption.
- Re-encryption dominates the cycle count because:
  - More computations.
  - More shares to compensate attacks.
- Its proportion decreases with target security.

Thanks !

@BronchainO