On the practical CPA^D security of "exact" and threshold FHE schemes and libraries

Marina Checri, Renaud Sirdey, Aymen Boudguiga, Jean-Paul Bultel

Université Paris-Saclay, CEA, List, F-91120, Palaiseau, France
{marina.checri, renaud.sirdey, jean-paul.bultel, aymen.boudguiga}@cea.fr

CRYPTO - August 18-22, 2024

France 2030 ANR Programs SecureCompute & TRUSTINCloudS
Horizon Europe Program

Table of contents

Table of contents

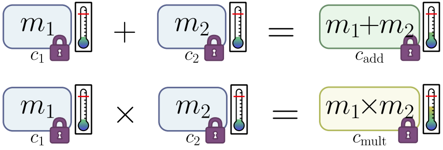
1. Introduction & Background
   - Homomorphic Encryption
   - Security model and CPA^D game
2. A CPA^D attack on “exact” FHE schemes
3. Impact on Threshold FHE
4. Countermeasures for “exact” and Threshold Schemes
5. Conclusion and Key takeaways

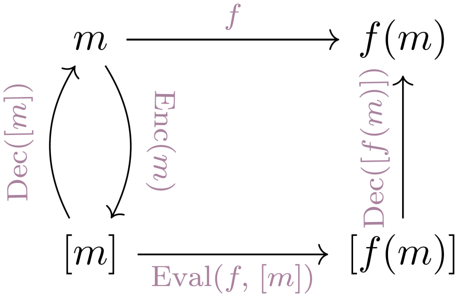
Table of contents

1. Introduction & Background
   - Homomorphic Encryption
   - Security model and CPA^D game
2. A CPA^D attack on "exact" FHE schemes
3. Impact on Threshold FHE
4. Countermeasures for "exact" and Threshold Schemes
5. Conclusion and Key takeaways

Homomorphic Encryption

What are we talking about?



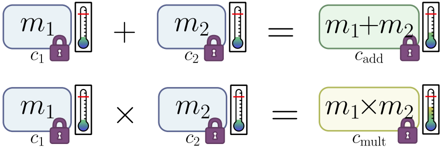


Ensure confidentiality *during* calculations

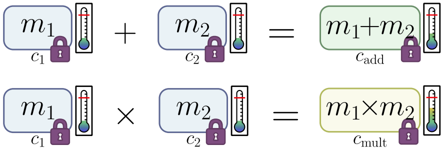
Typically, for Cloud Computing

Homomorphic Encryption

What are we talking about?



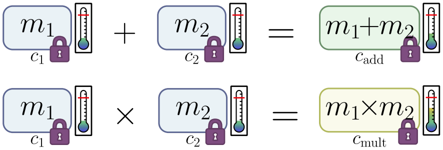
What are we talking about?



$$c = \left(a \ , \ b = m + \langle a, s \rangle + e \right)$$

FHE relies on **LWE**

What are we talking about?

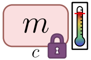


$$c = \left(a \;,\; b = m + \langle a, s \rangle + e \right)$$

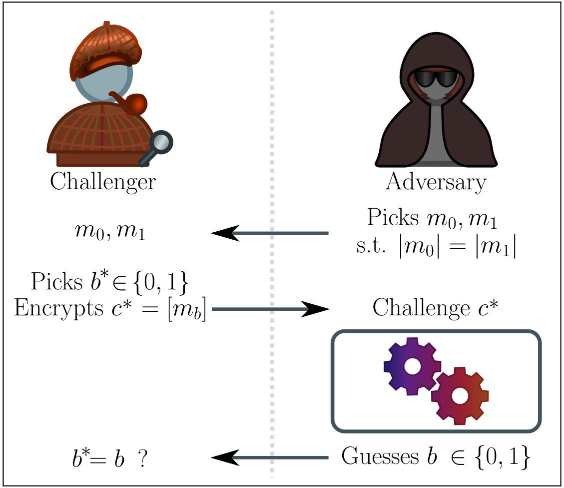
FHE relies on **LWE**

To ensure security, **noise is added during encryption**

but **it increases** at each homomorphic operation,

and may lead...  to **incorrect decryptions**!

CPA, CCA and CPA^D security game



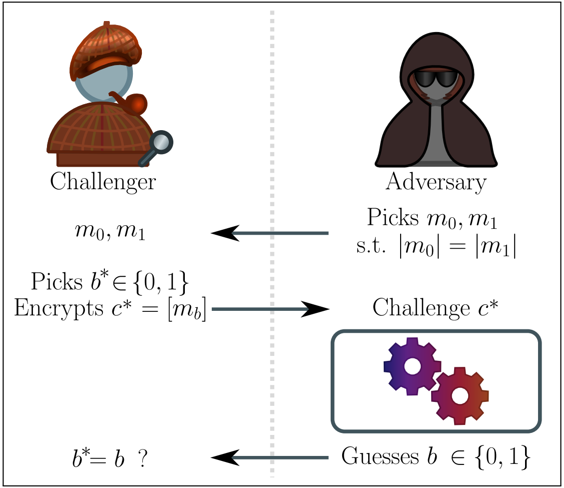
CPA, CCA and CPA^D security game

CPA Security Game

Chosen Plaintext Attack

- ➢ Encryption oracle

  ✔ FHE



CPA, CCA and CPA^D security game

CPA Security Game

Chosen Plaintext Attack

- ➢ Encryption oracle

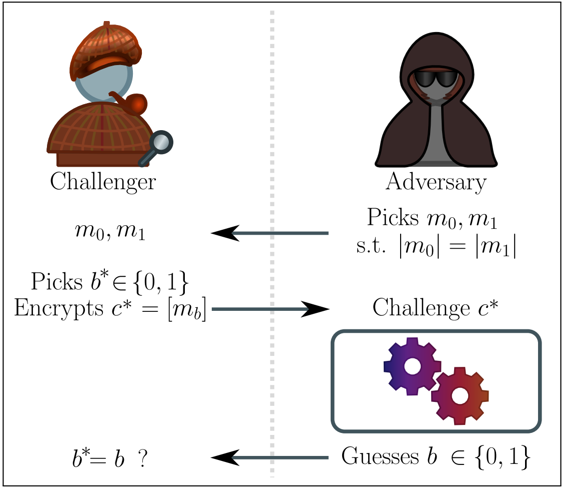
✔ FHE

CCA Security Game

Chosen Ciphertext Attack

- ➢ Encryption oracle
- ➢ Decryption oracle

✘ FHE



CPA, CCA and CPA^D security game

CPA Security Game

Chosen Plaintext Attack

➢ Encryption oracle

✔ FHE

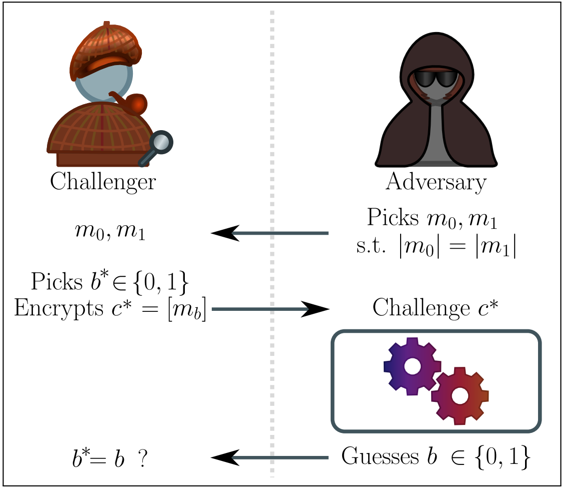
CCA Security Game

Chosen Ciphertext Attack

➢ Encryption oracle

➢ Decryption oracle

✘ FHE



CPA^D Security Game

Chosen Plaintext Attack with "Decryption oracle"

➢ Encryption oracle

➢ Evaluation oracle

➢ Limited Decryption oracle on well-formed ctxt

? FHE

Li & Micciancio. *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

CPA = CPA^D?

CPA = CPA^D?

- CPA^D = CPA + Limited Decryption Oracle
- The adversary seems to know all the output of the decryption oracle!

➢ CPA = CPA^D?

CPA = CPA^D?

- CPA^D = CPA + Limited Decryption Oracle
- The adversary seems to know all the output of the decryption oracle!

➢ CPA = CPA^D?

Li & Micciancio attack on **CKKS Approximate LWE Scheme** - EUROCRYPT'21

$$\begin{aligned}
&\text{CKKS.Encrypt}(m): &&\text{return} && (c_0, c_1) &= (m - a \cdot s + e,\ a), \qquad \text{with } a \xleftarrow{\$} \mathbb{Z}_q,\ e \xleftarrow{\$} \chi \\
&\text{CKKS.Decrypt}((c_0, c_1)): &&\text{return} && c_0 + c_1 \cdot s &= m - a \cdot s + e + a \cdot s \\
& && && &= m + \boxed{e} \\
& && && &\simeq m
\end{aligned}$$

CPA $=$ CPA^D?

- CPA^D = CPA + Limited Decryption Oracle
- The adversary seems to know all the output of the decryption oracle!

➢ CPA = CPA^D?

Li & Micciancio attack on **CKKS Approximate LWE Scheme** - EUROCRYPT'21

$$\begin{aligned}
&\text{CKKS.Encrypt}(m): && \text{return } (c_0, c_1) && = (m - a\cdot s + e,\ a), \quad \text{with } a \xleftarrow{\$} \mathbb{Z}_q,\ e \xleftarrow{\$} \chi \\
&\text{CKKS.Decrypt}((c_0, c_1)): && \text{return } c_0 + c_1 \cdot s && = m - a \cdot s + e + a \cdot s \\
&&&&& = m + \boxed{e} \\
&&&&& \simeq m
\end{aligned}$$

Compare to usual "Exact" LWE Schemes

$$\begin{aligned}
&\text{Encrypt}(m): && \text{return } (c_0, c_1) && = (Bm - a\cdot s + e,\ a), \quad \text{with } a \xleftarrow{\$} \mathbb{Z}_q,\ e \xleftarrow{\$} \chi \\
&\text{Decrypt}((c_0, c_1)): && \text{return } \lceil (c_0 + c_1 \cdot s)/B \rfloor && = \lceil (Bm - a\cdot s + e + a\cdot s)/B \rfloor \\
&&&&& = m
\end{aligned}$$

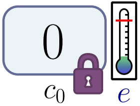
Table of contents

1. Introduction & Background
   - Homomorphic Encryption
   - Security model and CPA^D game
2. A CPA^D attack on “exact” FHE schemes
3. Impact on Threshold FHE
4. Countermeasures for “exact” and Threshold Schemes
5. Conclusion and Key takeaways

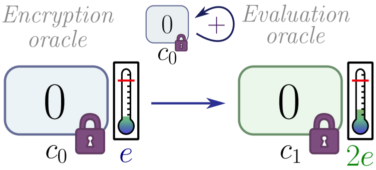
Find noiseless ciphertexts

Find noiseless ciphertexts

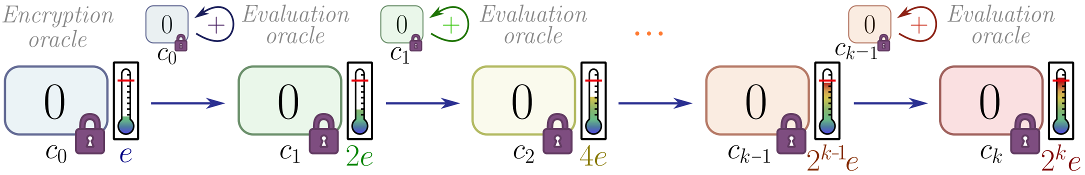
Encryption oracle



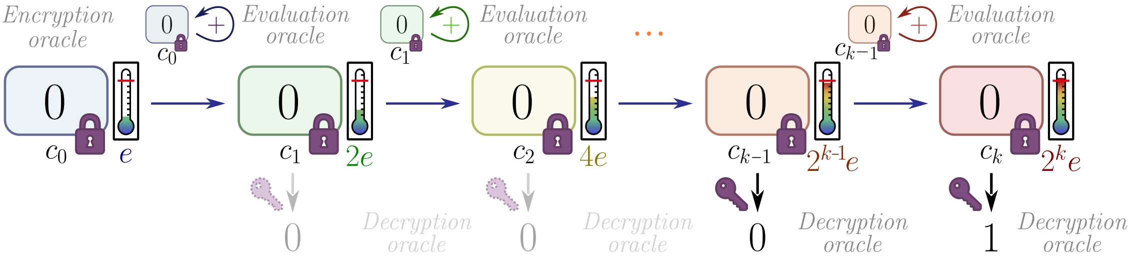
Find noiseless ciphertexts



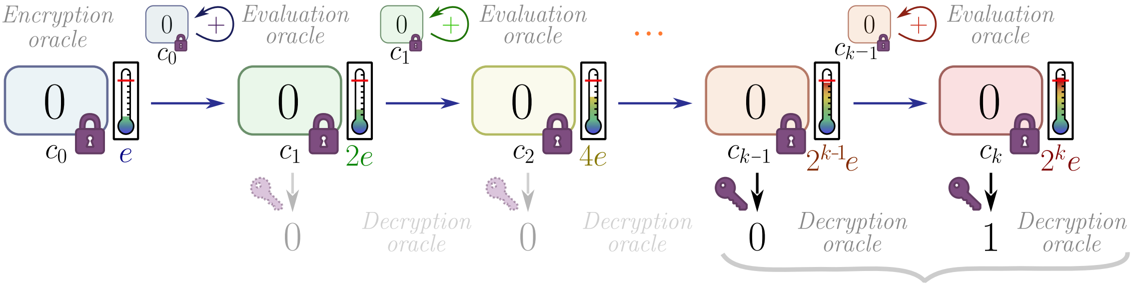
Find noiseless ciphertexts



Find noiseless ciphertexts



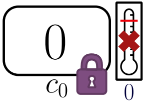
Find noiseless ciphertexts



$c_k = c_{k-1} + c_{k-1}$ until
$= (2^k a, \langle 2^k a, s\rangle + 2^k e)$

either $\mathsf{Dec}(c_{k-1})=0$ & $\mathsf{Dec}(c_k)=1$

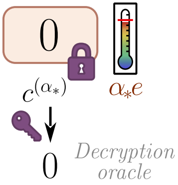
or $\mathsf{Dec}(c_k)=0$ & $2^k \geq q/2t$



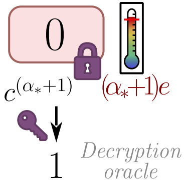
A CPAD Key recovery attack - Dichotomic search for $|e|$ (1/3)

A CPAD Key recovery attack - Dichotomic search for $|e|$ (1/3)

$c^{(\alpha_*)}$ encryption of 0 with noise $\alpha_* e$

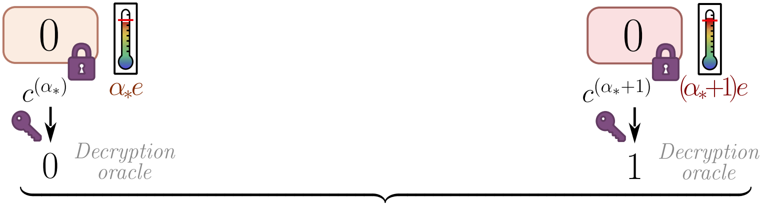


$c^{(\alpha_*+1)}$ encryption of 0 with noise $(\alpha_* + 1)e$



A CPAD Key recovery attack - Dichotomic search for $|e|$ (1/3)

$c^{(\alpha_*)}$ encryption of 0 with noise $\alpha_* e$ $c^{(\alpha_*+1)}$ encryption of 0 with noise $(\alpha_* + 1)e$



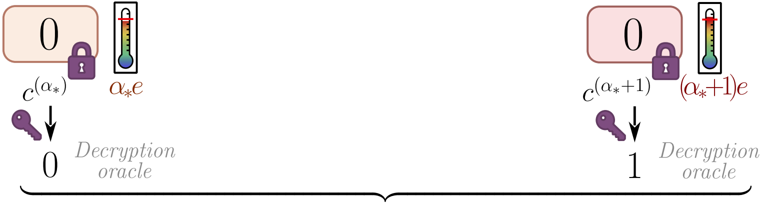
with $\frac{q}{2t(\alpha_*+1)} \leqslant |e| < \frac{q}{2t\alpha_*}$

➢ $|e|$ is uniquely determined when $\left\lceil \frac{q}{2t(\alpha_*+1)} \right\rceil = \left\lfloor \frac{q}{2t\alpha_*} \right\rfloor$

A CPAD Key recovery attack - Dichotomic search for $|e|$ (1/3)

$c^{(\alpha_*)}$ encryption of 0 with noise $\alpha_* e$

$c^{(\alpha_*+1)}$ encryption of 0 with noise $(\alpha_* + 1)e$

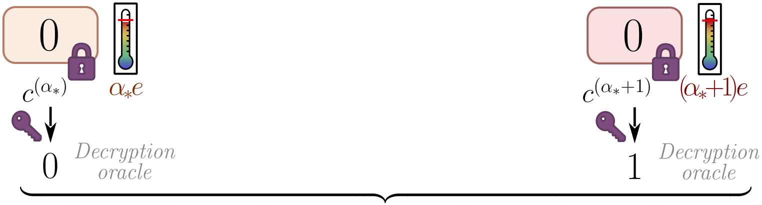


with $\frac{q}{2t(\alpha_*+1)} \leqslant |e| < \frac{q}{2t\alpha_*}$

- $|e|$ is uniquely determined when $\left\lceil \frac{q}{2t(\alpha_*+1)} \right\rceil = \left\lfloor \frac{q}{2t\alpha_*} \right\rfloor$ $\rightarrow$ Occurs when $|e| < \sqrt{\frac{q}{2t}}$

A CPAD Key recovery attack - Dichotomic search for $|e|$ (1/3)

$c^{(\alpha_*)}$ encryption of 0 with noise $\alpha_* e$ $c^{(\alpha_*+1)}$ encryption of 0 with noise $(\alpha_* + 1)e$

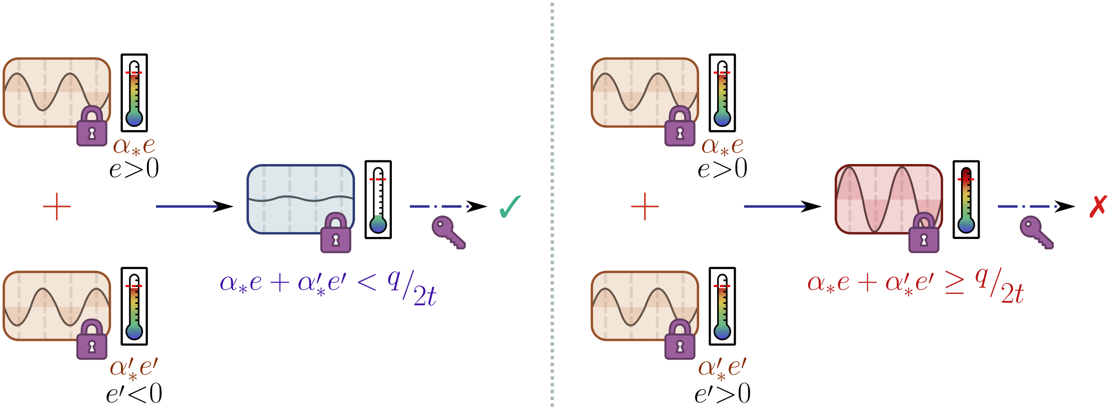


with $\frac{q}{2t(\alpha_*+1)} \leqslant |e| < \frac{q}{2t\alpha_*}$

- $|e|$ is uniquely determined when $\left\lceil \frac{q}{2t(\alpha_*+1)} \right\rceil = \left\lfloor \frac{q}{2t\alpha_*} \right\rfloor$ $\rightarrow$ Occurs when $|e| < \sqrt{\frac{q}{2t}}$
- Construct $c_k = c_{k-1} + c_{k-1}$ and $c^{(\alpha)} := \sum_k (\alpha_k = 1) c_k$, then $c^{(\alpha)} = (\alpha a, \langle \alpha a, s\rangle + \alpha e)$

Identify ciphertexts with same noise sign.

$\hookrightarrow$ *Evaluation and decryption oracles*



A CPAD Key recovery attack - Dichotomic search for $|e|$ (3/3)

Solve two systems of n linear equations to recover the key

$$\begin{cases} b_1 &= \langle a_1, s \rangle + |e_1| \\ b_2 &= \langle a_2, s \rangle + |e_2| \\ b_3 &= \langle a_3, s \rangle + |e_3| \\ b_4 &= \langle a_4, s \rangle + |e_4| \\ \cdots &= \cdots \\ b_n &= \langle a_n, s \rangle + |e_n| \end{cases} \qquad \begin{cases} b_1 &= \langle a_1, s \rangle - |e_1| \\ b_2 &= \langle a_2, s \rangle - |e_2| \\ b_3 &= \langle a_3, s \rangle - |e_3| \\ b_4 &= \langle a_4, s \rangle - |e_4| \\ \cdots &= \cdots \\ b_n &= \langle a_n, s \rangle - |e_n| \end{cases}$$

Try to decrypt fresh encryptions of 0 → the correct key always outputs 0

Win the CPA^D game by decrypting the challenge ciphertext c^*!

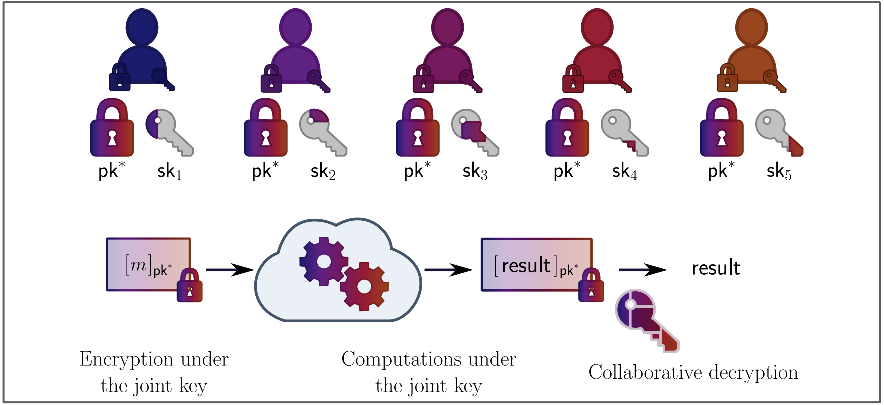
Some experimental results

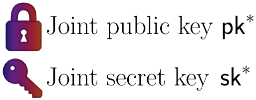
Library	Scheme	Parameters					Proportion of ctxt with $\|e\|$ recovered	Proportion of noisefree ctxt	Time
		λ	n	$\log_2(q)$	σ	t			
SEAL	BFV	95	4096	109	3.2	1024	1	6250/232858	$2m50s$
	BFV	227	4096	58	3.2	1024	1	1481/860557	$1m20s$
	BGV	227	4096	58	3.2	1024	1	124/65405	$52s$
OpenFHE	BFV	128	8192	120	3.19	1024	1	69/48929	$19m30s$
	BFV	256	16384	120	3.19	1024	1	173/130535	$75m30s$
	BGV	128	8192	69	3.19	1024	1	59/32811	$18m30s$
	BGV	256	16384	71	3.19	1024	1	80/65559	$68m50s$
TFHElib	TFHE	97	630	32	2^{17}	2	1295/5427	0	$0.245s$
	TFHE	128	700	32	81604.378	2	1363/3678	0	$0.195s$
	TFHE	128	1024	32	81604.378	4	2070/5608	0	$0.412s$
	TFHE	128	1024	32	279.172	16	2021/2041	11/2041	$0.237s$
Lattigo	BFV	95	4096	109	3.2	65537	1	785/29241	$5m50s$
	BFV	98	2048	54	3.2	65537	1	69/24518	$46s$
	BFV	106	4096	101	3.2	65537	1	829/32260	$6m40s$
	BFV	106	8192	202	3.2	65537	1	457/23943	$52m00s$
	BFV	217	4096	60	3.2	65537	1	828/31934	$1m25s$

Table of contents

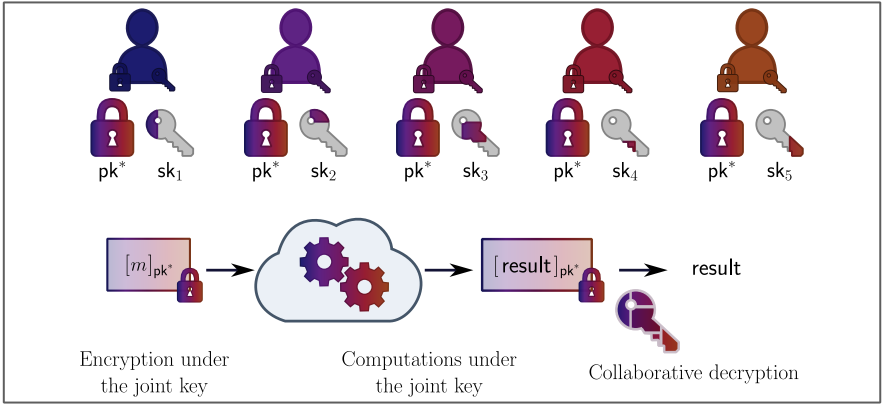
1. Introduction & Background
   - Homomorphic Encryption
   - Security model and CPA^D game
2. A CPA^D attack on “exact” FHE schemes
3. Impact on Threshold FHE
4. Countermeasures for “exact” and Threshold Schemes
5. Conclusion and Key takeaways

Impact on Threshold FHE





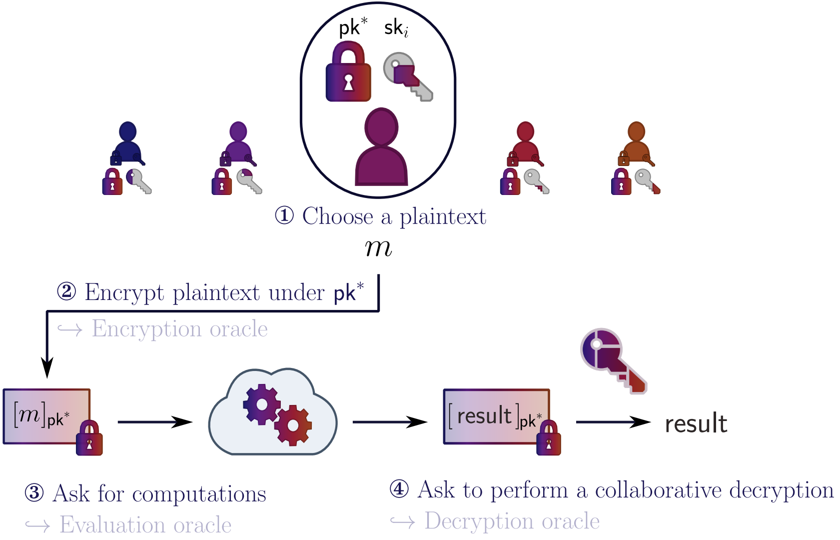
Impact on Threshold FHE



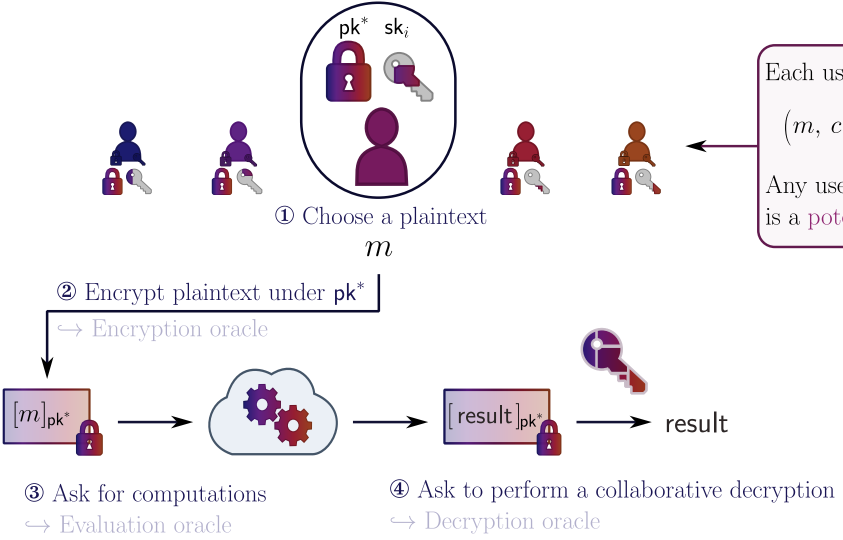
Joint public key pk^* — $\mathsf{pk}^* = \sum \mathsf{pk}_i$:

Joint secret key sk^* — $\mathsf{sk}^* = \sum \mathsf{sk}_i$:

Relationship between CPA^D and Threshold FHE



Relationship between CPA^D and Threshold FHE



Each user has

$$\big(m,\ c = \mathsf{Enc}_{\mathsf{pk}^*}(m),\ \mathsf{Dec}_{\mathsf{sk}^*}(c)\big)$$

Any user in a threshold scheme is a potential CPA^D attacker

Does the attack work against Threshold FHE schemes?

Algorithm 1: Collective Key Switch

Input: Ciphertext $ct = (c_0, c_1)$ of variance σ_{ct}^2
Private input: s_i, s'_i for each party P_i
Output: Key-switched ciphertext $ct' = (c'_0, c_1)$
Each *party* P_i
- Samples $e_i \leftarrow \chi_{CKS}(\sigma_{ct}^2)$
- Computes and Discloses $h_i = (s_i - s'_i) \cdot c_1 + e_i$

return $ct' = (c_0 + \sum_{P_i} h_i, c_1)$

Mouchet et al. *Multiparty Homomorphic Encryption from Ring-Learning-with-Errors*. PoPETs'21

Does the attack work against Threshold FHE schemes?

Algorithm 1: Collective Key Switch

Input: Ciphertext $ct = (c_0, c_1)$ of variance σ_{ct}^2
Private input: s_i, s_i' for each party P_i
Output: Key-switched ciphertext $ct' = (c_0', c_1)$
Each *party* P_i
  Samples $e_i \leftarrow \chi_{CKS}(\sigma_{ct}^2)$ ▷ Smudging noise sampled from $\chi_{\mathrm{CKS}} = \mathcal{N}(0, 2^{\lambda}\sigma_{\mathrm{ct}}^2)$
  Computes and Discloses $h_i = (s_i - s_i') \cdot c_1 + e_i$
return $ct' = (c_0 + \sum_{P_i} h_i, c_1)$

Mouchet et al. *Multiparty Homomorphic Encryption from Ring-Learning-with-Errors*. PoPETs'21

Does the attack work against Threshold FHE schemes?

Algorithm 1: Collective Key Switch

Input: Ciphertext $ct = (c_0, c_1)$ of variance σ_{ct}^2
Private input: s_i, s'_i for each party P_i
Output: Key-switched ciphertext $ct' = (c'_0, c_1)$
Each *party* P_i
  Samples $e_i \leftarrow \chi_{CKS}(\sigma_{ct}^2)$ ▷ Smudging noise sampled from $\chi_{\mathrm{CKS}} = \mathcal{N}(0, 2^\lambda \sigma_{\mathrm{ct}}^2)$
  Computes and Discloses $h_i = (s_i - s'_i) \cdot c_1 + e_i$
return $ct' = (c_0 + \sum_{P_i} h_i, c_1)$

Mouchet et al. *Multiparty Homomorphic Encryption from Ring-Learning-with-Errors*. PoPETs'21

$$c_k^{(\mathrm{smg})} = (2^k a, \langle 2^k a, s\rangle + 2^k e + e_{\mathrm{smg}}) \quad \text{indistinguishable from} \quad c_k = (2^k a, \langle 2^k a, s\rangle + e_{\mathrm{smg}}),$$

$$\text{where } \sigma_{\mathrm{smg}} = \sigma_{\mathrm{ct}}\sqrt{K}2^{\frac{\lambda}{2}} \text{ and } \sigma_{\mathrm{ct}} = 2^k\sigma$$

Table of contents

1 Introduction & Background
- Homomorphic Encryption
- Security model and CPA^D game

2 A CPA^D attack on "exact" FHE schemes

3 Impact on Threshold FHE

4 Countermeasures for "exact" and Threshold Schemes

5 Conclusion and Key takeaways

Countermeasures

- **Bootstrapping** ($\sim$ 50% cost)
  - **Bootstrap** after each homomorphic operation
  - Since bootstrapping resets the noise variance to a preset value, decryption errors cannot occur.
  - Choose FHE parameters such that bootstrapping errors occur with prob neg(λ).
- **Monitor & Block** ($\sim$ 35% cost)
  - Fix a noise deviation budget B.
  - Choose FHE parameters such that decryption error occur with prob neg(λ) at noise dev. B.
  - **Monitor** (worst-case) noise deviation during FHE execution.
  - **Block**: return $\perp$ when noise deviation $>$ B.
- **Monitor & Smudge** ($\sim$ 45% cost)
  - Prior to decryption, **flood/smudge** the ciphertext with a large λ-dependent and σ_{ct}-dependent variance.
  - Works for threshold scheme (and must not be optional)

Table of contents

1. Introduction & Background
   - Homomorphic Encryption
   - Security model and CPA^D game
2. A CPA^D attack on “exact” FHE schemes
3. Impact on Threshold FHE
4. Countermeasures for “exact” and Threshold Schemes
5. Conclusion and Key takeaways

Concurrent works

- **Guo et al.**
  Key recovery attacks on approximate homomorphic encryption with non worst-case noise flooding countermeasures. Usenix Security 2024
  - CPA^D attack on CKKS, when smudging based on non worst-case noise estimation

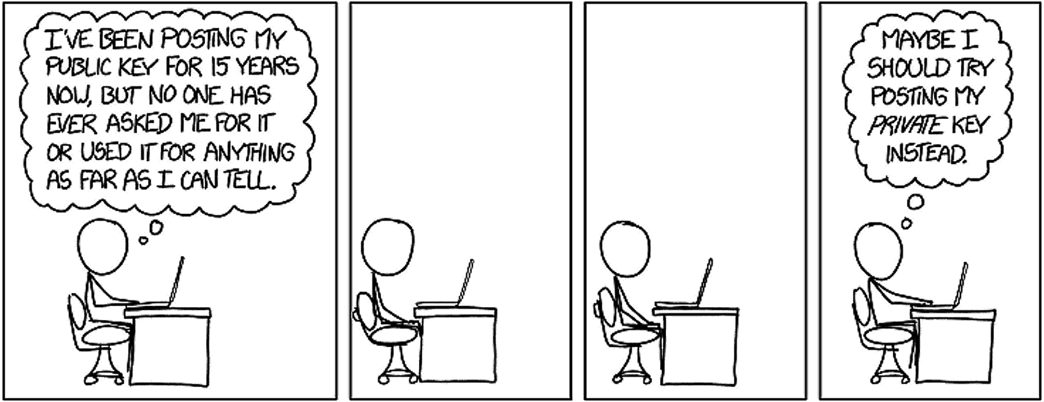
- **Cheon et al.**
  Attacks Against the IND-CPA^D Security of Exact FHE Schemes. IACR Eprint 2024/127
  - BGV/BFV CPA^D attack, migrate the noise polynomial in the plaintext domain
  - TFHE CPA^D attack, exploit bootstrapping error

- **Alexandru et al.**
  Application-aware approximate homomorphic encryption: configuring FHE for practical use. IACR Eprint 2024/203
  - Application-aware security: new *weaker* variant of CPA^D security
  - CPA^D security should be defined relative to a circuit class and a noise estimation strategy

Key Takeaways

- CPA^D is not just a theoretical threat, thus...
  CPA^D security must be carefully considered by all FHE schemes
- Simple CPA^D attacks can be implemented in most popular FHE libraries, but...
  Simple countermeasures can be devised, but have an impact on performance
- Recall that CPA^D is a natural security context in multi-user threshold FHE, so...
  Recall to have smudging appropriately implemented in your favorite threshold library

Thank you for your kind attention!



credit: xkcd.com

Generalization to RLWE

LWE

$$a \xleftarrow{\$} \mathbb{Z}_q,\, e \xleftarrow{\$} \chi(\mathbb{Z}_q)$$
$$c = (a, b := m + \langle a, s\rangle + e)$$

RLWE

$$A \xleftarrow{\$} \mathbb{Z}_q[X]/X^n{+}1,\, E \xleftarrow{\$} \chi(\mathbb{Z}_q[X]/X^n{+}1)$$
$$C = (A, B := M + A \cdot S + E)$$

➢ Just have to look at one coefficient of the RLWE polynomial: it is an LWE instance!

$$C = c_0 + \boxed{c_1} X + c_2 X + \cdots + c_{n-1} X^{n-1}$$

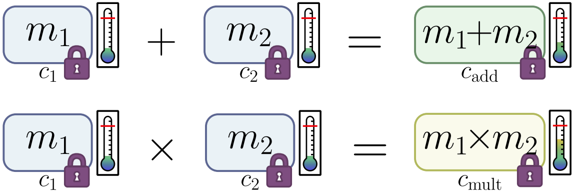
Bootstrapping

And boot...boot... What? Bootstrapping!

What is it and what for?

Bootstrapping (∼50% cost).

- Bootstrap after **each** homomorphic operation.
- Since bootstrapping resets the noise variance to a preset value, decryption errors cannot occur.
- Choose FHE parameters such that bootstrapping errors occur with prob neg(λ).



Noise grows with each homomorphic operations.
We need to regularly reduce the noise:
that's bootstrapping!

Monitor & Block

Monitor & Block (∼35% cost).

- Fix a noise deviation budget B.
- Choose FHE parameters such that decryption error occur with prob neg(λ) at noise dev. B.
- Monitor (worst-case) noise deviation during FHE execution.
- **Block** decryption when noise dev. $> B$.
- Scheme becomes "somewhat correct".

d	$\log_2(q)$	n	$\log_2(q)$	n	ratio
1	120	8192	131	8192	1,09
2	180	8192	181	8192	1,00
3	180	8192	237	16384	2,96
4	240	16384	289	16384	1,35
5	240	16384	341	16384	1,68
6	300	16384	392	16384	1,46
7	300	16384	444	16384	1,66
8	360	16384	516	32768	3,37
9	360	16384	570	32768	3,93
10	420	16384	624	32768	3,65

Illustration of the performance cost of the Monitor&Block countermeasure for OpenFHE/BFV.

Monitor & Smudge

Monitor & Smudge (∼45% cost).

- ➢ Prior to decryption, flood the ciphertext with a large λ-dependent and σ_{ct}-dependent variance.
- ➢ Works for threshold scheme
  $\hookrightarrow$ **must not be optional!**

d	$\log_2(q)$	n	$\log_2(q)$	n	ratio
1	120	8192	153	8192	1,28
2	180	8192	202	8192	1,12
3	180	8192	258	16384	3,22
4	240	16384	310	16384	1,45
5	240	16384	362	16384	1,79
6	300	16384	414	16384	1,55
7	300	16384	483	32768	3,99
8	360	16384	537	32768	3,70
9	360	16384	591	32768	4,30
10	420	16384	645	32768	3,95

Illustration of the performance cost of the Monitor&Smudge countermeasure for OpenFHE/BFV and K-out-of-K decryption, with $K = 5$.

Correctness and CPA^D

Correctness

A scheme is a correct/exact scheme if

$$\mathbf{P}\left(\mathsf{Dec}\left(\mathsf{Enc}(m,r)\right) \neq m\right) \leqslant \text{neg}(\lambda)$$

and

$$\mathbf{P}\left(\mathsf{Dec}\left(\mathsf{Eval}\left(f, \mathsf{Enc}(m_1, r_1), \ldots, \mathsf{Enc}(m_k, r_k)\right)\right) \neq f(m1, \ldots, m_k)\right) \leqslant \text{neg}(\lambda)$$

If the scheme is correct/exact, our attack is not applicable

Li & Micciancio, EUROCRYPT'21, **Lemma 1.**

"Any exact homomorphic encryption scheme $\mathscr{E}$ is IND-CPA secure if and only if it is IND-CPA^D secure."

CPA^D Security Game

Encryption scheme $\mathscr{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec}, \mathsf{Eval})$, plaintext domain $\mathscr{P}$ and security parameter λ. Adversary $\mathscr{A}$.

Game parameterized by $b^* \xleftarrow{\$} \{0, 1\}$ unknown to $\mathscr{A}$, and an initially empty state S of msg-msg-ctxt triplets:

- **Key generation.** Run $(\mathsf{ek}, \mathsf{dk}) \leftarrow \mathsf{KeyGen}(1^\lambda)$, and give ek to $\mathscr{A}$.
- **Encryption request.** $\mathscr{A}$ queries $(\texttt{test\_messages}, m_0, m_1)$, where $m_0, m_1 \in \mathscr{P}$.
  Compute $c = \mathsf{Enc}_{\mathsf{ek}}(m_{b^*})$, give c to $\mathscr{A}$ and do $S := [S; (m_0, m_1, c)]$.
- **Evaluation request.** $\mathscr{A}$ queries $(\texttt{eval}, f, l_1, \ldots, l_K)$.
  Compute $m'_0 = f(S[l_1].m_0, \ldots, S[l_K].m_0)$, $m'_1 = f(S[l_1].m_1, \ldots, S[l_K].m_1)$, and $c' = \mathsf{Eval}(f, S[l_1].c, \ldots, S[l_K].c)$. Update S as follows: $S := [S; (m'_0, m'_1, c')]$
- **Decryption request.** $\mathscr{A}$ queries $(\texttt{ciphertext}, l)$.
  If $S[l].m_0 \neq S[l].m_1$, return $\perp$. Otherwise return $\mathsf{Dec}_{\mathsf{dk}}(S[l].c)$.
- **Guessing stage.** $\mathscr{A}$ outputs $(\texttt{guess}, b)$.
  If $b = b^*$, $\mathscr{A}$ wins the game, otherwise $\mathscr{A}$ looses it.