ATTACKS AGAINST THE CPA-D SECURITY OF EXACT FHE SCHEMES

Damien Stehlé

May 25, 2024

Talk based on Eprint 2024/127
Joint work with J. H. Cheon, H. Choe, A. Passelègue & E. Suvanto

HEAAN CRYPTO LAB

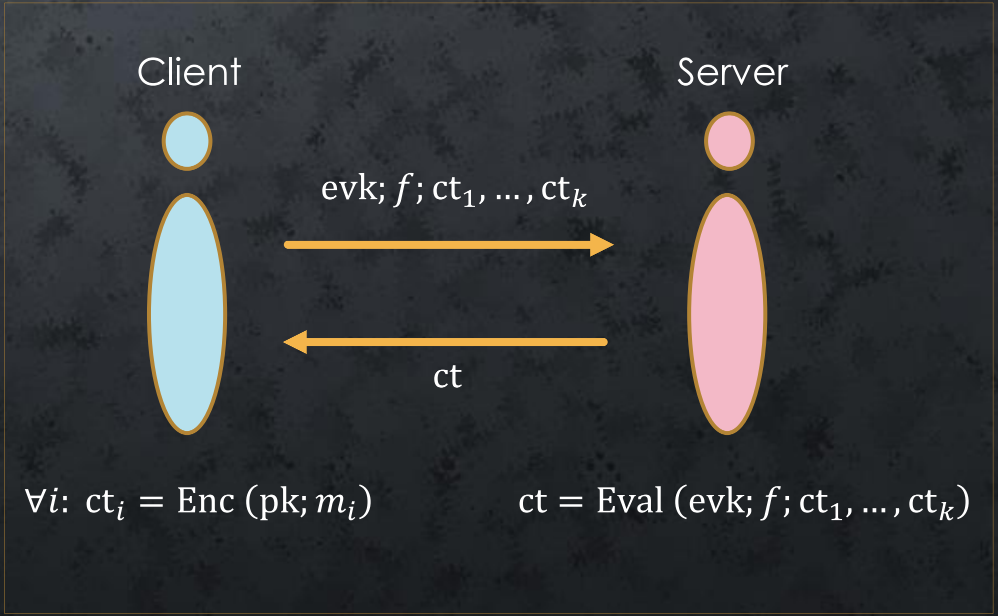
FULLY HOMOMORPHIC ENCRYPTION

An FHE scheme consists of $(\mathrm{KeyGen}, \mathrm{Enc}, \mathrm{Eval}, \mathrm{Dec})$:

- $\mathrm{KeyGen} \to (\mathrm{sk}, \mathrm{pk}, \mathrm{evk})$
- $\mathrm{Enc}\ (\mathrm{pk}; m) \to \mathrm{ct}$
- $\mathrm{Eval}\ (\mathrm{evk}; f; \mathrm{ct}_1, \ldots, \mathrm{ct}_k) \to \mathrm{ct}$
- $\mathrm{Dec}\ (\mathrm{sk}; \mathrm{ct}) \to m$

$$\forall f, m_1, \ldots, m_k :$$

$$\mathrm{Dec}\left(\mathrm{Eval}\left(f;\ \mathrm{Enc}(m_1), \ldots, \mathrm{Enc}(m_k)\right)\right) = f(m_1, \ldots, m_k)$$



MAIN FHE SCHEMES

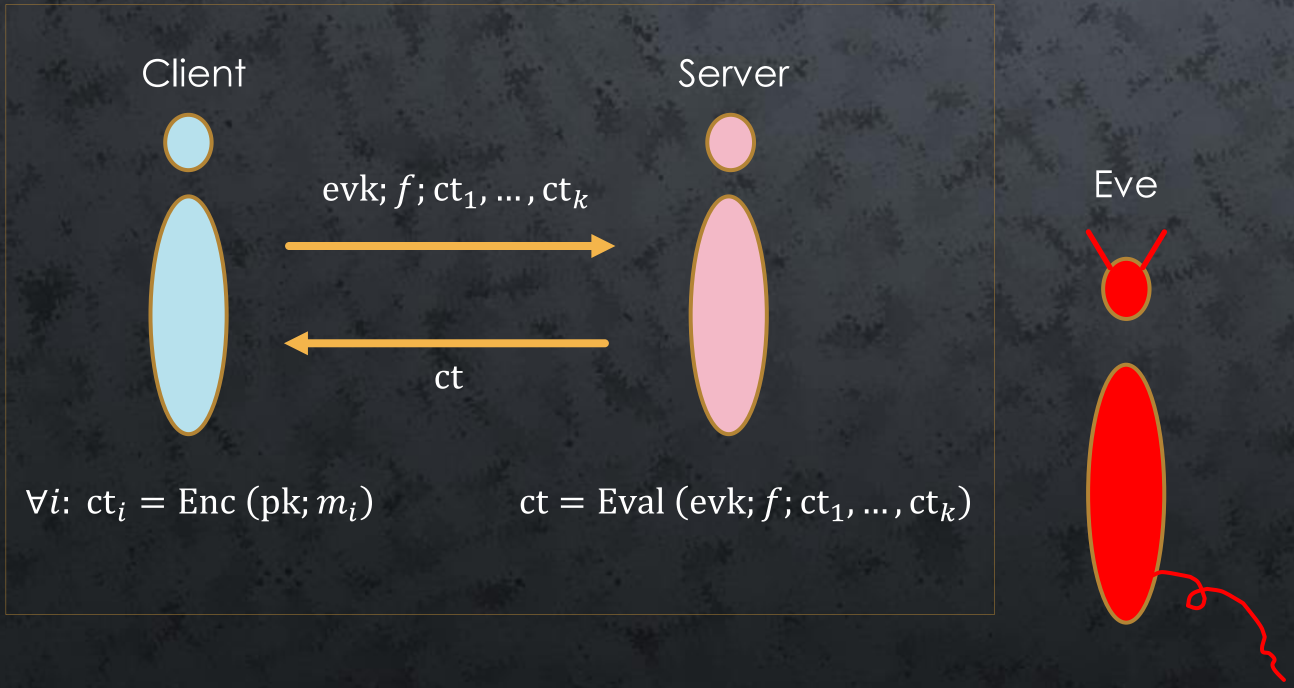
	Plaintext space	Basic operations	Ctxt format
BFV/BGV (2012)	$\left(\mathrm{F}_{p^k}\right)^{N/k}$	Add & Mult in // F_{p^k}-automorph. in // Slot rotate	RLWE
DM/CGGI (2015)	$\{0,1\}$	Binary gates	LWE (and RLWE internally)
CKKS (2017)	$\mathbb{C}^{N/2}$	Add & Mult in // Conj in // Slot rotate	RLWE

MAIN FHE SCHEMES

	Plaintext space	Basic operations	Ctxt format	
BFV/BGV (2012)	$\left(\mathbb{F}_{p^k}\right)^{N/k}$	Add & Mult in // $\mathbb{F}_{p^k}$-automorph. in // Slot rotate	RLWE	**EXACT**
DM/CGGI (2015)	$\{0,1\}$	Binary gates	LWE (and RLWE internally)	
CKKS (2017)	$\mathbb{C}^{N/2}$	Add & Mult in // Conj in // Slot rotate	RLWE	**APPROXIMATE** (there is an exact mode for CKKS, see you on Thursday)

$$\forall f, m_1, \ldots, m_k : \quad \mathrm{Dec}\left(\mathrm{Eval}\left(f;\ \mathrm{Enc}(m_1), \ldots, \mathrm{Enc}(m_k)\right)\right) \approx f(m_1, \ldots, m_k)$$

FHE SECURITY



IND-CPA security

one cannot distinguish between encryptions of two different plaintexts

B. Li, D. Micciancio: *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

IND-CPA-D SECURITY

IND-CPA security

one cannot distinguish between encryptions of two different plaintexts

IND-CPA-D security

Same, but the attacker may ask for decryption of ciphertexts for which it is supposed to know the underlying plaintext

Adversary has pk and evk

It can make queries:

- $\mathrm{Enc}\ (m) \rightarrow \mathrm{ct}$ // challenger knows the ptxts corresponding to all ctxts
- $\mathrm{ChallEnc}\ (m_0, m_1) \rightarrow \mathrm{ct}$ // challenge ctxts: m_b is encrypted
- $\mathrm{Eval}\ (\mathrm{evk}; f; \mathrm{ct}_1, \dots, \mathrm{ct}_k) \rightarrow \mathrm{ct}$ // for $\mathrm{ct}_1, \dots, \mathrm{ct}_k$ in the databasis
- $\mathrm{Dec}\ (\mathrm{sk}; \mathrm{ct}) \rightarrow m$ // for ct in the databasis **if the corresponding plaintext does not depend on b**

Adversary guesses b

THE TOPIC OF THIS TALK

B. Li, D. Micciancio: *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

IND-CPA security

one cannot distinguish between encryptions of two different plaintexts

IND-CPA-D security

Same, but the attacker may ask for decryption of ciphertexts for which it is supposed to know the underlying plaintext

"an approximate homomorphic encryption scheme can satisfy IND-CPA security and still be completely insecure"

"when applied to standard (exact) encryption schemes, IND-CPA-D is perfectly equivalent to IND-CPA"

CKKS is singled out as "insecure"

B. Li, D. Micciancio: *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

THE TOPIC OF THIS TALK

<u>IND-CPA security</u>

one cannot distinguish between encryptions of two different plaintexts

<u>IND-CPA-D security</u>

Same, but the attacker may ask for decryption of ciphertexts for which it is supposed to know the underlying plaintext

"an approximate homomorphic encryption scheme can satisfy IND-CPA security and still be completely insecure"

What does it mean?

"when applied to standard (exact) encryption schemes, IND-CPA-D is perfectly equivalent to IND-CPA"

Exact data?
Correct?
Heuristically?
Which error probability?

B. Li, D. Micciancio: *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

THE TOPIC OF THIS TALK

IND-CPA security

one cannot distinguish between encryptions of two different plaintexts

IND-CPA-D security

Same, but the attacker may ask for decryption of ciphertexts for which it is supposed to know the underlying plaintext

IND-CPA-D attacks on exact schemes

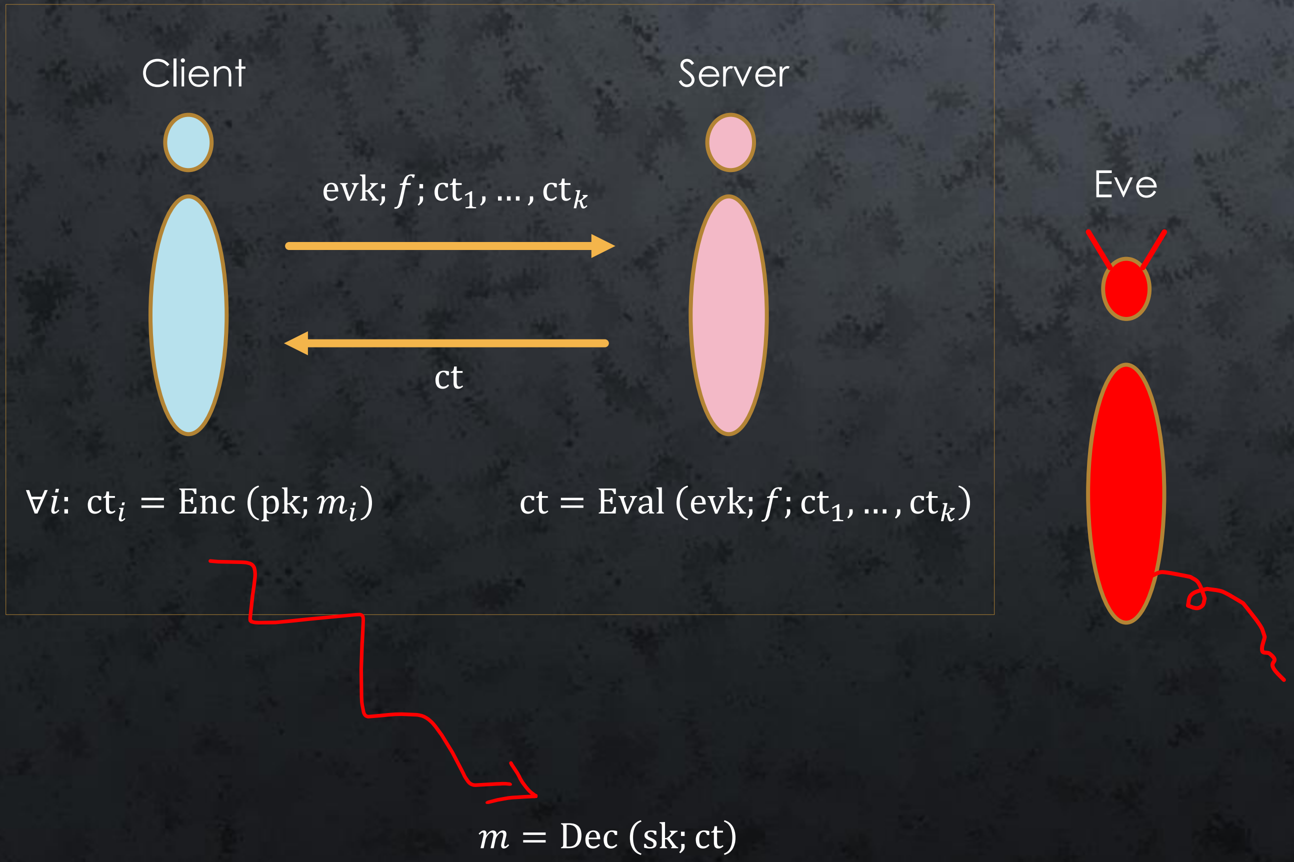
BGV / BFV
DM / CGGI
(Exact) CKKS

~~**"when applied to standard (exact) encryption schemes, IND-CPA-D is perfectly equivalent to IND-CPA"**~~

CKKS shouldn't be singled out

J. H. Cheon, S. Hong, D. Kim: *Remark on the security of CKKS scheme in practice*. Eprint 2020/1581

HOW RELEVANT IS IND-CPA-D SECURITY?

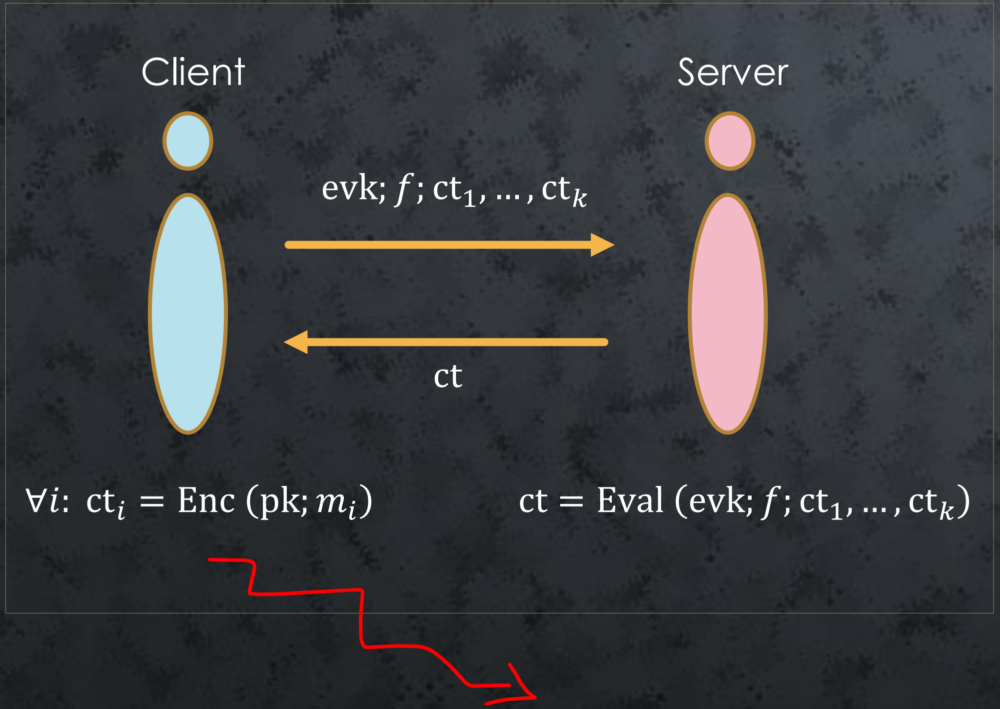


IND-CPA-D security

Same, but the attacker may ask for decryption of ciphertexts for which it is supposed to know the underlying plaintext

If the computation is **confidential**, why would the client make the output of a confidential computation **public**?

HOW RELEVANT IS IND-CPA-D SECURITY?

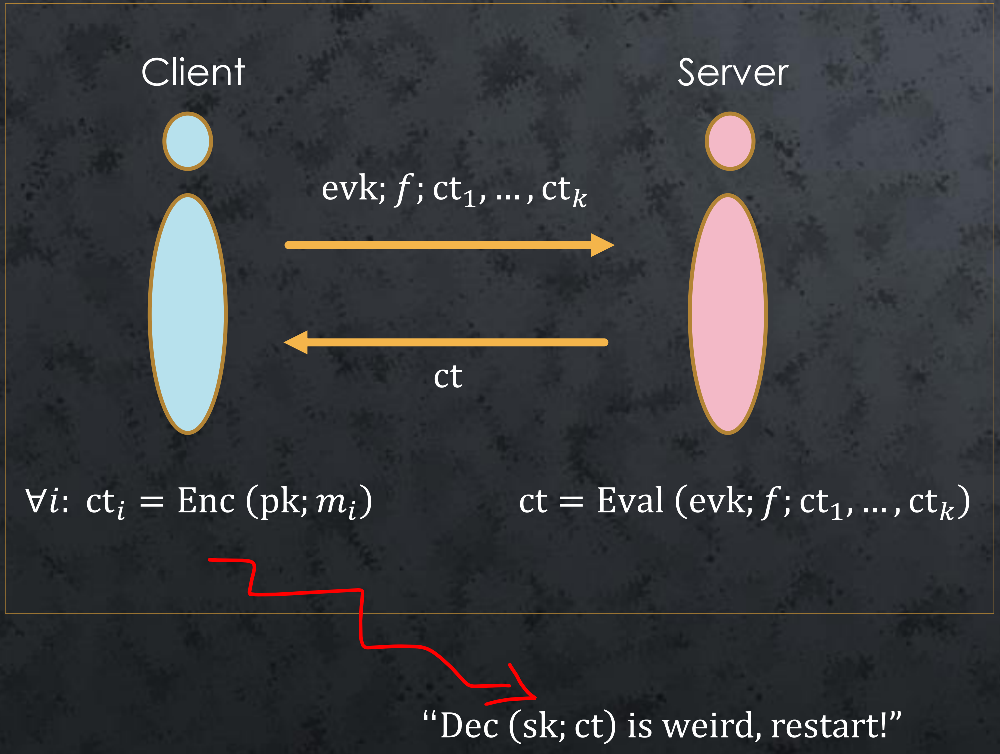


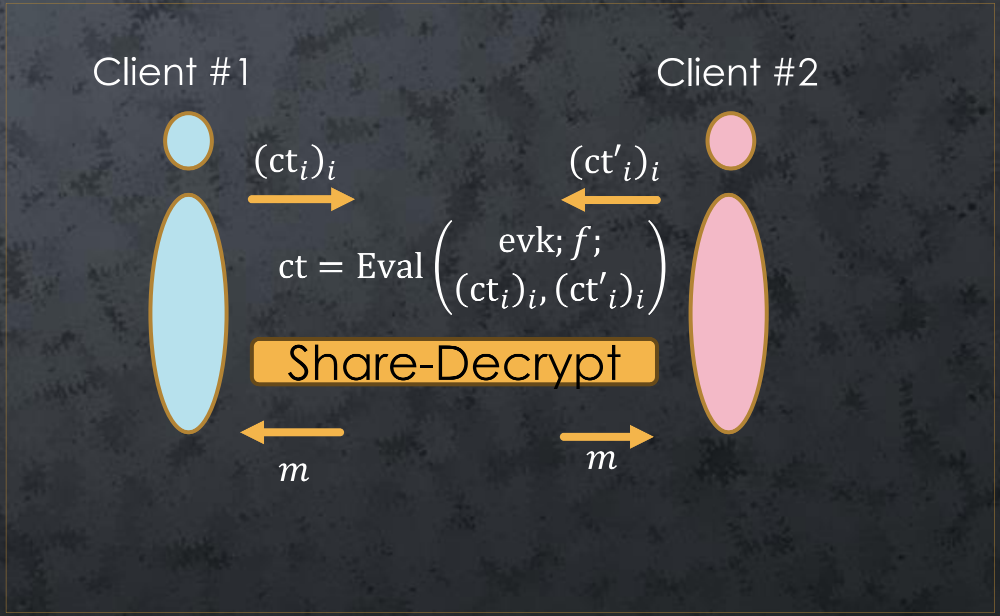
"Dec (sk; ct) is weird, restart!"

Weak variant of CVA security

If the result is weird,
the client could ask to redo the computation

HOW RELEVANT IS IND-CPA-D SECURITY?





Weak variant of CVA security

If the result is weird,
the client could ask to redo the computation

Threshold FHE

sk is shared across several clients
they collaborate to decrypt
and they all get to know the result

ROADMAP

1- Motivation

2- Attacks against CKKS

3- IND-CPA-D versus IND-CPA for exact schemes

4- An attack against BFV/BGV addition

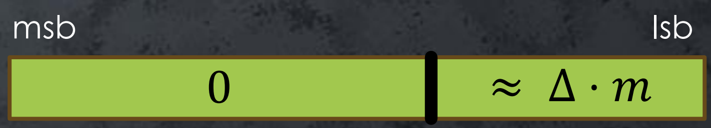
5- Attacks against bootstrapping algorithms

6- Concluding remarks

REMINDERS ON CKKS

Plaintext space: vectors of $\mathbb{C}^{N/2}$ (up to some precision)

- add in //
- multiply in //



A **ciphertext** is of the form $(a, b) \in R_q^2$ s.t. $a \cdot s + b \approx \Delta \cdot m$

- $s \in R_q$ is the secret key
- m is the (encoded) plaintext
- Δ is the scaling factor (precision)
- $R_q = \mathbb{Z}_q[x] / x^N + 1$

To **decrypt**: $(a, b) \mapsto (a \cdot s + b \mod q) / \Delta$

THE LI-MICCIANCIO ATTACK

To decrypt: $(a, b) \mapsto (a \cdot s + b \mod q) \,/\, \Delta$

Encrypt 0 and decrypt it:

=> We know (a, b) and $a \cdot s + b \mod q$
=> This reveals s

Key recovery

A COUNTERMEASURE

B. Li, D. Micciancio, M. Schultz, J. Sorrell: *Securing approximate homomorphic encryption using differential privacy*. CRYPTO'22

Noise flooding: $(a, b) \mapsto (a \cdot s + b \mod q) \,/\, \Delta + e$

1- Bound the contributions of all errors (due to encryption and evaluation), for all possible inputs

2- Add to the decrypted value a noise e that is $\geq 2^{\lambda/2}$ larger

Security

The output is simulatable from the knowledge of the expected ptxt

NECESSITY OF LARGE FLOODING

B. Li, D. Micciancio, M. Schultz, J. Sorrell: *Securing approximate homomorphic encryption using differential privacy*. CRYPTO'22

Noise flooding: $(a, b) \mapsto (a \cdot s + b \mod q) \,/\, \Delta + e$

If the noise is smaller, then there is an attack

$$f: \quad x_1, \dots, x_{2k} \quad \mapsto \quad x_1^2 + \cdots + x_k^2 - x_{k+1}^2 - \cdots - x_{2k}^2$$

$(0, \dots, 0)$ and $(1, \dots, 1)$ give the same result
But the noise for $(1, \dots, 1)$ is larger

(multiplication noise grows with plaintext)

If the flooding is too small, we can distinguish

Distinguishing attack

ROADMAP

1- Motivation

2- Attacks against CKKS

3- IND-CPA-D versus CPA-D for exact schemes

4- An attack against BFV/BGV addition

5- Attacks against bootstrapping algorithms

6- Concluding remarks

Passive Security

- IND-CPA security is typically sufficient to achieve passive security (for data privacy) for **exact** FHE schemes, including BGV, BFV, DM, and CGGI
- IND-CPA security is not sufficient for **approximate** FHE schemes
  - Li and Micciancio showed that CKKS is not secure if access to a decryption oracle is provided, i.e., when the decryption result is shared with parties that do not have the secret key [LM21]
  - They proposed a new definition IND-CPA^D that provides access to encryption, evaluation, and decryption oracles

(Borrowed from a talk by Y. Polyakov, given at NIST)

B. Li, D. Micciancio: *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

CPA / CPA-D

Assume the scheme is exact

The decryption queries do not help the adversary:

For any valid decryption query (i.e., the corresponding ptxt does not depend on the challenge b), the adversary already knows the underlying ptxt

B. Li, D. Micciancio: *On the security of homomorphic encryption on approximate numbers*. EUROCRYPT'21

CPA / CPA-D

Assume the scheme is exact

The decryption queries do not help the adversary:

For any valid decryption query (i.e., the corresponding ptxt does not depend on the challenge b), the adversary already knows the underlying ptxt

Caveat
The above requires perfect correctness

Let p_{err} be the maximum over all $f, m_1, \ldots, m_k$ of the probability that

$$\mathrm{Dec}\left(\mathrm{Eval}\left(f;\ \mathrm{Enc}(m_1), \ldots, \mathrm{Enc}(m_k)\right)\right) \neq f(m_1, \ldots, m_k)$$

The equivalency still holds if p_{err} **is extremely small**

(SEMI-)GENERIC ATTACK FOR INCORRECT SCHEMES

Let p_{err} be the maximum over all $f, m_1, \dots, m_k$ of the probability that

$$\text{Dec}\left(\text{Eval}\left(f;\ \text{Enc}(m_1), \dots, \text{Enc}(m_k)\right)\right) \neq f(m_1, \dots, m_k)$$

Assume that the adversary knows $f, m_1, \dots, m_k,\ m'_1, \dots, m'_k$ s.t.

- $f, m_1, \dots, m_k$ reaches p_{err}
- $f, m'_1, \dots, m'_k$ has much lower decryption error
- $f(m_1, \dots, m_k) = f(m'_1, \dots, m'_k)$

Then:

- request encryptions of $m_1, \dots, m_k$ $(b = 0)$ or $m'_1, \dots, m'_k$ $(b = 1)$
- request evaluation of f
- request decryption

If there is an error, it's more likely that $m_1, \dots, m_k$ were encrypted

Distinguishing attack

CORRECTNESS IN PRACTICE

In practice (all / most libraries):

- Failure probability from 2^{-15} to 2^{-50}
- It is derived from heuristic error analysis (probabilities without randomness)

Why?

1) Leads to more efficient schemes
2) For the primary use-case of FHE, IND-CPA (passive) security suffices

Next: how to exploit decryption errors to mount IND-CPA-D attacks on exact schemes!

ROADMAP

1- Motivation

2- Attacks against CKKS

3- IND-CPA-D versus IND-CPA for exact schemes

4- An attack against BFV/BGV addition

5- Attacks against DM/CGGI bootstrapping algorithms

6- Concluding remarks

REMINDERS ON BFV

Plaintext space: elements of $R_p = \mathbb{Z}_p[x] \,/\, x^N + 1$

- add in //

A **ciphertext** is of the form $(a, b) \in R_q^2$ s.t. $a \cdot s + b = \left(\frac{q}{p}\right) \cdot m + e$

- $s \in R_q$ is the secret key
- m is the plaintext
- e is the error
- $R_q = \mathbb{Z}_q[x] \,/\, x^N + 1$

To **decrypt**: $(a, b) \mapsto \left\lfloor (a \cdot s + b \mod q) \,/\, \left(\frac{\mathrm{q}}{\mathrm{p}}\right) \right\rceil$

AN ATTACK ON BFV

Theory

To get correctness,
bound the contributions of all errors
for all possible inputs

Practice (sometimes)

Use heuristic bounds

$$\mathrm{Noise}(\mathrm{ct}_1 + \mathrm{ct}_2) \approx \sqrt{\mathrm{Noise}(\mathrm{ct}_1)^2 + \mathrm{Noise}(\mathrm{ct}_2)^2}$$

AN ATTACK ON BFV

Theory

To get correctness,
bound the contributions of all errors
for all possible inputs

Practice (sometimes)

Use heuristic bounds

$$\mathrm{Noise}(\mathrm{ct}_1 + \mathrm{ct}_2) \approx \sqrt{\mathrm{Noise}(\mathrm{ct}_1)^2 + \mathrm{Noise}(\mathrm{ct}_2)^2}$$

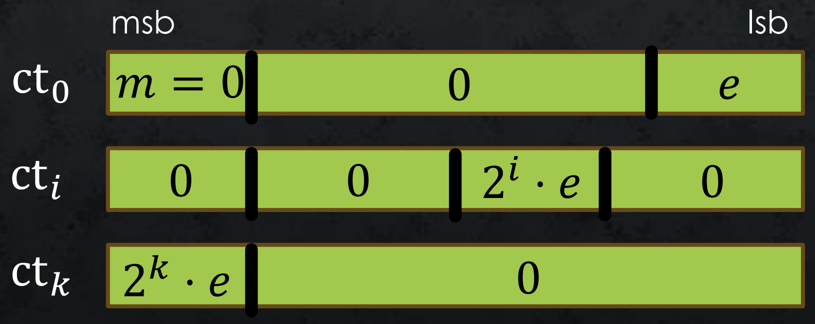
For $i = 1 \ldots k$: $x_{i+1} \leftarrow x_i + x_i$

Estimate noise $\approx 2^{k/2}$

=> The computation is deemed legitimate

Real noise $\approx 2^k$

Start with $\mathrm{ct} = \mathrm{Enc}(0)$



Key recovery

AN ATTACK ON BFV

Q. Guo, D. Nabokov, E. Suvanto, T. Johansson: *Key recovery attacks on approximate homomorphic encryption with non-worst-case noise flooding countermeasures.* USENIX'24

M. Checri, R. Sirdey, A. Boudguiga, J.-P. Bultel: On the practical CPAD security of "exact" and threshold FHE schemes and libraries. Eprint 2024/116

Adaptation of [GNSJ24] to BFV

Concurrently obtained in [CSBB24]

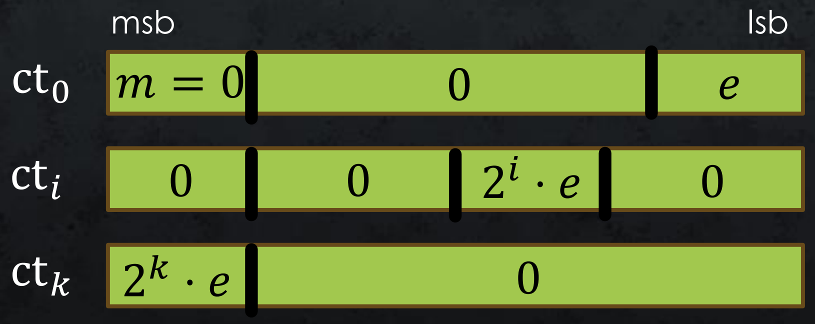
For $i = 1 \ldots k$: $x_{i+1} \leftarrow x_i + x_i$

Estimate noise $\approx 2^{k/2}$

=> The computation is deemed legitimate

Real noise $\approx 2^k$

Start with $\mathrm{ct} = \mathrm{Enc}(0)$



DOES IT WORK ON OPENFHE?

OpenFHE:

- claims to get IND-CPA-D security for CKKS,
- Has measures in place for correctness of exact schemes.

DOES IT WORK ON OPENFHE?

OpenFHE:

- claims to get IND-CPA-D security for CKKS,
- Has measures in place for correctness of exact schemes.

We tested the attack on **OpenFHE**'s BFV,

With: $N = 2^{12}$, $p = 2^{16} + 1$, $q = 2^{60}$, $\sigma \approx 2^{7.41}$

Start with an encryption of 0, and iterate $k = 44$ times

Estimated error probability $\approx 2^{-2^{27.5}}$

But decryption gives the initial noise, and we recover s

Only additions => attack is instantaneous

WHY DOES IT WORK ON OPENFHE?

Practice (sometimes)

Heuristic bounds

$$\mathrm{Noise}(\mathrm{ct}_1 + \mathrm{ct}_2) \approx \sqrt{\mathrm{Noise}(\mathrm{ct}_1)^2 + \mathrm{Noise}(\mathrm{ct}_2)^2}$$

OpenFHE

Triangular inequality

$$\mathrm{Noise}(\mathrm{ct}_1 + \mathrm{ct}_2) \leq \mathrm{Noise}(\mathrm{ct}_1) + \mathrm{Noise}(\mathrm{ct}_2)$$

But the attack **does** succeed!

WHY DOES IT WORK ON OPENFHE?

Practice (sometimes)

Heuristic bounds

$\mathrm{Noise}(\mathrm{ct}_1 + \mathrm{ct}_2) \approx \sqrt{\mathrm{Noise}(\mathrm{ct}_1)^2 + \mathrm{Noise}(\mathrm{ct}_2)^2}$

OpenFHE

Triangular inequality

$\mathrm{Noise}(\mathrm{ct}_1 + \mathrm{ct}_2) \leq \mathrm{Noise}(\mathrm{ct}_1) + \mathrm{Noise}(\mathrm{ct}_2)$

But the attack **does** succeed!

There is an error in the handling of addition error bounds in OpenFHE.

For k additions, OpenFHE multiplies the error by k.

For $i = 1 \ldots k$: $x_{i+1} \leftarrow x_i + x_i$ k additions but error grows as 2^k

ROADMAP

1- Motivation
2- Attacks against CKKS
3- IND-CPA-D versus IND-CPA for exact schemes
4- An attack against BFV/BGV addition
5- **Attacks against bootstrapping algorithms**
6- Concluding remarks

REMINDERS ON DM/CGGI

Plaintext space: elements of $\{0,1\}$

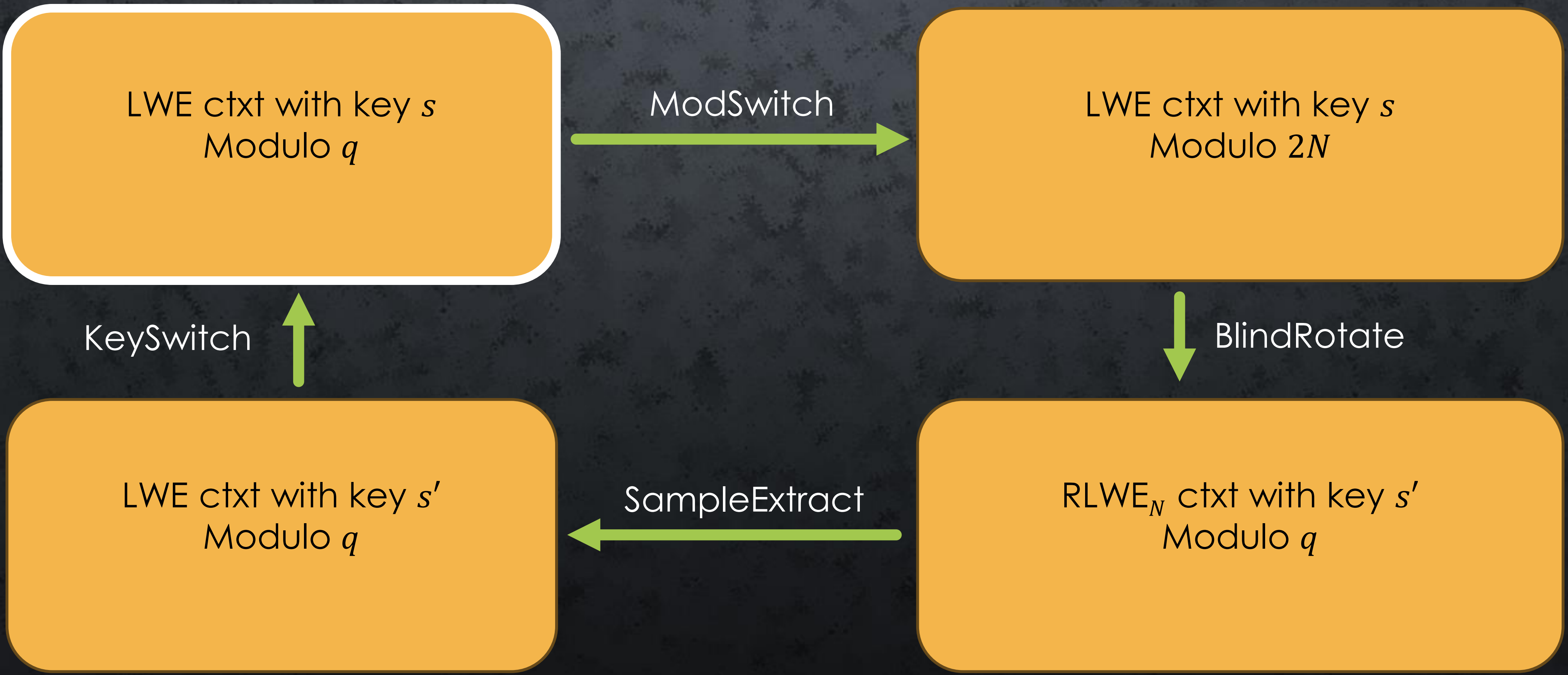
- Binary gates

A **ciphertext** is of the form $(a, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ s.t. $\langle a, s\rangle + b = \left(\frac{q}{8}\right) \cdot m + e$

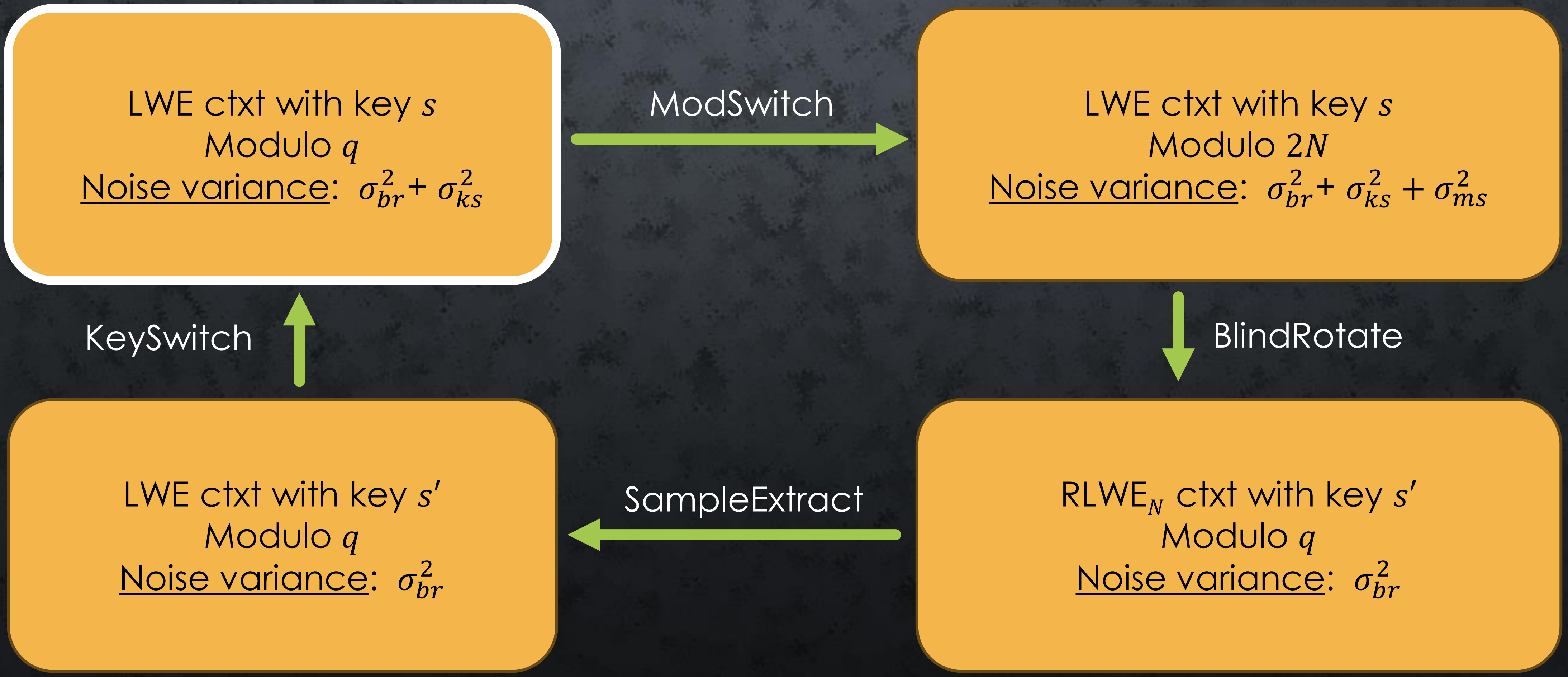
- $s \in \mathbb{Z}_q^n$ is the secret key
- m is the plaintext bit
- e is the error

To **decrypt**: $(a, b) \mapsto \left\lfloor (\langle a, s\rangle + b \mod q) \,/\, \left(\frac{q}{8}\right) \right\rceil$

DM/CGGI BOOTSTRAPPING



DM/CGGI BOOTSTRAPPING



DM/CGGI GATE BOOTSTRAPPING

Two LWE ctxts with key s
Modulo q
Noise variance: $\sigma_{br}^2 + \sigma_{ks}^2$

Add and ModSwitch

LWE ctxt with key s
Modulo $2N$
Noise variance: $4\sigma_{br}^2 + 4\sigma_{ks}^2 + \sigma_{ms}^2$

KeySwitch

BlindRotate

EXPLOITING DECRYPTION ERROR

Add and ModSwitch →

LWE ctxt with key s
Modulo $2N$
Noise variance: $4\sigma_{br}^2 + 4\sigma_{ks}^2 + \sigma_{ms}^2$

↓ BlindRotate

- Gate bootstrapping fails
if the noise spills over the ptxt

- After ModSwitch is where noise is largest

- If gate bootstrapping fails,
then the ModSwitch error must be large

EXPLOITING MODSWITCH ERROR

ModSwitch: $\mathrm{ct} \mod q \;\mapsto\; \mathrm{ct}' = \left\lfloor \left(\frac{2N}{q}\right) \cdot \mathrm{ct} \right\rceil \mod 2N$

$\langle \mathrm{ct}, \mathrm{sk} \rangle = e \;\Rightarrow\; \langle \mathrm{ct}', \mathrm{sk} \rangle = \langle e_{\mathrm{rnd}}, \mathrm{sk} \rangle + e$, where e_{rnd} is known

A failure tells that $\langle e_{\mathrm{rnd}}, \mathrm{sk} \rangle + e \geq \frac{2N}{16}$, for a known e_{rnd}

Attack can be completed with statistical analysis

IN PRACTICE

M. Dahl, D. Demmler, S. E. Kazdadi, A. Meyre, J.-B. Orfila, D. Rotaru, N. P. Smart, S. Tap, M. Walter: *Noah's ark: efficient threshold-FHE using noise flooding*. WAHC'23

We considered Zama's TFHE-rs

- For the default parameters, decryption error probability is $\approx 2^{-40}$
- We simulated that 256 decryption errors suffices
- Mounting the attack would take $\approx 2^{16}$ CPU years

- There are parameter sets with much poorer correctness
- The attack extends the [DDK+23] threshold-FHE scheme

AN ATTACK ON CKKS BOOTSTRAPPING

CKKS BTS has 4 steps:

1. S2C
2. ModRaise
3. C2S
4. EvalMod

AN ATTACK ON CKKS BOOTSTRAPPING

CKKS BTS has 4 steps:

1. S2C
2. ModRaise
3. C2S
4. EvalMod

Polynomial approximation to the mod-1 function, over a given number $2K + 1$ of periods.

- Higher K => more costly
- Smaller K => higher probability of error

AN ATTACK ON CKKS BOOTSTRAPPING

CKKS BTS has 4 steps:

1. S2C
2. ModRaise
3. C2S
4. EvalMod

Polynomial approximation to the mod-1 function, over a given number $2K+1$ of periods.

- Higher K => more costly
- Smaller K => higher probability of error

Input of EvalMod is not in the approximation range => Output is nonsense

When that happens, we have an equation

$\langle x, \mathbf{sk} \rangle + \mathbf{e} \geq \boldsymbol{bound}$, **where** x **is known.**

(like the DM/CGGI attack)

Example: OpenFHE
(claims INDCPA-D security for CKKS)

Probability of error ranges from 2^{-22} to 2^{-57}

ROADMAP

1- Motivation
2- Attacks against CKKS
3- IND-CPA-D versus IND-CPA for exact schemes
4- An attack against BFV/BGV
5- An attack against DM/CGGI
6- **Concluding remarks**

TAKE-AWAY

IND-CPA security:

one cannot distinguish between encryptions of two different plaintexts

IND-CPA-D security:

Same, but the attacker may ask for decryption of ciphertexts for which it is supposed to know the underlying plaintext

IND-CPA-D attacks on exact schemes

BGV / BFV
DM / CGGI
(Exact) CKKS

~~**"when applied to standard (exact) encryption schemes, IND-CPA-D is perfectly equivalent to IND-CPA"**~~

All competitive FHE schemes can suffer from IND-CPA-D attacks

ATTACKS OF DIFFERENT NATURES

Attack	Scheme	Decryption oracle or validity oracle?	Key recovery or distinguishing?
[LM21]	CKKS	Decryption	Key recovery
[LMSS22]	CKKS with limited decryption noise	Decryption	Distinguishing
[GNST24]	CKKS with heuristic error analysis	Decryption	Key recovery
Our work	FHE with imperfect correctness	Validity oracle	Distinguishing
Our work & [CSBB24]	BFV/BGV with heuristic error analysis	Can be adapted to validity oracle	Key recovery
Our work	DM/CGGI with large decryption error	Validity oracle	Key recovery
Our work	Exact CKKS	Validity oracle	Key recovery

ATTACKS OF DIFFERENT NATURES

Attack	Scheme	Decryption oracle or validity oracle?	Key recovery or distinguishing?
[LM21]	CKKS	Decryption	Key recovery
[LMSS22]	CKKS with limited decryption noise	Decryption	Distinguishing
[GNST24]	CKKS with heuristic error analysis	Decryption	Key recovery
Our work	FHE with imperfect correctness	Validity oracle	Distinguishing
Our work & [CSBB24]	BFV/BGV with heuristic error analysis	Can be adapted to validity oracle	Key recovery
Our work	DM/CGGI with large decryption error	Validity oracle	Key recovery
Our work	Exact CKKS	Validity oracle	Key recovery

The situation is arguably worse for exact schemes!

COUNTERMEASURES

For all schemes:

- **tiny failure probability**
- **no heuristic** noise analysis

For (approximate) CKKS:

- **high-precision** computation
- followed by **noise flooding**

} efficiency

COUNTERMEASURES

For all schemes:

- **tiny failure probability**
- **no heuristic** noise analysis

For (approximate) CKKS:

- **high-precision** computation
- followed by **noise flooding**

} efficiency

And be very diligent with the implementation:

- IND-CPA: be cautious about **KeyGen & Enc**
- IND-CPA-D: be cautious about **KeyGen, Enc, Eval & Dec**

QUESTIONS?