Attacks Against the IND-CPA^D Security of Exact FHE Schemes

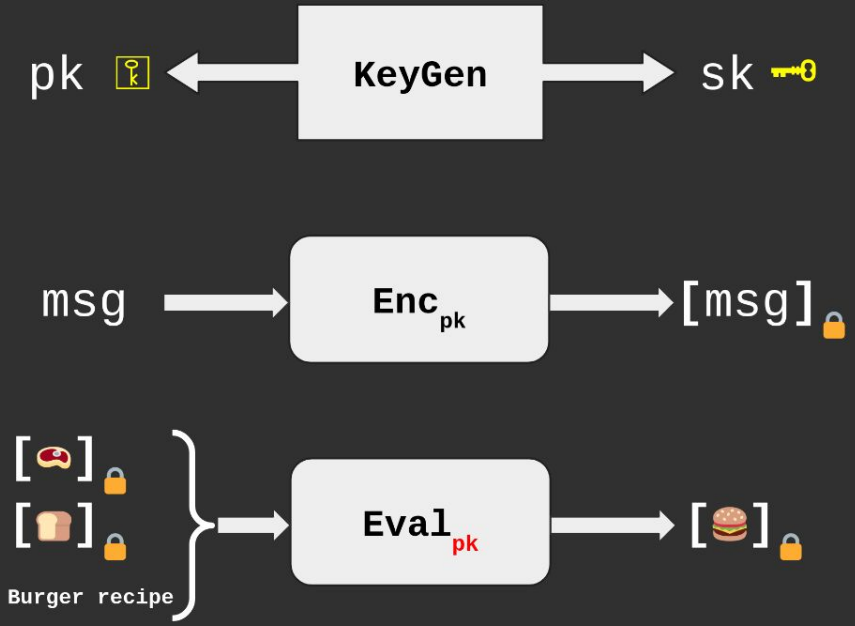
Jung Hee Cheon[12], Hyeongmin Choe[2], Alain Passelègue[1], Damien Stehlé[1], **Elias Suvanto**[13]

[1]CryptoLab Inc. [2]Seoul National University [3]University of Luxembourg

eprint.iacr.org/2024/127

Homomorphic encryption

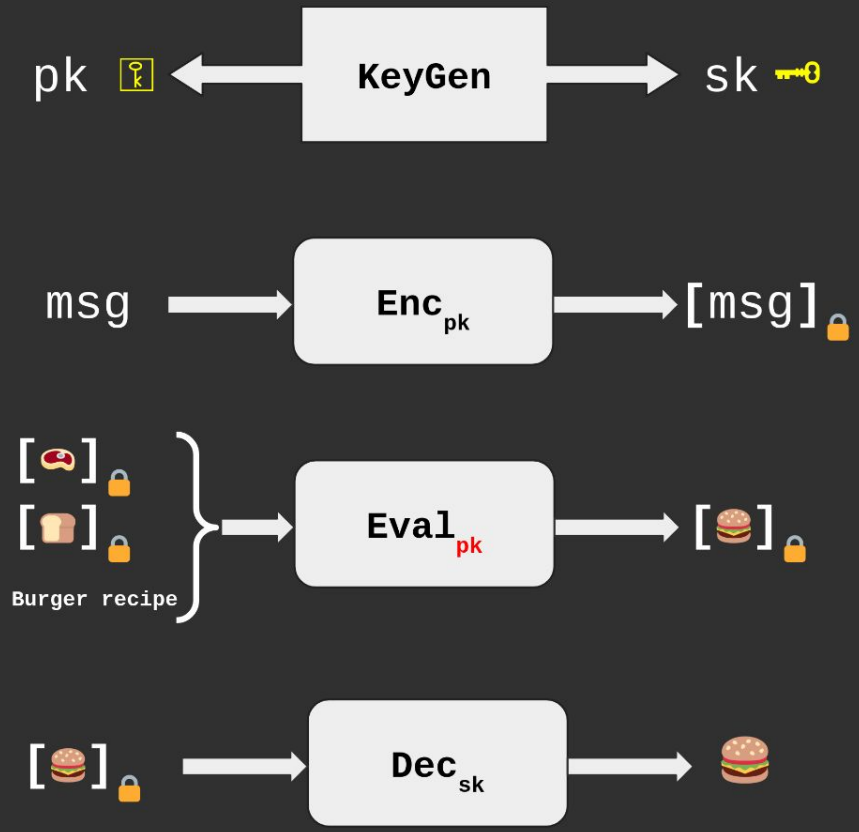
IND-CPA security:
The [*]🔒 do not leak any information about msg, 🥩, 🍞 and 🍔



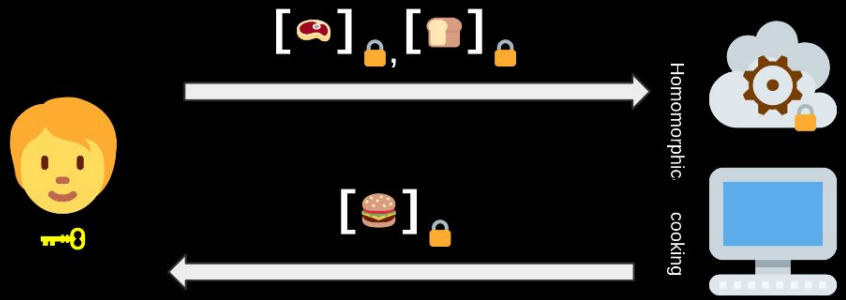
Homomorphic encryption

IND-CPA security:
The [*]🔒 do not leak any information about msg, 🥩, 🍞 and 🍔

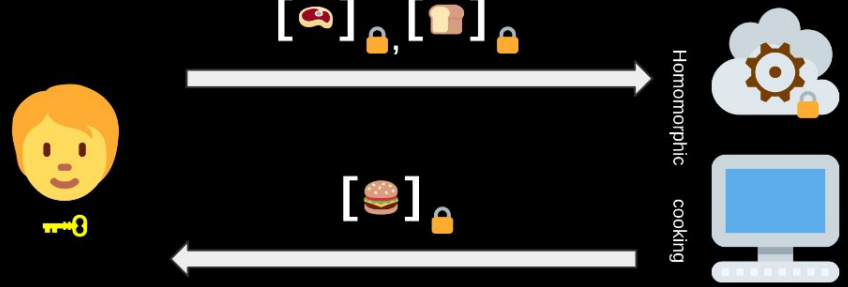
IND-CPA^D security:
Even if 🍔 is shared,
the [*]🔒 do not leak any additional information



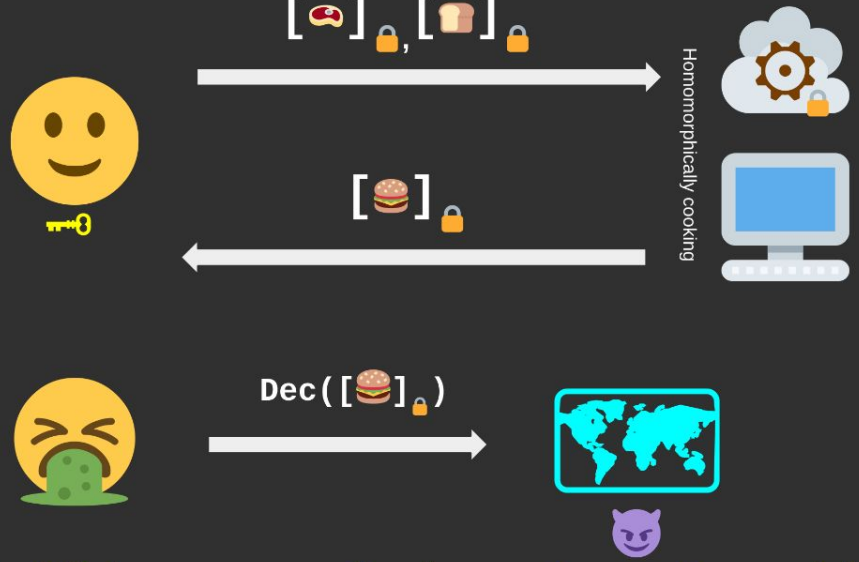
Secure outsourced computation



Secure outsourced computation



Secure outsourced computation with feedback



⚠ This scenario is not captured by IND-CPA security

Main FHE schemes

	Plaintext space	Basic operations	Ctxt format
BFV/BGV (2012)	$\left(\mathbb{F}_p^k\right)^{N/k}$	Arithmetic	RLWE
DM/CGGI (2015)	{0, 1}	Boolean gates	LWE
CKKS (2017)	$\mathbb{C}^{N/2}$	Arithmetic	RLWE

EXACT

APPROXIMATE

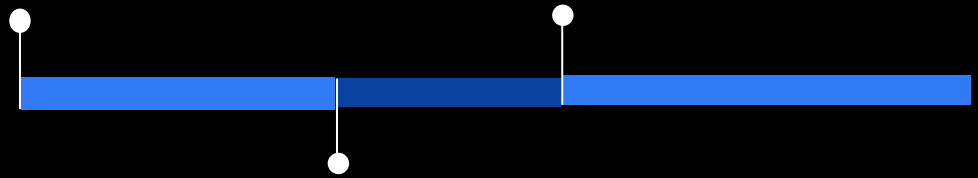
Previous works

EUROCRYPT'21
[LM21]

📜 IND-CPA^D definition
📜 **IND-CPA^D = IND-CPA for exact FHE schemes**

USENIX SECURITY'24
[GNSJ24]

📜 Attack against CKKS with noise-flooding



CRYPTO'22
[LMSS22]

📜 CKKS with noise-flooding

Previous works

EUROCRYPT'21
[LM21]

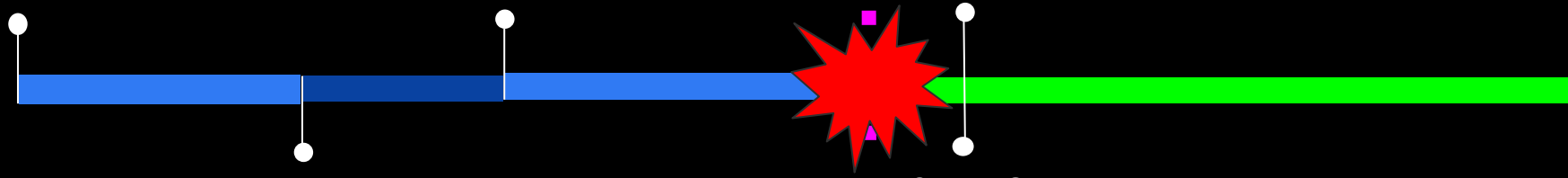
📜 IND-CPA^D definition
📜 **IND-CPA^D = IND-CPA for exact FHE schemes**

USENIX SECURITY'24
[GNSJ24]

📜 Attack against CKKS$^{\text{with noise-flooding}}$

CCS'24
This work

📜 **Non-adaptive key-recoveries against exact FHE schemes (BFV, CGGI, discrete CKKS)**
📜 **Generic CPA^D distinguisher**



CRYPTO'22
[LMSS22]

📜 CKKS$^{\text{with noise-flooding}}$

CRYPTO'24
[CSBB24]

📜 **Adaptive key-recoveries against BFV/BGV**

Our attacks

	Plaintext space	IND-CPA^D status belief	In many libraries	Reasons
BFV/BGV (2012)	$\left(\mathbb{F}_p^k\right)^{N/k}$	✅		
DM/CGGI (2015)	small integers	✅		
discrete-CKKS (2024)	small integers	✅		
CKKS (2017)	$\mathbb{C}^{N/2}$	❌		
CKKS (+ noise flooding)	$\mathbb{C}^{N/2}$	✅		

Our attacks

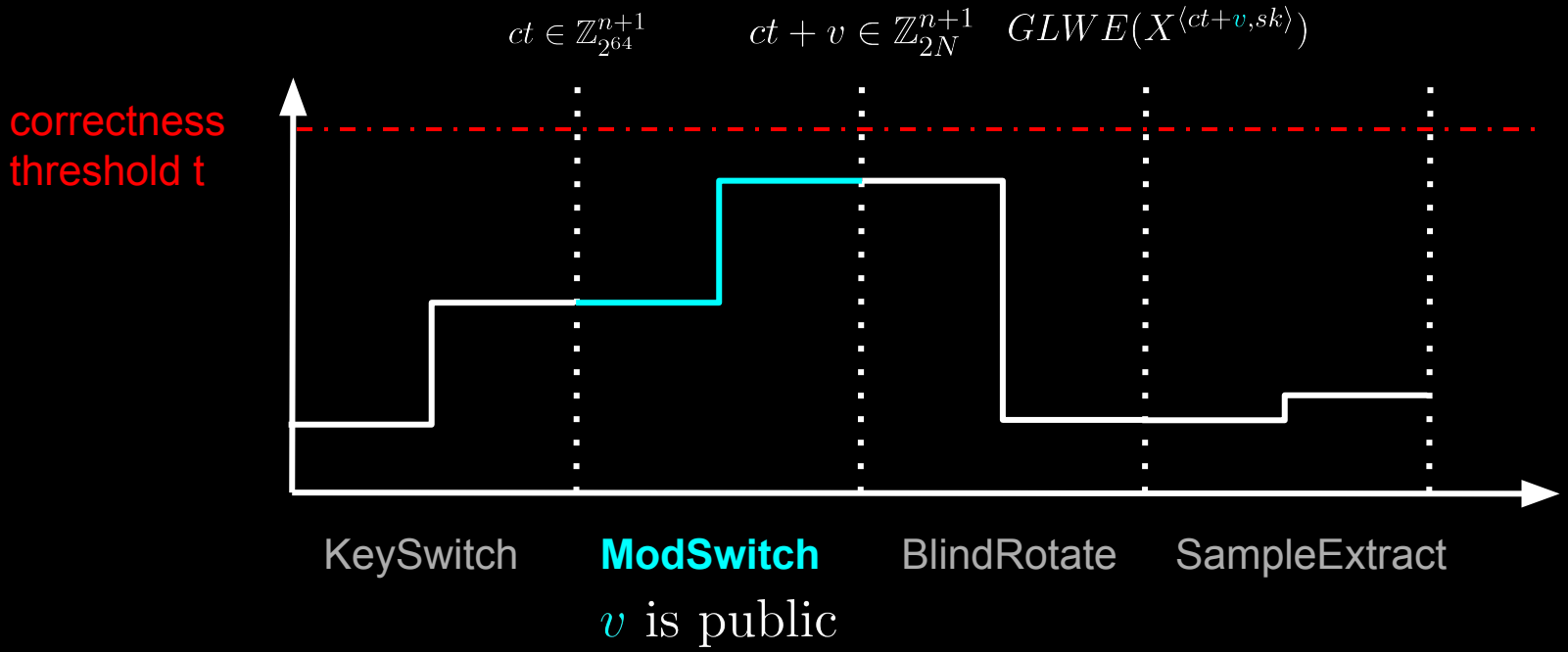
	Plaintext space	IND-CPA^D status belief	In many libraries	Reasons
BFV/BGV (2012)	$\left(\mathbb{F}_p^k\right)^{N/k}$	✅	❌	Incorrect noise upper bound
DM/CGGI (2015)	small integers	✅	❌	High failure probability
discrete-CKKS (2024)	small integers	✅	❌	High failure probability
CKKS (2017)	$\mathbb{C}^{N/2}$	❌		
CKKS (+ noise flooding)	$\mathbb{C}^{N/2}$	✅		

Our attacks

	Plaintext space	IND-CPA^D status belief	In many libraries	Reasons
BFV/BGV (2012)	$\left(\mathbb{F}_p^k\right)^{N/k}$	✅	❌	Incorrect noise upper bound
DM/CGGI (2015)	small integers	✅	❌	High failure probability
discrete-CKKS (2024)	small integers	✅	❌	High failure probability
CKKS (2017)	$\mathbb{C}^{N/2}$	❌	❌	High failure probability
CKKS (+ noise flooding)	$\mathbb{C}^{N/2}$	✅	❌	High failure probability

Attack on DM/CGGI

DM/CGGI boostrapping error at each step



A CPA^D adversary can detect **error overflow**! It **can lead to Key Recoveries**

Bootstrapping failure ***gives*** an LWE hint

Inequality hint

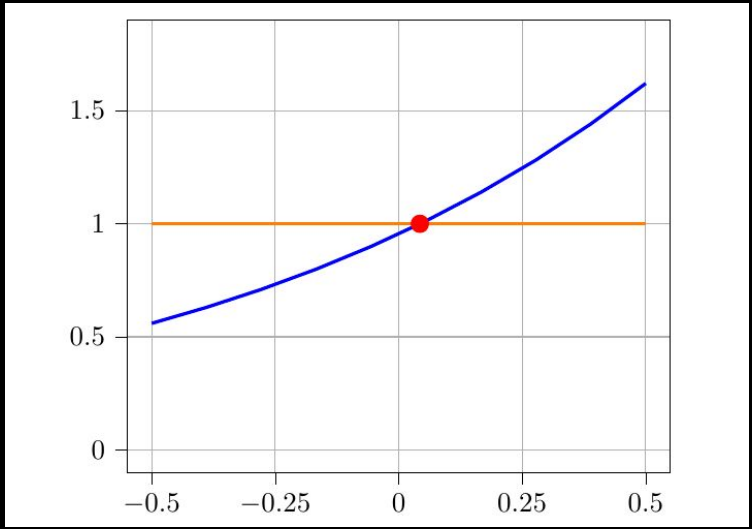
$$\langle v_{MS}, sk \rangle + e > t$$

observed unknown unknown public

Distribution of $v_{MS,i}$ during failures

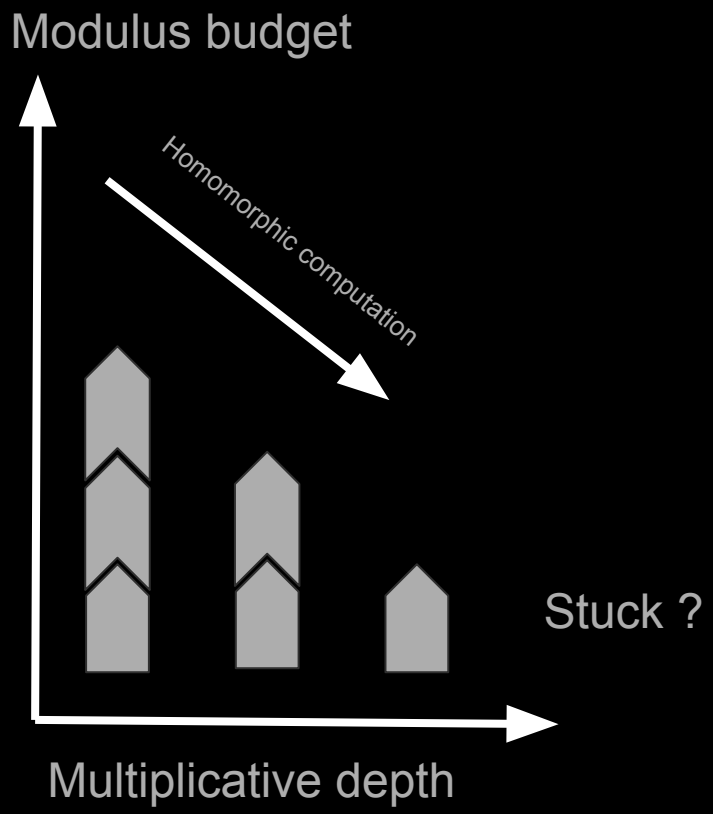
Orange curve displays $\Pr[v_{MS,i} \mid s_i=0, \text{failure}]$

Blue curve displays $\Pr[v_{MS,i} \mid s_i=1, \text{failure}]$

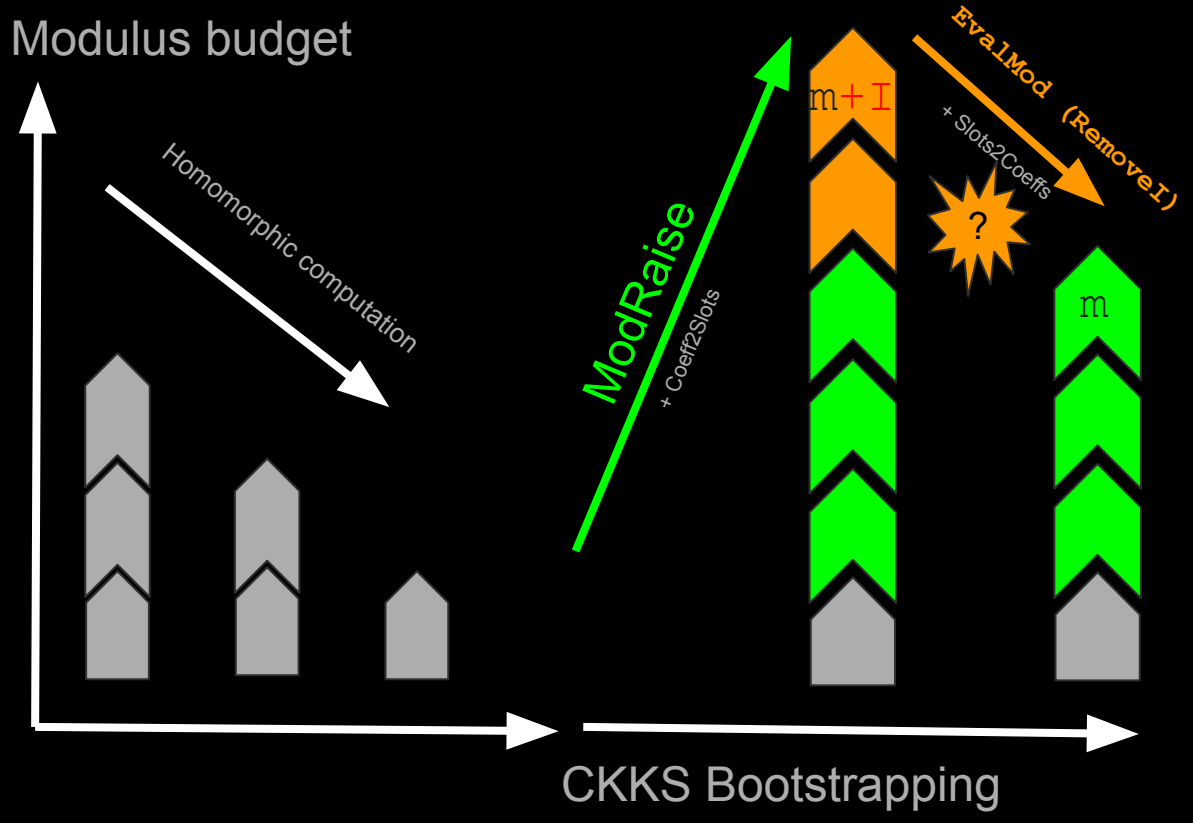


Attack on discrete-CKKS

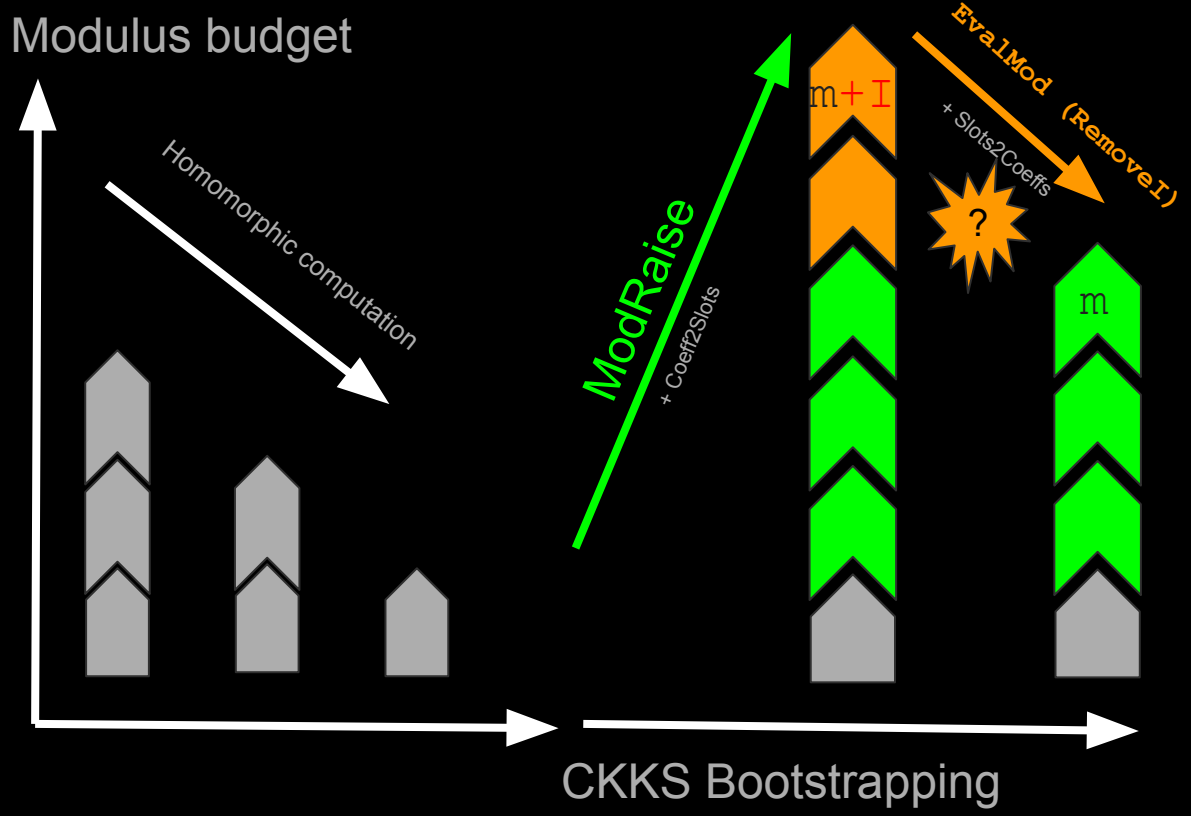
Key-recovery of CKKS from bootstrapping failures



Key-recovery of CKKS from bootstrapping failures



Key-recovery of CKKS from bootstrapping failures



1. The integer `I` comes from <ct, s>
2. EvalMod is correct iff -K < `I` < K
3. Incorrectness means `|I| >= K`
4. These hints lead to **Key-Recovery**

Bootstrapping failure probability in different libraries

DM/CGGI

Library	Parameters	Failure probability
OpenFHE	STD256	2^{-33}
TFHE-rs[1]	DEFAULT	2^{-40}
OpenFHE	STD192Q	2^{-101}
TFHE-rs	TFHE_LIB	2^{-165}
Concrete-python		2^{-17}

[1] now updated to 2^{-64}

Too high for 128 bits of **IND-CPA^D security**

Bootstrapping failure probability in different libraries

DM/CGGI

Library	Parameters	Failure probability
OpenFHE	STD256	2^{-33}
TFHE-rs[1]	DEFAULT	2^{-40}
OpenFHE	STD192Q	2^{-101}
TFHE-rs	TFHE_LIB	2^{-165}
Concrete-python		2^{-17}

[1] now updated to 2^{-64}

CKKS

Default parameters	Failure probability
OpenFHE	2^{-22}
HEaaN	2^{-35}
Lattigo	2^{-138}

Too high for 128 bits of **IND-CPA^D security**

Take-away

IND-CPA security 🥈

Difficult to distinguish between encryptions of two different plaintexts

IND-CPA^D security 🥇

Same with access to decryption of ciphertext for which the adversary is supposed to know the underlying plaintext

IND-CPA^D attacks on exact FHE schemes

BGV/BFV
DM/CGGI
(Exact) CKKS

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

Countermeasures 

For all schemes

- ☐ Small failure probability
- ☐ No heuristic noise analysis

For CKKS

- ☐ High precision computation
- ☐ With noise flooding after decryption

IND-CPA requires cautious implementation of **KeyGen & Enc**

IND-CPA^{D} requires cautious implementation of **KeyGen, Enc, Eval and Dec**

More in the paper

- Generic heuristic CPA^{D} distinguisher
- Connections to Threshold FHE
- Concrete experiments of the attack for $\text{IND-CPA}^{\text{D}}$ insecure parameter sets

Questions?