NTNU | Norwegian University of Science and Technology

Optimisations and Trade-Offs for HElib

Anamaria Costache[1], **Lea Nürnberger**[1], Rachel Player[2]

[1] Norwegian University of Science and Technology, Trondheim, Norway
[2] Royal Holloway, University of London, UK

I will talk about…

- One specific scheme: BGV[1]
- One specific problem: noise analysis

  - What is noise in FHE?
  - How can it by analysed?
  - Why is this important?

[1] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan: Fully Homomorphic Encryption Without Bootstrapping, ITCS‘ 12.

Quick Introduction to BGV[1]

- BGV was first proposed by **Brakerski, Gentry and Vaikuntanathan** in 2012.
- Second generation FHE scheme.
- Based on **R-LWE**.
- It is a **levelled scheme**.
- Multiple implementations exist.

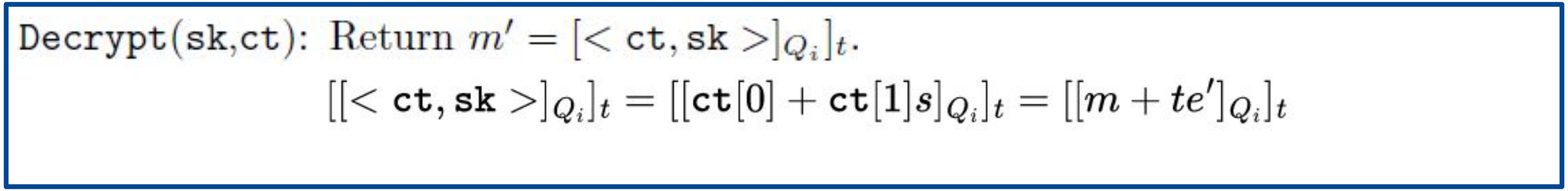
[1] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan: Fully Homomorphic Encryption Without Bootstrapping, ITCS‘ 12.

HElib

- HElib is a homomorphic encryption library by Shai Halevi and Victor Shoup, offering implementations of BGV and CKKS.
- It was first released in 2013.
- It is implemented in C++.

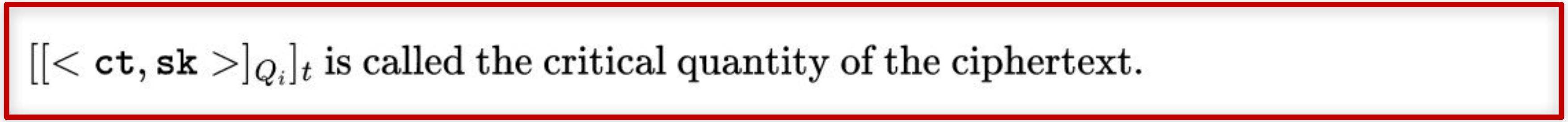
WHAT IS NOISE IN FHE?

Noise in BGV[1]

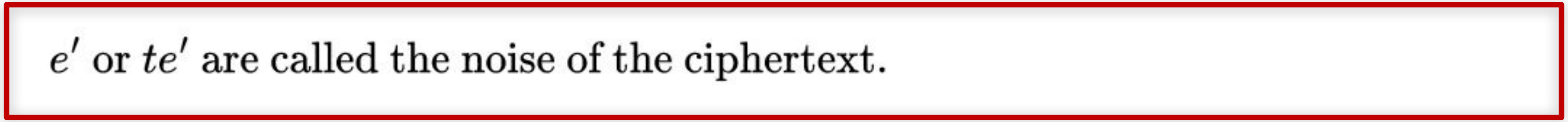


Decrypt(sk,ct): Return $m' = [< \mathtt{ct}, \mathtt{sk} >]_{Q_i}]_t$.

$$[[< \mathtt{ct}, \mathtt{sk} >]_{Q_i}]_t = [[\mathtt{ct}[0] + \mathtt{ct}[1]s]_{Q_i}]_t = [[m + te']_{Q_i}]_t$$



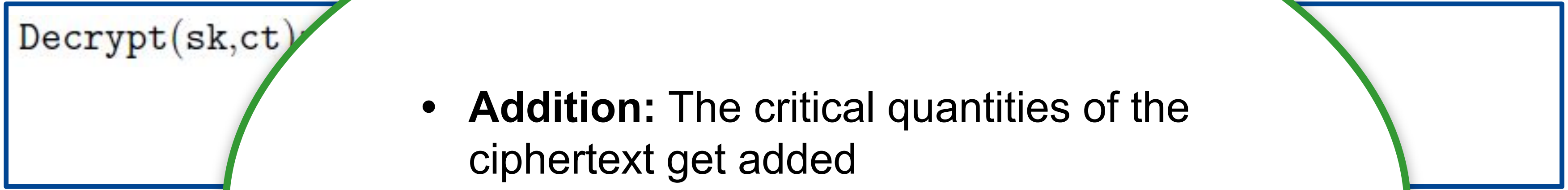
$[[< \mathtt{ct}, \mathtt{sk} >]_{Q_i}]_t$ is called the critical quantity of the ciphertext.



e' or te' are called the noise of the ciphertext.

[1] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan: Fully Homomorphic Encryption Without Bootstrapping, ITCS‘ 12.

Noise in BGV[1]



Decrypt(sk,ct)

$[[< \mathbf{ct}, \mathbf{sk} >$

e' or te' are called

- **Addition:** The critical quantities of the ciphertext get added
- **Multiplication:** The critical quantities get multiplied

[1] Zvika Brakerski, Craig Gentry, Vinod Vai than: Fully Homomorphic Encryption Without Bootstrapping, ITCS‘ 12.

What is Noise in FHE? - Summary

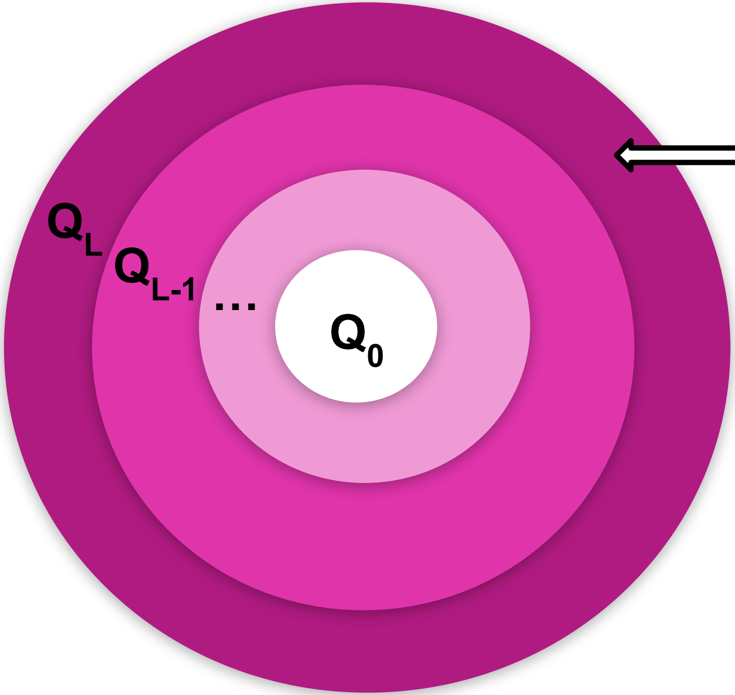
FHE Noise Dilemma

- Without noise the scheme would be insecure. But with too much noise eventually we will not be able to decrypt correctly.
- To know whether decryption is still correct, we need to know exactly how much noise the ciphertext has, but if we know it exactly the scheme is no longer secure.

BGV as a Levelled Scheme

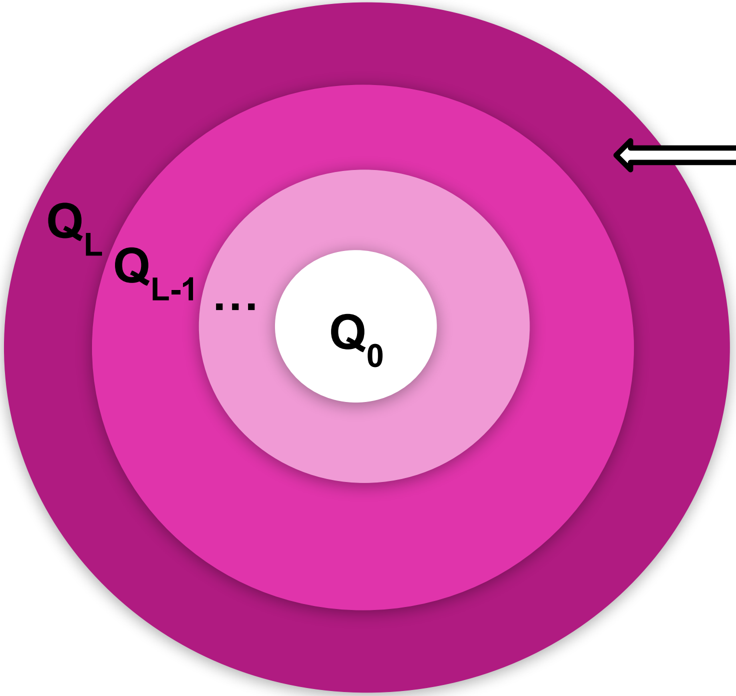
- A **chain of ciphertext moduli** is chosen, $Q_0 < Q_1 < \ldots < Q_L$, $Q_i | Q_{i+1.}$
- To **reduce noise** we can apply modulus switching.
- Modulus switching transforms a ciphertext ct_1 encrypting m with respect to Q_i to a ciphertext ct_2 encrypting m with respect to $Q_{i-1.}$ Modulus switching allows to reduce the noise by approximately a factor Q_{i-1}/Q_i.

BGV as a Levelled Scheme



Encrypt a fresh ciphertext at the top level.

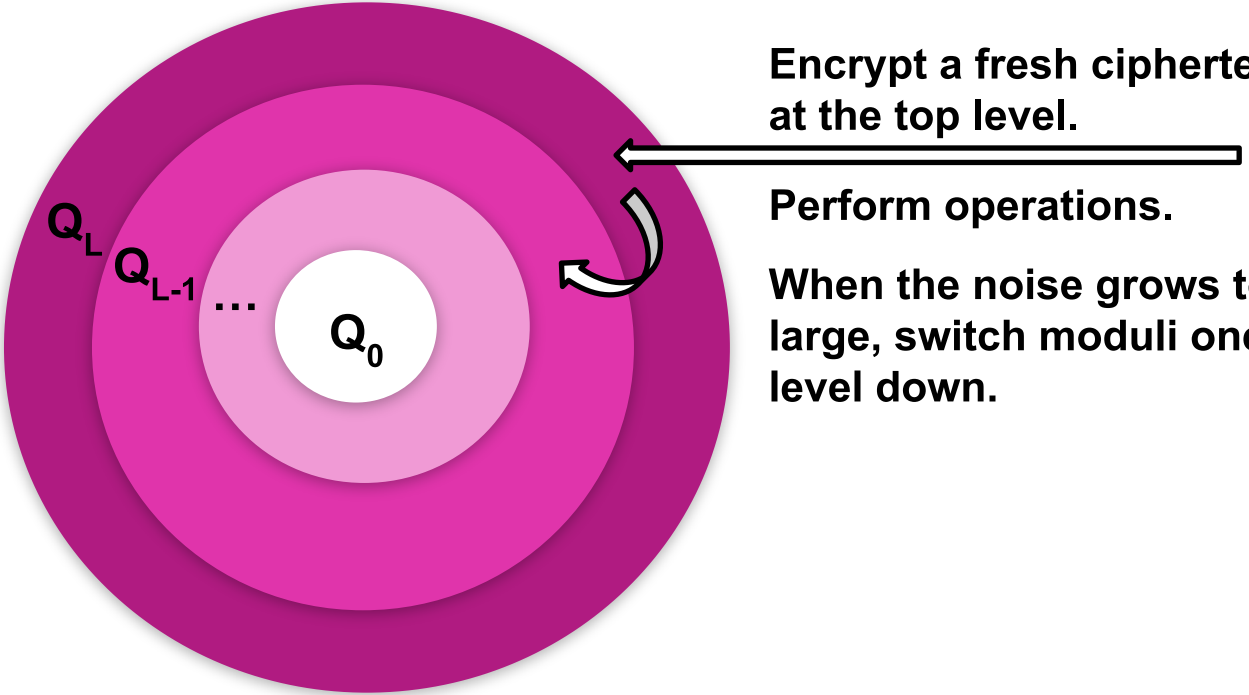
BGV as a Levelled Scheme



Encrypt a fresh ciphertext at the top level.

Perform operations.

BGV as a Levelled Scheme

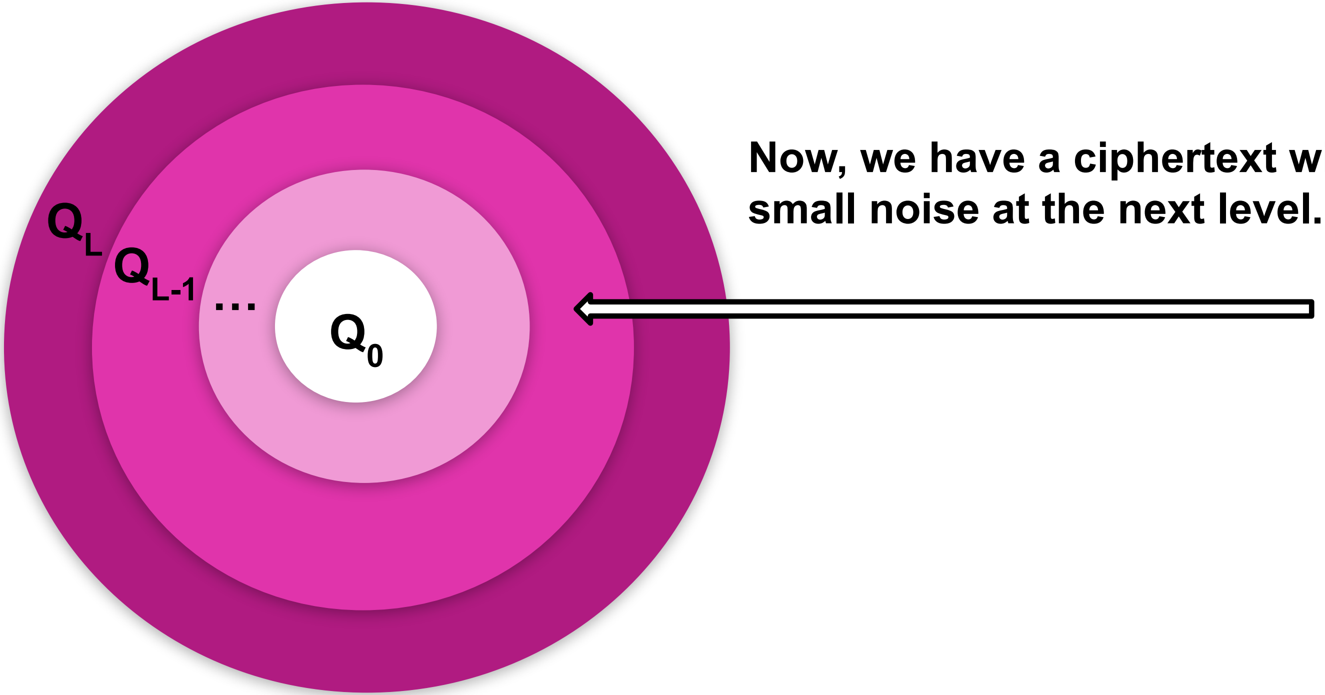


Encrypt a fresh ciphertext at the top level.

Perform operations.

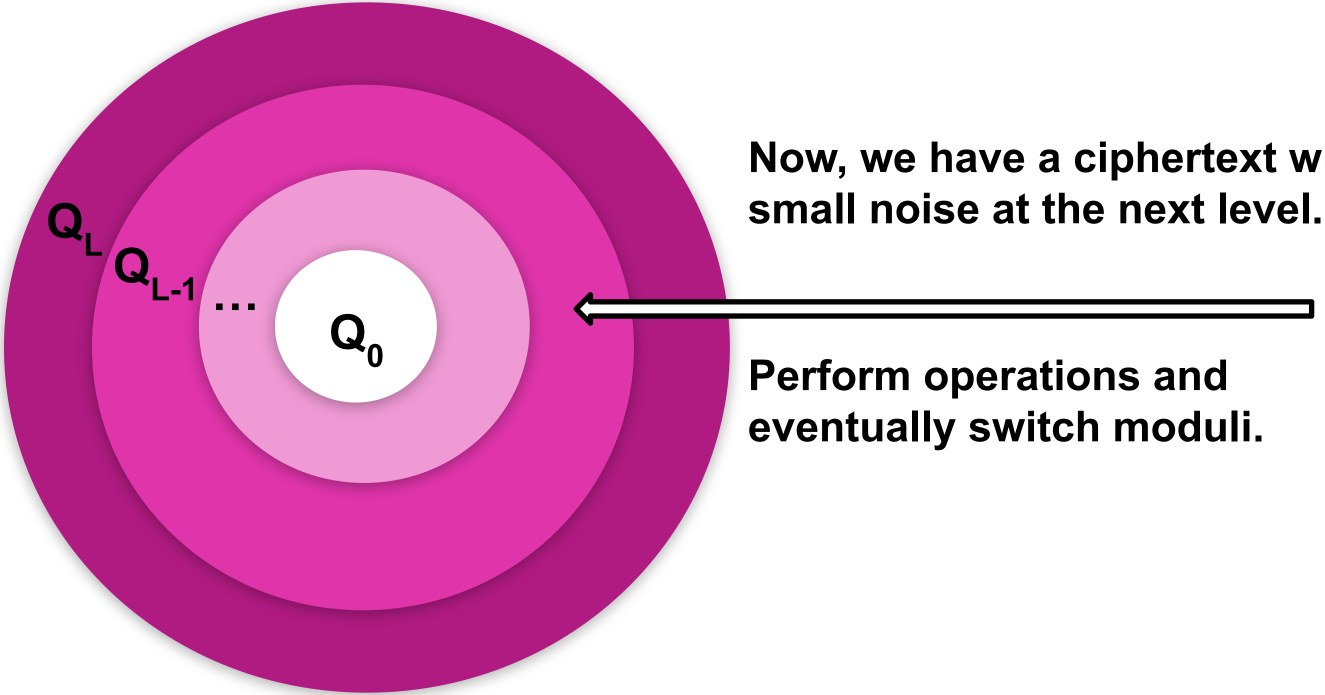
When the noise grows too large, switch moduli one level down.

BGV as a Levelled Scheme



Now, we have a ciphertext with small noise at the next level.

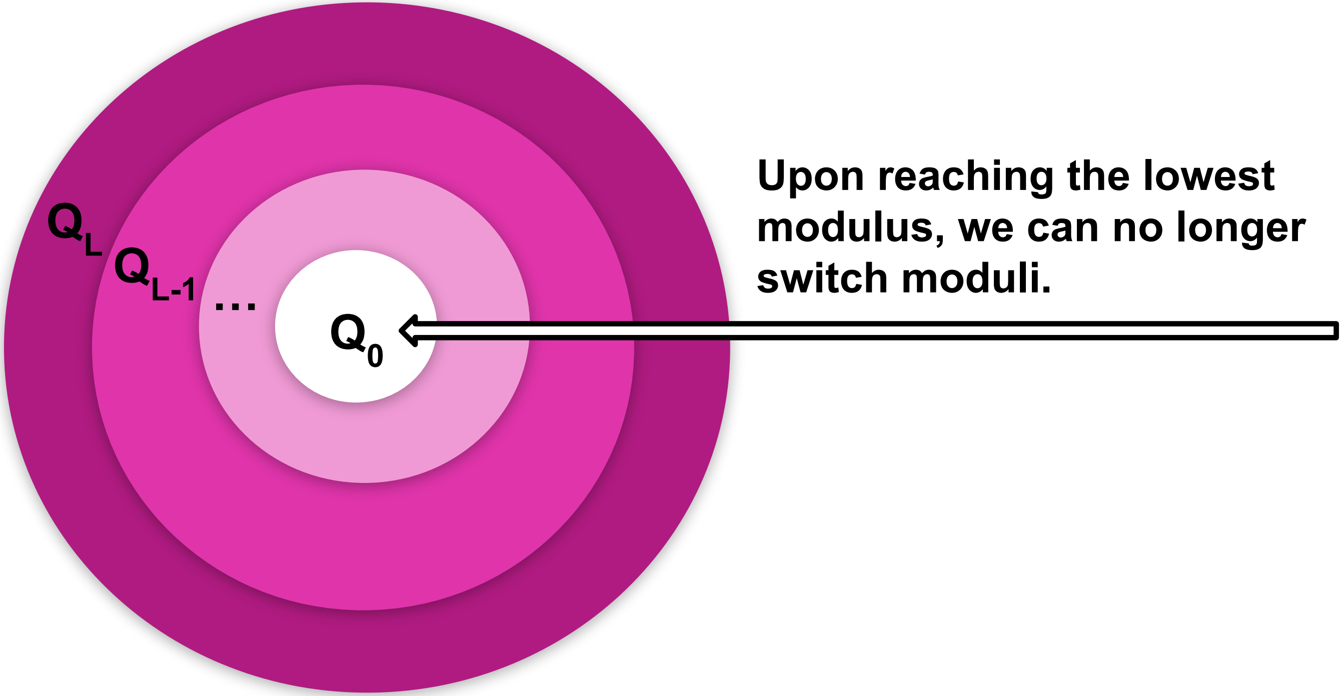
BGV as a Levelled Scheme



Now, we have a ciphertext with small noise at the next level.

Perform operations and eventually switch moduli.

BGV as a Levelled Scheme



Upon reaching the lowest modulus, we can no longer switch moduli.

WHY IS NOISE ANALYSIS IMPORTANT?

Why is noise analysis important?

- Each modulus switching **consumes** a level until we have no more. We want to **delay** modulus switches as long as possible.
- Tight noise estimates allow to perform **more operations** before modulus switching.
- The ratio between the noise and the ciphertext modulus determines the security level. **Tight estimates allow for better parameters**.

NOISE ANALYSIS TECHNIQUES

Noise Analysis Techniques

- Bounding the **canonical embedding norm** of the critical quantity after each step[2,3].
- Bounding the **infinity norm** of the critical quantity after each step[4] .
- Bounding the infinity norm after a **complete circuit**[5].

[2] Anamaria Costache, Kim Laine, Rachel Player: Evaluating the Effectiveness of Heuristc Worst-Case Noise Analysis in FHE, ESORICS 2020.

[3] Shai Halevi, Victor Shoup: Design and Implementation of HElib, https://eprint.iacr.org/2020/1481

[4] Andrey Kim, Yuri Polyakoff, Vincent Zucca: Revisiting Fully Homomorphic Encryption over the Finite Field, ASIACRYPT 2021.

[5] Anamaria Costache, Ben Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, Rachel Player: On the precision loss in approximate homomorphic encryption, https://eprint.iacr.org/2022/162

Our Work

- We apply the techniques from [5] to BGV.
- We provide an **implementation specific** noise analysis for HElib.
- We **compare** our results with previous analyses of BGV noise and show the need for an implementation specific analysis.
- Based on our analysis we provide **better parameter sets** for BGV and show them to be optimal for a given error probability.

[5] Anamaria Costache, Ben Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, Rachel Player: On the precision loss in approximate homomorphic encryption, https://eprint.iacr.org/2022/162

Our Noise Analysis Technique

- As in [5], we calculate the **variance of the critical quantity** after each step in the homomorphic operations.
- For pre-multiplication and modulus switching we show the coefficients of the critical quantity to be **normally distributed**. We can therefore bound the infinity norm of the critical quantity as

$$||v_{pm,ms}||_\infty \leq 10\sigma_{pm,ms}$$

, with error probability $\approx 2^{-75+\log_2(n)}$.

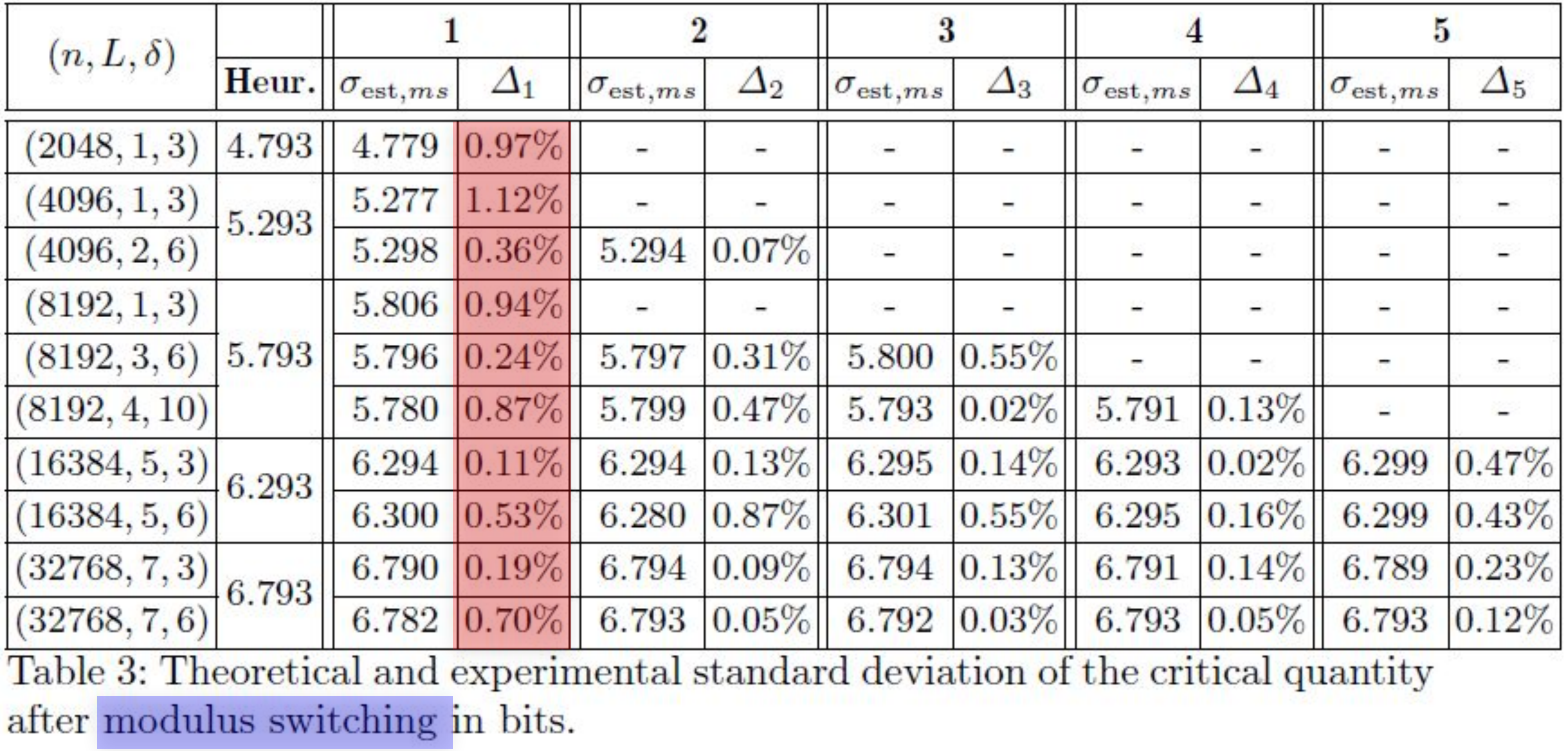
[5] Anamaria Costache, Ben Curtis, Erin Hales, Sean Murphy, Tabitha Ogilvie, Rachel Player: On the precision loss in approximate homomorphic encryption, https://eprint.iacr.org/2022/162

EXPERIMENTAL RESULTS

Experimental Results

- We **compared our theoretical bounds with experimentally obtained values** for the infinity norm of the critical quantity in HElib. We looked at 8 parameter sets after 1 – 5 multiplications, and calculated the average noise and standard deviation over 10,000 trials.

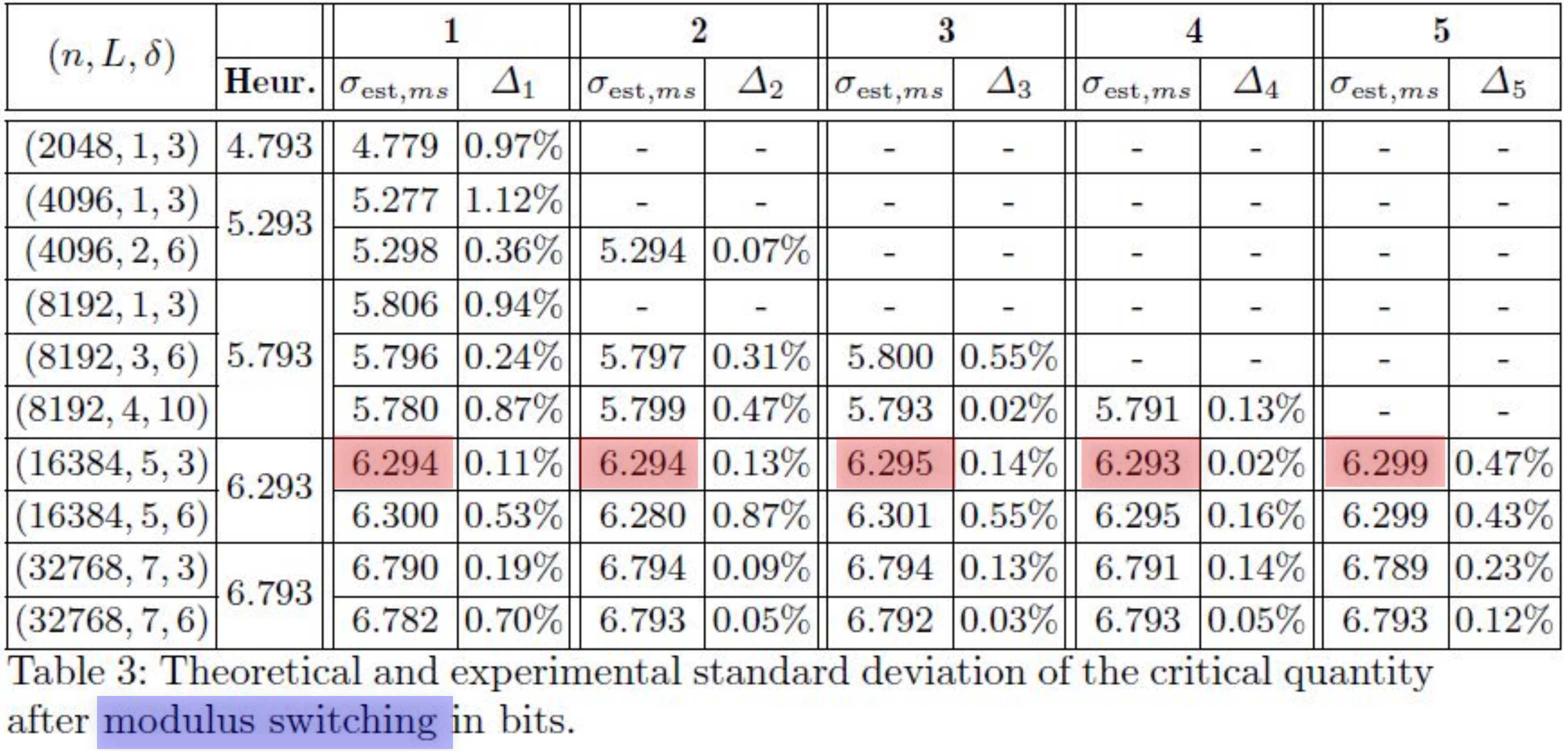
Experiments – Standard Deviation



(n, L, δ)		**1**		**2**		**3**		**4**		**5**	
	Heur.	$\sigma_{\mathrm{est},ms}$	Δ_1	$\sigma_{\mathrm{est},ms}$	Δ_2	$\sigma_{\mathrm{est},ms}$	Δ_3	$\sigma_{\mathrm{est},ms}$	Δ_4	$\sigma_{\mathrm{est},ms}$	Δ_5
$(2048, 1, 3)$	4.793	4.779	0.97%	-	-	-	-	-	-	-	-
$(4096, 1, 3)$	5.293	5.277	1.12%	-	-	-	-	-	-	-	-
$(4096, 2, 6)$		5.298	0.36%	5.294	0.07%	-	-	-	-	-	-
$(8192, 1, 3)$	5.793	5.806	0.94%	-	-	-	-	-	-	-	-
$(8192, 3, 6)$		5.796	0.24%	5.797	0.31%	5.800	0.55%	-	-	-	-
$(8192, 4, 10)$		5.780	0.87%	5.799	0.47%	5.793	0.02%	5.791	0.13%	-	-
$(16384, 5, 3)$	6.293	6.294	0.11%	6.294	0.13%	6.295	0.14%	6.293	0.02%	6.299	0.47%
$(16384, 5, 6)$		6.300	0.53%	6.280	0.87%	6.301	0.55%	6.295	0.16%	6.299	0.43%
$(32768, 7, 3)$	6.793	6.790	0.19%	6.794	0.09%	6.794	0.13%	6.791	0.14%	6.789	0.23%
$(32768, 7, 6)$		6.782	0.70%	6.793	0.05%	6.792	0.03%	6.793	0.05%	6.793	0.12%

Table 3: Theoretical and experimental standard deviation of the critical quantity after modulus switching in bits.

Experiments – Standard Deviation



(n, L, δ)		**1**		**2**		**3**		**4**		**5**	
	Heur.	$\sigma_{\text{est},ms}$	Δ_1	$\sigma_{\text{est},ms}$	Δ_2	$\sigma_{\text{est},ms}$	Δ_3	$\sigma_{\text{est},ms}$	Δ_4	$\sigma_{\text{est},ms}$	Δ_5
$(2048, 1, 3)$	4.793	4.779	0.97%	-	-	-	-	-	-	-	-
$(4096, 1, 3)$	5.293	5.277	1.12%	-	-	-	-	-	-	-	-
$(4096, 2, 6)$		5.298	0.36%	5.294	0.07%	-	-	-	-	-	-
$(8192, 1, 3)$	5.793	5.806	0.94%	-	-	-	-	-	-	-	-
$(8192, 3, 6)$		5.796	0.24%	5.797	0.31%	5.800	0.55%	-	-	-	-
$(8192, 4, 10)$		5.780	0.87%	5.799	0.47%	5.793	0.02%	5.791	0.13%	-	-
$(16384, 5, 3)$	6.293	6.294	0.11%	6.294	0.13%	6.295	0.14%	6.293	0.02%	6.299	0.47%
$(16384, 5, 6)$		6.300	0.53%	6.280	0.87%	6.301	0.55%	6.295	0.16%	6.299	0.43%
$(32768, 7, 3)$	6.793	6.790	0.19%	6.794	0.09%	6.794	0.13%	6.791	0.14%	6.789	0.23%
$(32768, 7, 6)$		6.782	0.70%	6.793	0.05%	6.792	0.03%	6.793	0.05%	6.793	0.12%

Table 3: Theoretical and experimental standard deviation of the critical quantity after modulus switching in bits.

Experiments – Standard Deviation

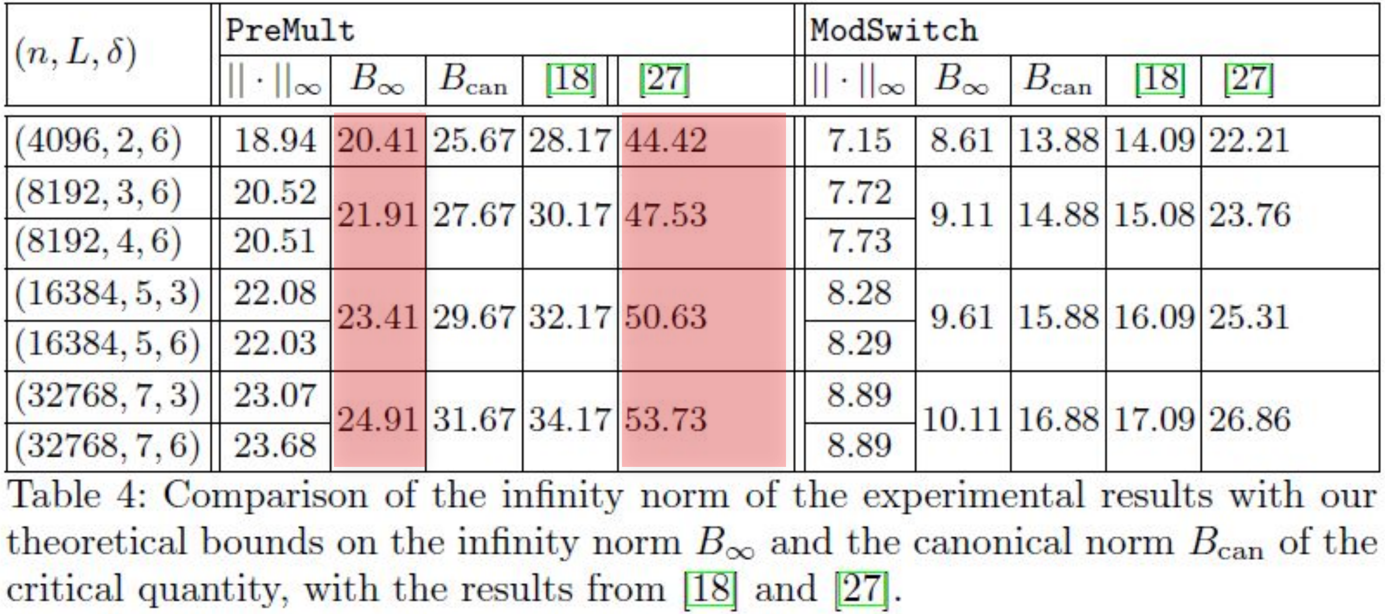
- For modulus switching our theoretical results deviate form the experimental results by at most 1.12%, but in most cases by no more than 1%, which is the expected standard error.

Our Observations

- The noise after modulus switching is **independent** of the input ciphertexts.
- It only depends on the **ring dimension and the hamming weight** of the secret key.
- We can hence give tighter estimations **for any number of multiplications** that have been performed.

Comparison with Related Work –

Critical Quantity

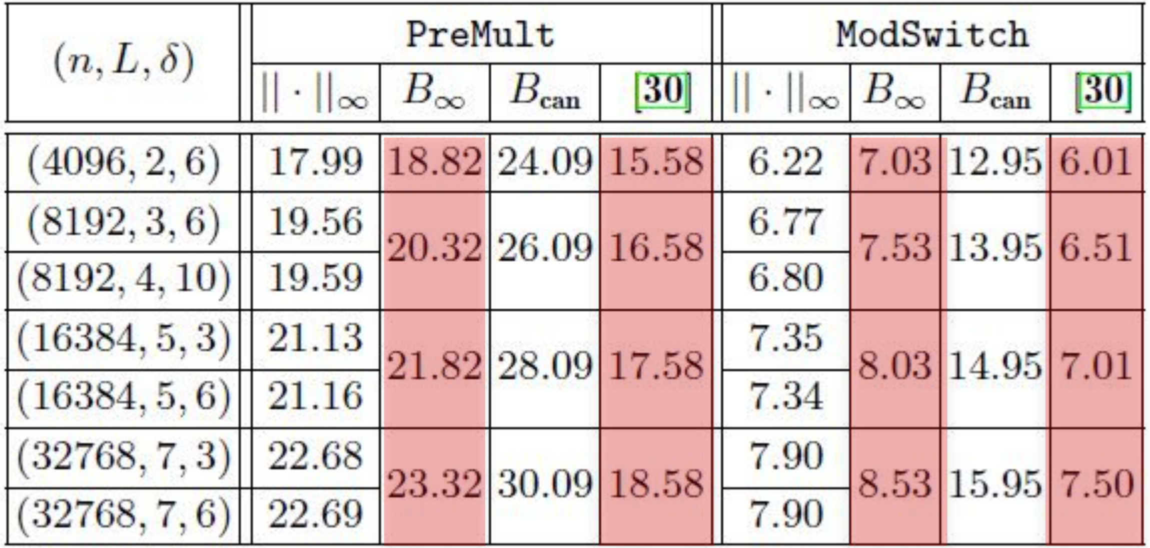


(n, L, δ)	PreMult					ModSwitch				
	$\|\| \cdot \|\|_\infty$	B_∞	B_{can}	[18]	[27]	$\|\| \cdot \|\|_\infty$	B_∞	B_{can}	[18]	[27]
$(4096, 2, 6)$	18.94	20.41	25.67	28.17	44.42	7.15	8.61	13.88	14.09	22.21
$(8192, 3, 6)$	20.52	21.91	27.67	30.17	47.53	7.72	9.11	14.88	15.08	23.76
$(8192, 4, 6)$	20.51					7.73				
$(16384, 5, 3)$	22.08	23.41	29.67	32.17	50.63	8.28	9.61	15.88	16.09	25.31
$(16384, 5, 6)$	22.03					8.29				
$(32768, 7, 3)$	23.07	24.91	31.67	34.17	53.73	8.89	10.11	16.88	17.09	26.86
$(32768, 7, 6)$	23.68					8.89				

Table 4: Comparison of the infinity norm of the experimental results with our theoretical bounds on the infinity norm B_∞ and the canonical norm B_{can} of the critical quantity, with the results from [18] and [27].

[18] Anamaria Costache, Kim Laine, Rachel Player: evaluating the effectiveness of worst-case noise heuristics, ESORICS 20.
[27] Shai Halevi, Victor Shoup: design and implementation of HElib, eprint.

Comparison With Related Work – Noise



(n, L, δ)	PreMult				ModSwitch			
	$\|\|\cdot\|\|_\infty$	B_∞	B_{can}	[30]	$\|\|\cdot\|\|_\infty$	B_∞	B_{can}	[30]
$(4096, 2, 6)$	17.99	18.82	24.09	15.58	6.22	7.03	12.95	6.01
$(8192, 3, 6)$	19.56	20.32	26.09	16.58	6.77	7.53	13.95	6.51
$(8192, 4, 10)$	19.59				6.80			
$(16384, 5, 3)$	21.13	21.82	28.09	17.58	7.35	8.03	14.95	7.01
$(16384, 5, 6)$	21.16				7.34			
$(32768, 7, 3)$	22.68	23.32	30.09	18.58	7.90	8.53	15.95	7.50
$(32768, 7, 6)$	22.69				7.90			

Table 6: Comparison of the bounds on the infinity norm of the noise after 2 multiplications for pre-multiplications and modulus switching with the results from [30] in bits.

[30] Andrey Kim, Yuriy Polyakov, Vincent Zucca: Revisiting Homomorphic Encryption Schemes for Finite Fields. ASIACRYPT 2021.

Comparison with Related Work

- Our bounds are tighter than the ones given in other sources
- Noise bounds developed for PALISADE[6] **underestimate** the noise, potentially leading to decryption errors.
- While these bounds may be tight for PALISADE, this illustrates the importance of implementation-specific noise analysis. Bounds developed for other implementations should not be used.

[6] Palisade lattice cryptography library (release 1.10.6) Dec 2020.

OPTIMISATIONS AND TRADE-OFFS

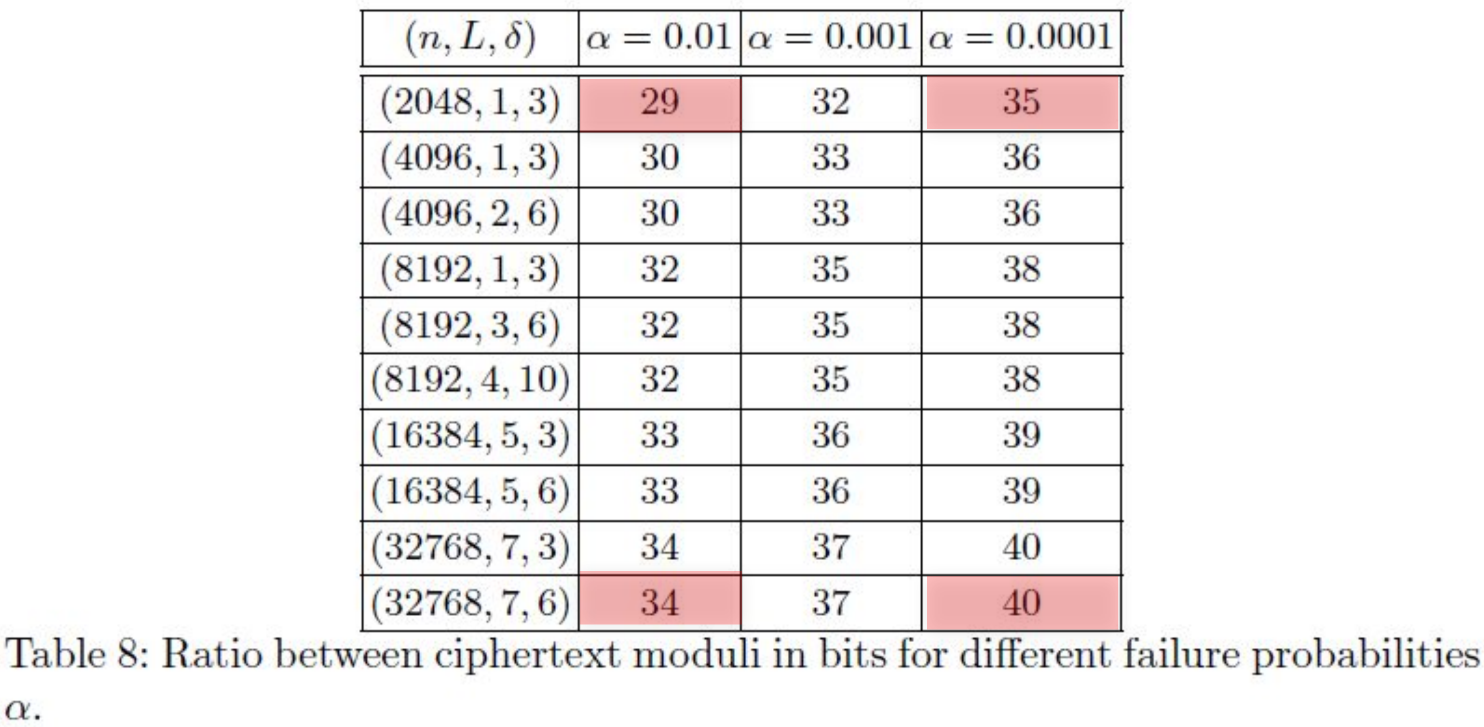
Optimisations and Trade-offs

- We want to obtain **constant noise** after modulus switching.
- EasyCalculations™ show we can either make the ciphertext moduli ratio smaller, or the special modulus kQ_i smaller.
- We fix both in turns.

Optimizations and Trade-offs

- Theoretically the smallest ratio between ciphertext moduli that can be observed are 36 bits. In practice, we always observed 54 or more.

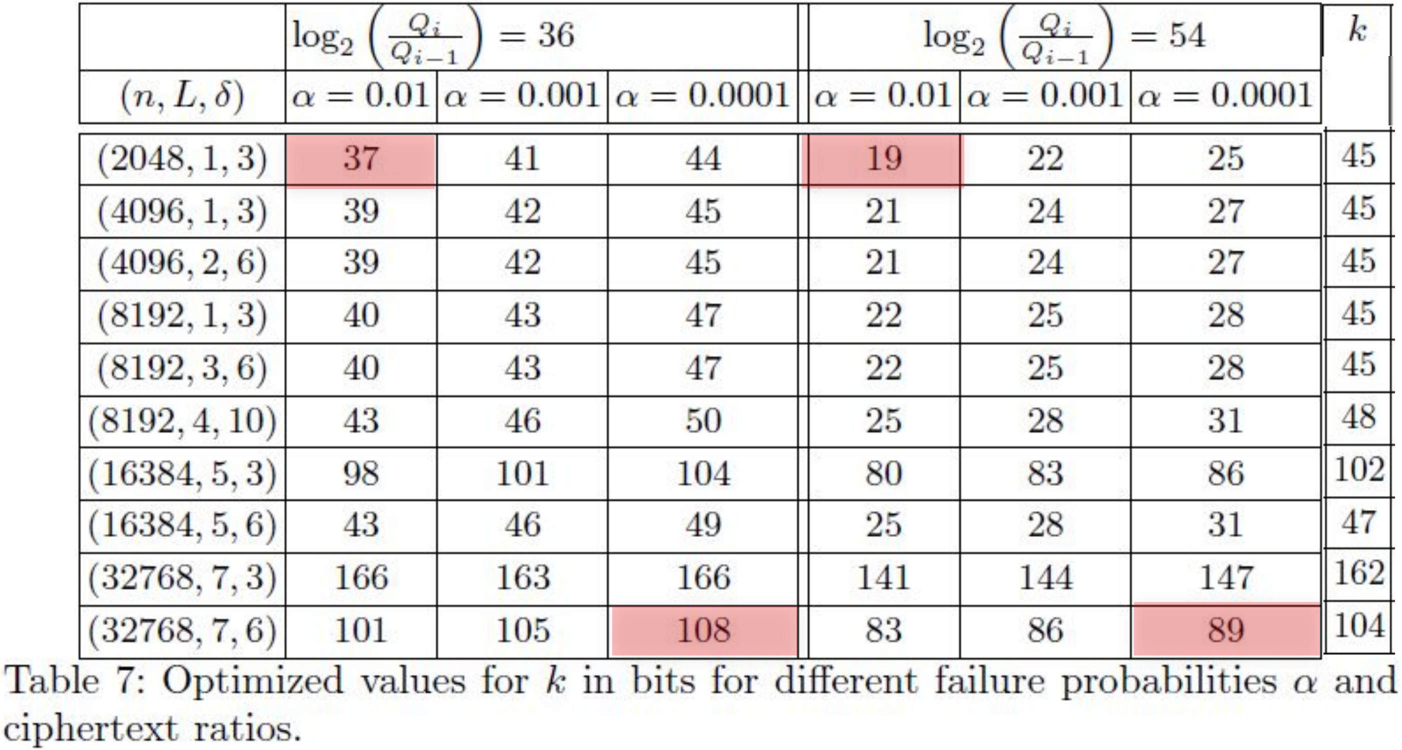
Optimisations for the Ciphertext Moduli Ratio



(n, L, δ)	$\alpha = 0.01$	$\alpha = 0.001$	$\alpha = 0.0001$
$(2048, 1, 3)$	29	32	35
$(4096, 1, 3)$	30	33	36
$(4096, 2, 6)$	30	33	36
$(8192, 1, 3)$	32	35	38
$(8192, 3, 6)$	32	35	38
$(8192, 4, 10)$	32	35	38
$(16384, 5, 3)$	33	36	39
$(16384, 5, 6)$	33	36	39
$(32768, 7, 3)$	34	37	40
$(32768, 7, 6)$	34	37	40

Table 8: Ratio between ciphertext moduli in bits for different failure probabilities α.

Optimisations for k



	$\log_2\left(\frac{Q_i}{Q_{i-1}}\right) = 36$			$\log_2\left(\frac{Q_i}{Q_{i-1}}\right) = 54$			k
(n, L, δ)	$\alpha = 0.01$	$\alpha = 0.001$	$\alpha = 0.0001$	$\alpha = 0.01$	$\alpha = 0.001$	$\alpha = 0.0001$	
$(2048, 1, 3)$	37	41	44	19	22	25	45
$(4096, 1, 3)$	39	42	45	21	24	27	45
$(4096, 2, 6)$	39	42	45	21	24	27	45
$(8192, 1, 3)$	40	43	47	22	25	28	45
$(8192, 3, 6)$	40	43	47	22	25	28	45
$(8192, 4, 10)$	43	46	50	25	28	31	48
$(16384, 5, 3)$	98	101	104	80	83	86	102
$(16384, 5, 6)$	43	46	49	25	28	31	47
$(32768, 7, 3)$	166	163	166	141	144	147	162
$(32768, 7, 6)$	101	105	108	83	86	89	104

Table 7: Optimized values for k in bits for different failure probabilities α and ciphertext ratios.

Trade-Offs

- Reducing the ciphertext moduli ratio allows us to „squeeze in“ **more levels**.
- Reducing k leads to **smaller evaluation keys** and key switching noise.
- Reducing k speeds up the **evaluation key generation** which can be slow.

Trade-Offs

- Obtaining more levels is of interest to make the evaluation of **deeper** circuits possible.
- Reducing k can be of interest in **client-aided protocols** where the ciphertext is sent to the client for reencryption.

Summary

- Tight noise analysis is important.
- Implementation-specific noise analysis is important.

Thank you!