



**HAL**  
open science

# Hitchhiker's Guide to a Practical Automated TFHE Parameter Setup for Custom Applications

Jakub Klemsa

► **To cite this version:**

Jakub Klemsa. Hitchhiker's Guide to a Practical Automated TFHE Parameter Setup for Custom Applications. FHE 2022, 1st Annual Conference on Fully Homomorphic Encryption, May 2022, Trondheim (Norvège), Norway. hal-04146488

**HAL Id: hal-04146488**

**<https://hal.science/hal-04146488>**

Submitted on 30 Jun 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Hitchhiker's Guide to a Practical Automated TFHE Parameter Setup for Custom Applications

Jakub Klemsa

jakub.klemsa@eurecom.fr

## Motivation

TFHE cipher is instantiated with **8 parameters**:

- determine **security level** and plaintext/evaluation **error rate**  
⇒ parameters are **application-specific**,
- vast impact on **performance** ⇒ optimization problem,

⇒ **need for a tool** for TFHE parameter setup.

Given application specifications:

- + **fully automated** TFHE parameter setup,
- + optimized for **best performance**,
- + unified approach ⇒ allows to **compare** different parameters:
  - same target security & error rates, aim for best parameters,
  - e.g., different homomorphization in digit-based arithmetics [1].

## Application Specifications

Specify application needs (security & error rates) by **three parameters**:

- bit-security level** denoted  $\lambda$ ;
- requested **cleartext space bit-precision** denoted  $\pi$ ; and
- bound on the number of homomorphic additions before the sample gets bootstrapped, denoted  $2^{2\Delta}$ , referred **quadratic weights**;

⇒ input parameters for our TFHE parameter setup tool.

### A. Bit-Security Level $\lambda$

*Observation 1.* At fixed security  $\lambda$ , the logarithm of stddev of LWE noise (den.  $\alpha$ ), is roughly linear in the LWE dimension  $n$  (with factor den.  $s_\lambda$ ):

$$-\log_2(\alpha) \approx s_\lambda \cdot n; \quad (1)$$

cf. Figure 1. Due to the **collision attack**, the relation is limited to  $n \geq 2\lambda$ , also the behavior changes for  $-\log_2(\alpha) > \tau$  with  $\tau$  the torus precision.

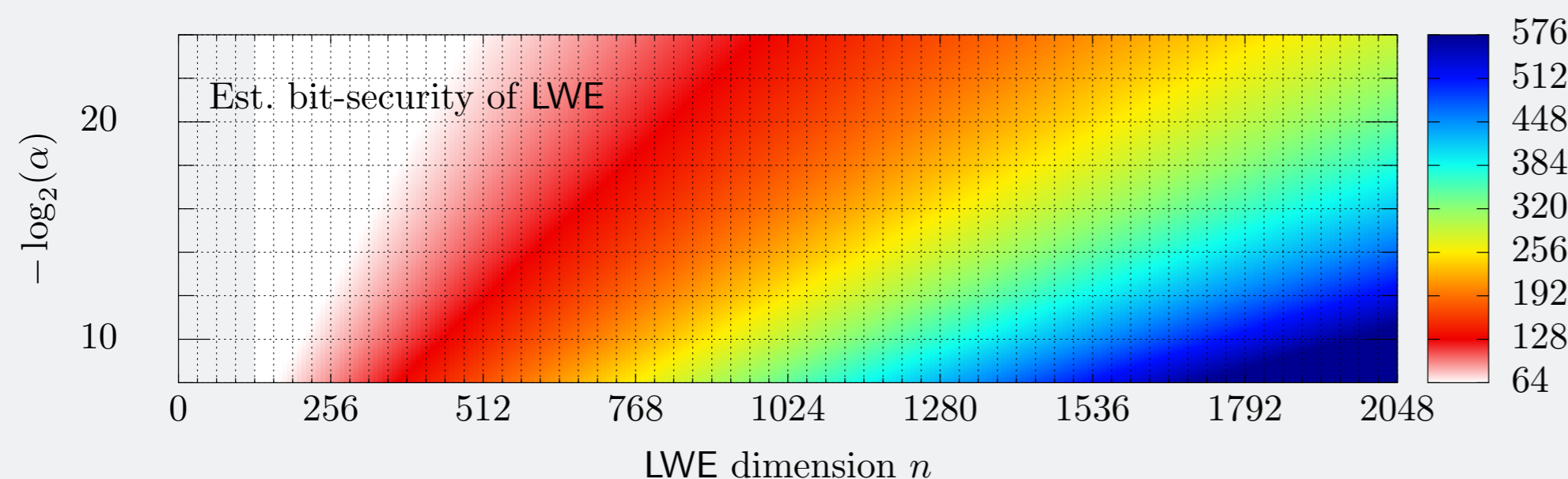


Figure 1: LWE bit-security level  $\lambda$  as estimated by the *LWE Estimator* by Albrecht et al. [2]. For  $\lambda = 128$  bits,  $s_\lambda \approx 0.0235$ .

### B. Cleartext Space Bit-Precision $\pi$

Before selecting an appropriate  $\pi$ , bare in mind:

- complexity of TFHE bootstrapping is roughly **exponential** in  $\pi$ ,
  - practical times for up to  $\pi \approx 6$  bits (cf. Figure 2),
- bootstrapping Look-Up Table (LUT) is inherently **negacyclic**:

$$LUT(2^{\pi-1} + m) = -LUT(m), \quad m \in [0, 2^{\pi-1}). \quad (2)$$

### C. Quadratic weights $2^{2\Delta}$

*Observation 2.* LWE noises **accumulate** with each homomorphic addition: for indep. TLWE samples  $\mathbf{c}_i$  with equal noise variance denoted  $V_0$ :

$$\text{Var}\left(\text{Err}\left(\sum w_i \cdot \mathbf{c}_i\right)\right) = \sum \underbrace{w_i^2}_{2^{2\Delta}} \cdot \underbrace{\text{Var}(\text{Err}(\mathbf{c}_i))}_{V_0}, \quad w_i \in \mathbb{Z}. \quad (3)$$

To ensure **correct LUT evaluation** during bootstrapping:

- **bound** on  $2^{2\Delta}$  before a sample gets bootstrapped (refresh noise),
- $2^{2\Delta}$  ... sum of squared weights;  $\Delta$  ... bits of stddev of addit'l noise.

## Acknowledgements

This work was supported by the MESRI-BMBF project UPCARE (ANR-20-CYAL-0003-01).

## Parameter Restrictions

Goal: **bound the noise** of a fresh(ly bootstrapped) sample, s.t.

- limited number of additions** can be performed (cf.  $2^{2\Delta}$ ); and
- the noise can be **refreshed correctly** during bootstrapping.

$2^\pi$  cleartext values ⇒ max error  $\stackrel{!}{<} 1/2^{\pi+1}$  ... by  $3\sigma$ -rule:

$$V_{\max} \leq 2^{2\Delta} V_0 + V_{\text{round}} \stackrel{!}{\leq} \frac{1}{3^2 \cdot 2^{2\pi+2}}, \quad \text{where } V_{\text{round}} = \frac{n+1}{48N^2}. \quad (4)$$

$V_0$  depends on implementation, for plain TFHE [3]:

$$V_0 \leq \underbrace{2nlN2^{2(\gamma-1)}V_{\text{BK}}(N)}_{(\heartsuit)} + \underbrace{n(1+N)2^{-2(\gamma l+1)}}_{(\diamondsuit)} + \underbrace{\text{Var}(\text{Err}(u,v))}_{=0} + \underbrace{tN2^{2(\kappa-1)}V_{\text{KS}}(n)}_{(\clubsuit)} + \underbrace{2^{-2(\kappa t+1)}N}_{(\spadesuit)}. \quad (5)$$

To derive **good TFHE parameters**, we need to:

- satisfy the bound** (4) (error budget), using (5); and
- check their quality in terms of **bootstrapping time**.

## Experimental Results


	$\mathbb{Z}/16\mathbb{Z}$ Demo: $\pi = 6, 2^{2\Delta} = 2$		Repo with exp. code
	Orig. param's [4]	New param's	
$N, n; \gamma, l$	2048, 750; 7, 3	2048, 766; 21, 1	
$\kappa, t; \log(\alpha_{\text{BK}, \text{KS}})$	2, 7; -52, -18	3, 5; -48, -18	
$\lambda; t_{BS}$	128.2; 199.6 ms	131.2; 124.6 ms	
$\eta_C, \eta_m [\%]$	86.0, 85.5	91.2, 90.5	

Table 1: Original and newly identified TFHE parameters for the  $\mathbb{Z}/16\mathbb{Z}$  demo [4] with  $\pi = 6$  bits. 500 runs with Concrete [5] on Intel Core i7-7800X.  $\eta_C$  and  $\eta_m$  stand resp. for the usage of the  $3\sigma_{\max}$  error budget as calculated by Concrete and as measured after decryption. Experimental code at <https://gitlab.eurecom.fr/fakub/tfhe-param-testing>.

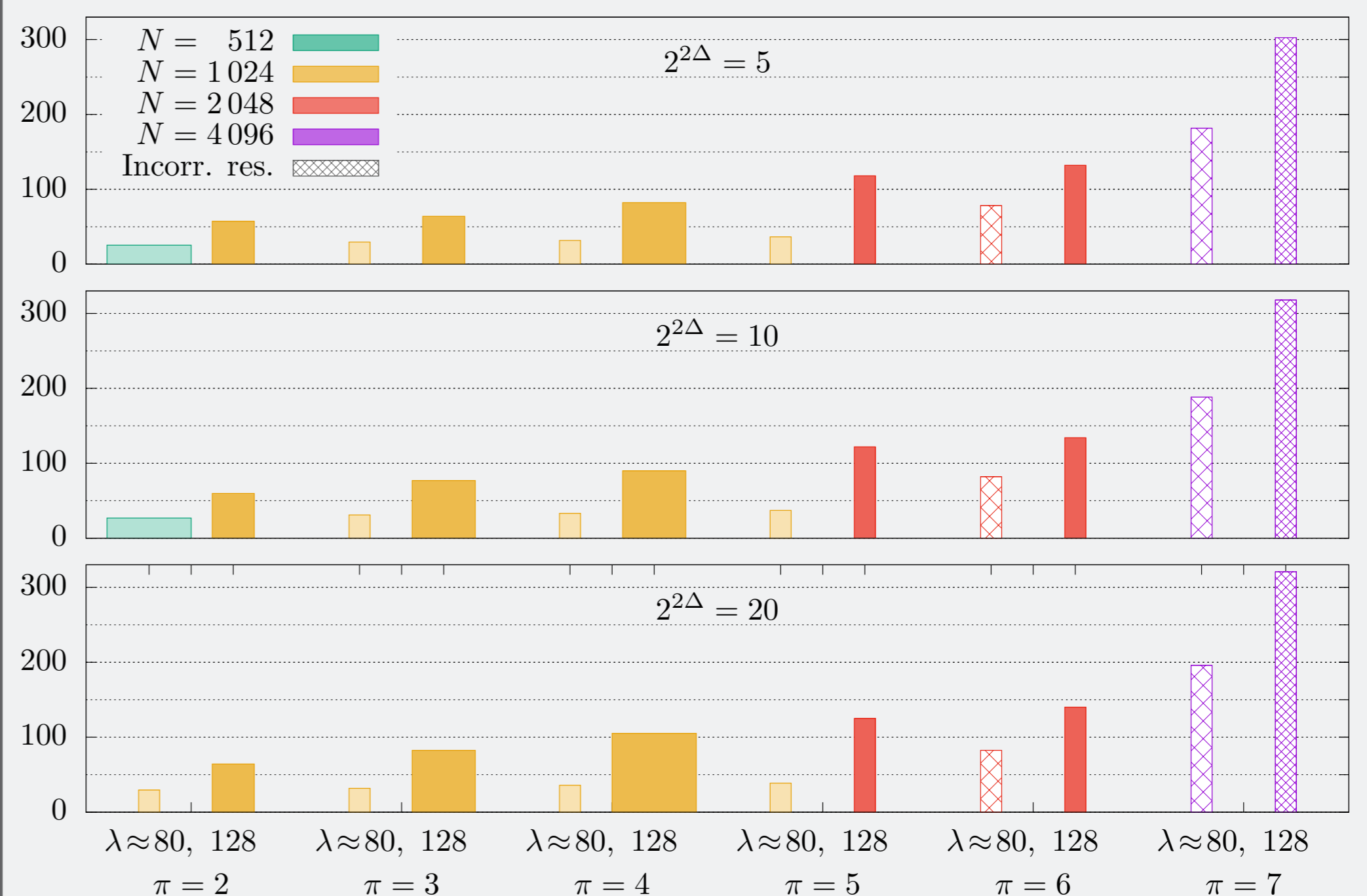


Figure 2: Bootstrapping times of best TFHE parameters with  $\eta_C < 100\%$  for various scenarios, chosen automatically. The width of the bars represents  $l \in [1, 4]$ . Hatched bars represent incorrect results, presumably due to  $\log(\alpha) < -64$  being out of Concrete's v0.1.11 supported range.

## References

- [1] J. Klemsa and M. Önen, "Parallel Operations over TFHE-Encrypted Multi-Digit Integers," ser. CODASPY '22, 2022.
- [2] M. R. Albrecht, B. R. Curtis, A. Deo, et al., *LWE Estimator*, <https://bitbucket.org/malb/lwe-estimator>, 2018.
- [3] I. Chillotti, N. Gama, M. Georgieva, and M. Izabachène, "TFHE: fast fully homomorphic encryption over the torus," *Journal of Cryptology*, vol. 33, no. 1, pp. 34–91, 2020.
- [4] Zama, *Demo Z/8Z*, [https://github.com/zama-ai/demo\\_z8z](https://github.com/zama-ai/demo_z8z), 2021.
- [5] Zama, *CONCRETE*, <https://concrete.zama.ai>, 2021.