Security Guidelines for Implementing Homomorphic Encryption

Presenter: Erin Hales at NIST WPEC (25 September 2024)
University of Edinburgh and Royal Holloway, University of London

In collaboration with: Jean-Philippe Bossuat, Rosario Cammarota, Jung Hee Cheon, Ilaria Chillotti, Benjamin R. Curtis, Wei Dai, Huijing Gong, Duhyeong Kim, Bryan Kumara, Changmin Lee, Xianhui Lu, Carsten Maple, Alberto Pedrouzo-Ulloa, Rachel Player, Yuriy Polyakov, Luis Antonio Ruiz Lopez, Yongsoo Song, Donggeon Yhee, Bahattin Yildiz

https://ia.cr/2024/463

Motivations to standardise FHE

FHE applications and commercialisation have been advancing rapidly in recent years.

Standardisation effort gives the opportunity to:

- Consider relevant security notions for FHE.
- Agree on recommended security levels for varying parameter sets.
- Offer FHE users and practitioners guidance on selecting parameters.
- Present relevant research on FHE security to practitioners.

FHE standardisation timeline

2009

First FHE proposal

Gentry proposes the first fully homomorphic encryption scheme.
Initial implementations were slow, 30 mins per bit operation.

2018

First Homomorphic Encryption (community) standard

Recommended LWE security parameters for BGV/BFV and CKKS.
Implementations of FHE schemes have improved significantly, 2 microseconds per bit operation.

2020

ISO/IEC standardisation process begins

Study period followed by preliminary work item begins in 2020.
The ISO/IEC standardisation process for FHE is ongoing.

2021

This work, Security Guidelines for Implementing Homomorphic Encryption

Update to first (community) standard, supporting the ISO/IEC standardisation process.

2024

This talk!

More detail on this work

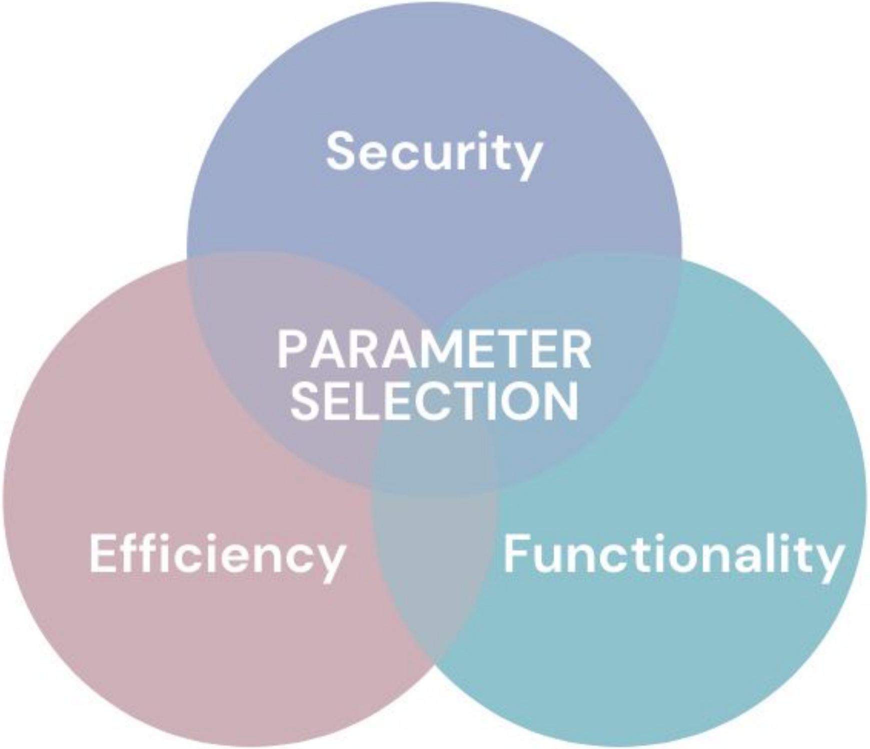
Security working group established Oct 2021, supporting ISO/IEC standardisation process, begun in Aug 2021.

20 collaborators in total from industry, academia, different libraries.

Initial goal: develop Annex to ISO/IEC documents on parameter selection.

Later goal: produce a separate white paper -- which became this work!

Parameter selection: the trade offs



Security, Correctness and Performance Tradeoffs

We need to maximise efficiency while ensuring security and correctness.

We use ‘bits of security’, which we obtain from brute force attack estimates.

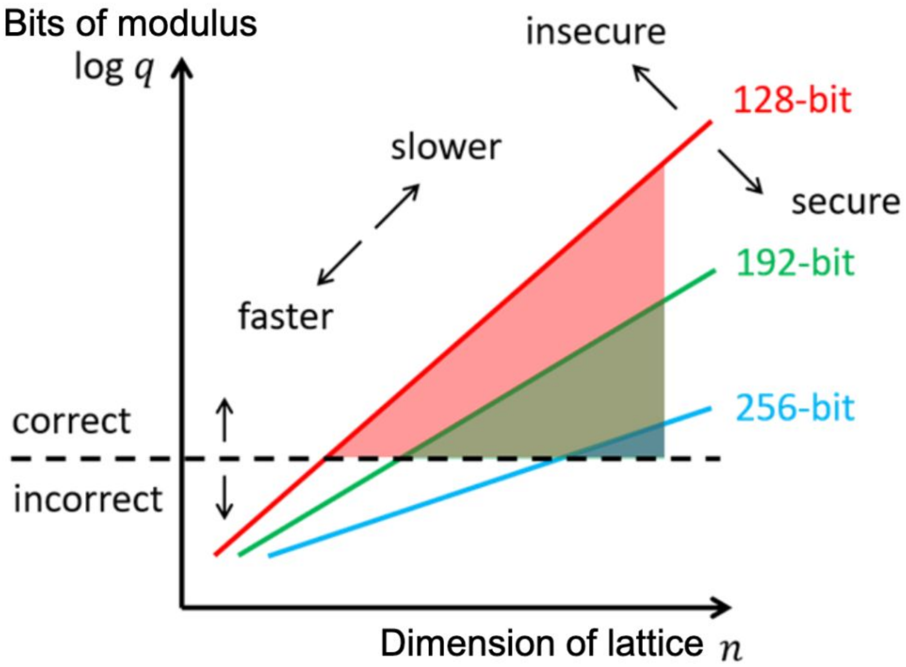
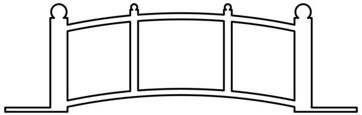
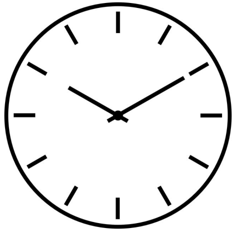


Image from here:
https://csrc.nist.gov/csrc/media/presentations/2023/stppa6-fhe/images-media/20230725-stppa6-he-fhe--kurt-rohloff.pdf

Goals of this work



Bridge the gap in security awareness between HE experts, engineers and users



Update the 2018 community standards [ACC+19]

Support ISO/IEC standardisation of FHE

Target schemes: BGV/BFV/CKKS/DM/CGGI

Outline of this work

Security Evaluation Methodology:

- Security analysis fixes a security notion and hardness assumptions.
- Target security levels.
- Security estimation tool.

Parameters:

- LWE parameter sets with target security levels.
- Scheme parameter sets as examples.
- Overview of parameter selection in open-sourced libraries and compilers.

[ACC+19]	This work
Dimensions 1024,...,32768	Dimensions 1024,...,131072
Uniform, ternary, Gaussian secrets No sparse secrets	Binary, ternary, Gaussian secrets No sparse secrets
Max log q for fixed σ	Max log q for fixed σ Min log σ for fixed q
Not easily reproducible Difficult to update	Code to reproduce all tables Can be rerun by users as needed
Only LWE parameters	Examples of full parameter sets
Describes various FHE schemes	Pointers to schemes and libraries
Describes various LWE algorithms	Pointers to cryptanalysis literature

[ACC+19] Martin Albrecht, Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Shai Halevi, Jeffrey Hoffstein, Kim Laine, Kristin Lauter, Satya Lokam, Daniele Micciancio, Dustin Moody, Travis Morrison, Amit Sahai, and Vinod Vaikuntanathan. Homomorphic encryption standard. Cryptology ePrint Archive, Paper 2019/939, 2019. https://eprint.iacr.org/2019/939.

Slide thanks to Rachel Player

Focus of security analysis

Security notion: IND-Chosen Plaintext Attack (IND-CPA).

Hardness Assumptions: Decision-Learning with Errors (LWE) and its variants, Ring-LWE (RLWE) and General-LWE* (GLWE).

Concrete security focus: parameters of the underlying LWE instances of HE.

Methodology: every instance of RLWE and GLWE is interpreted as an LWE instance.

*GLWE is often referred to as module-LWE (MLWE)

Target security levels

Category 128, 192, 256: any algorithm that solves the underlying LWE instance must require (classical) computational resources comparable to or greater than those required for key search on a block cipher with a 128-bit, 192-bit, 256-bit key respectively.

Our cost metric: (following the lattice-estimator) measure the workload in 'ring operations' (rop), which can be converted to CPU cycles for the classical computer setting if desired.

Concrete estimation: Lattice estimator

README

Security Estimates for Lattice Problems

launch binder | docs passing

This Sage module provides functions for estimating the concrete security of Learning with Errors instances.

The main purpose of this estimator is to give designers an easy way to choose parameters resisting known attacks and to enable cryptanalysts to compare their results and ideas with other techniques known in the literature.

Quick Start

We currently provide evaluators for the security of the LWE, NTRU, and SIS problems. Our estimator integrates simulators for the best known attacks against these problems, and provides bit-security estimates relying on heuristics to predict the cost and shape of lattice reduction algorithms. The default models are configured in conf.py.

It is possible to evaluate attacks cost individually, or using the helper functions:

https://github.com/malb/lattice-estimator

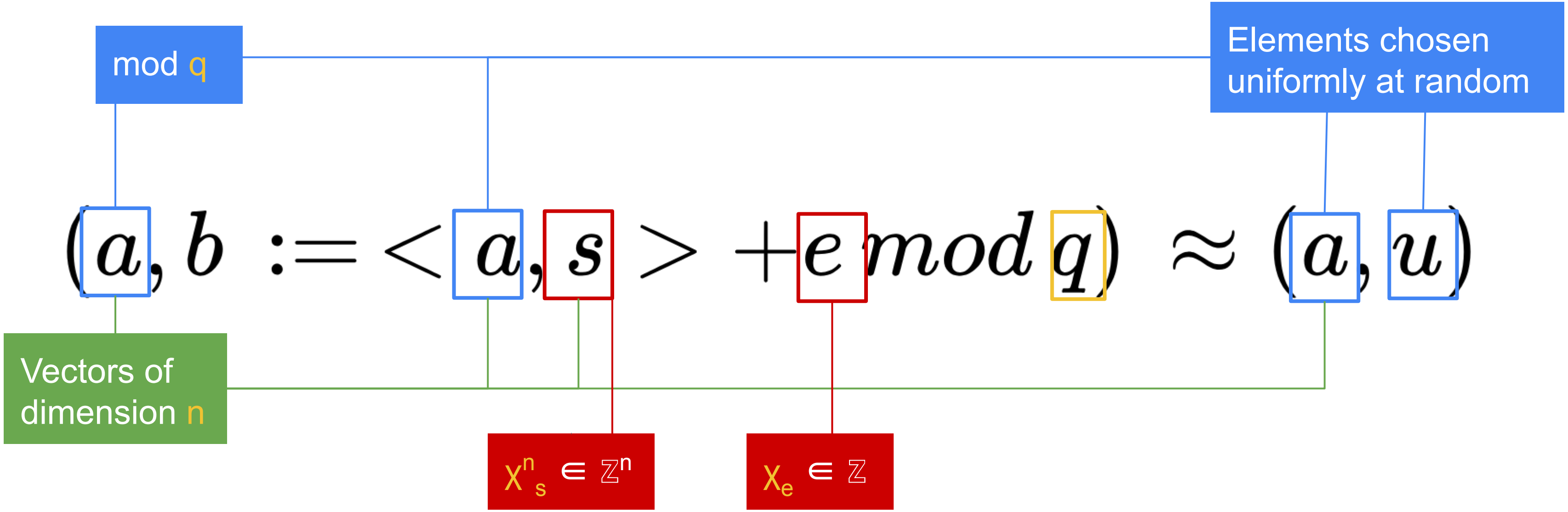
Estimator output

```
sage: param_1024_ternary_classic_128 = LWE.Parameters( n = 1024, q = 2**26, Xs
= ND.UniformMod(3), Xe = ND.DiscreteGaussian(3.19), m = oo, tag =
"param_1024_ternary_classic_128" )

sage: LWE.estimate(param_1024_ternary_classic_128)

{'arora-gb': rop: ≈2^inf, 'bkw': rop: ≈2^226.5, m: ≈2^212.3, mem: ≈2^213.3, b:
8, t1: 0, t2: 40, ℓ: 7, #cod: 933, #top: 0, #test: 91, tag: coded-bkw,
'usvp': rop: ≈2^134.1, red: ≈2^134.1, δ: 1.004234, β: 366, d: 1938, tag: usvp,
'bdd': rop: ≈2^132.3, red: ≈2^131.9, svp: ≈2^130.2, β: 358, η: 390, d: 1934,
tag: bdd,
'bdd_hybrid': rop: ≈2^132.5, red: ≈2^132.0, svp: ≈2^130.7, β: 358, η: 392, ζ:
0, |S|: 1, d: 2076, prob: 1, ↻: 1, tag: hybrid,
'bdd_mitm_hybrid': rop: ≈2^190.7, red: ≈2^189.7, svp: ≈2^189.7, β: 367, η: 2,
ζ: 142, |S|: ≈2^225.1, d: 1951, prob: ≈2^-53.1, ↻: ≈2^55.3, tag: hybrid,
'dual': rop: ≈2^137.2, mem: ≈2^88.1, m: 999, β: 373, d: 2023, ↻: 1, tag: dual,
'dual_hybrid': rop: ≈2^131.3, red: ≈2^131.3, guess: ≈2^125.2, β: 352, p: 3, ζ:
20, t: 40, β': 363, N: ≈2^74.1, m: 1024}
```

LWE



Search: Given an LWE sample (*a*,*b*), find s.

Decision: Decide if a pair (*a*,*b*) is from the LWE distribution, or uniformly random.

Slide thanks to Alberto Pedrouzo-Ulloa

Parameters

Parameter	Description
λ	Security level of the parameter set
n	Dimension of the (R)LWE instance
q	LWE modulus
σ	Standard deviation of LWE error distribution

Secure parameter sets

n	$\log_2(q)$	
	Ternary	Gaussian
$\lambda = 128$		
1024	26	28
2048	53	55
4096	106	108
8192	214	216
16384	430	432
32768	868	870
65536	1747	1749
131072	3523	3525

Maximal log of modulus q that can be used to achieve security level 128.

n	$\log_2(q)$	$\log_2(\sigma)$		
		Binary	Ternary	Gaussian
$\lambda = 128$				
630	32	18.5	17.2	14.6
1024		8.3	7.1	4.6
≥ 2048		2.0	2.0	2.0
630	64	50.5	49.2	46.6
750		47.4	46.2	43.5
870		44.3	43.1	40.3
1024		40.3	39.1	36.4
2048		13.7	12.4	10.0
≥ 4096		2.0	2.0	2.0

Minimal log of standard deviation σ that can be used to achieve security level 128.

Example parameter sets

λ	128	192	256
$\chi_{\mathbf{s}}$	Ternary	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19	3.19
t	65537	65537	786433
$\log_2(n)$	14	15	16
BFV parameters			
L[26]	10	15	18
$\log_2(Q)$	360	531	720
$\log_2(P)$	60	60	180
$\log_2(PQ)$	420	591	900
d_{num}	6	9	4
BGV parameters			
L[27]	8	13	16
$\log_2(Q)$	337	532	686
$\log_2(P)$	60	60	240
$\log_2(PQ)$	397	592	926
d_{num}	10	15	4

Table 5.5: Sample OpenFHE parameters for BFV/BGV without bootstrapping.

Example parameter sets

λ	128	128	128	128	128	128	128	128
Scheme	CGGI	CGGI	CGGI	CGGI	CGGI	CGGI	DM	DM
Library	TFHE-rs	TFHE-rs	Concrete	Concrete	OpenFHE	OpenFHE	OpenFHE	OpenFHE
n	841	785	805	687	503	556	447	556
$\log_2(N)$	11	9	11	9	10	10	10	10
k	1	4	1	3	1	1	1	1
q	2^{64}	2^{64}	2^{64}	2^{64}	$\approx 2^{27}$	$\approx 2^{27}$	$\approx 2^{28}$	$\approx 2^{27}$
q_{ks}	2^{64}	2^{64}	2^{64}	2^{64}	$\approx 2^{14}$	$\approx 2^{15}$	$\approx 2^{14}$	$\approx 2^{15}$
t	2^4	2	2^4	2	2	2	2	2
χ_{LWE}	Binary	Binary	Binary	Binary	Ternary	Ternary	Gaussian	Ternary
χ_{GLWE}	Binary	Binary	Binary	Binary	Ternary	Ternary	Gaussian	Ternary
β_{ks}	2^3	2^4	2^3	2^4	2^5	2^5	2^5	2^5
ℓ_{ks}	5	3	5	3	3	3	3	3
β_{pbs}	2^{22}	2^{23}	2^{15}	2^{18}	2^9	2^7	2^{10}	2^9
ℓ_{pbs}	1	1	2	1	3	4	3	3
σ_{LWE}	$2^{45.72}$	$2^{47.22}$	$2^{15.68}$	$2^{45.99}$	3.19	3.19	3.19	3.19
σ_{GLWE}	$2^{15.68}$	$2^{14.05}$	$2^{14.05}$	$2^{49.02}$	3.19	3.19	3.19	3.19
p_{error}	2^{-64}	2^{-64}	2^{-64}	2^{-64}	2^{-40}	2^{-220}	2^{-55}	2^{-120}

Table 5.6: Sample parameters for CGGI and DM. The first two parameter sets for CGGI (with $n =$

λ	128	192	256
$\log_2(n)$	14	15	16
$\log_2(q)$	424	585	920
$\log_2(t)$	20	20	20
$\chi_{\mathbf{s}}$	Ternary	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.2	3.2	3.2
L (BFV)	10	14	23
L (BGV)	8	12	19

Table 5.4: Sample SEAL parameters for BFV/BGV without bootstrapping.

λ	128	192	256
$\log_2(N)$	14	15	15
$\chi_{\mathbf{s}}$	Ternary	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19	3.19
Base Prime Size	40	43	40
L	7	9	7
$\log_2(PQ)$	427	592	434
$\log_2(Q)$	307	412	314
$\log_2(P)$	120	180	120
$\log_2$ (Scaling Factor)	38	41	39
Precision Bit	22.3	24.0	22.2

Table 5.7: Sample parameters for RNS-CKKS without bootstrapping

	Set I	Set II
λ	128	128
$\log_2(N)$	16	16
Number of Slots[32]	32768	32768
$\chi_{\mathbf{s}}$	Ternary	Ternary
σ ($\chi_{\mathbf{e}}$)	3.19	3.19
Base Prime Size	45	60
L (after bootstrapping)	10	6
$\log_2$(Scaling Factor)	35 [33]	58
$\log_2(PQ)$	1734	1691
$\log_2(Q)$	1464	1511
$\log_2(P)$	305	180
Level cost of SlotsToCoeffs	4	3
Level cost of EvalMod	12	13
$\log_2(\mathsf{Pr}[\|\|I(X)\|\| > K])$[34]	-37.65	-37.65
K	512	512
Level cost of CoeffsToSlots	3	3
Iterations[35]	1	1
Precision Bits[36]	15.9	12.0[37]

Table 5.8: Sample parameters for RNS-CKKS with bootstrapping.

Cryptanalytic advances: how to update?

Predicting future cryptanalytic progress is challenging. Instead of fixing a security margin t for the next x years, we offer scripts* which:

- can be **rerun to update parameters** if lattice-estimator is updated in the future.
- offer **flexible adjustments** if users wish to adopt a different cost model or include a new attack.

*Scripts for reproducing and verifying tables can be found at https://github.com/gong-cr/FHE-Security-Guidelines

Looking forward

Expand the scope: as FHE matures, include,

- more schemes
- diverse distributions
- broader attack scenarios

Parameter selection: develop advanced automated frameworks for systematic parameter selection that balance security, functionality, and efficiency.

Key Takeaways

Parameters can, and do, change as a result of advances in cryptanalysis.

For implementers, following up-to-date security guidelines is essential.

This work includes:

- Parameter set examples for major FHE schemes/libraries.
- New tools enabling users to independently update parameters.

Thank you!

For more details, see eprint: https://ia.cr/2024/463
Scripts for reproducing and verifying tables can be found at
https://github.com/gong-cr/FHE-Security-Guidelines
There will be a breakout session on FHE security at the 7th HES meeting, affiliated with CCS in Salt Lake City on October 13
https://homomorphicencryption.org/7th-homomorphicencryption-org-standards-meeting/