Security Guidelines for Implementing Homomorphic Encryption

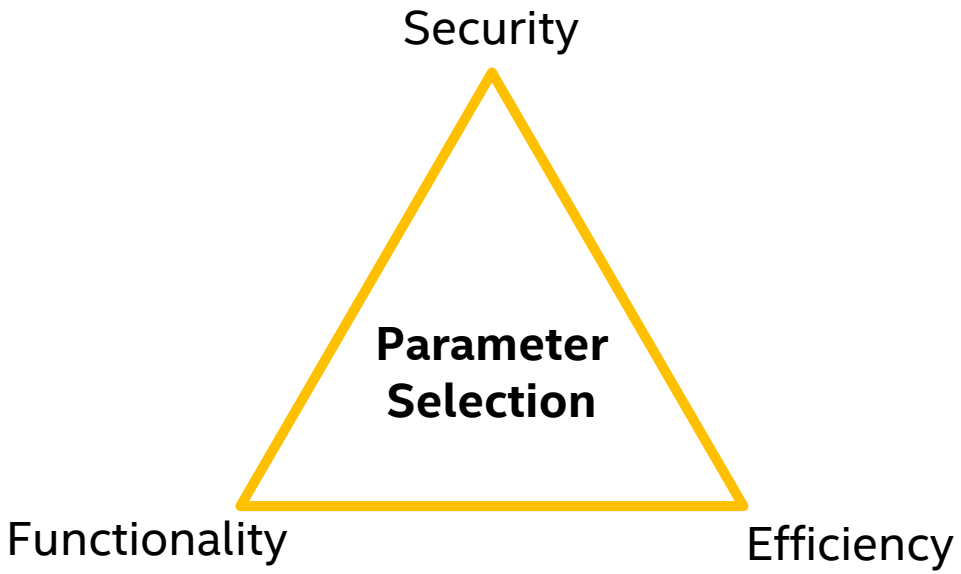
Presenter: Huijing Gong (Intel Labs)

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

FHE.org in Toronto
March 24th , 2024

The Rising Demands: Selecting Parameters Securely

- Challenges



- Bridging the gap in security awareness among HE experts, engineers and end-users.
- Updating the 2018 HE security white paper [ACC+19].
- Supporting ISO/IEC standardization on FHE.
  - Targeted schemes: BGV/BFV/CKKS/CGGI.

[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.

Evolving Security Guideline: Comparison to [ACC+19]

	Our Work	[ACC+19]
LWE security parameters	**Broader choices of dimensions and distributions**: • Expands secret distributions: Ternary, Gaussian, Binary (CGGI); • Broadens standard deviation (σ) range of error distribution. • Includes dimension up to 131072. **Applicable for BGV/BFV/CKKS/CGGI.**	• Excludes binary secret distribution. • Limited to fixed σ. • Max dimension = 32768. **Applicable to BGV/BFV/CKKS.**
	Updates with cryptanalysis. Provides open-source tools for individual parameter generation**.**	Potentially outdated cryptanalysis. Lacks tools for parameter updates.

[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.

Evolving Security Guideline: Comparison to [ACC+19]

	Our Work	[ACC+19]
Scheme parameter examples	BGV/BFV/CKKS/CGGI.	Not included.
Other contents	Only provides references for FHE constructions and attacks. Brief discussions includes: • NTRU-based FHE • Machine learning Attacks • Side-Channel Attacks • IND-CPA^D	Provides details for FHE constructions/Attacks Discussion on other features.

[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.

Outline of This Work

- Security Evaluation Methodology.
  - Focus of security analysis: notion and hardness assumptions.
  - Security levels.
  - Security estimation tool.
- Parameters.
  - LWE parameter sets with target security levels.
  - Scheme parameter sets as examples.
  - Parameter selection in open-sourced libraries and compilers.

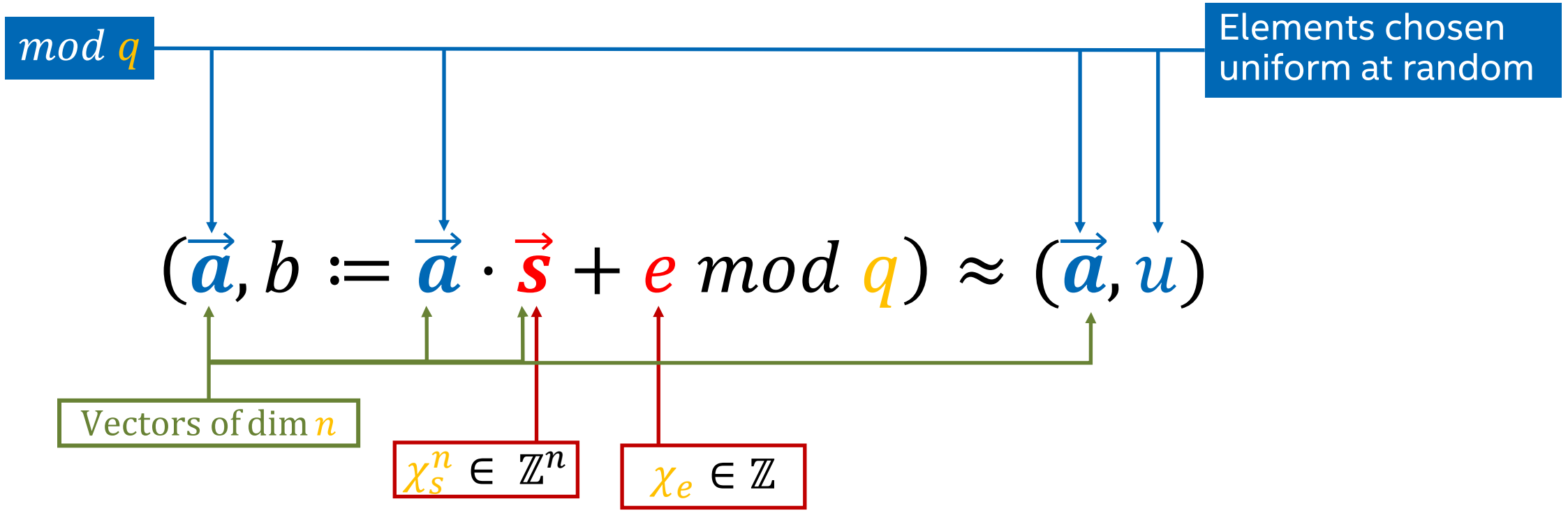
Focus of Security Analysis

- 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.
  - Every instance of RLWE and GLWE is interpreted as an LWE instance.
    - As their algebraic structures for practical applications has not yet been exploited.

*Module-LWE in many literatures.

LWE and GLWE (n, q, χ_s, χ_e)

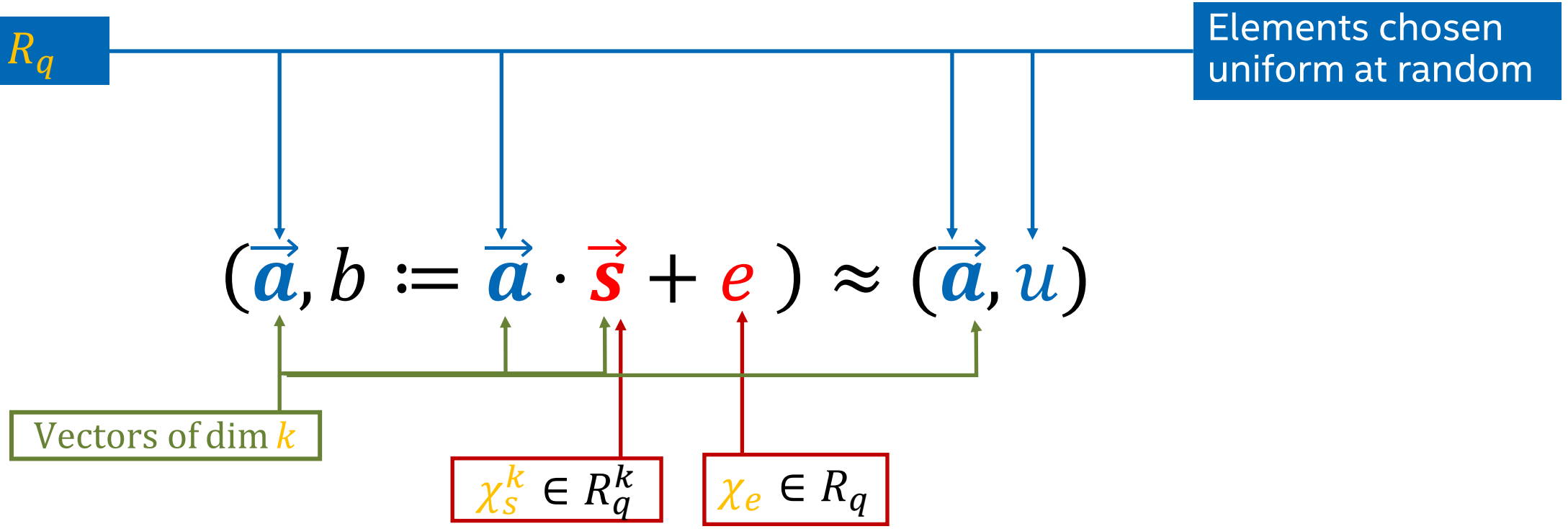
- **LWE**: Parametrized by (n, q, χ_s, χ_e). The computational indistinguishability between the following pairs of samples



Slide courtesy of Alberto Pedrouzo-Ulloa

LWE and **GLWE** $(n = (kN), q, \chi_s, \chi_e)$

- **GLWE***: Parametrized by (R_q, k, χ_s, χ_e), where R_q is an (e.g. cyclotomic) polynomial ring of degree N with modulus q.



Slide courtesy of Alberto Pedrouzo-Ulloa

Error Distributions χ_e

- **Hardness Assumption**: If the standard deviation (σ) of χ_e is $\Omega(\sqrt{n})$, the best-known algorithm to solve the LWE problem requires exponential time [Reg10].
- **Practical Choices**: much narrower distributions.
  - Standard deviations of **Gaussian distribution** (σ):
    - **Power of 2 cyclotomic ring**: a very common choice is $\sigma \approx 3.2$* [ACC+19,HS20].
    - Non power of 2 kth cyclotomic ring: $\sigma_{non} = \sigma\sqrt{k}$ [HS20].
  - An alternative from FIPS 203 (draft) [oST23]: a centered binomial distribution, with higher efficiency and constant-time sampling.

*σ = 3.19 for generating security parameter table in later section

Secret Distributions χ_s

- **Hardness Assumption:**
  - Uniform Secret: Coefficients are uniformly random from $\mathbb{Z}_q$.
  - Normal Form Secret: Coefficients follow the error distribution χ_e in hardness assumption.
- **Practical Choices:**
  - **Gaussian Secret: Coefficients are sampled from 0 centered narrow Gaussian distribution.**
  - **Ternary Secret: Coefficients are uniformly random from the set {−1, 0, 1}.**
  - **Binary Secret: Coefficients are uniformly random from the set {0, 1}.**
  - Fixed Hamming Weight Secret: "Exactly h coefficients are non-zero (either 1 or −1).
    - Sparse secret keys: when h is small (e.g., $h < 0.25 \cdot n$).
  - These distributions may account for different attacks and FHE scheme efficiencies.

Concrete Security Estimation

- Security levels: Adapted from NIST PQC standardization.
  - Quantum Security Levels (128Q, 192Q, 256Q): Equivalence to the cost of quantum computer required to break AES with corresponding key sizes.
  - Classical Security Levels (128, 192, 256): Equivalent values in the cost metric of classical computer.

Concrete Security Estimation

- Tool: Lattice-estimator (https://github.com/malb/lattice-estimator)
  - Cost models for lattice reduction core subroutine (BKZ):
    - Classical setting RC.BDGL16: $T_{BKZ}(\beta, d) = 8d \cdot 2^{0.292\beta+16.4}$.
    - Quantum setting RC.LaaMosPol14: $T_{BKZ}(\beta, d) = 8d \cdot 2^{0.265\beta+16.4}$.
  - Cost metric (as of used by lattice-estimator):
    - Measuring the workload in 'ring operations' (rop), can be converted to CPU cycles in classical computer setting.

Establishing Security LWE Parameters

- LWE Parameter sets for BGV/BFV/CKKS/CGGI targeting security levels 128/192/256 and 128Q/192Q/256Q.
- Parameters verified from the latest lattice-estimator as of 2/27/2024.
- Sets validated against key attacks: primal-usvp, primal-bdd, hybrid-bdd*, hybrid-dual.

*Up to n = 16384.

Security LWE Parameter Sets (BFV/BGV/CKKS)

n	$\log_2(q)$ (Classical)		$\log_2(q)$ (Quantum)	
	Ternary	Gaussian	Ternary	Gaussian
	$\lambda = 128$			
1024	26	29	25	27
2048	54	56	50	52
4096	108	110	101	103
8192	217	219	203	205
16384	438	439	409	411
32768	881	883	825	827
65536	1776	1778	1663	1665
131072	3576	3578	3348	3351

- Table 4.2 (max log q):
  - Fixed Gaussian error distribution ($\sigma = 3.19$) and **variable $log_2\ q$ upper bound**.
  - Ranges from $n = 1024$ (for 128/128Q)* up to 2^{17}.
  - Using ternary and Gaussian ($\sigma = 3.19$) secret distributions.
  - Security levels: 128/192/256/128Q/192Q/256Q.

*n ≥ 2048 for 192/192Q/256/256Q.

Security LWE Parameter Sets (CGGI)

n	$\log_2(q)$	$\log_2(\sigma)$ (Classical)			$\log_2(\sigma)$ (Quantum)		
		Binary	Ternary	Gaussian	Binary	Ternary	Gaussian
		$\lambda = 128$					
630	32	17.9	16.6	14.2	18.9	17.7	15.4
1024		7.6	6.3	4.5	9.2	8.0	6.3
≥ 2048		2.0	2.0	2.0	2.0	2.0	2.0
630	64	49.9	48.6	46.2	50.9	49.7	47.4
750		46.8	45.5	43.0	48.0	46.7	44.4
870		43.7	42.4	39.9	45.0	43.8	41.4
1024		39.6	38.3	36.1	41.2	40.0	37.9
2048		12.6	11.4	9.4	16.0	14.8	12.7
≥ 4096		2.0	2.0	2.0	2.0	2.0	2.0

- Table of 4.3 (min log σ):
  - Fixed $\log_2 q$ (32/64-bit), **variable lower bound of $log_2\sigma$.**
  - n is not restricted to a power of two.
  - Secret distributions: Binary, Ternary, Gaussian ($\sigma_s = 4$).
  - Security levels: 128/192/256/128Q/192Q/256Q.

Updating Tables in Response to Cryptanalysis Advances

- Predicting future cryptanalytic progress is challenging. Instead of fixing a security margin t for next x years, we offer:
- Scripts*:
  - Rerun to update parameters if lattice-estimator is updated in the future.
  - Flexible adjustments: Users can modify settings to adjust for various cost models or attacks.

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

Scheme Parameter Set Examples

- Provide scheme parameter examples to meet specific security targets:
  - BGV/BFV*: Example of somewhat HE (SHE) on SEAL.
  - CGGI: Examples by TFHE-rs, TFHElib, and OpenFHE.
  - CKKS*:
    - SHE: Examples on OpenFHE.
    - FHE: Examples on Lattigo and OpenFHE.

*Scripts for reproducing BGV/BFV/CKKS tables can be found in https://github.com/WeiDaiWD/SEAL-Depth-Estimator and https://github.com/gong-cr/FHE-Security-Guidelines.

Parameter Selection in Open-sourced FHE libraries

- Provide a survey for parameter selection among various FHE libraries and compilers.
- Survey highlights the critical need for a systematic approach to parameter selection across FHE libraries.

Library	Link	BFV	BGV	CKKS	CGGI	Note
blyss	blysprivacy/sdk					Combines GSW and basic LWE.
Cingulata	CEA-LIST/Cingulata	✓				Also a compiler toolchain for its own BFV implementation and for TFHElib.
Cupcake	facebookresearch/Cupcake					Only implements of the additive version of BFV.
FHE-DECK	FHE-Deck/fhe-deck-core					Contains only the basics for RLWE and NTRU infrastructure.
FHELib	Crypto-TII/fhelib		✓			
HEaaN	cryptolabinc/heaan		✓	✓		Proprietary. Free for non-commercial usage.
HELib	homenc/HElib		✓	✓		
HEHub	primihub/hehub		✓	✓	✓	
HEU	secretflow/heu			✓	✓	Contains additive homomorphic encryption. FHE algorithms still in development.
Lattigo	tuneinsight/lattigo	✓	✓	✓	✓	
Liberate. FHE	Desilo/liberate-fhe			✓		
NFLLib	quarkslab/NFLlib	✓				
OpenFHE	openfheorg	✓	✓	✓	✓	
Parmesan	crates/parmesan					Builds on TFHE-rs.
Phantom	encryptorion-lab/phantom-fhe	✓	✓	✓		
Poseidon	luhang-HPU/Poseidon	✓	✓	✓		
REDcuFHE	TrustworthyComputing/REDcuFHE				✓	
SEAL	microsoft/SEAL	✓	✓	✓		
TFHE-rs	zama-ai/tfhe-rs				✓	
TFHElib	tfhe/tfhe				✓	

ble 4.8: Open source homomorphic encryption libraries and the algorithms they supp

Journey of Collaboration: Sep. 2021 --

- Our team expanded to 19 researchers worldwide, from industry and academia
  - including Intel Labs, Royal Holloway University of London, Zama, and 10+ other institutions.
- Expertise in
  - Lattice cryptanalysis.
  - Major FHE schemes with their variants.
- Regularly meetings have integrated diverse expertise, fostering numerous consensus and innovative solutions.
- To be continued…

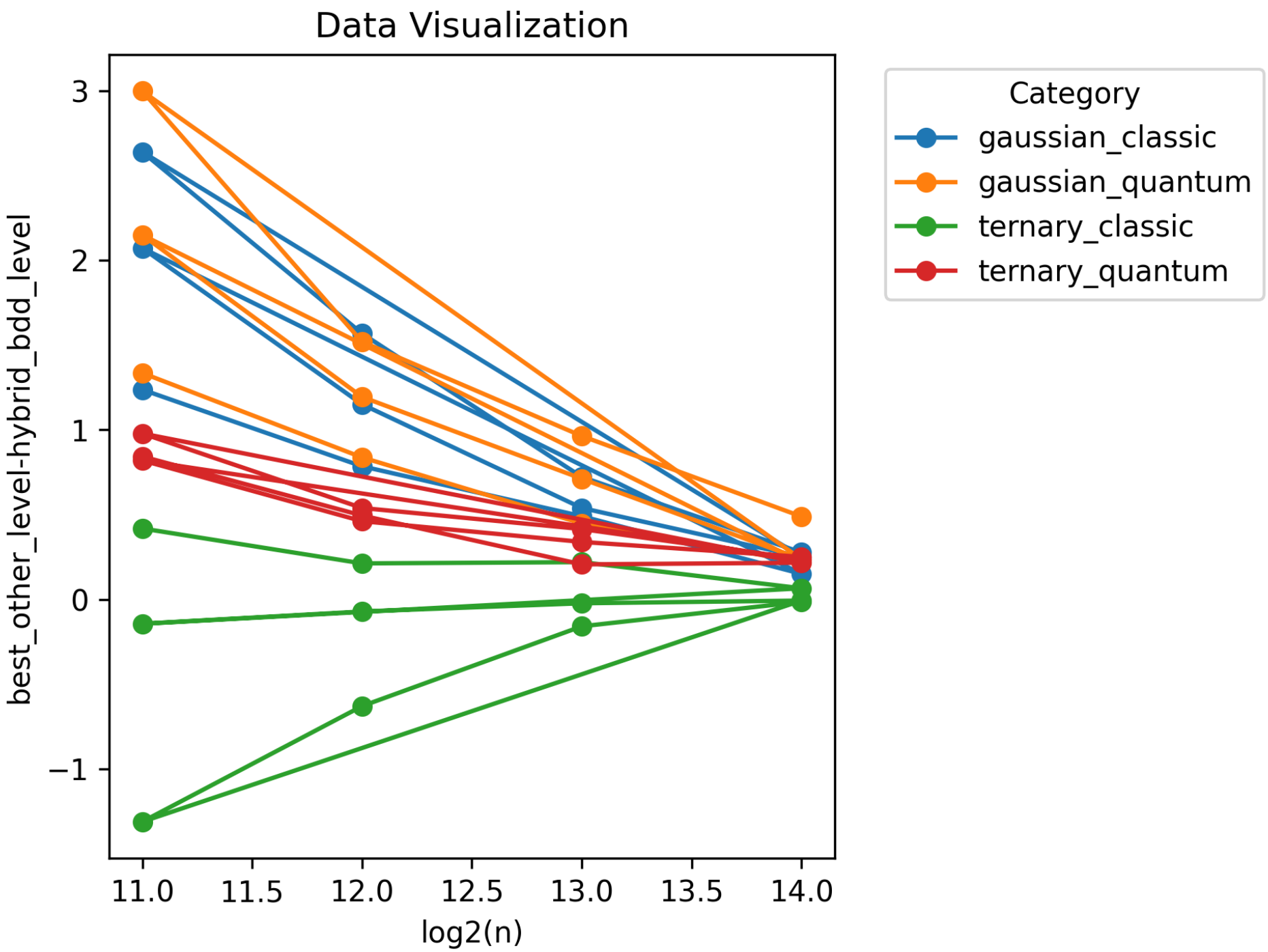
Conclusion & Future Directions

- Key Takeaways:
  - For those implementing FHE schemes, **up-to-date security guidelines are essential**.
    - Revitalized parameters in response to latest advanced cryptanalysis.
    - Scheme parameter set examples for major FHE schemes/libraries.
    - New tools enabling users to independently update parameters
- Future directions:
  - **Expand the scope**: as FHE matures, include more schemes (e.g. NTRU-based), diverse distributions, and broader attack scenarios.
  - **Parameter selection**: Develop advanced automated frameworks for **systematic parameter selection** that balances security, functionality, and efficiency.

THANK YOU

https://eprint.iacr.org/2024/463

https://github.com/gong-cr/FHE-Security-Guidelines.
https://github.com/WeiDaiWD/SEAL-Depth-Estimator


Data Visualization
best_other_level-hybrid_bdd_level
log2(n)
Category
gaussian_classic
gaussian_quantum
ternary_classic
ternary_quantum
3
2
1
0
−1
11.0
11.5
12.0
12.5
13.0
13.5
14.0