2018 IEEE Asia Pacific Conference on Circuits and Systems

Error Samplers for Lattice-Based Cryptography - Challenges, Vulnerabilities and Solutions

Ayesha Khalid*, Ciara Rafferty*, James Howe†, Séamus Brannigan*, Weiqiang Liu‡, Máire O'Neill*
* Centre for Secure Information Technologies (CSIT), ECIT, Queen's University Belfast, UK
† Department of Computer Science, University of Bristol, UK
‡ College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, China
Email: {a.khalid, c.m.rafferty, sbrannigan11, maire.oneill}@qub.ac.uk, james.howe@bristol.ac.uk, liuweiqiang@nuaa.edu.cn

***Abstract*—Lattice based cryptography (LBC) stands out today as one of the most promising types of post-quantum cryptography, and a strong contender in the ongoing NIST post-quantum cryptography standardisation process. LBC algorithms are advantageous due to their efficiency, versatility and the hardness of their underlying lattice problems. In this work, the practicality of LBC is explored by surveying one of the critical components, *the error samplers*, and highlighting the challenges associated with their efficient, secure implementation. Side channel attack (SCA) vulnerabilities and associated countermeasures are considered, concluding with error sampler recommendations, to aid the practicality, security and future widespread deployment of LBC.**

***Index Terms*—sampling, post-quantum cryptography, Gaussian, lattices, lattice-based cryptography**

I. Introduction

Current public-key security infrastructure will soon require a significant update, since its security may be compromised by a scalable quantum computer in the near future. Shor's algorithm, running on a quantum computer, can solve the integer factorization and discrete logarithm problems in polynomial time [1], on which currently used public key algorithms are based. This threat has resulted in an active area of research, known as *quantum-resilient* or *post quantum* cryptography (PQC), providing recommendations to transition to quantum-resistant public-key cryptography in the near future, from academia, industry and government agencies, including NSA and CESG [2], [3]. In 2016, NIST called for quantum-resilient cryptographic algorithms for standardisation [4].

Of the various flavours of quantum-resilient cryptography submitted to the NIST PQC competition, lattice-based cryptography (LBC) makes the most populous class (29 out of 71 PQC schemes). LBC stands out primarily because of the algorithmic hardness of the underlying lattice problems, efficient implementations due to inherent linear algebraic operations and extended functionality for advanced security services such as identity-based encryption (IBE) and fully-homomorphic encryption (FHE), in addition to the basic classical cryptographic primitives (encryption, signatures, key exchange). Google has demonstrated LBC key exchange in TLS [5].

Implementations of LBC raise several unique challenges. None of the commonly used underlying LBC building blocks, e.g., Discrete Gaussian samplers (DGS), fast Number Theoretic Transforms (NTTs) and compact cryptographic hashes, are part of traditional asymmetric cryptography used today. This survey focuses on the DGS used within LBC proposals. Secure implementation of DGS is challenging, due to inherent performance limitations and exploitable side channel vulnerabilities. This work surveys the requirements of error samplers for all LBC submissions to the NIST PQC competition. We chalk out the use of alternate schemes instead of DGS. Any DGS side channel vulnerabilities that have led to successful attacks are surveyed and appropriate countermeasures are discussed. Recommendations for efficient DGS on software and hardware platforms conclude the paper.

II. Background

A. Lattice-Based Primitives

Lattices are objects in n-dimensional Euclidean space characterized by a regular arrangement of points. A number of hard mathematical problems are used to construct lattice-based schemes, such as the Short Integer Solution (SIS) or the NTRU assumption. The most common problem is *Learning with Errors* (LWE), which involves finding a vector $\mathbf{s}$ when given a matrix $\mathbf{A}$ and a vector $\mathbf{b} = \mathbf{As} + \mathbf{e}$ where $\mathbf{e}$ is a small (unknown) error vector. The absence of this noise would give away secret information via Gaussian elimination. Discrete Gaussian samplers (DGS) are typically employed to generate this noise as they allow for efficient implementations, with smaller output sizes (ciphertexts or signatures).

There are several classes of lattices: Schemes based on LWE are known as **standard lattice-based** schemes. These schemes require matrix-vector multiplication using large memories and quadratic complexity. **Ideal or ring lattice-based** schemes alternatively represent the matrix by a single row, and the remaining rows are generated by cyclic shifts of the first row. This reduces matrix-vector multiplication to polynomial multiplication and is memory-computation efficient. To provide a trade-off between efficiency and security, **module lattices** are introduced [6]. The difference between module over standard lattices is that the associated module-lattice matrix has small dimensions and the matrix coefficients are no longer simple integers but polynomials. Polynomial schemes are a generalisation of ring lattice-based schemes. The security of ring and module lattice-based schemes is based on variants of the original mathematical problems, e.g. Ring-LWE, Module-SIS.

978-1-5386-8240-1/18/$31.00 ©2018 IEEE

III. Classification of LBC candidates in the NIST standardisation process

Table I presents a comprehensive summary of the lattice-based schemes submitted to the NIST PQC standardisation process [4] and their related lattice classes. The table shows for which key exchange (KEM)/ public key encryption (PKE) schemes the authors claim chosen ciphertext attack (CCA) or adaptive chosen ciphertext attack (CCA2) security in addition to the NIST PQC requirement of chosen plaintext attack (CPA) security. *CPA security* implies that the scheme is mathematically secure against an attacker with limited access to plaintext/ciphertext pairs; *CCA security* implies that an attacker also has access to a decryption oracle. This can be extended by assuming an adaptive attacker (*CCA2*). For most submitted signature schemes, the authors claim *EUF-CMA* security, which means that a signature is existentially unforgeable under chosen-message attacks. Thus, an attacker with access to a signing oracle is unable to forge a valid signature of a new message. Strong existential unforgeability under Chosen Message Attacks (SEUF-CMA) is a stronger security notion that assumes that an attacker is unable to forge a different signature of a message that he has already seen.

IV. Error samplers for Lattice-Based Cryptography

The key LBC components include linear algebraic operations and sampling from a discrete Gaussian-distributed random source. In addition to traditional *rejection sampling*, several other techniques have been proposed, including Cumulative Distribution Table (CDT) sampling (inversion sampling), Knuth-Yao sampling and Box Muller sampling. All schemes have advantages depending on the target application. However, tackling side-channel vulnerabilities is critical. Most of the proposed PKE and KEM LBC-schemes require a DGS with a small standard deviation ($\sigma < 10$, generally), as seen in Table I. σ controls the dispersion of the samples from the mean and depends on the modulus used. The three main types of DGS featuring in the NIST LBC submissions are:

- **Knuth-Yao**: The Knuth-Yao sampler is a tree-based algorithm for sampling from non-uniform distributions by using a minimal number of input uniform bits, close to the entropy of the probability distribution. The scheme has a very *compact* memory footprint, but needs additional data scrambling to make the generated Gaussian samples time-independent [7]–[10].
- **CDT**: The CDT sampler requires a precomputed table of discrete Gaussian cumulative distribution function (CDF) values and uses binary search with complexity of $log2(N)$ comparisons to generate a sample. The technique offers a reasonable data footprint and inherent constant time execution [10].
- **Box-Muller**: The Box-Muller transform produces a pair of Gaussian random numbers from a pair of uniform numbers. It generates the magnitude and phase of a vector of which the two Cartesian coordinates are the output Gaussian numbers [11], [12]. It requires calculation of log2, sin, cosine and consequently has platform limitations and limited precision. For higher precision and/or constant-time Gaussian samplers, the authors of NTRUEncrypt and pqNTRUSign schemes recommend using an alternate sampler [13].

An alternative way to approximate the normal distribution is via the **binomial distribution**. The binomial distribution enables easier, efficient sampling in constant time, in comparison with DGS. This approximation works well for *small sigmas*, as the exact distribution shape can be shown to have sufficient security equivalence under appropriate conditions using the Renyi divergence technique [14]. For qTESLA, Gaussian sampling from the Bernoulli distribution, a special case of the binomial distribution, is used for key generation [15]. DGS techniques are used to output a variety of distributions, i.e. discrete Gaussian, rounded continuous Gaussian, or binomial.

Efficient, side-channel resistant implementations of DGS schemes are non-trivial and to date limited research has been conducted. As seen in Table I, at least 9 candidates employ DGS, using a variety of aforementioned methods, with sigmas ranging from 1.2 to 107. Hardware designs of samplers have explored constant-time implementations [8], [10], [16]. For signature schemes, usually requiring large sigmas, hierarchical CDT sampling has been proposed for compact, efficient sampling in hardware [17]. Box-Muller sampling has been shown to be efficient on software platforms and has been implemented in constant time to produce rounded Gaussian samples for the BLISS LBC signature scheme [12].

A. SCA vulnerabilities and countermeasures for samplers

Physical attacks against lattice-based constructions are emerging. Physical attack resistance is also a fundamental parameter for the NIST standardization process. Timing attacks, introduced by Kocher [18], exploit the time difference required to perform specific operations, such as the non-constant time to execute different instructions, different data fetch times due to cache memory hit/miss, program behaviour due to branching, and optimisations that skip unnecessary operations.

Discrete Gaussian samplers have been shown to be vulnerable against timing attacks, for example information leaked via cache memory by a CDT based Gaussian sampler has been successfully extracted [19]. To unlink the timing information from sampling, Roy *et. al* proposed the use of Fisher-Yates shuffling [20] with Knuth-Yao DGS [8], [9]. Saarinen [21] suggested shuffling be carried out twice on the set of independently generated samples, before summation. Recent research shows that relying solely on two-stage shuffling may not be sufficient to protect against SCA attacks [22]. Consequently, *multiple sampling and shuffling stages* together with the use of *different convolution parameters* are recommended to ensure adequate protection [22], [23].

The simplest countermeasure against timing attacks is to ensure the execution time of an implementation is *independent* of the secret data. However, for Gaussian samplers, it is often expensive to ensure constant-time implementations. Several

TABLE I

CLASSIFICATION, CONSTRUCTION AND SALIENT FEATURES OF LATTICE BASED SUBMISSIONS TO NIST PQC COMPETITION. CLASS REFERS TO ENCRYPTION AND KEY EXCHANGE AS PKE AND KEM RESPECTIVELY. NIST SECURITY REFERS TO PARAMETER SECURITY AS DEFINED BY NIST FOR PQC STANDARDISATION, RANGING FROM 1 (EQUIVALENT TO AES128) TO 5 (EQUIVALENT TO AES256). NOTE THAT SOME SCHEMES BASE THEIR SECURITY ON MULTIPLE ASSUMPTIONS. NOTE THAT LEPTON IS BASED ON RING-LPN. THE **POLYNOMIAL LATTICES** CLASS IS VERY SIMILAR TO RING LATTICES AND FOR POWER-OF-TWO DIMENSIONS IS EQUIVALENT. CSPRNG STANDS FOR CRYPTOGRAPHICALLY SECURE PSEUDO RANDOM NUMBER GENERATOR. NIST PRNG IS A STANDARD FOR DETERMINISTIC RANDOM BIT GENERATION (DRBGS) [25]; AN ANALYSIS OF NIST DRBGS CAN BE FOUND IN [26]. DETAILS OF THE ROUND 1 SUBMITTED SCHEMES CAN BE FOUND IN [27]

		Lattice Type				Class				Salient Features		
		Standard	Ring	Module	Polynomial	PKE	KEM	Signatures	NIST Security	DGS/Binomial	CSPRNGs/ Hashes	Security
1.	Lepton		✔				✔		1,3, 5	none	SHAKE128/cSHAKE128 as XOF/PRNG	IND-CCA
2.	Odd Manhattan	✔					✔		1,3,5	none	NIST PRNGs	IND-CCA
3.	LOTUS	✔				✔	✔		1,2,3,4,5	DGS (Knuth-Yao), $\sigma = 3, \mu = 0$	NIST PRNG, SHA-512 as XOF/Hash	IND-CCA2
4.	Compact-LWE	✔				✔			3	none	NIST PRNGs	IND-CCA2
5.	Giophantus	✔				✔			1,3,5	none	NIST PRNGs, SHAKE256 as XOF/Hash	IND-CCA2
6.	FRODO	✔					✔		1,5	DGS (CDT), $\sigma = 2.3, 2.75, \mu = 0$	NIST PRNG, cSHAKE128/256 as XOF/Hash	IND-CCA
7.	DRS	✔						✔	1,3,5	none	NIST PRNGs, SHAKE512 as XOF/Hash	EUF-CMA
8.	Lizard	✔	✔			✔	✔		1,3,5	DGS (CDT, 9 entries)	NIST PRNGs, SHAKE512 as XOF/Hash	IND-CCA
9.	Round2	✔	✔			✔	✔		1,2,3,4,5	none	NIST AES XOF	IND-CCA KEM
10.	KCL	✔	✔	✔		✔	✔		3,5	Centered Binomial Distribution, $k = 2$	NIST PRNGs, SHA-1 as Hash	IND-CCA
11.	EMBLEM/ R. EMBLEM	✔	✔			✔	✔		1	DGS, (CDT, 54 entries), $\sigma = 25$	NIST PRNG, SHA-256 as XOF/Hash	IND-CCA2
12.	NTRU Prime		✔				✔		5	Wide DGS, $\sigma = 1.2 - 1.9$	NIST PRNG, SHA-512 as XOF/Hash	IND-CCA2
13.	NTRU Encrypt		✔			✔	✔		1,3,5	DGS (Box-Muller)	NIST PRNG (salsa20), SHA-256 as XOF/Hash	IND-CCA2
14.	Ding		✔				✔		1,3,5	DGS (CDT), $\sigma = 2.6, 4.19$	NIST PRNGs	IND-CPA
15.	KINDI			✔		✔	✔		2,4,5	none	NIST PRNGs, SHAKE256, SHAKE512 as XOF/Hash	IND-CCA
16.	LIMA		✔				✔		3	Centered Binomial Distribution, $\sigma = 3.16$	NIST PRNGs as XOF	IND-CCA
17.	NewHope		✔				✔		1,3	Centered Binomial Distribution	NIST PRNG, SHAKE128 and SHAKE256 as hash	IND-CCA
18.	HILA5		✔				✔		5	Centered Binomial Distribution	NIST PRNG, SHAKE256 as XOF/ hash	IND-CCA
19.	NTRU-HRSS-KEM		✔			✔	✔		-	Centered Binomial Distribution, $k = 2$	NIST PRNG, SHAKE128 as XOF/hash	IND-CCA2
20.	Mersenne -756839		✔				✔		5	DGS, $\sigma = 28.64$	NIST PRNG as XOF	IND-CCA
21.	qTESLA		✔					✔	1,3,5	Bernoulli distribution, $\sigma = 8.5, 10$	cSHAKE256 as PRNG, SHA-3 as Hash	EUF-CMA
22.	Falcon		✔					✔	1,2,3,4,5	Trapdoor sampling in Fourier domain, DGS-rejection sampling of bimodal distribution	ChaCha20, AES based NIST PRNG, SHAKE256 hash	EUF-CMA
23.	pqNTRUSign		✔			✔	✔	✔	1	DGS (Box-Muller), $\sigma = 107$	NIST PRNG, SHA-2, SHA3-512 as hash	IND-CCA2 KEM
24.	CRYSTALS-Kyber			✔			✔		1,3,5	none	NIST PRNG, SHA-3-256, SHA-3-512, SHAKE128, SHAKE512 as XOF, Hash	IND-CCA2
25.	SABER			✔			✔		1,3,5	none	NIST PRNG, SHAKE128 (as XOF), SHA3-256, SHA3-512 as Hash	IND-CCA
26.	THREEBEARS			✔			✔		-	none	cSHAKE256 as PRNG, XOF and Hash	IND-CCA
27.	CRYSTALS-Dilithium			✔				✔	1,2,3	none	NIST PRNG, SHAKE128, SHAKE256 as XOF, Hash	SEUF-CMA
28.	Titanium				✔	✔	✔		1,3,5	Binomial Difference Distribution	KMAC256 PRNG, SHA256 as Hash	IND-CCA KEM
29.	LAC				✔	✔	✔		1,3,5	Centered Binomial Distribution	NIST PRNG, SHA256, SHA384, SHA512 as Hash	IND-CCA

algorithms utilise uniform random numbers to return Gaussian distributed numbers and differ from each other in terms of implementation speed, memory, and precision. Constant-time hardware architectures for a wide range of samplers have been proposed [10], [24]. The binomial sampler is inherently protected against timing attacks. However, as it only samples from a binomial distribution instead of an exact Gaussian distribution it can only be used in encryption and key exchange schemes, as the security proof in signature schemes requires the sampler to have high precision.

V. Conclusions and Recommendations

Lattice based cryptographic primitives offer both efficiency and resilience against quantum attacks, highlighting the potential for SCA-resilient LBC implementations to replace current public key cryptography used today on existing commodity and custom hardware. This work surveys the samplers in LBC constructions, with consideration of SCA attacks that threaten the security of these implementations. It is recommended that:

- Ensure *constant time implementations* where possible.
- *Box-Muller sampling* is a suitable DGS candidate for software, offering efficient, constant time and scalable performance, and does not require precomputed tables.
- *CDT sampling* is a suitable DGS candidate for hardware, offering efficient, constant time and scalable performance, using precomputed tables.
- *Knuth-Yao sampling* is compact DGS candidate for hardware, but is inherently non- constant time. Scrambling can address this, incurring performance costs.
- *Binomial sampling* is a suitable sampling candidate for low sigma applications, and it runs in constant time.
- *Multiple threat vectors should be considered together*: to date, most SCA countermeasures for samplers address a specific threat, without consideration of other threats.
- *Performance overheads of countermeasures* should be considered to accurately assess LBC performance.

As more lattice-based designs emerge, further attacks will most likely surface and this will continue to be an important area of research.

References

[1] P. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Proceedings 35th Annual Symposium on Foundations of Computer Science*, p. , 1994.

[2] National Security Agency, "Commercial national security algorithm suite," August 2015. [Online]. Available: https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm

[3] CESG, "Quantum key distribution: A CESG white paper," February 2016. [Online]. Available: https://www.cesg.gov.uk/white-papers/quantum-key-distribution

[4] D. Moody, "Post-quantum cryptography: NIST's plan for the future," Talk given at PQCrypto '16 Conference, 23-26 February 2016, Fukuoka, Japan, February 2016. [Online]. Available: https://pqcrypto2016.jp/data/pqc2016_nist_announcement.pdf

[5] M. Braithwaite. (2016) Experimenting with post-quantum cryptography. [Online]. Available: https://security.googleblog.com/2016/07/experimenting-with-post-quantum.html

[6] A. Langlois and D. Stehlé, "Worst-case to average-case reductions for module lattices," *IACR Cryptology ePrint Archive*, vol. 2012, p. 90, 2012. [Online]. Available: http://eprint.iacr.org/2012/090

[7] D. Knuth and A. Yao, *Algorithms and Complexity: New Directions and Recent Results*. Academic Press, 1976, ch. The Complexity Of Nonuniform Random Number Generation.

[8] S. S. Roy, O. Reparaz, F. Vercauteren, and I. Verbauwhede, "Compact and side channel secure discrete Gaussian sampling," *ePrint Report 2014/591*, 2014, https://eprint.iacr.org/2014/591.

[9] S. S. Roy, F. Vercauteren, and I. Verbauwhede, "High precision discrete Gaussian sampling on FPGAs," in *Selected Areas in Cryptography - SAC 2013 - 20th International Conference, Burnaby, BC, Canada, August 14-16, 2013, Revised Selected Papers*, 2013, pp. 383–401. [Online]. Available: https://doi.org/10.1007/978-3-662-43414-7_19

[10] J. Howe, A. Khalid, C. Rafferty, F. Regazzoni, and M. O'Neill, "On Practical Discrete Gaussian Samplers For Lattice-Based Cryptography," *IEEE Transactions on Computers*, 2016.

[11] G. E. P. Box and M. E. Muller, "A note on the generation of random normal deviates," *Ann. Math. Statist.*, vol. 29, no. 2, pp. 610–611, 06 1958. [Online]. Available: https://doi.org/10.1214/aoms/1177706645

[12] A. Hülsing, T. Lange, and K. Smeets, "Rounded Gaussians - fast and secure constant-time sampling for lattice-based crypto," *IACR Cryptology ePrint Archive*, vol. 2017, p. 1025, 2017. [Online]. Available: http://eprint.iacr.org/2017/1025

[13] C. for Efficient Embedded Security, "Implementation aspects of NTRUEncrypt and pqNTRUSign," 2017. [Online]. Available: https://www.onboardsecurity.com/nist-post-quantum-crypto-submission

[14] S. Bai, T. Lepoint, A. Roux-Langlois, A. Sakzad, D. Stehlé, and R. Steinfeld, "Improved security proofs in lattice-based cryptography: using the rényi divergence rather than the statistical distance," *Journal of Cryptology*, vol. 31, no. 2, pp. 610–640, 2018.

[15] P. S. L. M. Barreto, P. Longa, M. Naehrig, J. E. Ricardini, and G. Zanon, "Sharper ring-lwe signatures," *IACR Cryptology ePrint Archive*, vol. 2016, p. 1026, 2016. [Online]. Available: http://eprint.iacr.org/2016/1026

[16] D. Micciancio and M. Walter, "Gaussian sampling over the integers: Efficient, generic, constant-time," *IACR Cryptology ePrint Archive*, vol. 2017, p. 259, 2017. [Online]. Available: http://eprint.iacr.org/2017/259

[17] A. Khalid, J. Howe, C. Rafferty, F. Regazzoni, and M. O'Neill, "Compact, scalable, and efficient discrete Gaussian samplers for lattice-based cryptography," in *2018 IEEE ISCAS*, 2018, pp. 1–5.

[18] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in *Annual International Cryptology Conference*. Springer, 1996, pp. 104–113.

[19] L. G. Bruinderink, A. Hülsing, T. Lange, and Y. Yarom, "Flush, Gauss, and reload–a cache attack on the BLISS lattice-based signature scheme," in *International Conference on Cryptographic Hardware and Embedded Systems*. Springer, 2016, pp. 323–345.

[20] R. A. Fisher, F. Yates *et al.*, *Statistical tables for biological, agricultural and medical research*. Edinburgh, 1938.

[21] M. O. Saarinen, "Arithmetic coding and blinding countermeasures for Ring-LWE," *IACR Cryptology ePrint Archive*, vol. 2016, p. 276, 2016. [Online]. Available: http://eprint.iacr.org/2016/276

[22] P. Pessl, "Analyzing the shuffling side-channel countermeasure for lattice-based signatures," in *Progress in Cryptology–INDOCRYPT 2016: 17th International Conference on Cryptology in India, Kolkata, India, December 11-14, 2016, Proceedings 17*. Springer, 2016, pp. 153–170.

[23] P. Pessl, L. G. Bruinderink, and Y. Yarom, "To BLISS-B or not to be: Attacking strongSwan's implementation of post-quantum signatures," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1843–1855.

[24] A. Khalid, J. Howe, C. Rafferty, and M. O'Neill, "Time-independent discrete gaussian sampling for post-quantum cryptography," in *2016 International Conference on Field-Programmable Technology, FPT 2016, Xi'an, China, December 7-9, 2016*, 2016, pp. 241–244. [Online]. Available: https://doi.org/10.1109/FPT.2016.7929543

[25] S. Keller and T. A. Hall, "The NIST SP 800-90A deterministic random bit generator validation system (DRBGVS)," 2015. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/cryptographic-algorithm-validation-program/documents/drbg/drbgvs.pdf

[26] S. Brannigan, N. Smyth, T. Oder, F. Valencia, E. O'Sullivan, T. Güneysu, and F. Regazzoni, "An investigation of sources of randomness within discrete Gaussian sampling," *IACR Cryptology ePrint Archive*, vol. 2017, p. 298, 2017. [Online]. Available: http://eprint.iacr.org/2017/298

[27] NIST, "NIST PQC Round 1 Submissions," https://csrc.nist.gov/Projects/Post-Quantum-Cryptography/Round-1-Submissions, 2018, accessed: 2018-09-04.