Using Learning with Rounding to Instantiate Post-Quantum Cryptographic Algorithms

ANDREA BASSO, IBM, Zürich, Switzerland
JOPPE BOS, NXP Semiconductors, Leuven, Belgium
JAN-PIETER D'ANVERS, Belfort, Leuven, Belgium
ANGSHUMAN KARMAKAR, Indian Institute of Technology Kanpur, Kanpur, India
JOSE BERMUDO MERA, PQShield, Oxford, United Kingdom of Great Britain and Northern Ireland
JOOST RENES, NXP Semiconductors, Austin, United States
SUJOY ROY, Graz University of Technology, Graz, Austria
FREDERIK VERCAUTEREN, imec-COSIC, KU Leuven, Leuven, Belgium
PENG WANG, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China
YUEWU WANG, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China
SHICONG ZHANG, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China
CHENXIN ZHONG, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China

The Learning with Rounding (LWR) problem, introduced as a deterministic variant of Learning with Errors (LWE), has become a promising foundation for post-quantum cryptography. This Systematization of Knowledge (SoK) article presents a comprehensive survey of the theoretical foundations, algorithmic developments, and practical implementations of LWR-based cryptographic schemes. We introduce LWR within the broader landscape of lattice-based cryptography and post-quantum security, highlighting its advantages such as reduced randomness, improved efficiency, and enhanced side-channel resistance. We explore the evolution of security reductions from LWR to LWE, including recent advances that support practical parameter regimes and address challenges in both bounded and unbounded sample settings. This article systematically reviews existing LWR-based schemes — including Saber, Lizard, Florete, Espada, Sable, and SMAUG — analyzing their design choices, parameter sets, and performance tradeoffs. Furthermore, we examine the impact of LWR on side-channel resistance, failure probabilities, and masking efficiency, demonstrating its suitability for secure

Author list in alphabetical order; see the AMS culture statement.
Authors' Contact Information: Andrea Basso, IBM, Zürich, ZH, Switzerland; e-mail: Andrea.Basso@ibm.com; Joppe Bos (corresponding author), NXP Semiconductors, Leuven, Belgium; e-mail: joppe.bos@nxp.com; Jan-Pieter D'Anvers, Belfort, Leuven, Belgium; e-mail: janpieter.danvers@esat.kuleuven.be; Angshuman Karmakar, Indian Institute of Technology Kanpur, Kanpur, India; e-mail: angshu99@gmail.com; Jose Bermudo Mera, PQShield, Oxford, United Kingdom of Great Britain and Northern Ireland; e-mail: josebmera@gmail.com; Joost Renes, NXP Semiconductors, Austin, Texas, United States; e-mail: joost.renes@nxp.com; Sujoy Roy, Graz University of Technology, Graz, Austria; e-mail: sujoyetc@gmail.com; Frederik Vercauteren, imec-COSIC, KU Leuven, Leuven, Belgium; e-mail: frederik.vercauteren@gmail.com; Peng Wang, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China; e-mail: p-wang@ucas.ac.cn; Yuewu Wang, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China; e-mail: wangyuewu@ucas.ac.cn; Shicong Zhang, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China; e-mail: zhangshicong20@mails.ucas.ac.cn; Chenxin Zhong, School of Cryptology, University of the Chinese Academy of Sciences, Beijing, China; e-mail: chenxin.zhong@outlook.com.



This work is licensed under a Creative Commons Attribution 4.0 International License.

© 2026 Copyright held by the owner/author(s).
ACM 1539-9087/2026/07-ART71
https://doi.org/10.1145/3807514

and efficient implementations. By consolidating the research spanning theory and practice, this SoK aims at guiding future cryptographic design and standardization efforts leveraging LWR.

CCS Concepts: • **Security and privacy → Public key (asymmetric) techniques;**

Additional Key Words and Phrases: Post-quantum cryptography, learning with rounding, lattice-based cryptography

ACM Reference Format:
Andrea Basso, Joppe Bos, Jan-Pieter D'Anvers, Angshuman Karmakar, Jose Bermudo Mera, Joost Renes, Sujoy Roy, Frederik Vercauteren, Peng Wang, Yuewu Wang, Shicong Zhang, and Chenxin Zhong. 2026. Using Learning with Rounding to Instantiate Post-Quantum Cryptographic Algorithms. *ACM Trans. Embedd. Comput. Syst.* 25, 4, Article 71 (July 2026), 33 pages. https://doi.org/10.1145/3807514

1 Introduction

It is well-known that the recent advancements in quantum computing pose a significant threat to the classical public-key cryptographic systems, such as RSA [105] and ECC [62, 79], which rely on the hardness of problems like integer factorization and discrete logarithms. As quantum algorithms [97, 111] can efficiently solve these problems, the cryptographic community has been working to develop quantum-resistant alternatives. This realm of **post-quantum cryptography** (**PQC**) aims at designing cryptographic schemes secure against both classical and quantum adversaries.

In response to this challenge, the **National Institute of Standards and Technology** (**NIST**) initiated a multi-year standardization process to evaluate and select quantum-safe cryptographic algorithms. This effort culminated in the selection of several algorithms for standardization, including the recommended schemes ML-KEM [80] (based on the CRYSTALS-Kyber [19] key encapsulation mechanism) and ML-DSA [81] (based on the CRYSTALS-Dilithium [38] digital signature scheme). Both of these schemes are grounded in the **Learning with Errors** (**LWEs**) problem [102], a lattice-based hardness assumption that has become a cornerstone of modern PQC due to its strong security guarantees and versatility.

This SoK focuses on a derandomization technique for LWE introduced in 2012 by Banerjee, Peikert and Rosen [9] denoted ***Learning with Rounding*** (**LWR**). LWR offers a deterministic, noise-free alternative to LWE, replacing the addition of a random error with a rounding operation. This subtle shift yields several practical advantages, including simpler implementations, reduced randomness requirements, and improved efficiency and side-channel resistance. Initially introduced for the construction of pseudorandom function families, it has been used (among others) to construct deterministic encryption [120], an IND-CCA2 secure KEM [32] (but see also earlier work such as the NTRU-base KEM [12] which used rounding) and a digital signature scheme [92]. Over the past decade, LWR has evolved from a theoretical notion into a practical foundation for cryptographic design. Its deterministic nature makes it particularly attractive for constrained environments and high-assurance implementations, where minimizing randomness and maximizing resistance to side-channel attacks are critical. Moreover, the ability to instantiate LWR with power-of-two moduli enables efficient implementations without the need for expensive modular reduction, further enhancing its appeal for real-world deployment.

This practical relevance is highlighted by its adoption in major standardization efforts. Notably, the LWR-based Saber key encapsulation mechanism [32] is one of the finalists in the NIST Post-Quantum Cryptography standardization process, reflecting its strong balance of security, efficiency, and implementation simplicity. More recently, SMAUG-T [47], a hybrid scheme that uses LWR for encapsulation and LWE for key generation, has been selected as the national post-quantum cryptographic standard in South Korea. These developments demonstrate that LWR is not only

of theoretical interest but also a viable and competitive foundation for real-world cryptographic deployments.

Systematization Contributions. This SoK article makes several key contributions to the study and application of LWR in post-quantum cryptography. First, it provides a comprehensive survey of the theoretical foundations of LWR, including its security reductions to LWE, parameter selection strategies, and recent advances in both bounded and unbounded sample settings. Second, it systematically reviews a wide range of LWR-based cryptographic schemes — including (u)Saber, Lizard, Florete, Espada, Sable, and SMAUG(-T) — highlighting their design choices, performance characteristics, and tradeoffs. Third, the article offers a detailed analysis of implementation aspects, including side-channel resistance, masking techniques, and decryption failure probabilities, with Saber serving as a representative case study. Fourth, it evaluates the practical efficiency of countermeasures against side-channel attacks and discusses the implications of LWR's deterministic structure for high-assurance implementations. A primary goal of this SoK is to consolidate the fragmented research across theory and practice, providing a valuable resource for cryptographic designers and standardization bodies aiming at building secure, efficient, and quantum-resistant systems based on LWR.

Selection criteria. We include LWR-based schemes that are (1) standardization-relevant, i.e., submitted to or discussed in major PQC standardization efforts, or selected in regional processes; (2) representative of distinct design points in the LWR design space; (3) supported by mature public cryptanalysis, meaning there exists a substantive peer-reviewed or community-evaluated literature beyond an initial proposal; and (4) accompanied by high-quality, publicly available implementations, prioritizing reference and optimized implementations and, where available, implementations that explicitly address side-channel concerns. This emphasis follows standardization guidance that values schemes whose resistance to side-channel attacks can be achieved with minimal overhead and that treats optimized side-channel-aware implementations as more meaningful evidence of deployability [84, 4.A.6], [86, 4.B.4], [54, 1.(3)]. We also incorporate standardization-style practicality considerations: such as resource consumption (software memory and hardware area/throughput/energy) and platform flexibility (including constrained environments) [84, 4.C.1], [54, 1.(6), and 2.(3)].

To keep the SoK focused and interpretable, we consider the following out of scope. (1) Schemes where LWR is not central to the security claim, e.g., constructions that primarily rely on LWE/RLWE and only use rounding as a compression/encoding detail rather than as the defining hardness assumption. (2) Protocol- and system-integration topics such as full TLS/VPN/PKI integration.

2 Preliminaries

2.1 Notation

The ring of integers modulo and integer q is denoted as $\mathbf{Z}_q = \mathbf{Z}/q\mathbf{Z}$. Similarly, the polynomial quotient ring is denoted as $\mathbf{R}_q = \mathbf{Z}_q[x]/(f(x))$ where $f(x)$ in this article is chosen to be the cyclotomic polynomial $f(x) = x^n + 1$ for a power-of-two value of n. $\zeta_{\hat{n}}$ denotes a primitive $\hat{n}$th root of unity. In algorithm blocks, bold lowercase letters, such as $\mathbf{a}$, represent vectors of polynomials in the ring, while bold uppercase letters, such as $\mathbf{A}$, represent matrices. The uniform distribution over a set S is denoted by $\mathcal{U}(S)$. The inner product between two vectors a and s is written as $\langle \mathbf{a}, \mathbf{s} \rangle$. The rounding operation is denoted as $\lfloor a \rceil_p \in \mathbf{Z}_p$.

2.2 Learning with Errors

Understanding the theoretical and practical aspects of LWR requires an introduction to the LWEproblem [102] and their more efficient popular variants: ring learning with errors [73] and

module learning with errors [20, 68]. We refer the interested reader to the comprehensive surveys related to LWE [94, 103].

Let $\mathbf{K}$ be a number field of degree n and let $\mathbf{R} = \mathcal{O}_K$ be its ring of integers and denote $\mathbf{R}_q = \mathbf{R}/q\mathbf{R}$. Let the module rank d be a positive integer and let χ denote a distribution over $\mathbf{R}$. Let $s \in \mathbf{R}_q^d$ be a secret vector and define the distribution $A_{s,\chi}$ as

$$A_{s,\chi} = (a, b = \langle a, s \rangle + e \bmod q) \in \mathbf{R}_q^d \times \mathbf{R}_q,$$

where $a \leftarrow \mathcal{U}(\mathbf{R}_q^d)$ is sampled uniformly at random and $e \leftarrow \chi$ is the error term.

The *search LWE problem* is given access to polynomially many independent samples from $A_{s,\chi}$, recover the secret s, while the *decision LWE problem* is to distinguish between samples drawn from $A_{s,\chi}$ and samples drawn uniformly from $\mathbf{R}_q^d \times \mathbf{R}_q$. In practice, one often takes $\mathbf{R}_q = \mathbf{Z}_q[x]/(f(x))$ with $f(x)$ the cyclotomic polynomial

$$f(x) = \Phi_{\hat{n}}(x) = x^n + 1,$$

with $\hat{n} = 2n$ and n a power of two. In this Ring-LWE variant all operations are performed within the ring $\mathbf{R}_q$. A single Ring-LWE sample embeds n Plain-LWE samples worth of information while drastically reducing the size of cryptographic keys. Furthermore, the choice of the cyclotomic polynomial allows for the polynomial multiplication to be computed very efficiently using the **Number Theoretic Transform** (**NTT**) [96], a variant of the Fourier Transform that involves only integer arithmetics.

The Module-LWE problem [20, 68] generalizes both Plain-LWE and Ring-LWE, offering a flexible intermediate structure which enables a tradeoff between security and efficiency. In Module-LWE, the secrets are vectors of polynomials and a Module-LWE sample consists of $(\mathbf{a}, b)$, where $b = \langle \mathbf{a}, \mathbf{s} \rangle + e$. Hence, Module-LWE is a generalization of the Plain- and Ring-LWE versions. Setting the rank $d = 1$ particularizes to the Ring-LWE problem, whereas setting the ring to be $\mathbf{Z}$ and the rank $d = n$ particularized to Plain-LWE problem. Additionally, for a given parameter choice of a Module-LWE, the difficulty of the problem, i.e., the security of the scheme, can be adjusted by simply tuning the module rank.

The practical values of the key ingredients from the LWE variants are listed in the table below.

	Plain-LWE	**Ring-LWE** (2^k-cyclotomics)	**Module-LWE**
Number field K	$\mathbf{Q}$	$\mathbf{Q}(\zeta_{\hat{n}})$	General number field K
Ring $\mathbf{R}$	$\mathbf{Z}$	$\mathbf{Z}[\zeta_{\hat{n}}] = \mathbf{Z}[x]/(\Phi_{\hat{n}}(x))$	$\mathcal{O}_K$ (ring of integers)
Module rank d	n	1	$d > 1$
Secret $s \in$	$\mathbf{Z}_q^n$	$\mathbf{R}_q$	$\mathbf{R}_q^d$
Public $a \in$	$\mathbf{Z}_q^n$	$\mathbf{R}_q$	$\mathbf{R}_q^d$

It should be noted that more algebraically structured LWE variants have been introduced and studied. This includes polynomial-LWE [107, 113], order-LWE [17], and middle-product LWE [106]. A general framework of all these LWE variants over commutative base rings is given in [95].

In practice, the most well-known (module-)LWE based post-quantum cryptographic algorithms are CRYSTALS-Kyber [19] (which became the ML-KEM NIST standard [80]) and CRYSTALS-Dilithium [38] (which became the ML-DSA NIST standard [81]): the recommended post-quantum secure key-encapsulation mechanism and digital signature algorithms, respectively.

3 LWR Security and Parameter Selection

Introduced in [9], learning with rounding can be viewed as a derandomization technique for LWE. Recall from Section 2.2 that in LWE one computes $\langle a, s\rangle + e \bmod q$: the error term e is *added* to the inner product (modulo q). Instead of this addition, the idea of LWR is to *scale and round* this inner product to some smaller group $\mathbf{Z}_p$. Specifically, this rounding divides the elements in $\mathbf{Z}_q$ up into p contiguous intervals and maps an element of $\mathbf{Z}_q$ to its corresponding index in $\mathbf{Z}_p$. The value of p needs to be smaller than q otherwise no error is introduced. This rounded inner product $\lfloor \frac{p}{q} \langle a, s\rangle \rceil$ is denoted as

$$\lfloor \langle a, s\rangle \rceil_p = \left\lfloor \frac{p}{q} \langle a, s\rangle \right\rceil \in \mathbf{Z}_p.$$

Here $\mathbf{Z}_q$ (resp. $\mathbf{Z}_p$) is represented by $[0, q[$ (resp. $[0, p[$). Note that when $p \mid q$, the rounded version of the uniform distribution on $\mathbf{Z}_q$ is again uniform on $\mathbf{Z}_p$. The rounding operation can be extended to vectors / polynomials by applying it coefficient wise.

Analogously to the ring- and module-variants of learning with errors, one can define ring- and module-versions of LWR. A table showing the differences (and similarities) between plain-, ring- and module LWR can be found below and follows the one for LWE.

	Plain-LWR	**Ring-LWR** (cyclotomics)	**Module-LWR**
Number field K	$\mathbf{Q}$	$\mathbf{Q}(\zeta_n)$	General number field K
Ring $\mathbf{R}$	$\mathbf{Z}$	$\mathbf{Z}[\zeta_n] = \mathbf{Z}[X]/\Phi_n(X)$	$\mathcal{O}_K$ (ring of integers)
Module rank d	n	1	$d > 1$
Secret $s \in$	$\mathbf{Z}_q^n$	$\mathbf{R}_q$	$\mathbf{R}_q^d$
LWR-sample $(a, b) \in$	$\mathbf{Z}_q^n \times \mathbf{Z}_p$	$\mathbf{R}_q \times \mathbf{R}_p$	$\mathbf{R}_q^d \times \mathbf{R}_p$

The decision version of the Learning with Rounding problem is as follows:

Definition 3.1 (Decision Learning with Rounding, based on [9, Definition 3.1]). Let $n \geq 1$ and moduli $q > p \geq 2$ be integers. Let $\mathbf{R}_q = \mathbf{Z}_q[x]/f(x)$ be as defined above.

- **(Plain-LWR)**: For a vector $s \in \mathbf{Z}_q^n$, define the LWR distribution L_s to be the distribution over $\mathbf{Z}_q^n \times \mathbf{Z}_p$ obtained by choosing a vector $a \leftarrow \mathbf{Z}_q^n$ uniformly at random, and outputting $(a, b = \lfloor \langle a, s\rangle \rceil_p)$.
- **(Ring-LWR)**: For a polynomial $s \in \mathbf{R}_q$, define the ring-LWR distribution L_s to be the distribution over $\mathbf{R}_q \times \mathbf{R}_p$ obtained by choosing $a \leftarrow \mathbf{R}_q$ uniformly at random and outputting $(a, b = \lfloor a \cdot s \rceil_p)$.
- **(Module-LWR)**: Fix a module rank $d > 1$. For $s \in \mathbf{R}_q^d$, define the module-LWR distribution L_s to be the distribution over $\mathbf{R}_q^d \times \mathbf{R}_p$ obtained by choosing $a \leftarrow \mathbf{R}_q^d$ uniformly at random and outputting $(a, \lfloor \langle a, s\rangle \rceil_p)$.

The **decision (ring/module-) LWR**$_{n,q,p}$ problem is to distinguish (with non-negligible advantage in n) between polynomially many independent samples $(a_i, b_i) \leftarrow L_s$ and the same number of samples drawn uniformly and independently from $\mathbf{Z}_q^n \times \mathbf{Z}_p$, $\mathbf{R}_q \times \mathbf{R}_p$, or $\mathbf{R}_q^d \times \mathbf{R}_p$, respectively.

The high-level idea for the security of LWR is to show that the security is statistically close to that of solving decision-LWE (distinguishing from random $\mathbf{u}$) [5]:

$$(a, \lfloor \langle a, s\rangle \rceil_p) \overset{\text{stat}}{\approx} (a, \langle a, s\rangle + e) \overset{\text{comp}}{\approx} (a, \mathbf{u}).$$

The security of LWR was initially studied in [9]. This article shows the following result.

Theorem 3.2 (Reduction to LWE [9, Theorem 3.2]). *Let χ_n be any efficiently sampleable family of probability distributions, indexed by the security parameter n, over $\mathbf{Z}$ which is $\beta(n)$-bounded (that is, $Pr_{x\leftarrow\chi}[|x| > \beta] \leq negl(n)$). Let $q \geq p \cdot \beta \cdot n^{\omega(1)}$. Then for any distribution over the secrets $s \in \mathbf{Z}_q^n$ solving decision-$LWR_{n,q,p}$ is at least as hard as solving decision-$LWE_{n,q,\chi}$ for the same distribution over s.*

This security result requires that the modulus $q \geq p \cdot \beta \cdot n^{\omega(1)}$ is super-polynomial. This result was improved by [5] by providing a security reduction that works for a larger range of parameters: the size of the modulus q used is not super-polynomial but polynomial [5, Thm. 4.1+Cor. 4.2]. Specifically, the authors show the hardness of LWR (using parameters n, m (the number of samples), q, and p) assuming the hardness of LWE parameters n', m, q, β as long as

$$n \geq \frac{\log(q)}{\log(2\gamma)} \cdot n' \quad \text{and} \quad q \geq \gamma(nm\beta p),$$

for some parameter $\gamma \geq 1$. For the reduction to work the modulus q needs to be much larger than the number of LWR samples m. Hence, when an attacker is allowed to choose the number of LWR samples (and is not fixed *a priori*) the modulus q becomes super-polynomial again. The modulus q needs to be prime or the above inequality should hold for the largest prime divisor $q_{\max}$ of q. Moreover, the result requires q and $q/q_{\max}$ to be co-prime: this implies that the result does not hold when the modulus q is a power of two.

This restriction is lifted by the results from [16]. Firstly, the bound on the modulus q is improved to

$$q \geq 2m\beta p,$$

and secondly, this result holds without any number-theoretic restrictions on q: specifically, the results holds when the modulus q is a power of two [16, Thm. 1 + Thm. 3].

Theorem 3.3 (One-wayness of LWR [16, Theorem 1]). *Let p, q, n, m, and β be integers such that $q > 2p\beta$. For every algorithm* `Learn`*,*

$$\Pr_{A,s,e}\left[\mathtt{Learn}(A, \lfloor As + e \rceil_p) = s\right] \geq \frac{\Pr_{A,s}[\mathtt{Learn}(A, \lfloor As \rceil_p) = s]^2}{(1 + 2p\beta/q)^m},$$

where $A \leftarrow \mathbf{Z}_q^{m\times n}$, the noise e is independent over all m coordinates, β-bounded and balanced in each coordinate, and s is chosen from any distribution supported on $\mathbf{Z}_q^{n*}$.

This result proves the hardness of LWR when the number of samples m are bounded. More recently, the authors of [87] studied the unbounded setting and in this regime identify a barrier using "pointwise" reductions from LWE to LWR. The main result of this work and the associated conjecture require the modulus q to be prime [87, Thm. 1 + Conj. 1]: something which in practical instantiations is not the case (cf. Section 4). It should be noted that the results of [87] indicate new ideas are necessary to prove the security in the unbounded sample setting and *does not* suggest that LWR is easy.

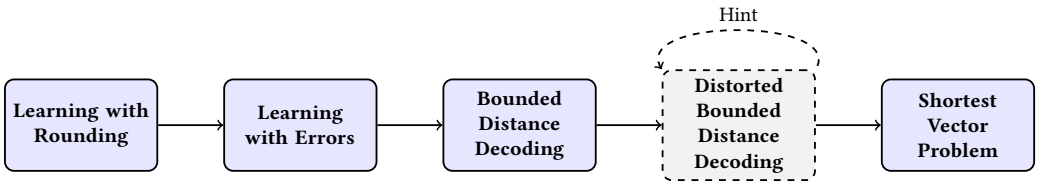
3.1 Parameter Selection in Practice

In practice, parameters for all lattice-based cryptographic schemes are chosen in such a way that the scheme is functionally correct with overwhelming probability (see Section 7 on failure probabilities), efficiency is optimized (in terms of running time as well as memory usage and key sizes) and the security goals are met (with some margin for future improvements in compute power and algorithm advancements). In general in cryptography, this is not a trivial task [69]. As shown in the previous section, the security of LWR can be reduced to LWE, which in turn can be related to quantumly

solving the unique shortest vector problem, which is believed to be a hard problem [77, 116]. However, to select *practical* parameters one often estimates how long a *practical* attack would take and relates this to a concrete security level.

The security of lattice problems has a vast body of literature starting with [3] which linked the average case complexity of lattice problems to the worst case: i.e., a random instance of the shortest vector problem in a lattice is hard. Many algorithms for solving LWR and LWE rely on lattice reduction as the central step. Depending on the block size used, and the quality of output required, one typically estimates the running time of the lattice reduction algorithm BKZ [110] and improvements including [23, 44].

The most popular approach for such an assessment is using the "`lattice-estimator`" which provides functions for estimating the concrete security of LWE and LWR instances [4]. For LWR such a high-level chain of reductions look like



where the dashed box is optional and integrates hints into lattice reduction attacks as explained in [29]. The best known attacks are then used to solve this shortest vector problem and practical parameters are based on the running times for such algorithms.

Conclusions and Future Directions

The theoretical foundations of Learning with Rounding have matured significantly since its introduction as a derandomized variant of Learning with Errors. In this section, we have highlighted the core security reductions from LWR to LWE, the evolution of these reductions to support more practical parameter regimes, and the challenges that arise in both bounded and unbounded sample settings. The evolution of the modulus q from being super-polynomial (in terms of the security parameter) in size and prime to polynomial and allowing the usage of a powers-of-two modulus. The unbounded sample setting remains an open challenge.

Future work should aim at refining the existing reductions to yield tighter bounds and better concrete security estimates, especially for parameter sets used in practice. Moreover, the security in the unbounded sample regime should be investigated further. The limitations of pointwise reductions in this unbounded setting call for novel proof techniques or alternative hardness assumptions that can bridge this gap.

4 LWR-based Key Establishment and Signatures

Mathematically hard problems such as LWR are a great source for post-quantum secure cryptographic constructions, such as key establishment and digital signatures. To define a key establishment protocol based on LWR (or any suitable hard problem), the high-level approach is to first construct a **Public-Key Encryption** (**PKE**) scheme, and then transform it into a **Key Encapsulation Mechanism** (**KEM**) scheme by using a variant of the **Fujisaki-Okamoto** (**FO**) transform [41, 42]. The FO transform turns a passively (IND-CPA) secure PKE scheme into an efficient chosen-ciphertext attack (IND-CCA) secure KEM. This design approach has been followed for the two NIST PQC KEM-finalists Saber [32] and Kyber [19] based on LWR and LWE, respectively. As the choice of FO transform is largely orthogonal to the selection of the PKE, we discuss these

ALGORITHM 1: Generic Public-Key Encryption (PKE) scheme based on module-LWR.

`MLWR-PKE.Keygen`

Input: System parameters $q, p, \ell, D_{\mathbf{s}}$
Output: (pk, sk)
1: $\mathbf{A} \leftarrow \mathbf{R}_q^{\ell \times \ell}$
2: $\mathbf{s} \leftarrow D_{\mathbf{s}}$ ▷ $\mathbf{s} \in \mathbf{R}_q^{\ell}$
3: $\mathbf{b} = \lfloor \frac{p}{q}\mathbf{A}^T\mathbf{s} \rceil \in \mathbf{R}_p^{\ell}$
4: **return** $(pk = (\mathbf{b}, \mathbf{A}), sk = \mathbf{s})$

`MLWR-PKE.Encrypt`

Input: $pk = (\mathbf{b}, \mathbf{A}), m \in \mathbf{R}_{2^B}$
Output: c
1: $\mathbf{s}' \leftarrow D_{\mathbf{s}}$ ▷ $\mathbf{s}' \in \mathbf{R}_q^{\ell}$
2: $\mathbf{b}' = \lfloor \frac{p}{q}\mathbf{A}\mathbf{s}' \rceil \in \mathbf{R}_p^{\ell}$
3: $v' = \langle \mathbf{b}, \mathbf{s}' \rangle + \frac{q}{2 \cdot p} \in \mathbf{R}_p$
4: $c_m = \lfloor \frac{t}{p} \cdot (v' + p/(2^B) \cdot m) \bmod p \rceil \in \mathbf{R}_t$ ▷ Encode B bits of m per coefficient
5: **return** $c = (c_m, \mathbf{b}')$

`MLWR-PKE.Decrypt`

Input: $(\mathbf{s}, c = (c_m, \mathbf{b}'))$
Output: m'
1: $v = \mathbf{b}'^T\mathbf{s} \in \mathbf{R}_p$
2: $m' = \lfloor \frac{2^B}{p}((v - \frac{p}{t} \cdot c_m + (\frac{p}{2 \cdot t} - \frac{q}{2 \cdot p})) \bmod p) \rceil \in \mathbf{R}_{2^B}$
3: **return** m'

steps separately in Section 4.2. While the number of digital signature designs based on LWR is much smaller than for KEMs, we include them for completeness.

4.1 Public-Key Encryption (PKE)

There are several choices involved in the design of a public-key encryption scheme based on a KEM. Generally speaking, modern public-key encryption schemes based on LWR (and LWE) follow the approach from Regev [102]. We refer to Algorithm 1 for a general description. In the remainder of this section, we will introduce the various aspects of this public key encryption scheme and how they can be instantiated. We then return to the different proposals that have been defined in recent years, see Table 2 for an overview.

4.1.1 Security Reduction. A security reduction for Algorithm 1 can be constructed similar to the security reduction of Saber [35]. [1] This proof reduces the security of the public-key encryption scheme to the hardness of LWR, using similar secret distributions, ring structures, and moduli q and p. Algorithm 1 introduces a minor variation that allows multiple bits to be encrypted per coefficient; this modification does not impact the underlying proof strategy since it only impacts the decryption failure probability (increasing with the number of encrypted bits B). A machine-verifiable version of the Saber proof has later been provided in [53], which also contains a very detailed hand-written security proof.

[1] The original proof contained a minor error, which has been corrected in the updated ePrint version.

Informally, the security reduction can be stated as follows:

Theorem 4.1. *In the random oracle model where* $\mathbf{A}$ *is assumed to be generated through a random oracle, for any adversary* $\mathcal{A}$*, there exist two adversaries* $\mathcal{B}_0$ *and* $\mathcal{B}_1$ *with roughly the same running time such that* $Adv^{ind\text{-}cpa}_{Saber.PKE}(\mathcal{A}) \leqslant Adv^{mod\text{-}lwr}_{keygen}(\mathcal{B}_0) + Adv^{mod\text{-}lwr}_{encrypt}(\mathcal{B}_1)$*, if* $q/p \leqslant p/t$.

where $\text{Adv}^{\text{mod-lwr}}_{\text{keygen}}$ and $\text{Adv}^{\text{mod-lwr}}_{\text{encrypt}}$ are the LWR samples constructed in the key generation and encryption procedure respectively. For a more formal analysis, we refer to the Saber article.

The security proof does not impose restrictions on the choice of ring or module rank, nor does it require particularly strict conditions on the secret distribution. However, it does rely on the condition that p divides q; otherwise, the public component $\mathbf{b}$ would not be uniformly distributed, and consequently can not be used as public input to generate an LWR sample as this requires the public value to be uniformly distributed. The proof also requires $q/p \leqslant p/t$, but this condition is typically met in practice, as optimal parameter choices generally involve a small t to minimize bandwidth. For this same reason ML-KEM does not apply rounding to the public key, whereas the initial Kyber proposal did.

Another subtlety in the proof lies in the computation of c_m. Instead of applying a standard rounding operation, the constant $q/(2p)$ is added during encryption. This adjustment ensures that the rounding operation used to compute $\mathbf{b'}$ aligns with the one used to compute c_m (in the derivation of v'), and is required for the proof. The resulting deviation from standard rounding is later corrected during decryption (line 2), which compensates for the altered mean and thus reduces the noise. This has no significant effect on the efficiency of the scheme, but it does make it slightly more complicated to implement.

The above restrictions (namely, that p divides q and the use of an adapted rounding constant) are artifacts of the game-based security proof, and are required to make the statistical distance between the different games in the proof negligible (we refer to [53] for the definition of the game sequence). As an example: in the first game-hop $\mathbf{b}$ is replaced by a uniform random sample in $\mathbf{R}^{\ell}_p$. If $p \nmid q$, then clearly the result of the scale-and-round operation is far from uniform, making this game-hop invalid. Note that this requirement precludes the use of a prime modulus q. Saber accommodates these requirements by working with power-of-two moduli and adding the necessary rounding constants.

We note that there are currently no known attacks that exploit variants of the protocol that do not enforce these constraints. It remains an open research question whether and by how much the use of a prime q in combination with public key rounding, or the use of standard rounding in the computation of c_m, affects the security of the encryption scheme. However, until these research questions have a satisfactory answer, it seems prudent to closely adhere to these constraints, as was done in the design of Saber.

4.1.2 (Module) Learning With Rounding: Ring and Modulus Considerations. The first and most critical design aspect of a Ring/Module-LWR based encryption scheme is its underlying ring. For the ring $\mathbf{R}_q = \mathbf{Z}_q[x]/(f(x))$ this comes down to the selection of q and f. As explained in Section 4.1.1 the security reduction requires q to be divisible by p, which is typically achieved by selecting q as a power of two (which is also supported by the theoretical security reduction as explained in Section 3). Hence, the selection can be made by choosing the smallest powers of two q and $p < q$ (the rounding modulus) such that the intended security level is achieved (see Section 3.1). A notable difference compared to many LWE-based designs, where q can be selected to be a prime, is that the NTT does not apply natively due to the lack of appropriate roots of unity in $\mathbf{Z}_q$. This affects the efficiency of polynomial arithmetic in $\mathbf{R}_q$, leading to generally slower implementations of LWR than LWE. However, it was first shown in [27] that the asymptotically faster NTTs can still be

used by embedding $\mathbf{R}_q$ into a larger NTT-friendly ring with prime modulus q'. While this would increase the bit length of the modulus by at least a factor 2, it can still lead to improved arithmetic compared to Toom-Cook or Karatsuba on certain platforms. For example, [27] uses 6 layers of size-256 negacyclic NTTs followed by 4x4 schoolbook multiplications using

$$q' = 25166081 = 196610 \cdot 128 + 1,$$

for parameter sets Saber and Firesaber resulting in approximately 60 percent faster multiplication than in $\mathbf{R}_{8192}$. Ultimately, while there is certainly a measurable difference in the efficiency of LWR and LWE, they are close enough in performance that it is not significant for many practical applications. The selection of f and optionally the module rank ℓ must be made such that the LWR problem is sufficiently hard. Typically f is chosen as $x^n + 1$ for a certain choice of n (e.g., $n = 256, 512, 1024$) in combination with ℓ to achieve varying security levels.

Having established the ring, the distribution $D_{\mathbf{A}}$ is fixed to the uniform distribution $\mathcal{U}_q$ on $\mathbf{R}_q^{\ell \times \ell}$. The final choice that remains is that of the secret distribution $D_{\mathbf{s}}$. As the security of LWR depends on the variance of $D_{\mathbf{s}}$, and less so on its precise distribution, this leaves more room for choice. The Saber proposal follows a centered binomial distribution β_μ with coefficients in $[-\mu, \mu]$, similar to the distribution used in Kyber. Other proposals include for example the uniform distribution (uSaber) or a sparse binary distribution (RLizard); we refer to Section 4.2.1 for more details. While sparse distributions can have major advantages in terms of secret key size and efficiency of computations, their security is not as well understood.

4.1.3 Pseudo-Random Generation. While ring and module LWR reduce secret, public key and ciphertext sizes, they can still be prohibitively large for distributing or storing. As such, pseudo-random generation is used to generate their values on-the-fly. For the public key matrix $\mathbf{A}$, this typically means that it is replaced by a seed $\mathsf{seed}_{\mathbf{A}}$ from which $\mathbf{A}$ is generated. The bit length of the seed should be sufficiently large, e.g., 32 or 64 bytes. The expansion from $\mathsf{seed}_{\mathbf{A}}$ to $\mathbf{A}$ consists of the bit expansion of $\mathsf{seed}_{\mathbf{A}}$, combined with a sampling routine that uses the generated bits to sample from $D_{\mathbf{A}}$. For LWR-based proposals this second step is trivial: as $D_{\mathbf{A}} = U_q$ for a power of two q, the sampling can be completed by simply selecting $\epsilon_q = \log(q)$ bits per coefficient. This is in contrast to primes q where rejection sampling (or equivalent) is required, leading in particular to non-constant execution times.

Similarly, secret key sizes can be reduced by storing a seed $\mathsf{seed}_{\mathbf{s}}$ instead of the expanded value $\mathbf{s}$. This tradesoff storage size for efficiency during decryption, and it will be dependent on the application which is favored. Since the secret key format does not require compatibility between different parties, this selection can be made on a case by case basis. Generally the secret key generation begins by sampling a seed $\mathsf{seed}_{\mathbf{s}}$ internal to the key generation algorithm from a true randomness source, after which it is expanded to the full secret $\mathbf{s}$ sampled in $D_{\mathbf{s}}$. The sampling routine strongly depends on the choice of $D_{\mathbf{s}}$, which affects the efficiency of the implementation. In particular, as the secret key must potentially be protected against timing and other side-channel attacks, complexity in the sampling routines can impact the effectiveness of masking techniques.

The choice of symmetric primitive is orthogonal to the selection of the sampling distributions themselves. Often SHA-3 or SHAKE are deployed due to the convenient nature of SHAKE as an **eXtendable Output Function** (**XOF**) that allows to generate an arbitrary number of bits, and due to their approval in the NIST PQC competition. However, any sufficiently robust symmetric primitive could be used for this purpose.

4.1.4 Decryption Failures. After an execution of an LWR encryption protocol as given in Algorithm 1, there is a small probability that the decrypted message m' is different from the input message m. A *decryption failure* is defined as an occurrence where $m \neq m'$. To calculate the

probability of such an event happening we define the noise introduced by the various rounding operations:

$$u = \frac{q}{p} \cdot \mathbf{b} - \mathbf{A}^T\mathbf{s}; \quad u' = \frac{q}{p} \cdot \mathbf{b'} - \mathbf{A} \cdot \mathbf{s'}; \quad u'' = \frac{q}{t} \cdot \left\lfloor \frac{t}{q} \cdot \mathbf{bs'} \right\rceil - \mathbf{bs'}. \tag{1}$$

We can then compute the condition for correct decryption as

$$-\frac{q}{2 \cdot 2^B} \leq (\mathbf{u'}^T\mathbf{s} - \mathbf{u}^T\mathbf{s'} - \mathbf{u''})_i < \frac{q}{2 \cdot 2^B}, \tag{2}$$

for each coefficient i of the polynomial. As proposed by Bos et al. [18], one can compute an upper bound on this decryption failure probability by computing the full distribution of the coefficients of $(\mathbf{u'}^T\mathbf{s} - \mathbf{u}^T\mathbf{s'} - \mathbf{u''})$, and assuming independence between the coefficients, which is a reasonable assumption as long as there is no error correction as shown in [37]. Note that decryption failures depend on the secret key, and as such will leak information about the secret key. This means that for encryption schemes where a secret key is reused, the decryption failure rate typically needs to be negligible. We summarize the decryption failure probabilities for existing schemes in Table 2 and will cover attacks that make use of decryption failures in Section 7.

4.2 Key Encapsulation Mechanisms (KEMs)

The FO transform [41, 42, 48] turns a passively-secure PKE scheme into an efficient **chosen-ciphertext attack (CCA)**-secure KEM. In [48] several variants are analyzed that are similar yet different. They define the transforms $\mathsf{FO}^{\not\perp}$, $\mathsf{FO}^{\not\perp}_m$ (the ciphertext is not included in the shared secret creation), $\mathsf{FO}^{\not\perp}_{pk}$ (the public key is included in the derivation of random coins for multi-user security) and $\mathsf{FO}^{\not\perp}_{pk,m}$, all of which *implicitly* reject a ciphertext when re-encryption fails by returning a key that is indistinguishable from random. For each variant there is an option to *explicitly* reject by aborting for all rejected ciphertexts. While early proposals in the NIST competition selected $\mathsf{FO}^{\not\perp}$ or $\mathsf{FO}^{\not\perp}_{pk}$ or their explicit variants, subsequent work from [14] showed that the inclusion of the ciphertext (or its digest) into the key derivation is not necessary.

In the more recent work [39] the authors look into yet another variant $\mathsf{FO}^{\not\perp}_{\mathrm{ID(pk)},m}$ that includes the use of prefix hashing. The core idea behind the prefix hashing is to feed a small unpredictable part of the public-key into the hash function instead of the entire large public-key itself. This significantly speeds up the computation of the KEM where the cost of this hash computation often dominates (in software implementations). An overview of the security advantages of such approaches is given in Table 1 (from [39]). We refer to Table 2 for an overview of the use of FO per LWR scheme.

4.2.1 Plain-, Ring-, Module-LWR Instantiations. Finally, we describe various instantiations of LWR-based KEMs that have appeared over the years. We focus on Ring and Module based KEMs due to their favorable key and ciphertext sizes. We highlight the unique design features and their advantages and disadvantages compared to alternatives. We summarize a high-level comparison in Table 2.

All schemes have reported performance results on popular `x86` computer architectures: including reference and optimized implementations often using the single instruction, multiple data programming paradigm. Unfortunately, these benchmarks are all done on a variety of different platforms (with platform specific optimizations) making direct comparisons difficult. Furthermore, the majority of the LWR-based schemes have not been submitted to the **ECRYPT benchmarking of cryptographic systems (eBACS)** [1] for benchmark and comparison. An exception is when targeting embedded devices where the popular `pqm4` [57, 60] project is used: a reference framework for benchmarking implementations of PQC KEM and signature schemes for Arm Cortex-M4

Table 1. Security Reduction of $\mathsf{Adv}^{(n,q_C)\text{-IND-CCA}}_{\mathsf{KEM}}$ to $\mathsf{Adv}^{(n,q_C)\text{-IND-CPA}}_{\mathsf{PKE}}$ (Denoted as Adv in the Table) in the ROM and QROM Models (Simplified) for the Different Fujisaki-Okamoto Variants

FO Variant	ROM	QROM
$\mathsf{FO}^{\not\perp}_{\mathsf{ID(pk)},m}$ [39]	$\mathsf{Adv} + q_F\delta(n) + \frac{n^2}{2^e}$	$\sqrt{q_F\mathsf{Adv}} + q_F^2\delta(n) + \frac{n^2}{2^e}$
$\mathsf{FO}^{\not\perp}_{\mathsf{pk},m}$ [48]	$\mathsf{Adv} + q_F\delta(n)$	$\sqrt{q_F\mathsf{Adv}} + q_F^2\delta(n)$
$\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$ [14, 48]	$\mathsf{Adv} + q_F\delta(n)$	$\sqrt{q_F\mathsf{Adv}} + q_F^2\delta(n)$
$\mathsf{FO}^{\not\perp}_{m}$ [48]	$\mathsf{Adv} + n \cdot q_F\delta(1)$	$nq_C \cdot \left(\sqrt{q_F\mathsf{Adv}} + q_F^2\delta(1)\right)$
$\mathsf{FO}^{\not\perp}$ [14, 48]	$\mathsf{Adv} + n \cdot q_F\delta(1)$	$nq_C \cdot \left(\sqrt{q_F\mathsf{Adv}} + q_F^2\delta(1)\right)$

Here, $\delta(n)$ is the n-user correctness error of PKE, q_C is the number of challenges, q_F is the number of (Q)ROM queries, and e is the min-entropy of ID(pk). The majority of the table is taken from [39, Figure 1] with the addition of $\mathsf{FO}^{\not\perp}$ and $\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$ following the result of [14, Theorem 5].

Table 2. Comparison of Parameters for Ring / Module LWR-based Post-quantum Cryptographic Schemes Targeting 128-bit Security

Scheme	$f(x)$	ℓ	ϵ_q	ϵ_p	ϵ_t	$D_{\mathbf{A}}$	$D_{\mathbf{s}}$	B	FPR	C	Q	FO
RLiz.-104*	$x^{512}+1$	1	10	8	5	$\mathcal{U}_q$	$\mathcal{H}$	1	-167	-	104	$\mathsf{FO}^{\not\perp}$
RLiz.-128*	$x^{1024}+1$	1	10	8	5	$\mathcal{U}_q$	$\mathcal{H}$	1	-145	-	128	$\mathsf{FO}^{\not\perp}$
Saber	$x^{256}+1$	3	13	10	4	$\mathcal{U}_q$	β_4	1	-136	189	172	$\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$
uSaber	$x^{256}+1$	3	12	10	4	$\mathcal{U}_q$	$\mathcal{U}_2$	1	-167	182	165	$\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$
Florete	$x^{768}-x^{384}+1$	1	10	9	3	$\mathcal{U}_q$	β_4	1	-131	–	157	$\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$
Espada	$x^{64}+1$	12	15	13	3	$\mathcal{U}_q$	β_4	4	-167	–	128	$\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$
Sable	$x^{256}+1$	3	11	9	4	$\mathcal{U}_q$	β_1	1	-143	–	169	$\mathsf{FO}^{\not\perp}_{\mathsf{pk}}$
SMAUG-T*	$x^{256}+1$	3	11	9	4	$\mathcal{U}_q$	spCBD	1	-179	180	159	$\mathsf{FO}^{\not\perp}_{\mathsf{m}}$

Using the MLWR parameters with ring $\mathbf{R} = \mathbf{Z}[x]/(f(x))$, matrix distribution $D_{\mathbf{A}}$, (ephemeral) secret key distribution $D_{\mathbf{s}}$ for $\mathbf{s}$ and/or $\mathbf{s}'$, and B bits per input message. with ϵ_q the log of q, ϵ_p the log of p, ϵ_t the log of t, ℓ the module rank (with $\ell = 1$ for ring-based schemes), and FPR the log of the failure probability. SVP classical (C) and quantum (Q) bit security are included.

* Uses LWE for key generation and LWR for Encaps/Decaps.

based embedded systems. This results in benchmark results (when available) on the same platform measured in a similar fashion. This allows for an apples-to-apples comparison. The available performance results, key and ciphertext sizes and are reported in Table 3.

(Ring) Lizard. Lizard [25] uses a mixed LWE and LWR approach. The public key is a LWE sample (with error terms sampled from a discrete Gaussian distribution) and the encryption procedure uses deterministic error: learning with rounding. Note that Lizard uses the plain version of LWE and LWR, but the proposals also include variants called **Ring-Lizard** (**RLizard**) that rely on the ring version of LWE and LWR, leading to smaller public keys and ciphertexts. A distinguishing feature of Lizard is its use of sparse binary secrets sampled from the distribution $\mathcal{H} = \mathcal{HWT}_m(h)$ sampling m elements from $\{-1, 0, 1\}$ of which exactly $m - h$ are non-zero.

Table 3. Comparison of Ring/Module LWR-based Post-quantum Cryptographic Schemes Targeting 128-bit Security

Scheme	PK	SK	CT	KeyGen		Encaps		Decaps	
				perf	stack	perf	stack	perf	stack
Saber	992	2,304	1,088	645	6.5	821	7.1	774	7.2
uSaber	992	2,208	1,088	-	-	-	-	-	-
Florete	896	1,152	1,248	439	17.8	814	18.0	953	18.0
Espada	1,280	1,728	1,304	2,343	2.8	2,568	2.1	2,497	2.0
Sable	896	1,152	1,024	745	6.0	1,004	5.9	1,028	5.4
SMAUG-T	1,088	1,312	992	-	-	-	-	-	-

Sizes for the public-key (PK), secret key (SK) and ciphertext (CT) are in bytes. Performance results (perf) and stack usage (stack) for KeyGen, Encapsulation and Decapsulation are reported in thousands of cycles and KiB, respectively. These figures are measured on a Cortex-M4 using the `pqm4` benchmarking framework [59].

There exist various instances of Lizard and RLizard. In 2016 a proposal was submitted to the NIST PQC competition [26] containing parameter sets for both Lizard and RLizard target the different NIST security levels. A subsequent article [25] contains a subset of the NIST parameter sets, in particular a version of RLizard targeting the NIST Level-I parameters. This selects $n = 1024$, $\epsilon_q = 10$, $\epsilon_p = 8$ and $\epsilon_t = 8$ (not explicitly stated, but inferred from the protocol description) leading to a failure probability of 2^{-154} and is referred to as **RLizard128**. A follow-up article [25] proposes further improvements in the implementation and introduces a variant **RLizard104** targeting 104-bit quantum security. A notable change is the introduction of $\epsilon_t = 5$ (denoted p' in the article) in the protocol description to also include rounding on the ciphertext.

Saber. The Saber scheme [32] is the first KEM that is fully based on (module) LWR. It was submitted to the NIST PQC competition in 2016 and made it to the third round [33]. By fully relying on LWR no sampling from an error distribution is needed, as opposed to RLizard, which still relies on LWE for key generation. The Saber design aims for simplicity and efficiency. The choice of ring $\mathbf{Z}_q[x]/(x^{256}+1)$ with $q = 2^{13}$ is fixed across all parameter sets, varying instead the module dimension, ciphertext modulus and secret key distribution, and the arithmetic modulo a power of two q allows for straightforward implementation without modular reduction or rejection sampling. The NIST proposal defines three parameter sets LightSaber, Saber, and FireSaber corresponding to NIST security level 1, 3, and 5, respectively. Secret keys are selected from a binomial distribution β_η with $\eta = 3, 4, 5$. A drawback of the choice of q is that NTTs are not natively supported in $\mathbf{Z}_q$ and the implementations therefore initially relied on Toom-Cook. However, by embedding into a larger ring with appropriately selected modulus (about twice the bit length of q) the polynomial multiplication can still be implemented with NTTs, albeit not quite as efficient as for example in Kyber. We refer to Section 4.1.2 for more details.

uSaber. Uniform Saber (**uSaber**) [33, Section A.2] is a variant of Saber proposed in the Round 3 submission of Saber that is not part of the main submission. The main difference is that the secret vectors $\mathbf{s}$ and $\mathbf{s}'$ are sampled from the uniform distribution $\mathcal{U}_2$ instead of the binomial distribution β_μ. This means that the secret coefficients are in the interval $[-2, 1]$ instead of the interval $[-\mu, \mu]$ in Saber and can be represented with exactly 2 bits each.

The main advantage of uSaber is its simplified secret generation process, which is used during key generation, encryption, and consequently in encapsulation and decapsulation for a KEM. A key benefit is the significant reduction in the amount of (pseudo-)random bits needed to achieve a

comparable secret variance, and thus a similar security level. For instance, uSaber parameter sets require only two random bits per secret coefficient, compared to the six to ten bits required by standard Saber when using binomial sampling. This leads to faster hashing, which in turn reduces both computational overhead and the attack surface. Since hashing is a highly non-linear process, this reduction is especially valuable for masked implementations.

Furthermore, uSaber eliminates the need to compute a binomial distribution. This simplification makes implementations more straightforward, particularly in masked settings where binomial sampling, being non-linear, adds considerable complexity. By removing it, uSaber not only enhances efficiency but also lowers the attack surface, resulting in a more secure and practical design.

Scabbard Suite: Florete, Espada, and Sable. Building upon the improvements of LWR-based schemes in many aspects such as multiplication routines, cryptanalysis, parameter selection, and so on, Scabbard [67, 74] was proposed during the course of NIST standardization effort as a suite of three KEMs. Each of the three KEMs is tailored to improve the state-of-the-art in different aspects such as speed, area requirements, memory footprint, and parallel implementation.

The first scheme "Florete" is a Ring-LWR scheme. As opposed to Saber, which uses module rank $\ell = 3$ and polynomial degree $n = 256$ over $\mathbf{Z}_q$ for $q = 2^{13}$, Florete relies on the ideal lattice defined by the ring $\mathbf{Z}_q[x]/(x^{768} - x^{384} + 1)$ for $q = 2^{10}$ and fixes the module rank to be $\ell = 1$. Aiming for the 192-bit security level of Saber, this reduces the matrix dimension of $\mathbf{A}$ from 3×3 to a single ring element which removes the need for matrix-vector products. An advantage of this approach is that fewer pseudo-random bytes are required for the generation of $\mathbf{A}$, which can be a bottleneck of performance (particularly in software). The design is done in such a way that the multiplication routines from Saber can be fully re-used. A downside of this design is that the security compared to Saber is reduced by approximately 15 bits [74, Table 2]. A similar approach is followed to match the security levels of LightSaber and FireSaber using rings with polynomial degree 512 and 1,024 respectively, as proposed in a later work [67].

The second scheme is called "Espada" and aims at improving efficiency by allowing for more parallelization in the algorithm. The main idea is to reduce the larger polynomial multiplication of 256 coefficients into multiple smaller multiplications consisting of 64 coefficients each. Although a 256×256 polynomial multiplication can be reduced to 64×64 polynomial multiplications through Toom-Cook or Karatsuba, the use of such algorithms does not enable the same level of parallelization and also increases the size of the multiplier due to the evaluation and interpolation stages of Toom-Cook and Kartasuba. By making $n = 64$ the native polynomial degree of the underlying lattice, the implementation cost of the base multiplication routine is reduced, particularly in hardware [75] and microcontrollers. It also improves the granularity of the parameter selection. As a result, the module rank ℓ is increased from 3 to 12 to achieve a 128-bit security level (and similarly for other security levels). While this increases the total number of 64×64 polynomial multiplications that are required compared to Saber, their independence makes parallel implementation more straightforward. The reduction of n from 256 to 64 requires the selection of $B = 4$ bits per coefficient, affecting the failure probability and ultimately requiring to increase the size of q to 15 bits.

The final scheme "Sable" follows the design of Saber most closely, but changes the secret key distribution. Whereas LightSaber, Saber, and FireSaber change their choice of η to select secret key coefficients in the range from the centered binomial distribution β_μ in $[-\eta, \eta]$ for $\eta = 5, 4, 3$ respectively, Sable fixes $\eta = 1$ across all parameter sets, which makes switching between different schemes more seamless. In this case, only 2 bits are required to store a secret key coefficient, making the storage requirement equal to uSaber and helping with masked implementations. Overall, this change reduces the complexity of the binomial multiplier and makes the multiplier design uniform across parameter sets if it is implemented using the maximum values of p and q. To make up for

the loss in security the values of p and q are decreased accordingly. The article remarks that the selection of p and q for Saber was using erroneous estimations as it was considering a *continuous* as opposed to the *discrete* uniform distribution in the rounding error. The LightSable, Sable, and FireSable parameter sets all assume the discrete uniform distribution which enables more accurate parameter selection and security estimations.

SMAUG and Friends. SMAUG-T [47], a merger of SMAUG [24] and TiGER [93], is the most recent LWR-based proposal and has been selected as a winner in the Korean PQC standardization effort. Similar to Lizard and RLizard, its key generation routine reduces to LWE while the Encaps/Decaps rely on LWR. For the generation of the ephemeral encapsulation secret they use a sparse centered binomial distribution, more precisely providing a ternary value which is zero with probability $1-2r$, -1 with probability r and 1 with probability r for $0 \leq r < 1/2$. The value r is selected as $1/8$ or $3/16$, where the power of two in the denominator is specifically chosen to ease implementation. It should be noted that the design has been updated in between v2.0 to v4.0 as a result of non-constant-time implementation of some of its samplers, so careful implementation is required. The advantage of the sparse secrets is that they reduce decryption failures and improve the efficiency of the KEM. In v3.0 the new "TiMER" parameter set was added to the specification, aiming at furthering lower the decryption failure probability due to the D2 error reconciliation mechanism. By encoding each message bit into two polynomial coefficients the probability of failure reduces at the cost of additional encoding/decoding.

4.3 Digital Signatures

The work on digital signature schemes based on LWR is more limited than for KEMs, and at the time of writing only a small number of designs explicitly place LWR at the core of the security claim. Existing work explores two complementary directions: (i) keeping the Dilithium-style structure but simplifying arithmetic by moving to a power-of-two moduli [78], and (ii) modifying Dilithium more fundamentally so that key generation and signing equations are expressed in terms of (module-)LWR rather than (module-)LWE [92].

A first line of work focuses on the observation that some practical advantages often associated with LWR, in particular efficient modular reduction and simpler bit operations enabled by power-of-two moduli, can already be used *without* changing the underlying assumption from LWE to LWR. Migliore et al. [78] study masking Dilithium and argue that replacing the prime modulus by a power of two can simplify arithmetic (e.g., eliminating rejection sampling modulo a prime) and replace Dilithium's decomposition steps (bases γ_1, γ_2) by operations that are essentially bit truncations. Importantly, this direction does not claim an LWR-based security reduction; rather, it illustrates that a large fraction of the implementation complexity (especially in masked settings) arises from modulus/representation choices and the resulting data-dependent procedures.

Okada et al. [92] propose a more significant redesign that aims at replacing Dilithium's module-LWE component by module-LWR. Whereas the public key for Dilithium is generated as an LWE sample $\mathbf{t} = \mathbf{A}\mathbf{s}_1 + \mathbf{s}_2$, the proposal is to generate $t = \lfloor \frac{p}{q}\mathbf{A}\mathbf{s}_1 \rceil$ instead, where $q = 2^{23}$ and p are powers of two. The error term $\mathbf{s}_2$ can be recovered deterministically as $\mathbf{s}_2 = t - \frac{p}{q}\mathbf{A}\mathbf{s}_1$. Similar changes are made to the computation of $\mathbf{w} = \mathbf{A}\mathbf{y}$ and $c\mathbf{s}_2$. As $\mathbf{s}_2$ is chosen not to be stored as part of the private key, but rather to be generated on-the-fly during signing, this results in a signature scheme with a smaller private key and faster key generation, but slower verification and signing than Dilithium (see [92, Table 3]).

Overall, the literature suggests that "LWR-ifying" Dilithium-like signatures is plausible, but the design space is narrower than for KEMs.

Conclusions and Future Directions

LWR has proven to be a strong foundation for post-quantum key establishment and, to a lesser extent, digital signatures. Key establishment schemes, particularly those following the PKE-to-KEM paradigm via the Fujisaki-Okamoto transform, have shown strong potential for post-quantum cryptography. Many schemes have been proposed demonstrating that LWR enables efficient, compact, and side-channel-resistant constructions, especially when using power-of-two moduli.

While KEMs dominate the LWR landscape, signature schemes are still underrepresented. Future work should explore native LWR-based signature designs and further optimize existing ones. In particular, it would be interesting if an LWR-based signature scheme could avoid the re-computation of $\mathbf{s}_2$ during signature generation. Additionally, more consistent benchmarking across platforms and deeper analysis of rounding strategies, secret distributions, and FO variants will help guide both theoretical and practical advancements.

5 Using LWR in High-Assurance Implementations

Since the late 1990s, it is publicly known that the statistical analysis of a power trace obtained when executing a cryptographic primitive might correlate to the secret key material used during execution [63]. This area of *side-channel* attacks and associated countermeasures has since then grown into a large subfield of cryptology. Examples of more advanced attacks include (but are not limited to) higher order attacks [76], correlation power analyses [21] and template attacks [22]. Such powerful attacks which use meta-data to deduce information about the secret key material are not always in scope. The adversary needs (for example) to be able to obtain a power trace to execute a power analysis attack. However, for many everyday applications (e.g., payment, transit, or identity solutions) such attacks are relevant and countermeasures need to be devised. For example, NIST in their submission requirements and evaluation criteria for the post-quantum cryptography standardization process emphasized this property [85]:

> Schemes that can be made resistant to side-channel attacks at minimal cost are more desirable than those whose performance is severely hampered by any attempt to resist side-channel attacks. We further note that optimized implementations that address side-channel attacks (e.g., constant-time implementations) are more meaningful than those that do not.

Not surprisingly, among the LWR-based schemes, Saber KEM stands out as the most studied scheme since it appeared early. Hence, for a side-channel analysis of LWR-based schemes we consider Saber as a representative example.

To organize our discussion, we divide this subsection into the following parts: First, we present a classification of SCAs relevant to Saber KEM. Then, we provide two subsubsections covering countermeasures in the CPA-security and CCA-security settings respectively. Next, we comment on SCAs targeting schemes beyond Saber. Finally, we evaluate the practical efficiency of these countermeasures.

5.1 Classification of SCA in Saber KEM

The Saber algorithm consists of two components: a CPA-secure public-key encryption scheme and a CCA-secure KEM. The key exchange protocol typically involves three stages, where we suppose Alice and Bob run this protocol. First, Alice runs key generation to produce a public-secret key pair and sends the public key to Bob. Second, Bob performs the encapsulation procedure using the received public key, generating a ciphertext and a shared session key, and sends the ciphertext to Alice. Third, Alice runs the decapsulation procedure using the secret key as well as received ciphertext to recover the same shared session key. See as an example Figure 1 (from [11]) which

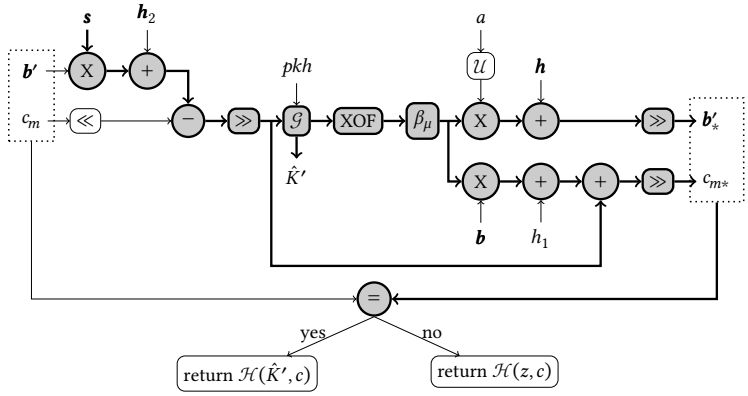


Fig. 1. Decapsulation of Saber. In grey the operations that are influenced by the long term secret $\boldsymbol{s}$ and thus vulnerable to side-channel attacks [11].

Table 4. Classification of Side-Channel Attacks in Saber's CPA-design Following the Framework from [100] Targeting Either Key Generation (KeyGen), Encapsulation (Encaps) or Decapsualtion (Decaps) using Single-trace Attacks with Ephemeral Keys

Attack Scenario	**Target Component**
Keygen	polynomial multiplication
Encaps	hashing and key derivation
Encaps	message encoding
Decaps	message decoding

displays the decapsulation of Saber and in grey all the operations that are vulnerable to side-channel attacks.

We follow the general analytical framework proposed in [100], originally developed for Kyber and Dilithium, to structure our classification of side-channel threats in the context of Saber.

5.2 Countermeasure in the Single Trace SCA

In this section, we classify side-channel vulnerabilities in Saber's CPA-secure design based on four critical implementation components as outlined in Table 4. Each of these components introduces specific leakage surfaces and attack opportunities for single-trace adversaries.

Polynomial multiplication. In the original Saber approach, polynomial multiplication is implemented using the Toom-Cook algorithm [28, 115]. In [70], it is shown that a single **electromagnetic** (**EM**) trace from Toom-Cook multiplication can be used to recover the secret key under a statistical and algebraic **side-channel analysis** (**SASCA**) model. The attack has two phases. In the *profiling phase*, the attacker collects EM traces from a cloned device to build templates of key-dependent operations. In the *key recovery phase*, a single trace from the target is matched to these templates.

Countermeasures include applying masking to operands to break leakage correlations. Operation-level shuffling, dummy executions, and encoding randomization can add noise and uncertainty. Despite being constant-time, Toom-Cook's regular structure remains vulnerable in high-resolution side-channel settings.

Hashing and key derivation. Like Kyber, Saber uses the Keccak-based SHAKE-128 XOF for seed expansion and key derivation, inheriting similar side-channel risks. Leakage from Keccak's internal state can reveal input or seed data as demonstrated in [58]. To mitigate such SASCA-style attacks, countermeasures like internal state shuffling, timing jitter, and targeting low-SNR platforms have been proposed to reduce trace alignment and increase noise.

Message encoding. During encryption, Saber encodes the message into polynomial coefficients, which are then compressed. The scaled coefficients differ in Hamming weight based on bit values. Early attacks on NewHope [6] inspired a line of work targeting message encoding in lattice-based KEMs [112, 121]. More recent studies focus specifically on Saber [7, 88, 90], showing that these attacks remain effective under first-order and even higher-order masking, and exploit leakage from all message shares using profiling or deep learning. Thus, masking alone is insufficient: constant-time encoding, share re-randomization, or shuffling are recommended for high-security implementations.

Message decoding. During decryption, Saber reconstructs the message by scaled subtraction and rounding. Like encoding, this step manipulates each coefficient, making it vulnerable to single-trace side-channel attacks. These attacks exploit Hamming weight differences and remain effective even on masked implementations [7, 88, 90, 99]. Thus, shuffling or re-randomizing decoding operations is recommended for stronger protection.

a In the multi-trace setting, the non-ephemeral key usage in Saber allows the attacker to observe multiple executions of the decapsulation procedure under the same secret. Compared to single-trace scenarios, the primary advantage is that an attacker can aggregate information across traces, either passively or through chosen ciphertexts. For a passive attacker, attacks such as correlation power analysis are accessible. In this case, the attack focuses on repeated polynomial multiplications, where intermediate values maintain a consistent secret-dependent structure. As demonstrated by [82], even sanity checks cannot fully eliminate leakage from such operations. In this case, masking remains the most effective defense.

For an active attacker, the threat surface expands further. In this case, **side-channel-assisted chosen ciphertext attacks** (**SCA-assisted CCA**) emerge, which can be classified using the four oracle models.

Binary Plaintext-Check Oracle. The attacker crafts highly sparse ciphertexts so that a specific bit of the decrypted plaintext depends exclusively on one secret-coefficient. By observing side-channel leakage during decryption, the attacker determines whether the decrypted bit equals their guess about the secret coefficient, effectively instantiating a one-bit oracle. The attacker applies this strategy repeatedly to learn the secret coefficients one-by-one. Such SCA-assisted CCA attack was introduced in [36], targeting LAC KEM via timing leakage in non-constant-time BCH decoding. It was generalized in [101] to perform key recovery in several constant-time LWE and LWR-based KEMs by observing EM or power side-channel leakage.

In the CCA-secure KEM setting, the attacker crafts a sparse ciphertext $\mathrm{ct} = (u, v)$ with:

$$u_i = \begin{cases} U \cdot x^0 & \text{if } i = 0, \\ 0 & \text{otherwise,} \end{cases} \qquad v = V \cdot x^0,$$

where $U, V \in \mathbf{Z}^+$ are attacker-chosen. The decrypted bits m'_i then satisfy:

$$m'_i = \begin{cases} \mathrm{Decode}(V - U \cdot s_0[0]) & \text{if } i = 0, \\ \mathrm{Decode}(-U \cdot s_0[i]) & \text{for } i \geq 1. \end{cases}$$

Each bit of m' thus depends on a single coefficient of s_0. If the adversary can detect whether m' matches a guess (e.g., via power leakage), a bitwise recovery oracle is formed.

Masking the decapsulation process mitigates this attack. While higher-order attacks remain possible, they require exponentially more traces.

Parallel Plaintext-Check Oracle. Parallel PC Oracle attacks extend the binary case by extracting multiple key bits per query. By crafting ciphertexts that induce secret-dependent variations in the first P bits of the decrypted message m', the attacker reduces the number of required traces by a factor of P. The general strategy is to set the ciphertext components as

$$u_i = \begin{cases} U \cdot x^0 & \text{if } i = 0, \\ 0 & \text{otherwise,} \end{cases} \qquad v = V \cdot \sum_{i=0}^{P-1} x^i,$$

with $U, V \in \mathbf{Z}^+$. This yields $m'_i = \text{Decode}(V - U \cdot s_0[i])$, for $i \in [0, P-1]$, where each m'_i acts as a binary distinguisher for $s_0[i]$. This P-way oracle reduces trace complexity, though profiling cost grows exponentially with P due to 2^P output combinations. With access to a clone device, profiling can be done offline. Experiments show $P \leq 10$ is feasible [98, 114], and full key recovery on Saber is possible with as few as 1,152 traces for $P = 8$ [114].

As in the binary case, masking the decapsulation procedure mitigates this attack. Higher-order masking increases resistance by requiring exponentially more traces.

Decryption Failure (DF) Oracle. DF oracle-based attacks exploit the fact that decryption failures can leak information about the secret key. By crafting ciphertexts that trigger failures depending on specific secret coefficients, an attacker can recover the key through repeated queries. The first such attack, targeting FrodoKEM, used timing leakage in non-constant-time comparison logic [45]. Later work extended this to power/EM side channels [13], showing that even masked implementations can leak distinguishable patterns during failure events. While initially demonstrated on LWE-based schemes, the same principles may apply to LWR-based KEMs like Saber, provided side-channel leakage can distinguish successful from failed decryptions. Although more traces are needed than in the binary oracle case (around 6k), the attack remains practical in unprotected, low-SNR environments.

Full Decryption (FD) Oracle. FD Oracle-based attacks represent the strongest form of chosen-ciphertext side-channel attacks, aiming at recovering the entire decrypted message m' in a single query, far surpassing Binary and Parallel PC attacks. [121] showed that message recovery techniques exploiting leakage from encoding can be adapted to the CCA setting, enabling full key recovery on Kyber512 in as few as 6 queries. Similar attacks on Saber and other LWR-based schemes [89, 117, 118] exploit leakage from bitwise decoding, achieving full key recovery within 20 traces. In Saber, rounding during decapsulation causes bit-dependent leakage. [99] demonstrated that even with shuffling, controlled ciphertext perturbations and Hamming weight comparisons can recover each bit. Recent work [7] further showed that profiling-based attacks can bypass combined masking and shuffling given enough queries. The only robust countermeasure is frequent key refresh, preventing attackers from accumulating sufficient traces per key instance.

Apart from the abovementioned four oracles, the polynomial multiplication where the non-ephemeral secret key is multiplied with the ciphertext can be exploited by correlation power analysis. It was first demonstrated in [83]. In this work, the authors demonstrated the attack on all three lattice-based KEMs, which were the finalists of the NIST procedure. For Kyber, the multiplication happens between the secret coefficients, which are in the NTT domain. Due to this, each secret expands to a 13-bit number. However, for Saber, the secrets also expand to a larger range due to the Toom-Cook and Karatsuba transformation. But this expanded range is much smaller

Table 5. Cycle Overhead Factor of a Masked (tth Order) Versus an Unprotected Implementation on Cortex-M4 [65]

Masking Order t	Saber	uSaber	Kyber
1	2.7×	2.3×	5.6×
2	5.0×	4.2×	9.5×
3	7.7×	6.5×	13.0×

compared to Kyber. Due to this, a major observation in that work was that the correlation power attacks are much easier to mount on a LWR-based scheme where Karatsuba- and Toom-Cook-based multiplication is used compared to the LWE-based schemes where NTT is used.

5.3 Practical Efficiency of Countermeasures

Among countermeasures against side-channel attacks, masking is the most computationally expensive but also the most effective one against **differential power analysis** (**DPA**), especially when targeting static keys across multiple traces. Masking splits a secret s into t+1 random shares $(s_0, \ldots, s_t)$ such that $s = \bigoplus_i s_i$ (Boolean) or $s = \sum_i s_i$ (arithmetic). Computations are performed on the shares independently, randomizing leakage and requiring (t+1)th-order statistical moments for successful DPA. Early masked implementations, such as the first-order ring-LWE scheme in [104], used arithmetic masking in $\mathbf{R}_q$. While polynomial operations scale linearly with t, decoding became the main bottleneck, requiring up to 16 share refreshes per coefficient. [91] improved this with a more efficient masked decoder using distribution shifting and **arithmetic to Boolean** (**A2B**) mask conversion.

Despite these advances, first-order masking still incurs significant overhead: polynomial multiplication roughly doubles in cost, and bitwise operations like error sampling and randomness generation can be over 5× slower. [11] observed that in masked implementations, non-linear operations (e.g., binomial sampling, hashing) dominate the cost. These often require Boolean masking, while polynomial arithmetic uses arithmetic masking: requiring mask conversions. LWR-based schemes, like Saber, benefit from two key advantages:

— Power-of-two moduli simplify mask conversions.
— Rounding is easier to mask than noise sampling.

These properties motivated the first-order masked Saber framework, achieving a 2.5× overhead on ARM Cortex-M4: compared to 5× for LWE-based schemes. Higher-order masked implementations of Saber and uSaber [65] show that LWR schemes scale better than LWE counterparts with increasing t. uSaber is even more efficient due to its uniform secret distribution, which simplifies sampling. See Table 5.

The masked implementations of LWR-based KEMs Florete, Espada, and Sable from the suite, Scabbard, also perform better than Kyber due to the power-of-two moduli, and these schemes are portrayed as masking-friendly schemes in [66]. Masked Florete and masked Sable are even faster than masked Saber for any order masking due to their efficient parameter choices. Although uSaber is shown as a mask-friendly version of Saber in [65], Sable performs better than uSaber, and that also remains the same after masking, as masked Sable showed slight improvement over uSaber in performance for any order masking.

In hardware, [40] proposed RISC-V instruction set extensions for masking Kyber and Saber. Their results confirm Saber's masking efficiency over LWE schemes. [2] further optimized masked polynomial multiplication for power-of-two moduli. Their Saber hardware achieves first-order

resistance with only 1.7× area overhead and completes multiplication in just 340 cycles: significantly outperforming generic masking approaches.

Conclusions and Future Directions

LWR-based schemes, especially when using power-of-two moduli, are efficient on various software and hardware platforms. Nevertheless, these schemes remain susceptible to various side-channel attacks. Masking is one of the most effective algorithmic countermeasures against side-channel attacks, though it introduces performance costs. The deterministic structure of LWR-based schemes and the use of power-of-two moduli facilitate simpler and more efficient masking, and allows for optimized arithmetic operations. Compared to their LWE-based counterparts, LWR-based schemes generally achieve better side-channel resistance at a lower cost, making them attractive in constrained environments.

Future research on LWR-based cryptographic schemes should focus on enhancing the efficiency and scalability of masking techniques, particularly for higher-order protection. Expanding side-channel analysis beyond Saber to include other LWR-based schemes is essential for a broader understanding of their resilience. Additionally, improving automated tools for detecting side-channel vulnerabilities and verifying countermeasures, especially through formal methods, will support more robust implementations. Optimizing design choices, including secret distributions and rounding strategies, can further balance security and performance.

6 Implementation and Deployment Guidance

This section provides practical recommendations for deploying LWR-based schemes on constrained platforms. We focus on consolidating modulus selection guidelines, multiplication strategies, embedded bottlenecks, and masking overheads, based on the analysis presented throughout this work.

6.1 Modulus Choice: Practical Implications

Power-of-two moduli ($q = 2^k$). Although not mandatory, this is the usual choice for LWR-based schemes (such as Saber and the Scabbard suite). The main advantages of going for a power-of-two modulus include: (i) easy constant-time sampling and arithmetic due to trivial modular reduction, (ii) simple rounding and compression operations, (iii) masking-friendly arithmetic with efficient conversions between Boolean and arithmetic masks. These points follow from the properties discussed in Section 4 and the implementation behavior documented throughout Section 5. One drawback of 2^k-based rings is the absence of native NTT-based polynomial multiplication, which has the fastest asymptotic time complexity. However, as detailed in Section 4.1.2, the NTT method can still be used as an "embedded NTT" by temporarily lifting computations into an NTT-friendly prime modulus.

Prime moduli (for comparison). Prime moduli q can enable asymptotically fast native NTTs but require rejection sampling and more complex constant-time coding. We include this option here only for contrast with LWE deployments; for LWR-based schemes there is typically no reason to adopt prime moduli. For microcontroller-class devices or deployments requiring masking, power-of-two moduli are generally preferable mainly due to simpler and cheaper arithmetic-Boolean conversions compared to prime moduli. Embedded NTTs can close much of the multiplication-performance gap while retaining the benefits of $q = 2^k$: especially when high-assurance implementations are in scope (see Section 5).

6.2 Polynomial Multiplication Strategy

Polynomial multiplication dominates the arithmetic cost of all LWR-based constructions surveyed in Section 4. The optimal polynomial multiplication strategy depends on the parameter sets of the LWR-based scheme, architectural or platform constraints, and not just on asymptotic complexity.

NTT multiplication. The classical NTT enables polynomial multiplication in $\mathcal{O}(n \log n)$ time provided that the modulus admits a suitable primitive root of unity. In practice, this typically requires a prime modulus in the polynomial ring. As LWR-based schemes typically choose power-of-two moduli, polynomial rings do not support native NTTs. Following Section 4.1.2, one possible way to use embedded NTT where we lift the computation into a larger transform-friendly modulus q', perform multiplication in $R_{q'}$ using an NTT, and then reduce the result back to the original ring. For parameters used in Saber and FireSaber, this achieves noticeable speedups for polynomial multiplication on software platforms (up to 60% in the reported measurements) at the cost of using increased memory usage due to the larger modulus q' used [27]. On hardware platforms, where custom data-path design is the convention, the embedded NTT may not be the best choice over other types of polynomial multiplication method.

Toom-Cook and Karatsuba multiplication. These divide-and-conquer approaches are typically the usual options for polynomial multiplication in rings where NTTs are not possible (due to a lack of proper roots). Asymptotically, these algorithms are slower than NTT-based multiplication. However, when the polynomial ring has a small degree, e.g., $n = 256$, the asymptotic disadvantages of the Toom-Cook-k ($\mathcal{O}(n^{\log_k(2k-1)})$) and Karatsuba (which can be seen as a Toom-Cook-2 with a time complexity of $\mathcal{O}(n^{\log_2(3)}) \approx \mathcal{O}(n^{1.585})$) methods do not become visible in practice.

On resource-constrained software platforms and custom hardware platforms, Toom-Cook and Karatsuba methods can be competitive to NTT-based multiplication. The two algorithms have a highly linear memory access structure, unlike NTT. Furthermore, with power-of-two moduli, the modular reductions become free, significantly reducing arithmetic cost. The ARM Cortex-M study on implementing Saber [61] shows that carefully engineered combinations of Toom-Cook, Karatsuba, and optimized schoolbook multiplication can compete with or outperform NTT-based implementations.

Schoolbook multiplication. The classical schoolbook algorithm has $\mathcal{O}(n^2)$ time complexity, slower than both NTT and divide-and-conquer approaches. Yet, schoolbook multiplication remains highly relevant in LWR-based implementations, particularly in hardware due to its very simple structure. The memory access pattern is entirely linear making it easy to unroll the multiplication loops and use on-chip flip-flops for storing the two operand and one result polynomials. For example, on FPGA and ASIC platforms, fully parallel schoolbook multipliers [10, 108] have been shown to compute a complete degree-256 polynomial multiplication in 256 cycles by instantiating one MAC unit per coefficient. Because modular reduction modulo 2^k is free, no additional reduction circuitry is required.

Rule of thumb. Assume a power-of-two modulus is used. On a typical embedded platform where memory footprint, stack usage, and code size are the primary constraints, a Karatsuba/Toom-Cook strategy with an optimized schoolbook base case is a natural starting point. These methods preserve linear memory access patterns, exploit the smallness of secret coefficients and efficient modular reduction, and avoid the coefficient growth introduced by transform-based approaches.

Lifting to a larger modulus and using an embedded NTT can reduce cycle counts, particularly on platforms with SIMD acceleration. However, this improvement comes at the cost of increased memory usage, more complex reduction logic, and potentially higher masking overhead. Therefore,

NTT-based approaches are most attractive when cycle count dominates over memory, area, and implementation simplicity.

6.3 Common Embedded Bottlenecks

Polynomial multiplication. As usual in virtually all LWR and LWE approaches the hashing evaluations require most of the run-time. Next, (polynomial) multiplication requires significant cycle counts in key generation, encapsulation, and decapsulation. On embedded platforms with limited SRAM, the main bottleneck is often not the number of multiplications but the number of memory accesses. If intermediate polynomials are repeatedly written back to memory, the gain from reduced multiplication count can be offset by load/store overhead. Careful in-place variants and shallow recursion levels are therefore critical for performance stability. Techniques described in Section 4 and Table 3 provide measurable guidance.

Pseudo-random expansion and sampling. The XOF cost (typically SHAKE-128) becomes the main component in LWE- and LWR-based schemes. With $q = 2^k$ in LWR-based schemes, sampling requires exactly k bits per coefficient and no rejection, thus saving pseudo-randomness generation. Embedded microcontrollers often provide only a few kilobytes of RAM. Full storage of the public pseudorandom matrix in algorithms can exceed available stack space. Hence, seed-based on-the-fly generation of the matrix, just-in-time computation, and streaming accumulation are therefore essential techniques. As discussed in Section 4.1.3, storing only seeds for A and s reduces the persistent memory footprint while retaining predictable cycle counts.

Message encoding and compression. LWR schemes rely on rounding and bit truncation for ciphertext compression. Although each individual operation is inexpensive, repeated packing and unpacking of coefficients can introduce non-negligible overhead on embedded platforms. Bit extraction and recombination must therefore be implemented carefully to minimize shifts, masks, and temporary buffers, When strong side-channel resistance is required, additional countermeasures such as coefficient shuffling or re-randomization (see Section 5) are required.

Platform-specific arithmetic units. Finally, the availability of DSP instructions or MAC units strongly influences the bottleneck profile. On platforms with efficient MAC instructions, optimized schoolbook or shallow Karatsuba can be highly efficient. Consequently, multiplication strategy and bottleneck analysis must be co-designed with the target micro-architecture.

6.4 Masking: Expected Overheads and Practical Reductions

In masked KEM decapsulation, the dominant overhead is often not polynomial arithmetic itself, but the integration layer between arithmetic and bit-level routines. For t-th order masking, arithmetic operations such as additions and multiplications typically scale close to the expected $(t + 1)$ share factor. In contrast, bit-wise components, especially Boolean-arithmetic share conversions, masked comparisons, sampling routines, and hash-related gadgets, grow significantly faster and increasingly dominate runtime and randomness consumption at higher orders.

Table 5 in Section 5 reports measured overhead factors on a Cortex-M4. Saber and uSaber consistently exhibit lower masking overhead than many NTT-centric LWE schemes. One reason is the structure of LWR decapsulation: masked re-encryption does not require integrating additional masked noise polynomials, thereby reducing the number of costly masked sampling and combination steps. Parameter choices further influence masking efficiency, e.g., choosing power-of-two moduli simplifies Boolean↔arithmetic mask conversions [11]. For hardware deployments, masking efficiency also depends on whether the small-secret multiplier structure can be preserved. Naive

masking removes this advantage, whereas dedicated masked multiplication techniques [2] retain compact architectures.

In summary, reductions in masking overhead for LWR-based schemes are achieved by exploiting the LWR decapsulation structure, minimizing Boolean↔arithmetic conversions, carefully selecting secret distributions and parameters, and on hardware using masking techniques that preserve compact multiplier architectures.

Implementation and Deployment Summary

- **Modulus:** Prefer $q = 2^k$ for constant-time implementations and masking.
- **Multiplication:** Use Karatsuba/Toom–Cook when compactness is the main criteria; consider embeddings in a larger polynomial ring which enable NTT computations for significant speedups.
- **Bottlenecks:** Efficients hash evaluations possibly with hardware accelerators. (Polynomial) multiplication is the next dominant factor. Batch XOF use; keep encoding/decoding constant-time with shuffling if needed.
- **Masking:** Expect ≈ 2.3×–7.7× overhead (1st–3rd order). Use power-of-two moduli and simple secret distributions for best performance.

7 Failure Probabilities

In lattice-based encryption schemes, security is achieved by introducing an error term. The combination of these error terms, e.g., the one in the public key combined with the one from encryption, can sometimes result in protocol failures. For encryption schemes and KEMs, these failures constitute an attack vector and as such it is important to understand the potential threat of decryption failures.

7.1 Decryption Failures

This section will look at decryption failures in encryption schemes, which happen when there is excessive noise due to the error terms. The main issue with these failures is that they reveal information about the secret key, as their occurrence depends on it. A failure happens if the message cannot be correctly decoded. Following the formula in Section 4.1.4, a decryption is successful if:

$$-\frac{q}{2 \cdot 2^B} \leq (\mathbf{u'}^T\mathbf{s} - \mathbf{u}^T\mathbf{s'} - \mathbf{u''})_i < \frac{q}{2 \cdot 2^B}.$$

Here, the s-terms represent the secret terms, sampled as part of the LWR problem. In contrast, the u terms are the errors introduced as part of the LWR rounding. This term is similar to the failure probability in LWE schemes, where one replaces the $\mathbf{u}$-terms with $(\mathbf{e} + \mathbf{u})$-terms where $\mathbf{e}$ is the error introduced due to the LWE problem.

Some of these terms are generated by Alice, while others originate from Bob. Grouping them, we can define

$$\boldsymbol{\mathcal{S}} = \begin{pmatrix} -\mathbf{s} \\ \mathbf{u} \end{pmatrix}, \ \boldsymbol{\mathcal{C}} = \begin{pmatrix} \mathbf{u'} \\ \mathbf{s'} \end{pmatrix}, \text{ and } \boldsymbol{\mathcal{G}} = \mathbf{u''}.$$

In this breakdown, $\boldsymbol{\mathcal{S}}$ is generated by Alice and encompasses all her secret elements within the encryption scheme. Conversely, $\boldsymbol{\mathcal{C}}$ and $\boldsymbol{\mathcal{G}}$ are generated by Bob. With these definitions, we can rewrite the failure equation more concisely (and using a slight approximation) as

$$\|\boldsymbol{\mathcal{S}}^T\boldsymbol{\mathcal{C}} + \boldsymbol{\mathcal{G}}\|_\infty > \frac{q}{2^B}. \tag{3}$$

As this equation demonstrates, the occurrence of a decryption failure reveals information about the secret key $\boldsymbol{\mathcal{S}}$. In most scenarios, all terms within the failure equation are small, preventing

modulus reduction. The above equation thus yields a linear inequality involving the secret, which can subsequently be exploited to reduce the security of the LWE or LWR sample holding the secret key.

Comparing LWE or LWR, security is defined based on the $\mathbf{e}$ and $\mathbf{u}$ terms respectively. The advantage of LWR is that the error introduced for security naturally comes with ciphertext compression. For LWE one can do additional ciphertext compression but this increases the noise with an additional $\mathbf{u}$ term and thus the failure rate.

Type of attacks. For schemes that achieve IND-CPA security but lack protection against chosen-ciphertext attacks, an adversary can readily craft ciphertexts designed to yield specific values for $\mathcal{C}$ and $\mathcal{G}$. Typically, such an adversary would generate ciphertexts with an approximate 50% probability of decryption failure. By submitting these crafted ciphertexts for decryption and observing whether a failure occurs, the adversary gains a linear equation involving the secret. By accumulating a sufficient number of these equations, the secret can be trivially recovered.

These types of chosen-ciphertext attacks were initially described in [55] for NTRU, and were subsequently extended in [52] and [43]. Similar attacks targeting lattice-based homomorphic schemes have since been developed [71], and specifically on LWE-based schemes [8] and other works. Importantly, these techniques are adaptable for use against LWR schemes. The clear implication of these attacks is that for schemes lacking IND-CCA security, secret keys should never be reused.

The situation differs significantly for schemes offering IND-CCA security (also known as security against chosen-ciphertext attacks). The aforementioned attacks rely on specially crafted ciphertexts, and can therefore be prevented by restricting an adversary's ability to choose ciphertexts. This is commonly achieved through a chosen-ciphertext transform (see Section 4.2), In such cases, attacks become considerably more complex, thereby permitting the safe reuse of secret keys. We will delve further into decryption failure attacks on IND-CCA schemes in the remainder of this section.

Using decryption failures. If a decryption failure is discovered, that is if a ciphertext is found with $\mathcal{C}$ and $\mathcal{G}$ that triggers such a failure, a linear inequality in the secret is obtained, as shown in Equation (3). Since the coefficients of all terms are typically small, no modular reduction occurs. This ensures the sample is useful and not merely another LWE/LWR sample. In [31], it is demonstrated that these inequalities can be used to reduce the hardness of the primary sample protecting the secret key. The reduction in hardness of samples under hints, such as inequalities, was further formalized and generalized [29], which also provided an open-source codebase for calculating security after hints are applied.

One of the challenges with this approach is that the specific coefficient that caused the failure, and whether the failure was positive or negative, remain unknown. One method for determining the relative position of errors leverages the fact that two failing ciphertexts are both aligned with the secret, and thus also roughly aligned (and correlated) with each other, as shown in [31]. This effect becomes more pronounced with lower failure probabilities. The authors in [31] used this correlation to estimate the relative failure position for schemes with very low failure probability. Later, [34] improved upon this technique by employing a belief propagation algorithm.

Interestingly, [15] demonstrated that while decryption failures indeed offer information about the secret key, the same holds true for successful decryptions. However, given their significantly higher probability of occurrence, the information gleaned from these successful decryptions is considerably less. Currently, all existing decryption failure attacks still depend on identifying decryption failures and do not directly process successful decryptions to extract information.

Finding decryption failures. To locate decryption failures, one can generate ciphertexts and submit them to Alice for decryption, subsequently observing her response to ascertain if a failure occurred. Due to the FO-transformation, it is crucial that ciphertexts are generated via the correct

procedure, otherwise they will be rejected. A straightforward approach of generating random valid ciphertexts would result in decryption failures with a probability equivalent to the underlying scheme's intrinsic failure probability.

In [31] it is demonstrated that it is possible to generate valid ciphertexts with an increased failure probability. These are termed 'weak ciphertexts' and are produced by generating a large number of ciphertexts and then selecting those with high $\mathcal{C}$ and $\mathcal{G}$ norms. This method offers three key advantages: firstly, a quantum computer can accelerate this search (whereas the search for decryption failures on Alice's machine cannot be quantumly sped up); secondly, if there is a limit on the number of queries that can be submitted, precomputation can enhance the failure rate of each query; and thirdly, if multiple targets are available, precomputation can identify weak ciphertexts applicable to multiple targets. This latter advantage is less pronounced for LWR compared to LWE. In LWE, the s', e', and e'' error terms are generated independently of the public key. Conversely, in LWR, only the s' term is independent of the secret key, with the error terms u' and u'' being generated based on a, which is part of the secret key.

Similar to the concept of weak ciphertexts, one can also search for weak keys. If an attacker aims at compromising the secret key of one among several parties, they can probe all parties and identify one possessing a larger-than-average $\mathcal{S}$, which consequently leads to a higher decryption failure probability. In [31], an attack leveraging this property was successfully mounted while carefully limiting the number of required queries. This attack was subsequently generalised and extended in [30], which currently also provides the most state-of-the-art scripts for estimating the efficiency of decryption failure attacks against generic schemes.

In some encryption schemes, error correction is used to reduce the failure probability. While this does indeed reduce the failure probability and make attacks more expensive, [37] showed that one has to be careful when calculating the failure probability, as the individual coefficient inputs to the error correction are correlated and thus the error correcting code is less efficient. In case of the LWE scheme LAC [72], the calculation assuming independence lead to an underestimation of the failure probability of 2^{48}. Later, [46] mounted an attack on the LAC scheme that used the fact that weak ciphertexts are also more effective for schemes with ECC.

For a successful decryption failure attack, multiple failing ciphertexts are required. In [34], it is demonstrated that once a single failure is identified, it can significantly reduce the cost of finding subsequent failures through a procedure known as failure boosting. The expense of discovering failures in a generic setting is primarily attributed to finding the initial one. In [15], it is further shown that even successful decryptions provide information that aids in more efficient discovery of decryption failures, which was later improved by [119]. However, in certain specific scenarios, such as the multi-target setting in [30], the cost of finding a second failure (in one particular target) is significantly larger than the cost of finding the first failure across all potential targets. As a result, the attack cost is generally the cost of finding the first failure; however, in the case of multi-target attacks, the attack cost is the combined cost of finding the first and the follow-up failures.

Provable security. Decryption failures are accounted for in the security proofs of KEM systems, as illustrated in Table 1. Specifically, the proofs include terms like $\delta \cdot q_F$ for classical attackers and $\delta \cdot q_F^2$ for quantum attackers. Here, δ represents the failure probability of the underlying encryption scheme, and q_F denotes the number of queries made to a random oracle or to the decryption oracle (e.g., hash function executions or queries to the server). Throughout the rest of this section, we will refer to these queries as operations.

In the classical scenario, roughly $q_F \approx \delta$ operations are needed to reach the proven security bound. Yet, in the quantum setting, only about $q_F \approx \sqrt{\delta}$ operations are required. This quadratic speedup is attributed to a worst-case scenario where Grover's algorithm could potentially accelerate

the search for decryption failures. However, we do not see a comparable quadratic speedup in real attacks, where the speedup using a quantum computer leans more toward the linear case than the quadratic case. The reason an attacker cannot fully utilize Grover's algorithm is that, without precise knowledge of the secret key, they cannot accurately predict if an error will occur. This limits an attacker to searching for weak ciphertext attacks.

Bridging the gap between practical attacks and provable security remains an interesting open question for lattice-based schemes. Improvements toward tighter security proofs have been made in several articles [14, 48–51, 56, 64, 109]. A significant complication is that attackers gain small pieces of information about the failure probability through successful queries, as noted by Nina Bindel and John Schanck [15]. This makes the search progressively more efficient with more queries, a factor that must be considered when developing a tighter security proof.

Conclusions and Future Directions

Decryption failures in LWR-based encryption schemes pose a subtle but serious threat, as even rare failures can leak information about secret keys. These failures can be exploited through chosen-ciphertext attacks. While current security proofs account for such failures, they often rely on conservative bounds that may not reflect practical attack scenarios.

Looking ahead, research should focus on narrowing the gap between theoretical security guarantees and real-world attack models. This includes refining failure probability estimates, improving the robustness of schemes against failure-based leakage, and developing tighter, quantum-aware security proofs. Additionally, practical tools for simulating and analyzing decryption failures will be essential for guiding secure parameter selection and implementation.

8 Conclusions and Looking Forward

This Systematization of Knowledge has demonstrated the maturation of LWR as a foundational alternative for lattice-based cryptographic constructions. It has gone from a theoretical variant of LWE into a practical and competitive cornerstone of post-quantum cryptography.

The theoretical foundations have proven to be as mature as those of LWE as well as enable more practical parameter selection. We have surveyed the landscape of LWR constructions and analyzed the security and implementation tradeoffs. Moreover, they have shown their suitability to achieve security, efficiency and compactness of implementation, and usability of their constructions.

Currently, the landscape of LWR schemes is dominated by KEMs where particularly power-of-two moduli constructions outstand in efficiency of implementations as well as protection against side-channel attacks. Moving forward, there is a need to develop native LWR-based signature schemes to complete the post-quantum cryptographic needs.

In conclusion, LWR has firmly established its place in the post-quantum toolkit. By consolidating the last years' research, this SoK provides a clear foundation and a roadmap for the continued development and deployment of LWR-based cryptography in the post-quantum era.

References

[1] Daniel J. Bernstein and Tanja Lange (Eds.). 2025. *eBACS: ECRYPT Benchmarking of Cryptographic Systems*. Retrieved from https://bench.cr.yp.to/

[2] Aikata, Andrea Basso, Gaëtan Cassiers, Ahmet Can Mert, and Sujoy Sinha Roy. 2023. Kavach: Lightweight masking techniques for polynomial arithmetic in lattice-based cryptography. *IACR TCHES* 2023, 3 (2023), 366–390. DOI: https://doi.org/10.46586/tches.v2023.i3.366-390

[3] Miklós Ajtai. 1996. Generating hard instances of lattice problems (extended abstract). In *Proceedings of the 28th ACM STOC*. ACM, 99–108. DOI: https://doi.org/10.1145/237814.237838

[4] Martin R. Albrecht, Rachel Player, and Sam Scott. 2015. On the concrete hardness of Learning with Errors. *Journal of Mathematical Cryptology* 9, 3 (2015), 169–203. DOI: https://doi.org/doi:10.1515/jmc-2015-0016

[5] Joël Alwen, Stephan Krenn, Krzysztof Pietrzak, and Daniel Wichs. 2013. Learning with rounding, revisited - new reduction, properties and applications. In *Proceedings of the CRYPTO 2013*, Ran Canetti and Juan A. Garay (Eds.). LNCS, Vol. 8042, Springer, Berlin, 57–74. DOI: https://doi.org/10.1007/978-3-642-40041-4_4

[6] Dorian Amiet, Andreas Curiger, Lukas Leuenberger, and Paul Zbinden. 2020. Defeating newhope with a single trace. In *Proceedings of the 11th International Conference on Post-Quantum Cryptography*, Jintai Ding and Jean-Pierre Tillich (Eds.). Springer, Cham, 189–205. DOI: https://doi.org/10.1007/978-3-030-44223-1_11

[7] Linus Backlund, Kalle Ngo, Joel Gärtner, and Elena Dubrova. 2023. Secret key recovery attack on masked and shuffled implementations of CRYSTALS-Kyber and saber. In *Proceedings of the International Conference on Applied Cryptography and Network Security*. Springer-Verlag, Berlin, 159–177. DOI: https://doi.org/10.1007/978-3-031-41181-6_9

[8] Ciprian Băetu, F. Betül Durak, Loïs Huguenin-Dumittan, Abdullah Talayhan, and Serge Vaudenay. 2019. Misuse attacks on post-quantum cryptosystems. In *Proceedings of the EUROCRYPT 2019*, Yuval Ishai and Vincent Rijmen (Eds.). LNCS, Vol. 11477, Springer, Cham, 747–776. DOI: https://doi.org/10.1007/978-3-030-17656-3_26

[9] Abhishek Banerjee, Chris Peikert, and Alon Rosen. 2012. Pseudorandom functions and lattices. In *Proceedings of the EUROCRYPT 2012*, David Pointcheval and Thomas Johansson (Eds.). LNCS, Vol. 7237, Springer, Berlin, 719–737. DOI: https://doi.org/10.1007/978-3-642-29011-4_42

[10] Andrea Basso and Sujoy Sinha Roy. 2021. Optimized polynomial multiplier architectures for post-quantum KEM saber. In *Proceedings of the 58th ACM/IEEE Design Automation Conference*. IEEE, 1285–1290. DOI: https://doi.org/10.1109/DAC18074.2021.9586219

[11] Michiel Van Beirendonck, Jan-Pieter D'anvers, Angshuman Karmakar, Josep Balasch, and Ingrid Verbauwhede. 2021. A side-channel-resistant implementation of SABER. *ACM Journal on Emerging Technologies in Computing Systems* 17, 2 (2021), 26 pages. DOI: https://doi.org/10.1145/3429983

[12] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. 2017. NTRU prime: Reducing attack surface at low cost. In *Proceedings of the SAC 2017*, Carlisle Adams and Jan Camenisch (Eds.). LNCS, Vol. 10719, Springer, Cham, 235–260. DOI: https://doi.org/10.1007/978-3-319-72565-9_12

[13] Shivam Bhasin, Jan-Pieter D'Anvers, Daniel Heinz, Thomas Pöppelmann, and Michiel Van Beirendonck. 2021. Attacking and defending masked polynomial comparison. *IACR TCHES* 2021, 3 (2021), 334–359. DOI: https://doi.org/10.46586/tches.v2021.i3.334-359

[14] Nina Bindel, Mike Hamburg, Kathrin Hövelmanns, Andreas Hülsing, and Edoardo Persichetti. 2019. Tighter proofs of CCA security in the quantum random oracle model. In *Proceedings of the TCC 2019*, Dennis Hofheinz and Alon Rosen (Eds.). LNCS, Vol. 11892, Springer, Cham, 61–90. DOI: https://doi.org/10.1007/978-3-030-36033-7_3

[15] Nina Bindel and John M. Schanck. 2020. Decryption failure is more likely after success. In *Proceedings of the 11th International Conference on Post-quantum Cryptography*, Jintai Ding and Jean-Pierre Tillich (Eds.). Springer, Cham, 206–225. DOI: https://doi.org/10.1007/978-3-030-44223-1_12

[16] Andrej Bogdanov, Siyao Guo, Daniel Masny, Silas Richelson, and Alon Rosen. 2016. On the hardness of learning with rounding over small modulus. In *Proceedings of the TCC 2016-A*, Eyal Kushilevitz and Tal Malkin (Eds.). LNCS, Vol. 9562, Springer, Berlin, Heidelberg, 209–224. DOI: https://doi.org/10.1007/978-3-662-49096-9_9

[17] Madalina Bolboceanu, Zvika Brakerski, Renen Perlman, and Devika Sharma. 2019. Order-LWE and the hardness of ring-LWE with entropic secrets. In *Proceedings of the ASIACRYPT 2019*, Steven D. Galbraith and Shiho Moriai (Eds.). LNCS, Vol. 11922, Springer, Cham, 91–120. DOI: https://doi.org/10.1007/978-3-030-34621-8_4

[18] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle. 2018. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *Proceedings of the 2018 IEEE European Symposium on Security and Privacy*. 353–367.

[19] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 2018. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *Proceedings of the 2018 IEEE European Symposium on Security and Privacy*. IEEE Computer Society Press, 353–367. DOI: https://doi.org/10.1109/EuroSP.2018.00032

[20] Zvika Brakerski, Craig Gentry, and Vinod Vaikuntanathan. 2012. (Leveled) fully homomorphic encryption without bootstrapping. In *Proceedings of the ITCS 2012*, Shafi Goldwasser (Ed.). ACM, 309–325. DOI: https://doi.org/10.1145/2090236.2090262

[21] Eric Brier, Christophe Clavier, and Francis Olivier. 2004. Correlation power analysis with a leakage model. In *Proceedings of the CHES 2004*, Marc Joye and Jean-Jacques Quisquater (Eds.). LNCS, Vol. 3156, Springer, Berlin 16–29. DOI: https://doi.org/10.1007/978-3-540-28632-5_2

[22] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. 2003. Template attacks. In *Proceedings of the CHES 2002*, Burton S. Kaliski, Jr., Çetin Kaya Koç, and Christof Paar (Eds.). LNCS, Vol. 2523, Springer, Berlin, 13–28. DOI: https://doi.org/10.1007/3-540-36400-5_3

[23] Yuanmi Chen and Phong Q. Nguyen. 2011. BKZ 2.0: Better lattice security estimates. In *Proceedings of the ASIACRYPT 2011*, Dong Hoon Lee and Xiaoyun Wang (Eds.). LNCS, Vol. 7073, Springer, Berlin, 1–20. DOI: https://doi.org/10.1007/978-3-642-25385-0_1

[24] Jung Hee Cheon, Hyeongmin Choe, Dongyeon Hong, and MinJune Yi. 2024. SMAUG: Pushing lattice-based key encapsulation mechanisms to the limits. In *Proceedings of the SAC 2023*, Claude Carlet, Kalikinkar Mandal, and Vincent Rijmen (Eds.). LNCS, Vol. 14201, Springer, Cham, 127–146. DOI: https://doi.org/10.1007/978-3-031-53368-6_7

[25] Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song. 2018. Lizard: Cut off the tail! a practical post-quantum public-key encryption from LWE and LWR. In *Proceedings of the SCN 18*, Dario Catalano and Roberto De Prisco (Eds.). LNCS, Vol. 11035, Springer, Cham, 160–177. DOI: https://doi.org/10.1007/978-3-319-98113-0_9

[26] Jung Hee Cheon, Sangjoon Park, Joohee Lee, Duhyeong Kim, Yongsoo Song, Seungwan Hong, Dongwoo Kim, Jinsu Kim, Seong-Min Hong, Aaram Yun, et al. 2017. *Lizard*. Technical Report. National Institute of Standards and Technology. Retrieved July 1, 2025 from https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-1-submissions

[27] Chi-Ming Marvin Chung, Vincent Hwang, Matthias J. Kannwischer, Gregor Seiler, Cheng-Jhih Shih, and Bo-Yin Yang. 2021. NTT multiplication for NTT-unfriendly rings. *IACR TCHES* 2021, 2 (2021), 159–188. DOI: https://doi.org/10.46586/tches.v2021.i2.159-188

[28] S. Cook. 1966. *On The Minimum Computation Time of Functions*. Ph. D. Dissertation. Harvard University.

[29] Dana Dachman-Soled, Léo Ducas, Huijing Gong, and Mélissa Rossi. 2020. LWE with side information: Attacks and concrete security estimation. In *Proceedings of the CRYPTO 2020*, Daniele Micciancio and Thomas Ristenpart (Eds.). LNCS, Vol. 12171, Springer, Cham, 329–358. DOI: https://doi.org/10.1007/978-3-030-56880-1_12

[30] Jan-Pieter D'Anvers and Senne Batsleer. 2022. Multitarget decryption failure attacks and their application to saber and kyber. In *Proceedings of the PKC 2022*, Goichiro Hanaoka, Junji Shikata, and Yohei Watanabe (Eds.). LNCS, Vol. 13177, Springer, Cham, 3–33. DOI: https://doi.org/10.1007/978-3-030-97121-2_1

[31] Jan-Pieter D'Anvers, Qian Guo, Thomas Johansson, Alexander Nilsson, Frederik Vercauteren, and Ingrid Verbauwhede. 2019. Decryption failure attacks on IND-CCA secure lattice-based schemes. In *Proceedings of the PKC 2019*, Dongdai Lin and Kazue Sako (Eds.). LNCS, Vol. 11443, Springer, Cham, 565–598. DOI: https://doi.org/10.1007/978-3-030-17259-6_19

[32] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. 2018. Saber: Module-LWR based key exchange, CPA-secure encryption and CCA-secure KEM. In *Proceedings of the AFRICACRYPT 18*, Antoine Joux, Abderrahmane Nitaj, and Tajjeeddine Rachidi (Eds.). LNCS, Vol. 10831, Springer, Cham, 282–305. DOI: https://doi.org/10.1007/978-3-319-89339-6_16

[33] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, Frederik Vercauteren, Jose Maria Bermudo Mera, Michiel Van Beirendonck, and Andrea Basso. 2020. *SABER*. Technical Report. National Institute of Standards and Technology. Retrieved July 1, 2025 from https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions

[34] Jan-Pieter D'Anvers, Mélissa Rossi, and Fernando Virdia. 2020. (One) failure is not an option: Bootstrapping the search for failures in lattice-based encryption schemes. In *Proceedings of the EUROCRYPT 2020*, Anne Canteaut and Yuval Ishai (Eds.). LNCS, Vol. 12107, Springer, Cham, 3–33. DOI: https://doi.org/10.1007/978-3-030-45727-3_1

[35] Jan-Pieter D'Anvers, Angshuman Karmakar Sujoy Sinha Roy, and Frederik Vercauteren. 2018. Saber: Module-LWR Based Key Exchange, CPA-Secure Encryption and CCA-Secure KEM. Cryptology ePrint Archive, Report 2018/230. Retrieved from https://eprint.iacr.org/2018/230

[36] Jan-Pieter D'Anvers, Marcel Tiepelt, Frederik Vercauteren, and Ingrid Verbauwhede. 2019. Timing attacks on error correcting codes in post-quantum schemes. In *Proceedings of the Workshop on Theory of Implementation Security Workshop*. Association for Computing Machinery, New York, NY, USA, 2–9. DOI: https://doi.org/10.1145/3338467.3358948

[37] Jan-Pieter D'Anvers, Frederik Vercauteren, and Ingrid Verbauwhede. 2019. The impact of error dependencies on ring/Mod-LWE/LWR based schemes. In *Proceedings of the 10th International Conference on Post-Quantum Cryptography*, Jintai Ding and Rainer Steinwandt (Eds.). Springer, Cham, 103–115. DOI: https://doi.org/10.1007/978-3-030-25510-7_6

[38] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. 2018. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR TCHES* 2018, 1 (2018), 238–268. DOI: https://doi.org/10.13154/tches.v2018.i1.238-268

[39] Julien Duman, Kathrin Hövelmanns, Eike Kiltz, Vadim Lyubashevsky, and Gregor Seiler. 2021. Faster lattice-based KEMs via a generic fujisaki-okamoto transform using prefix hashing. In *Proceedings of the ACM CCS 2021*, Giovanni Vigna and Elaine Shi (Eds.). ACM, 2722–2737. DOI: https://doi.org/10.1145/3460120.3484819

[40] Tim Fritzmann, Michiel Van Beirendonck, Debapriya Basu Roy, Patrick Karl, Thomas Schamberger, Ingrid Verbauwhede, and Georg Sigl. 2022. Masked accelerators and instruction set extensions for post-quantum cryptography. *IACR TCHES* 2022, 1 (2022), 414–460. DOI: https://doi.org/10.46586/tches.v2022.i1.414-460

[41] Eiichiro Fujisaki and Tatsuaki Okamoto. 1999. Secure integration of asymmetric and symmetric encryption schemes. In *Proceedings of the CRYPTO'99*, Michael J. Wiener (Ed.). LNCS, Vol. 1666, Springer, Berlin, 537–554. DOI: https://doi.org/10.1007/3-540-48405-1_34

[42] Eiichiro Fujisaki and Tatsuaki Okamoto. 2013. Secure integration of asymmetric and symmetric encryption schemes. *Journal of Cryptology* 26, 1 (2013), 80–101. DOI: https://doi.org/10.1007/s00145-011-9114-1

[43] Nicolas Gama and Phong Q. Nguyen. 2007. New chosen-ciphertext attacks on NTRU. In *Proceedings of the PKC 2007*, Tatsuaki Okamoto and Xiaoyun Wang (Eds.). LNCS, Vol. 4450, Springer, Berlin, 89–106. DOI: https://doi.org/10.1007/978-3-540-71677-8_7

[44] Nicolas Gama, Phong Q. Nguyen, and Oded Regev. 2010. Lattice enumeration using extreme pruning. In *Proceedings of the EUROCRYPT 2010*, Henri Gilbert (Ed.). LNCS, Vol. 6110, Springer, Berlin, 257–278. DOI: https://doi.org/10.1007/978-3-642-13190-5_13

[45] Qian Guo, Thomas Johansson, and Alexander Nilsson. 2020. A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM. In *Proceedings of the CRYPTO 2020*, Daniele Micciancio and Thomas Ristenpart (Eds.). LNCS, Vol. 12171, Springer, Cham, 359–386. DOI: https://doi.org/10.1007/978-3-030-56880-1_13

[46] Qian Guo, Thomas Johansson, and Jing Yang. 2019. A novel CCA attack using decryption errors against LAC. In *Proceedings of the ASIACRYPT 2019*, Steven D. Galbraith and Shiho Moriai (Eds.). LNCS, Vol. 11921, Springer, Cham, 82–111. DOI: https://doi.org/10.1007/978-3-030-34578-5_4

[47] Jung Hee Cheon, Hyeongmin Choe, Jungjoo Seo, and Hyoeun Seong. 2024. SMAUG (-T), revisited: Timing-secure, more compact, less failure. *IEEE Access* 12 (2024), 188386–188397. DOI: https://doi.org/10.1109/ACCESS.2024.3511346

[48] Dennis Hofheinz, Kathrin Hövelmanns, and Eike Kiltz. 2017. A modular analysis of the Fujisaki-Okamoto transformation. In *Proceedings of the TCC 2017*, Yael Kalai and Leonid Reyzin (Eds.). LNCS, Vol. 10677, Springer, Cham, 341–371. DOI: https://doi.org/10.1007/978-3-319-70500-2_12

[49] Kathrin Hövelmanns, Andreas Hülsing, and Christian Majenz. 2022. Failing gracefully: Decryption failures and the Fujisaki-Okamoto transform. In *Proceedings of the ASIACRYPT 2022*, Shweta Agrawal and Dongdai Lin (Eds.). LNCS, Vol. 13794, Springer, Cham, 414–443. DOI: https://doi.org/10.1007/978-3-031-22972-5_15

[50] Kathrin Hövelmanns, Eike Kiltz, Sven Schäge, and Dominique Unruh. 2020. Generic authenticated key exchange in the quantum random oracle model. In *Proceedings of the PKC 2020*, Aggelos Kiayias, Markulf Kohlweiss, Petros Wallden, and Vassilis Zikas (Eds.). LNCS, Vol. *12111*, Springer, Cham, 389–422. DOI: https://doi.org/10.1007/978-3-030-45388-6_14

[51] Kathrin Hövelmanns and Christian Majenz. 2024. A note on failing gracefully: Completing the picture for explicitly rejecting Fujisaki-Okamoto transforms using worst-case correctness. In *Proceedings of the International Conference on Post-Quantum Cryptography*, Markku-Juhani Saarinen and Daniel Smith-Tone (Eds.). Springer, Cham, 245–265. DOI: https://doi.org/10.1007/978-3-031-62746-0_11

[52] Nick Howgrave-Graham, Phong Q. Nguyen, David Pointcheval, John Proos, Joseph H. Silverman, Ari Singer, and William Whyte. 2003. The impact of decryption failures on the security of NTRU encryption. In *Proceedings of the CRYPTO 2003*, Dan Boneh (Ed.). LNCS, Vol. 2729, Springer, Berlin, 226–246. DOI: https://doi.org/10.1007/978-3-540-45146-4_14

[53] Andreas Hülsing, Matthias Meijers, and Pierre-Yves Strub. 2022. Formal verification of saber's public-key encryption scheme in EasyCrypt. In *Proceedings of the CRYPTO 2022*, Yevgeniy Dodis and Thomas Shrimpton (Eds.). LNCS, Vol. 13507, Springer, Cham, 622–653. DOI: https://doi.org/10.1007/978-3-031-15802-5_22

[54] Institute of Commercial Cryptography Standards (ICCS). 2025. Evaluation Criteria for Public-Key Cryptographic Algorithms. Retrieved July 1, 2025 from https://www.niccs.org.cn/niccs/Notice/1975896137741635584/tT7TSQiz.pdf

[55] Éliane Jaulmes and Antoine Joux. 2000. A chosen-ciphertext attack against NTRU. In *Proceedings of the CRYPTO 2000*, Mihir Bellare (Ed.). LNCS, Vol. 1880, Springer, Berlin, 20–35. DOI: https://doi.org/10.1007/3-540-44598-6_2

[56] Haodong Jiang, Zhenfeng Zhang, Long Chen, Hong Wang, and Zhi Ma. 2018. IND-CCA-secure key encapsulation mechanism in the quantum random oracle model, revisited. In *Proceedings of the CRYPTO 2018*, Hovav Shacham and Alexandra Boldyreva (Eds.). LNCS, Vol. 10993, Springer, Cham, 96–125. DOI: https://doi.org/10.1007/978-3-319-96878-0_4

[57] Matthias J. Kannwischer, Markus Krausz, Richard Petri, and Shang-Yi Yang. 2024. pqm4: Benchmarking NIST Additional Post-Quantum Signature Schemes on Microcontrollers. Cryptology ePrint Archive, Report 2024/112. Retrieved July 1, 2025 from https://eprint.iacr.org/2024/112

[58] Matthias J. Kannwischer, Peter Pessl, and Robert Primas. 2020. Single-trace attacks on Keccak. *IACR TCHES* 2020, 3 (2020), 243–268. DOI: https://doi.org/10.13154/tches.v2020.i3.243-268

[59] Matthias J. Kannwischer, Richard Petri, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. [n. d.]. PQM4: Post-quantum crypto library for the ARM Cortex-M4. Retrieved July 1, 2025 from https://github.com/mupq/pqm4

[60] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. 2019. pqm4: Testing and Benchmarking NIST PQC on ARM Cortex-M4. Cryptology ePrint Archive, Report 2019/844. Retrieved July 1, 2025 from https://eprint.iacr.org/2019/844

[61] Angshuman Karmakar, Jose Maria Bermudo Mera, Sujoy Sinha Roy, and Ingrid Verbauwhede. 2018. Saber on ARM CCA-secure module lattice-based key encapsulation on ARM. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2018, 3 (2018), 243–266. DOI: https://doi.org/10.13154/TCHES.V2018.I3.243-266

[62] N. Koblitz. 1987. Elliptic curve cryptosystems. *Mathematics of Computation* 48 (1987), 203–209.

[63] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential power analysis. In *Proceedings of the CRYPTO'99*, Michael J. Wiener (Ed.). LNCS, Vol. 1666, Springer, Berlin, 388–397. DOI: https://doi.org/10.1007/3-540-48405-1_25

[64] Veronika Kuchta, Amin Sakzad, Damien Stehlé, Ron Steinfeld, and Shifeng Sun. 2020. Measure-rewind-measure: Tighter quantum random oracle model proofs for one-way to hiding and CCA security. In *Proceedings of the EUROCRYPT 2020*, Anne Canteaut and Yuval Ishai (Eds.). LNCS, Vol. 12107, Springer, Cham, 703–728. DOI: https://doi.org/10.1007/978-3-030-45727-3_24

[65] Suparna Kundu, Jan-Pieter D'Anvers, Michiel van Beirendonck, Angshuman Karmakar, and Ingrid Verbauwhede. 2022. Higher-order masked saber. In *Proceedings of the SCN 22*, Clemente Galdi and Stanislaw Jarecki (Eds.). LNCS, Vol. 13409, Springer, Cham, 93–116. DOI: https://doi.org/10.1007/978-3-031-14791-3_5

[66] Suparna Kundu, Angshuman Karmakar, and Ingrid Verbauwhede. 2023. On the masking-friendly designs for post-quantum cryptography. In *Proceedings of the 13th International Conference on Security, Privacy, and Applied Cryptography Engineering*, Francesco Regazzoni, Bodhisatwa Mazumdar, and Sri Parameswaran (Eds.). Lecture Notes in Computer Science, Vol. 14412, Springer, 162–184. DOI: https://doi.org/10.1007/978-3-031-51583-5_10

[67] Suparna Kundu, Quinten Norga, Angshuman Karmakar, Shreya Gangopadhyay, Jose Maria Bermudo Mera, and Ingrid Verbauwhede. 2025. Scabbard: An exploratory study on hardware aware design choices of learning with rounding-based key encapsulation mechanisms. *ACM Transactions on Embedded Computing Systems* 24, 1 (2025), 10:1–10:40. DOI: https://doi.org/10.1145/3696208

[68] Adeline Langlois and Damien Stehlé. 2015. Worst-case to average-case reductions for module lattices. *DCC* 75, 3 (2015), 565–599. DOI: https://doi.org/10.1007/s10623-014-9938-4

[69] Arjen K. Lenstra and Eric R. Verheul. 2001. Selecting cryptographic key sizes. *Journal of Cryptology* 14, 4 (2001), 255–293. DOI: https://doi.org/10.1007/s00145-001-0009-4

[70] Yanbin Li, Jiajie Zhu, Yuxin Huang, Zhe Liu, and Ming Tang. 2022. Single-trace side-channel attacks on the toom-cook: The case study of Saber. *IACR TCHES* 2022, 4 (2022), 285–310. DOI: https://doi.org/10.46586/tches.v2022.i4.285-310

[71] Jake Loftus, Alexander May, Nigel P. Smart, and Frederik Vercauteren. 2012. On CCA-secure somewhat homomorphic encryption. In *Proceedings of the SAC 2011*, Ali Miri and Serge Vaudenay (Eds.). LNCS, Vol. 7118, Springer, Berlin, 55–72. DOI: https://doi.org/10.1007/978-3-642-28496-0_4

[72] Xianhui Lu, Yamin Liu, Dingding Jia, Haiyang Xue, Jingnan He, Zhenfei Zhang, Zhe Liu, Hao Yang, Bao Li, and Kunpeng Wang. 2019. *LAC*. Technical Report. National Institute of Standards and Technology. Retrieved July 1, 2025 from https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-2-submissions

[73] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. 2010. On ideal lattices and learning with errors over rings. In *Proceedings of the EUROCRYPT 2010*, Henri Gilbert (Ed.). LNCS, Vol. 6110, Springer, Berlin, 1–23. DOI: https://doi.org/10.1007/978-3-642-13190-5_1

[74] Jose Maria Bermudo Mera, Angshuman Karmakar, Suparna Kundu, and Ingrid Verbauwhede. 2021. Scabbard: A suite of efficient learning with rounding key-encapsulation mechanisms. *IACR TCHES* 2021, 4 (2021), 474–509. DOI: https://doi.org/10.46586/tches.v2021.i4.474-509

[75] Jose Maria Bermudo Mera, Furkan Turan, Angshuman Karmakar, Sujoy Sinha Roy, and Ingrid Verbauwhede. 2020. Compact domain-specific co-processor for accelerating module lattice-based KEM. In *Proceedings of the 57th ACM/IEEE Design Automation Conference*. IEEE, 1–6. DOI: https://doi.org/10.1109/DAC18072.2020.9218727

[76] Thomas S. Messerges. 2000. Using second-order power analysis to attack DPA resistant software. In *Proceedings of the CHES 2000*, Çetin Kaya Koç and Christof Paar (Eds.). LNCS, Vol. 1965, Springer, Berlin, 238–251. DOI: https://doi.org/10.1007/3-540-44499-8_19

[77] Daniele Micciancio. 1998. The shortest vector in a lattice is hard to approximate to within some constant. In *Proceedings of the 39th FOCS*. IEEE Computer Society Press, 92–98. DOI: https://doi.org/10.1109/SFCS.1998.743432

[78] Vincent Migliore, Benoît Gérard, Mehdi Tibouchi, and Pierre-Alain Fouque. 2019. Masking dilithium - efficient implementation and side-channel evaluation. In *Proceedings of the ACNS 19th International Conference on Applied Cryptography and Network Security*, Robert H. Deng, Valérie Gauthier-Umaña, Martín Ochoa, and Moti Yung (Eds.). LNCS, Vol. 11464, Springer, Cham, 344–362. DOI: https://doi.org/10.1007/978-3-030-21568-2_17

[79] Victor S. Miller. 1986. Use of elliptic curves in cryptography. In *Proceedings of the CRYPTO'85*, Hugh C. Williams (Ed.). LNCS, Vol. 218, Springer, Berlin, 417–426. DOI: https://doi.org/10.1007/3-540-39799-X_31

[80] MLKEM 2024. Module-Lattice-Based Key-Encapsulation Mechanism Standard. NIST FIPS PUB 203, U.S. Department of Commerce. DOI: https://doi.org/10.6028/NIST.FIPS.203

[81] MLSA 2024. Module-Lattice-Based Digital Signature Standard. NIST FIPS PUB 204, U.S. Department of Commerce. DOI: https://doi.org/10.6028/NIST.FIPS.204

[82] Catinca Mujdei, Lennert Wouters, Angshuman Karmakar, Arthur Beckers, Jose Maria Bermudo Mera, and Ingrid Verbauwhede. 2024. Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication. *ACM Transactions on Embedded Computing Systems* 23, 2 (2024), 23 pages. DOI: https://doi.org/10.1145/3569420

[83] Catinca Mujdei, Lennert Wouters, Angshuman Karmakar, Arthur Beckers, Jose Maria Bermudo Mera, and Ingrid Verbauwhede. 2024. Side-channel analysis of lattice-based post-quantum cryptography: Exploiting polynomial multiplication. *ACM Transactions on Embedded Computing Systems* 23, 2 (2024), 27:1–27:23. DOI: https://doi.org/10.1145/3569420

[84] National Institute of Standards and Technology. 2016. *Submission Requirements and Evaluation Criteria for the Post-Quantum Cryptography Standardization Process.* Call for Proposals. National Institute of Standards and Technology (NIST). Retrieved July 1, 2025 from https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[85] National Institute of Standards and Technology. 2016. Submission Requirements and Evaluation Criteria for the Post-quantum Cryptography Standardization Process. Retrieved July 1, 2025 from https://csrc.nist.gov/CSRC/media/Projects/Post-Quantum-Cryptography/documents/call-for-proposals-final-dec-2016.pdf

[86] National Institute of Standards and Technology. 2022. *Call for Additional Digital Signature Schemes for the Post-Quantum Cryptography Standardization Process.* Call for Proposals. National Institute of Standards and Technology (NIST). Retrieved from https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf

[87] Parker Newton and Silas Richelson. 2023. A lower bound for proving hardness of learning with rounding with polynomial modulus. In *Proceedings of the CRYPTO 2023*, Helena Handschuh and Anna Lysyanskaya (Eds.). LNCS, Vol. 14085, Springer, Cham, 805–835. DOI: https://doi.org/10.1007/978-3-031-38554-4_26

[88] Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. 2021. A side-channel attack on a masked IND-CCA secure Saber KEM implementation. *IACR TCHES* 2021, 4 (2021), 676–707. DOI: https://doi.org/10.46586/tches.v2021.i4.676-707

[89] Kalle Ngo, Elena Dubrova, and Thomas Johansson. 2021. Breaking masked and shuffled CCA secure saber KEM by power analysis. In *Proceedings of the 5th Workshop on Attacks and Solutions in Hardware Security.* Association for Computing Machinery, New York, NY, USA, 51–61. DOI: https://doi.org/10.1145/3474376.3487277

[90] Kalle Ngo, Ruize Wang, Elena Dubrova, and Nils Paulsrud. 2022. Side-Channel Attacks on Lattice-Based KEMs Are Not Prevented by Higher-Order Masking. Cryptology ePrint Archive, Report 2022/919. Retrieved July 1, 2025 from https://eprint.iacr.org/2022/919

[91] Tobias Oder, Tobias Schneider, Thomas Pöppelmann, and Tim Güneysu. 2018. Practical CCA2-Secure Masked Ring-LWE Implementations. *IACR TCHES* 2018, 1 (2018), 142–174. DOI: https://doi.org/10.13154/tches.v2018.i1.142-174

[92] Hiroki Okada, Atsushi Takayasu, Kazuhide Fukushima, Shinsaku Kiyomoto, and Tsuyoshi Takagi. 2020. A compact digital signature scheme based on the module-LWR problem. In *Proceedings of the ICICS 20*, Weizhi Meng, Dieter Gollmann, Christian Damsgaard Jensen, and Jianying Zhou (Eds.). LNCS, Vol. 11999, Springer, Cham, 73–90. DOI: https://doi.org/10.1007/978-3-030-61078-4_5

[93] Seunghwan Park, Chi-Gon Jung, Aesun Park, Joongeun Choi, and Honggoo Kang. 2022. TiGER: Tiny bandwidth key encapsulation mechanism for easy miGration based on RLWE(R). Cryptology ePrint Archive, Report 2022/1651. Retrieved from https://eprint.iacr.org/2022/1651

[94] Chris Peikert. 2016. A Decade of Lattice Cryptography. *Foundations and Trends® in Theoretical Computer Science* 10, 4 (2016), 283–424. DOI: https://doi.org/10.1561/0400000074

[95] Chris Peikert and Zachary Pepin. 2024. Algebraically structured LWE, revisited. *Journal of Cryptology* 37, 3 (2024), 28. DOI: https://doi.org/10.1007/s00145-024-09508-3

[96] John M. Pollard. 1971. The fast Fourier transform in a finite field. *Mathematics of Computation* 25, 114 (1971), 365–374.

[97] J. Proos and C. Zalka. 2003. Shor's discrete logarithm quantum algorithm for elliptic curves. *Quantum Information and Computation* 3 (2003), 317–344. Retrieved from https://cds.cern.ch/record/602816

[98] Gokulnath Rajendran, Prasanna Ravi, Jan-Pieter D'Anvers, Shivam Bhasin, and Anupam Chattopadhyay. 2023. Pushing the limits of generic side-channel attacks on LWE-based KEMs - Parallel PC oracle attacks on Kyber KEM and beyond. *IACR TCHES* 2023, 2 (2023), 418–446. DOI: https://doi.org/10.46586/tches.v2023.i2.418-446

[99] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay. 2022. On exploiting message leakage in (Few) NIST PQC candidates for practical message recovery attacks. *IEEE Transactions on Information Forensics and Security* 17 (2022), 684–699. DOI: https://doi.org/10.1109/TIFS.2021.3139268

[100] Prasanna Ravi, Anupam Chattopadhyay, Jan Pieter D'Anvers, and Anubhab Baksi. 2024. Side-channel and fault-injection attacks over lattice-based post-quantum schemes (Kyber, Dilithium): Survey and new results. *ACM Transactions on Embedded Computing Systems* 23, 2 (2024), 54 pages. DOI: https://doi.org/10.1145/3603170

[101] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. 2020. Generic side-channel attacks on CCA-secure lattice-based PKE and KEMs. *IACR TCHES* 2020, 3 (2020), 307–335. DOI: https://doi.org/10.13154/tches.v2020.i3.307-335

[102] Oded Regev. 2005. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the 37th ACM STOC*, Harold N. Gabow and Ronald Fagin (Eds.). ACM. 84–93. DOI: https://doi.org/10.1145/1060590.1060603

[103] Oded Regev. 2010. The learning with errors problem (invited survey). In *Proceedings of the IEEE Conference on Computational Complexity*. 191–204. DOI: https://doi.org/10.1109/CCC.2010.26

[104] Oscar Reparaz, Sujoy Sinha Roy, Frederik Vercauteren, and Ingrid Verbauwhede. 2015. A masked ring-LWE implementation. In *Proceedings of the CHES 2015*, Tim Güneysu and Helena Handschuh (Eds.). LNCS, Vol. 9293, Springer, Berlin, 683–702. DOI: https://doi.org/10.1007/978-3-662-48324-4_34

[105] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. 1978. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the Association for Computing Machinery* 21, 2 (1978), 120–126. DOI: https://doi.org/10.1145/359340.359342

[106] Miruna Rosca, Amin Sakzad, Damien Stehlé, and Ron Steinfeld. 2017. Middle-product learning with errors. In *Proceedings of the CRYPTO 2017*, Jonathan Katz and Hovav Shacham (Eds.). LNCS, Vol. 10403, Springer, Cham, 283–297. DOI: https://doi.org/10.1007/978-3-319-63697-9_10

[107] Miruna Rosca, Damien Stehlé, and Alexandre Wallet. 2018. On the ring-LWE and polynomial-LWE problems. In *Proceedings of the EUROCRYPT 2018*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). LNCS, Vol. 10820, Springer, Cham, 146–173. DOI: https://doi.org/10.1007/978-3-319-78381-9_6

[108] Sujoy Sinha Roy and Andrea Basso. 2020. High-speed instruction-set coprocessor for lattice-based key encapsulation mechanism: Saber in hardware. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2020, 4 (2020), 443–466. DOI: https://doi.org/10.13154/TCHES.V2020.I4.443-466

[109] Tsunekazu Saito, Keita Xagawa, and Takashi Yamakawa. 2018. Tightly-secure key-encapsulation mechanism in the quantum random oracle model. In *Proceedings of the EUROCRYPT 2018*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). LNCS, Vol. 10822, Springer, Cham, 520–551. DOI: https://doi.org/10.1007/978-3-319-78372-7_17

[110] Claus-Peter Schnorr and M. Euchner. 1994. Lattice basis reduction: Improved practical algorithms and solving subset sum problems. *Mathematical Programming* 66 (1994), 181–199. DOI: https://doi.org/10.1007/BF01581144

[111] Peter W. Shor. 1994. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the 35th FOCS*. IEEE Computer Society Press, 124–134. DOI: https://doi.org/10.1109/SFCS.1994.365700

[112] Bo-Yeon Sim, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Tae-Ho Lee, Jaeseung Han, Hyojin Yoon, Jihoon Cho, and Dong-Guk Han. 2020. Single-trace attacks on message encoding in lattice-based KEMs. *IEEE Access* 8 (2020), 183175–183191. DOI: https://doi.org/10.1109/ACCESS.2020.3029521

[113] Damien Stehlé, Ron Steinfeld, Keisuke Tanaka, and Keita Xagawa. 2009. Efficient public key encryption based on ideal lattices. In *Proceedings of the ASIACRYPT 2009*, Mitsuru Matsui (Ed.). Springer, Berlin, 617–635. DOI: https://doi.org/10.1007/978-3-642-10366-7_36

[114] Yutaro Tanaka, Rei Ueno, Keita Xagawa, Akira Ito, Junko Takahashi, and Naofumi Homma. 2023. Multiple-valued plaintext-checking side-channel attacks on post-quantum KEMs. *IACR TCHES* 2023, 3 (2023), 473–503. DOI: https://doi.org/10.46586/tches.v2023.i3.473-503

[115] A. Toom. 1963. The complexity of a scheme of functional elements realizing the multiplication of integers. Soviet Mathematics-Doklady. 7 (1963), 714–716.

[116] Peter van Emde Boas. 1981. Another NP-complete problem and the complexity of computing short vectors in a lattice. *Tecnical Report, Department of Mathmatics, University of Amsterdam* (1981).

[117] Ruize Wang, Kalle Ngo, and Elena Dubrova. 2022. Making Biased DL Models Work: Message and Key Recovery Attacks on Saber Using Amplitude-Modulated EM Emanations. Cryptology ePrint Archive, Paper 2022/852. Retrieved from https://eprint.iacr.org/2022/852

[118] Ruize Wang, Kalle Ngo, and Elena Dubrova. 2022. Side-channel analysis of Saber KEM using amplitude-modulated EM emanations. In *Proceedings of the Conference on Digital System Design*. 488–495. DOI: https://doi.org/10.1109/DSD57027.2022.00071

[119] Han Wu and Guangwu Xu. 2022. More accurate geometric analysis on the impact of successful decryptions for IND-CCA secure Ring/Mod-LWE/LWR based schemes. In *Proceedings of the CT-RSA 2022*, Steven D. Galbraith (Ed.). LNCS, Vol. 13161, Springer, Cham, 196–222. DOI: https://doi.org/10.1007/978-3-030-95312-6_9

[120] Xiang Xie, Rui Xue, and Rui Zhang. 2012. Deterministic public key encryption and identity-based encryption from lattices in the auxiliary-input setting. In *Proceedings of the SCN 12*, Ivan Visconti and Roberto De Prisco (Eds.). LNCS, Vol. 7485, Springer, Berlin, 1–18. DOI: https://doi.org/10.1007/978-3-642-32928-9_1

[121] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, David Oswald, Wang Yao, and Zhiming Zheng. 2022. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of Kyber. *IEEE Transactions on Computers* 71, 9 (2022), 2163–2176. DOI: https://doi.org/10.1109/TC.2021.3122997

Received 19 September 2025; revised 28 February 2026; accepted 30 March 2026