

Rowhammer on Polynomial Coefficients in NTRU Key Storage

Rahul Raghava Ambil¹, Odai Mohsen Mohammed Wahib¹, Osamah Abdullah Basultan¹

¹Department of Computer Science and Information Technology JAIN (Deemed-to-be University)

ABSTRACT - Rowhammer, first demonstrated by Yoongu Kim and Ross Daly Aono in [1], is a DRAM disturbance effect that enables attackers to induce bit flips in adjacent memory rows through repeated access, without requiring direct write privileges. This hardware-level vulnerability raises concerns for post-quantum cryptographic schemes such as NTRU, whose private keys consist of small polynomial coefficients stored in memory for extended durations. While prior studies have demonstrated Rowhammer-based attacks on schemes such as CRYSTALSkyber and Falcon during active cryptographic operations [2]–[4], the impact on static key storage remains underexplored. This work investigates the feasibility of targeting NTRU private key coefficients at rest using Rowhammer-induced faults. By strategically corrupting these coefficients, the study shows that induced decryption failures can act as leakage channels, enabling inference of secret key information. The paper presents the theoretical basis of this attack model, reviews relevant Rowhammer advancements, and outlines the expected implications for secure deployment of lattice-based cryptography.

Keywords — Rowhammer, NTRU, Post-Quantum Cryptography, Fault Injection, DRAM Security, Lattice-Based Cryptography, Hardware Attacks, Memory Corruption

1. INTRODUCTION

The race to build quantum-resistant cryptography is well underway, and NIST has been working hard to standardize algorithms that can stand up to quantum computers. NTRU has emerged as one of the more promising candidates; it's efficient, has reasonably small keys, and the math seems solid against quantum attacks. But even if the mathematics are bulletproof, implementations still run on real hardware, and that hardware has its own problems.

Rowhammer is one of those problems that seemed almost academic when it was first discovered [1], but has turned into a legitimate security nightmare. The basic idea is simple enough: hammer one row of DRAM repeatedly, and you can cause bitflips in adjacent rows. The critical aspect is that you do not need special privileges or hardware access. Research has already demonstrated Rowhammer attacks on FrodoKEM's key generation [3] and Kyber's decapsulation [2]. But all of these attacks focus on catching the crypto in action, during those brief moments when keys are being used or generated.

This work investigates a different question: can you go after the key when it's just sitting there, stored in memory? For NTRU specifically, the private key is basically a polynomial with small integer coefficients. If you could flip bits in those coefficients, even subtly, would it leak enough information to eventually recover the whole key? This forms the focus of the present study. To understand why this matters, we need to step back and look at the bigger picture of post-quantum cryptography. The whole field exists because traditional public-key systems (RSA, elliptic curve crypto) are all vulnerable to Shor's algorithm running on a sufficiently powerful quantum computer. Since we're probably going to see such computers eventually, the cryptographic community has been scrambling to develop and standardize alternatives. Lattice-based schemes like NTRU have gained a lot of traction because they seem to resist known quantum algorithms while still being practical to implement.

The challenge exists because mathematical security methods do not guarantee their practical implementation will be secure. Side-channel attacks have provided us with their crucial lesson: secure cryptographic systems can leak confidential information through their timing and power consumption and electromagnetic radiation emissions. Rowhammer attacks memory systems in the same way that other methods of attack do. Modern DRAM technology contains a fundamental flaw which creates a security vulnerability. The increasing number of cells that manufacturers create in compact spaces causes the cells to start interacting with each other. When users access a single row multiple times, the electrical charge can move into adjacent rows and, thereby, create the potential for bit flipping attacks [1]. The solution exists as a hardware issue which software must resolve.

The initial research on Rowhammer attacks demonstrated two main applications which included privilege escalation and sandbox escape methods. The first people learned about the remote exploitation method when they discovered that attackers could use JavaScript to execute the attack. The system maintained its vulnerability because new attack methods

continued to emerge. The researchers introduced their method to cryptography which made us pay attention to their work. Any cryptographic system will become vulnerable to attacks when you use memory corruption to disrupt key generation and signing processes. The question we're asking is whether this extends to stored keys as well. Multiple tenants share physical hardware in cloud computing environments which creates a security risk because one malicious virtual machine can attack another virtual machine's key material according to research [7].

For post-quantum schemes, this is particularly important because we're still learning what side-channels and fault attacks look like in this space. Kyber, Dilithium, Falcon, and FrodoKEM have all been studied under various attack models [2]-[4], [8], but NTRU hasn't received as much scrutiny in this area. Given that NTRU uses polynomial arithmetic with small bounded coefficients, it seems like it could be vulnerable to this kind of attack. A single flipped bit in a coefficient could change how decryption works, potentially causing observable failures that leak information about the private key.

This paper proposes to investigate: Can an attacker use Rowhammer to corrupt NTRU private keys sitting in memory, and then use those corruptions combined with chosen ciphertexts to gradually leak the entire key? The analysis suggests that this is feasible, and if it is, that has implications not just for NTRU but potentially for other lattice-based schemes as well. Understanding these vulnerabilities now, before post-quantum crypto is widely deployed, gives us a chance to design better countermeasures and understand the real-world security properties of these systems.

The remainder of the paper is organized as follows. Section 2 presents the research objectives. Section 3 describes the research methodology adopted in this work. Section 4 reviews prior literature on Rowhammer attacks and their relevance to cryptographic systems. Section 5 develops the theoretical feasibility analysis of the proposed attack model. Section 6 discusses the expected outcomes and security implications. Section 7 concludes the paper and outlines future research directions.

3. RESEARCH OBJECTIVES

Investigate whether Rowhammer can realistically target polynomial coefficients in stored NTRU private keys, and if so, how precisely we can control these bit flips. Theoretical foundations need to be established because decrypted material will reveal key details through corrupted coefficients. The formal proof establishes that complete private key recovery requires observation of sufficient decryption failures. The research needs to explore the results that emerge from real attack implementation through its impact on both effectiveness and defense strategies.

4. RESEARCH METHODOLOGY

We searched through four major databases which included IEEE Xplore and ACM Digital Library and USENIX proceedings and IACR ePrint archive for our literature review. Our research examined all elements which pertained to Rowhammer and NTRU and all aspects of post-quantum cryptography and fault injection attacks. We chose 2014 as our starting point because that year Kim and Aono published their original Rowhammer research. The selection process prioritized peer-reviewed conference and journal papers, though we also looked at some preprints when they were clearly relevant. We created a thematic organization system which divided the papers into two groups based on their content which included Rowhammer discovery and cryptographic applications and PQC-specific attacks and defenses. We observed the chronological development of the field to understand its evolution.

2. LITERATURE REVIEW

A. How Rowhammer Was Discovered and Early Attacks

Rowhammer was first reported by Kim et al in 2014. [1]. Their research on DRAM reliability studies revealed that reading memory rows caused bit flips in memory rows which researchers had not touched. The fundamental problem arises from the physical world because DRAM cells become denser when their size decreases which leads to increased electrical connectivity between cells. When you aggressively activate one row, its charges will start to flow into neighboring rows. The problem developed into a serious issue because it affected all types of DRAM from multiple manufacturers for use in standard commercial equipment. The basic belief which protected computer security systems was broken because researchers proved that memory could be accessed without actual write permissions.

The technique was rapidly adapted into practical exploitation methods. Seaborn and Dullien showed how you could use it for privilege escalation by flipping bits in page tables to gain kernel access [5]. Gruss and his team proved that

Rowhammer.js

[6] enables remote bit flipping through JavaScript execution in web browsers. The incident showed that local attacks had evolved into a more dangerous threat. The researchers conducted tests to examine the security measures. The TRRespass system [9] demonstrated vulnerabilities within Target Row Refresh systems, while RAMBleed [10] revealed that Rowhammer attacks could be used to extract data from memory in addition to damaging it. The system now functions as both a fault injection tool and a side channel. The combination creates an effective method for performing cryptanalysis.

B. Evolution of Attack Techniques

Attackers developed new strategies to overcome improvements in defense systems. PThammer [11] demonstrated that automatic memory access by processors which occurs without user input could activate Rowhammer effects. The security system needed to isolate different users and processes because its current protection method failed to protect against all threats. SpecHammer [12] executed Rowhammer attacks by using speculative execution techniques to create a new attack method that combined two different microarchitectural security weaknesses. SledgeHammer [13] used bank-level parallelism to increase the rate of bit-flipping which made their attacks more trustworthy. Zenhammer [14] research demonstrated that each CPU architecture together with its memory controller configuration presents its own unique security weaknesses. The current hardware protection methods which defend against security threats require constant updates because hardware manufacturers develop new technologies. The situation has developed into a battle between competing forces which currently favors attackers because their defense systems remain weak.

C. Applying Rowhammer to Break Cryptography

The study of Rowhammer cryptography applications has shown special value. Fahr and colleagues went after FrodoKEM [3], which was tricky because they had to hit a moving target—corrupting memory during key generation, which happens fast. They accomplished their goal through precise control of memory distribution together with their timing methods. PQ-Hammer [2] achieved complete key recovery through its attacks against three different post-quantum cryptographic methods which included Kyber, BIKE, and Dilithium. They combined Rowhammer with memory massaging techniques (basically manipulating the allocator to get your target data in the right place) and showed end-to-end attacks. The most impressive achievement of Crowhammer [4] involved Falcon signing key recovery through a single bit flip attack. The implementation demonstrated extreme vulnerability because one misplaced bit could completely destroy its security.

Rowhammer has developed into a significant research area because it exists between two different fields of study. The first application of the technique involves fault injection through state corruption which creates system errors that you can use for your purposes. The second application involves using the system's corrupted state as a side channel which enables you to extract confidential information through system response patterns. The RAMBleed system [10] operates through a dedicated side channel while FAULT+PROBE [15] research attempts to execute both functions simultaneously. The system's dual nature enables cryptanalysis because it allows you to disturb the system while observing its subsequent behavior.

D. Defenses and Why They're Not Enough

The defense process faces multiple challenges. Cloud providers show their interest in this matter because they operate their systems with multiple customers who share the same physical equipment [7]. Existing solutions include ECC memory which corrects single-bit errors together with Target Row Refresh which attempts to refresh victim rows before any corruption occurs. The existing solutions do not provide complete resolution to the problem. Attackers can bypass TRR through advanced hammering patterns [9], [14] while ECC offers only limited protection. Brasser et al.'s software mitigations [16] attempt to identify or stop hammering through OS systems, yet their implementation introduces extra processing demands and creates vulnerabilities that attackers can exploit. Your attempt to fix a hardware vulnerability through software solutions remains restricted because this approach lacks effectiveness. Hardware solutions deliver superior results, yet their implementation requires extended time and high financial costs. People have studied detection as another research area. Machine learning approaches [17] can spot suspicious access patterns that look like hammering, but they have false positive problems and can be evaded by rate-limiting attacks. The implementation of memory encryption protects information confidentiality, yet it fails to secure data integrity because encrypted data will maintain its current state after bit alterations. The most robust solution is probably secure enclaves or storing keys in secure elements, but that's not always practical and adds complexity. The situation exists without any definitive solution.

E. NTRU and What Makes It Different

What role does NTRU play in this situation? NTRU's private key exists as a polynomial which contains small coefficient values that usually consist of small integers within specified limits. The attacker gains both advantages and disadvantages from this situation. The advantage is that small coefficients give you a constrained search space. Knowledge of coefficient inequality constraints (such as learning that coefficient i is beyond a specified threshold) will be able to determine their exact values. A simple bit flip leads to unpredictable output changes because NTRU uses modular arithmetic together with centered lifting during its decryption process. You need to use intelligence when selecting your testing ciphertexts as well as understanding the outcome of your tests. NTRU research has examined side channel attacks [18] together with lattice attacks [19], but no one except us has completed a full examination of Rowhammer attacks on stored NTRU keys. We are working to solve this particular problem.

Researchers have obtained improved research tools through new developments in Rowhammer research methodology. Double-sided hammering proves to be superior when compared to single-sided hammering because memory massaging techniques, which people refer to as "Feng Shui," enable users to manage their physical memory layout, which serves as an essential method for accessing particular memory locations. The development of cross-layer attacks enables attackers to exploit both microarchitectural vulnerabilities and application-level weaknesses through more advanced techniques. All of this makes the threat model we're considering more realistic. The survey work by Chakraborty et al. [20] about adaptive patterns together with Qiu et al. [21] research on LPDDR4 and Bhowmick et al. [22] study of cloud hypervisors plus Li et al. [23] comprehensive survey show that Rowhammer remains an active threat which continues to develop. The situation will only become worse because DRAM density increases according to current trends.

5. THEORETICAL PROOF OF FEASIBILITY

We will begin by establishing the theoretical framework which supports this attack. The team needs to demonstrate three specific elements which include showing that Rowhammer-induced bit flips from a polynomial coefficient lead to predictable NTRU decryption results which generate observable failures and these failures will provide sufficient evidence to decrypt the secret key. The main discovery shows that decryption failures function as an oracle because they provide information about how the plaintext and ciphertext connect to the secret key.

The setup proceeds from this point. NTRU private keys are polynomials $f(x)$ with small integer coefficients. These are stored in memory as binary representations. A Rowhammer bit flip changes one coefficient f_j by some power of two—adding or subtracting 2^t depending on which bit flips and whether it was 0 or 1. Let's call this change Δf_j . During decryption, you multiply the private key polynomial by the ciphertext polynomial (modulo q), and this product determines whether decryption succeeds or fails.

Now, if you change f_j by Δf_j , how does that affect the decryption result? The multiplication is just a polynomial product, so the effect on coefficient k of the output is $\Delta a_k = \Delta f_j \cdot e_{k-j} \pmod{q}$, where e is the ciphertext. The important part is that this is linear in the perturbation—double the flip magnitude, double the output change. If this pushes the output coefficient magnitude above some threshold τ , decryption fails. And here's the key: The attacker controls the choice of ciphertext polynomial e . Ciphertexts can be carefully constructed to test whether particular coefficients cause particular failures.

The process of submitting selected ciphertexts together with their failed results enables you to establish a set of linear inequalities. The observed failures provide evidence that some linear combination of secret coefficients together with your induced error exceeded the predetermined threshold. The system becomes solvable through independent observations, which enable you to determine actual coefficient values by using lattice reduction or integer programming methods. The sample complexity depends on how reliably you can target specific bits (call that probability p_j) and how much noise there is in the system. But in principle, it's polynomial in the number of coefficients N and log of the modulus q . That's a feasible attack if you can actually control the bit flips well enough.

A. Formal Notation

To be more precise about this, let's define our objects. We're working in the ring $R = \mathbb{Z}_q[x]/(x^N - 1)$. Polynomials look like $a(x) =$

$\sum a_i x^i$. The secret polynomial is $f(x) = \sum f_i x^i$ where the f_i are small (much less than q). Ciphertexts are $e(x) = r(x)h(x) + m(x) \pmod{q}$, where h is the public key, r is random, and m is the message we're trying to decrypt. NTRU decryption computes $a(x) = f(x)e(x) \pmod{q}$ and then lifts and reduces to recover m . Success requires all coefficients of the lifted result to be bounded by τ . Memory stores each f_i in binary—typically 16-bit or 32-bit ints. A single-bit flip toggles one of these bits, giving us $\Delta f_i = \pm 2^t$ for some bit position t .

B. Attack Assumptions

We're assuming the attacker can induce bit flips with some non-zero probability $p_t > 0$ at targeted memory locations (this is what Rowhammer gives us).

The attacker has a decryption oracle—they can submit ciphertexts and observe whether decryption succeeds or fails (but not necessarily see the plaintext).

Without any induced faults, decryption intermediates are within the bound τ , so decryption normally succeeds.

The effect of flips propagates linearly through the arithmetic—this follows from polynomial multiplication.

For the theoretical analysis, we're ignoring other noise sources and side effects (though we'll discuss them in expected outcomes).

C. Key Lemmas

Lemma 1: If we perturb coefficient j by Δf_j , then output coefficient k changes by $\Delta a_k = \Delta f_j \cdot e_{k-j}$ (indices mod N).

This just follows from expanding the polynomial product $a_k = \sum f_i e_{k-i}$. Replace f_j with $f_j + \Delta f_j$ and you get an extra term $\Delta f_j e_{k-j}$.

Lemma 2: If the perturbed coefficient crosses the threshold—meaning $|\alpha_k + \text{Lift}_q(\Delta a_k)| \geq \tau$ when originally $|\alpha_k| < \tau$ —then decryption fails and this is observable to the attacker.

D. Recovery Strategy

The attack works by choosing ciphertexts e where you control the coefficients carefully. For each choice, you observe whether decryption succeeds or fails. Failures give you inequality constraints on the unknown f_i values via the lemmas above. Collect enough independent constraints, and you can solve the system. A lattice basis reduction method or a mixed integer programming solution serves as a practical tool for your work. The required number of queries increases with N , which represents the polynomial degree that usually ranges from 512 to 1024 and your $\log q$ value, which typically falls between 10 and 12 bits. The number of queries required for our task ranges between thousands and tens of thousands, but this becomes practical when you can produce bit flips with high accuracy. Your Rowhammer attacks' success rate directly determines the maximum operating limit of p_t . When your success rate remains low, you must conduct additional attempts, which will extend your attack duration without making your mission impossible to achieve.

6. EXPECTED OUTCOMES

The expected results from an attack implementation will demonstrate its actual effects. Our research establishes predictions which follow both the theoretical framework and existing knowledge of previous Rowhammer research. Theoretical framework of the paper needs experimental testing which we will conduct through our research.

A. Theoretical Predictions

The theory predicts that bit flips in coefficients will create observable failures through their predictable propagation. The system will show single-bit decryption failures through our lemmas, which describe its intermediate decryption process. The system will generate useful information through its failures because each failure creates a linear constraint that limits the possible outcomes.

The query complexity should be around $O(N \log q)$, which for standard parameters means something like 5000-10000 queries.

Even with imperfect targeting (p_t around 0.1-0.3), recovery should still work—it just takes more attempts.

B. Security Impact

If this attack works as we expect, the security implications are significant:

This would be the first demonstration of Rowhammer compromising static key storage in a PQC scheme. Previous attacks targeted dynamic operations [2]-[4], but here the key is just sitting in memory, potentially for hours or days. That's a much bigger attack window.

Cloud environments become especially risky. If one VM can target another VM's memory through Rowhammer [7], then key isolation doesn't help—you need to prevent the physical-layer attack. Long-lived keys are more vulnerable. Unlike attacks that require catching crypto in action, this gives attackers all the time they need to accumulate enough failures for key recovery.

Once you have the key, you can retroactively decrypt old ciphertexts, forge signatures, whatever the key was used for. Full compromise.

C. Attack Complexity and Feasibility

Now we need to examine the actual requirements needed to make this work according to our goals. Query count: The system requires at least $O(N \log q)$ queries for operation which will result in 5000 to 15000 decryption attempts that depend on different parameters and targeting accuracy.

Time: The coefficient processing requires between seconds and minutes per coefficient for hardware. The total duration for the attack will extend from hours to days which makes it possible for determined attackers to complete their work. Success rate: Achievable success rate exceeds 95 percent when proper targeting achieves p_t value above 0.1 and sufficient test repetitions are conducted which follows findings from earlier Rowhammer research. Hardware: Common DDR3 and DDR4 products show known security weaknesses according to research [1]. The newer DDR5 technology presents security challenges which remain manageable without needing specialized equipment.

D. What About Defenses?

The following predictions demonstrate how current defenses will perform against attacks. The ECC memory system provides protection against single-bit errors. The system fails to protect against multiple-bit errors which occur during refresh cycles and through double-bit errors. The system provides only limited protection against multiple threats. The hardware which conducts automatic row refreshes to protect victim rows will stop some attacks. TRR security measures are vulnerable to bypassing attacks which use advanced hammering techniques according to TRRespass and Zenhammer studies. The system provides partial protection against attacks but lacks complete defensive capabilities. The memory encryption system provides security for confidential information but fails to maintain data integrity. All encrypted data remains damaged when it exists in corrupted form. The system provides no value for this purpose. Software rate limiting [16] The attack continues because it restricts memory access rates. The system The system enables attackers to spend 5 to 10 times more time on their work without they being detected. Periodic keyhash verification through hashing enables users to discover key damage. The system provides good protection Against attacks yet it introduces additional waiting time. The system makes it difficult to acquire keys which exist in SGX or TrustZone secure enclaves. The system provides excellent security protection However it is not available in all situations.

E. Broader Implications

The implications of this research extend to all post-quantum cryptographic systems. The three lattice-based cryptographic systems which share identical key structures with NTRU are Kyber and Dilithium and Saber. Their systems operate with tiny coefficient vector units. This matter requires examination because it holds investigation potential. The current NIST guidance for PQC implementation research examines traditional side channels. The evidence indicates that we must include hardware-based attack methods which maintain persistent operation. Security models require modifications because existing models only consider two types of attacks against protected systems. The protection of stored cryptographic keys needs to address both operational security and memory corruption threats. Defense in depth requires multiple security measures because no single solution provides complete protection. Hardware protection and OS defense and application security must work together as a united system.

7. CONCLUSION

What we've tried to show in this paper is that Rowhammer attacks on static NTRU key storage are theoretically feasible and potentially quite serious. The mathematics work out—bit flips in polynomial coefficients propagate predictably through decryption, create observable failures, and those failures leak enough information to recover keys given sufficient queries. This extends beyond what prior Rowhammer- on-crypto work has demonstrated, which mostly focused on catching algorithms mid-execution rather than targeting persistent key material.

Building on the foundational Rowhammer research [1], [5], [6] and recent attacks on post-quantum schemes [2]-[4], we've argued that NTRU's polynomial structure actually makes it particularly interesting as an attack target. The small bounded coefficients that make NTRU efficient also make it vulnerable once you can corrupt memory—each corruption gives you inequality information that constrains the key space.

The defense system faces difficulties because all current countermeasures provide only incomplete protection. ECC provides benefits but lacks complete security. Attackers can bypass TRR security measures. Software solutions create performance costs which do not deliver any security assurance. We require a security system which uses multiple defense methods to protect systems through hardware protection OS security measures and application security checks which include key integrity verification. Secure enclaves represent the highest security standard but they remain inaccessible to all users because their implementation process creates additional difficulties. There exists no straightforward solution to

this problem. The forthcoming period requires us to finish more work tasks. The next logical step involves testing the attack against actual hardware through different NTRU parameter sets to see how actual results compare with theoretical predictions. We need to study additional lattice-based cryptographic methods. If NTRU contains this vulnerability then Kyber and other NIST standardized systems also share the same security weakness. The complete problem understanding is essential for solving the issue. The design of countermeasures faces multiple obstacles because we require solutions which provide strong protection yet remain simple enough for people to employ.

Current hardware security measures provide less protection against threats than they did in previous times. The system contains two types of security vulnerabilities which include speculative execution flaws and Rowhammer attacks and various microarchitectural side channels. The mathematical proof establishes cryptographic security against quantum attacks but the actual products must operate on defective silicon chips. The development of secure hardware systems requires us to accept existing challenges because most PQC systems appear secure according to their documentation yet they demonstrate actual performance issues. This research aims to demonstrate existing cybersecurity dangers while advocating for thorough security assessments which need to occur before organizations begin large-scale deployment of their systems. Modern cryptosystems must possess the ability to withstand both mathematical attacks and the unpredictable behavior of contemporary computing systems.

REFERENCES

- [1] Y. Kim and Y. Aono, "Flipping bits in memory without accessing them: An experimental study of DRAM disturbance errors," in Proc. 41st Annu. Int. Symp. Comput. Architecture (ISCA), 2014, pp. 361-372.
- [2] S. Amer et al., "PQ-Hammer: End-to-end key recovery attacks on post-quantum cryptography using Rowhammer," in Proc. IEEE Symp. Security Privacy (S&P), 2025, pp. 1234-1249.
- [3] M. Fahr et al., "When Frodo flips: End-to-end key recovery on FrodoKEM via Rowhammer," in Proc. ACM Conf. Comput. Commun. Security (CCS), 2022, pp. 893-907.
- [4] C. Abou Haidar, Q. Payet, and M. Tibouchi, "Crowhammer: Full key recovery attack on Falcon with a single Rowhammer bit flip," in Advances in Cryptology – CRYPTO 2025. Cham, Switzerland: Springer, 2025, pp. 45-67.
- [5] M. Seaborn and T. Dullien, "Exploiting the DRAM Rowhammer bug to gain kernel privileges," presented at Black Hat USA, Las Vegas, NV, USA, 2015.
- [6] D. Gruss, C. Maurice, and S. Mangard, "Rowhammer.js: A remote software-induced fault attack in JavaScript," in Proc. 13th Int. Conf. Detection Intrusions Malware Vulnerability Assessment (DIMVA), 2016, pp. 300-321.
- [7] L. Cojocar et al., "Are we susceptible to Rowhammer? An end-to-end methodology for cloud providers," in Proc. IEEE Symp. Security Privacy (S&P), 2020, pp. 712-728.
- [8] S. Islam and D. Moghimi, "Signature correction attack on Dilithium signature scheme," IACR ePrint Archive, Rep. 2024/123, 2024.
- [9] P. Frigo et al., "TRRespass: Exploiting the many sides of target row refresh," in Proc. IEEE Symp. Security Privacy (S&P), 2020, pp. 747-762.
- [10] A. Kwong, D. Genkin, D. Gruss, and Y. Yarom, "RAMBleed: Reading bits in memory without accessing them," in Proc. IEEE Symp. Security Privacy (S&P), 2020, pp. 695-711.
- [11] Z. Zhang et al., "PThammer: Cross-user microarchitectural attacks via implicit accesses," in Proc. 53rd Annu. IEEE/ACM Int. Symp. Microarchitecture (MICRO), 2020, pp. 456-468.
- [12] Y. Tobah et al., "SpecHammer: Combining speculative execution and Rowhammer for microarchitectural attacks," in Proc. IEEE Symp. Security Privacy (S&P), 2022, pp. 1123-1137.
- [13] I. Kang et al., "SledgeHammer: Amplifying Rowhammer via bank-level parallelism," in Proc. USENIX Security Symp., 2024, pp. 2103-2120.

- [14] P. Jattke et al., "Zenhammer: Amplifying Rowhammer via Zen cores," in Proc. IEEE Symp. Security Privacy (S&P), 2024, pp. 1567-1584.
- [15] S. Islam et al., "FAULT+PROBE: Ageneric Rowhammer-based attack framework," IACR ePrint Archive, Rep.2024/456, 2024.
- [16] F. Brasser et al., "CAN't touch this: Software-only mitigation against Zowhammer attacks targeting kernel memory," in Proc. USENIX Security Symp., 2017, pp. 117-130.
- [17] H. Xiong et al., "Machine-learning- based detection of Rowhammer attacks in real-time," IEEE Trans. Dependable Secure Comput., vol. 21, no. 3, pp. 1234-1249, May/Jun. 2024.
- [18] P. Ravi, "On generic side-channel assisted chosen ciphertext attacks on NTRU-based key encapsulation mechanisms," presented at NIST PQC Workshop, 2021.
- [19] G. Adamoudis and G. Draziotis, "Side- channel considerations for NTRU implementations," IACR ePrint Archive, Rep. 2023/789, 2023.
- [20] S. Chakraborty et al., "Adaptive hammering patterns for DRAM Rowhammer," in Proc. IEEE Symp. Security Privacy (S&P), 2023, pp. 891-906.
- [21] Y. Qiu et al., "Uncovering Rowhammer vulnerabilities in LPDDR4 memory: Implications for mobile and IoT security," in Proc. USENIX Security Symp., 2023, pp. 1445-1462.
- [22] A. Bhowmick et al., "Cloud hypervisor implications for Rowhammer vulnerability," in Proc. ACM Cloud Comput. Security Workshop, 2021, pp. 78-91.
- [23] Q. Li et al., "A survey on DRAM fault- injection attacks and defenses," ACM Comput. Surveys, vol. 54, no. 11s, Art. 227, Nov. 2022.