5-2023

Reverse Engineering Post-Quantum Cryptography Schemes to Find Rowhammer Exploits

Sam Lefforge
University of Arkansas, Fayetteville

Reverse Engineering Post-Quantum Cryptography Schemes to Find Rowhammer Exploits

Reverse Engineering Post-Quantum Cryptography Schemes to Find Rowhammer Exploits

An Undergraduate Honors College Thesis

in the

Department of Computer Science and Computer Engineering
College of Engineering
University of Arkansas
Fayetteville, AR
May, 2023

by

Sam Lefforge

Abstract

Post-quantum cryptography is a necessary countermeasure to protect against attacks from quantum computer. However, the post-quantum cryptography schemes are potentially vulnerable to side channel attacks. One such method of attacking involves creating bit-flips in victim memory through a process called Rowhammer. These attacks can vary in nature, but can involve rowhammering bits to raise the encryption scheme's decryption failure rate, or modifying the scheme's random seed. With a high enough decryption failure rate, it becomes feasible to generate sufficient information about the secret key to perform a key recovery attack. This thesis proposed two attacks on proposed post-quantum cryptography algorithms, namely Kyber and BIKE. This process involves profiling the memory to determine which bits can be flipped, massaging a victim page into the correct spot in physical memory, and degrading the cores so that our attack timing coincides with the refreshing of the system's DRAM. The thesis demonstrates both of these attacks in simulation, and further work will execute these attacks on real hardware.

THESIS DUPLICATION RELEASE

I hereby authorize the University of Arkansas Libraries to duplicate this thesis when needed for research and/or scholarship.

Agreed_____

Sam Lefforge

Refused_____

Sam Lefforge

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deepest gratitude to Dr. Alexander Nelson for his invaluable guidance and support throughout my academic journey. Dr. Nelson's mentorship has been instrumental in shaping my research skills and inspiring my intellectual curiosity. I am grateful for the numerous hours Dr. Nelson has spent reviewing my work, providing feedback, and pushing me to think critically and deeply about my research questions and methodologies.

I would like to thank Dr. Miaoqing Huang and Dr. David Andrews agreeing to serve on my thesis committee. Their willingness to devote their time and expertise to review and evaluate my thesis is greatly appreciated.

I would also like to express my deepest gratitude to my family for their unwavering love, encouragement, and support throughout my academic journey.

Thank you again for all of your support, guidance, and encouragement throughout this process.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 Introduction

It is vitally important for companies and governments to protect their data. If data is compromised there can be huge consequences. There was a data breach in 2017 in which over 147 million customers' data was stolen from Equifax's databases [1]. Equifax settled with the Federal Trade Commission and others. Ultimately, they agreed to give up 425 million dollars in the settlement. The cautionary tale of Equifax should serve as a grim reminder of the cost of insecure data, as well as the value data holds.

The most common methods of encryption used today are vulnerable to quantum computers. One such example of this is RSA, a cryptosystem based on the factoring of large numbers. The keys generated by RSA are vulnerable to attacks by quantum computers because quantum computers can run Shor's Algorithm more effectively than a traditional computer which allows them to factor large numbers quickly, therefore cracking the encryption [2].

Another standard that is somewhat vulnerable to attacks by quantum computers is the Advanced Encryption Standard, or AES. AES is a widely used encryption standard that is used to protect sensitive data. Grover's Algorithm is a quantum algorithm that allows one to search a database in $O(\sqrt{n})$ time [3], which could be a powerful tool against AES. However, AES can be made sufficiently strong by merely increasing the key length. As it stands, we already have an implementation of AES with a key length such that the future of quantum computing does not pose a threat.

Post-quantum cryptography uses mathematical problems that are difficult for quantum computers to solve. Researchers have come up with several cryptography algorithms that are quantum safe. NIST hopes to choose one of these algorithms to be the new cryptographic standard. CRYSTALS-Kyber [4] is one of four public-key encryption and key-establishment algorithms that has made it

to round three of NIST's competition [5]. BIKE [6] and SIKE [7] are also post-quantum cryptographic algorithm feautred in the NIST competition. They were not selected after round 4.

While CRYSTALS-Kyber and BIKE may be less vulnerable to the type of attacks possible with quantum computing, it is more vulnerable to another type of attack. In fact, an attack capable of breaking CRYSTALS-Kyber and BIKE is possible with current technology. Broadly, the attack in question is a rowhammer attack.

1.1 Current Vulnerability

The current standards for public-key encryption are at risk of breaking under the stress of the techniques that are possible with quantum computing. The same is true for private-key encryption.

1.1.1 Public-Key Cryptography

Public-key cryptography, also known as asymmetric key cryptography, uses two different keys to encrypt and decrypt data. One key is called the public key, and it is freely available to anyone who wants to send an encrypted message. The other key is called the private key, and it is kept secret by the recipient of the encrypted message. Each user has their own public key which anyone can use to encrypt messages. Each user also has their own private key which is available only to them, and is used to decrypt the messages encrypted by that user's public key. [8]. This process is shown in Figure 1.1.

RSA (Rivest-Shamir-Adleman) is a popular and widely-used algorithm for public key cryptography. RSA works by generating two large prime numbers, which are then used to create a public key and a private key. RSA is built on the difficulty of computing the prime factors of large numbers, a task which would be theoretically quite easy for a quantum computer [9]. Shor's algorithm on a quantum computer can exploit this weakness in RSA's security.

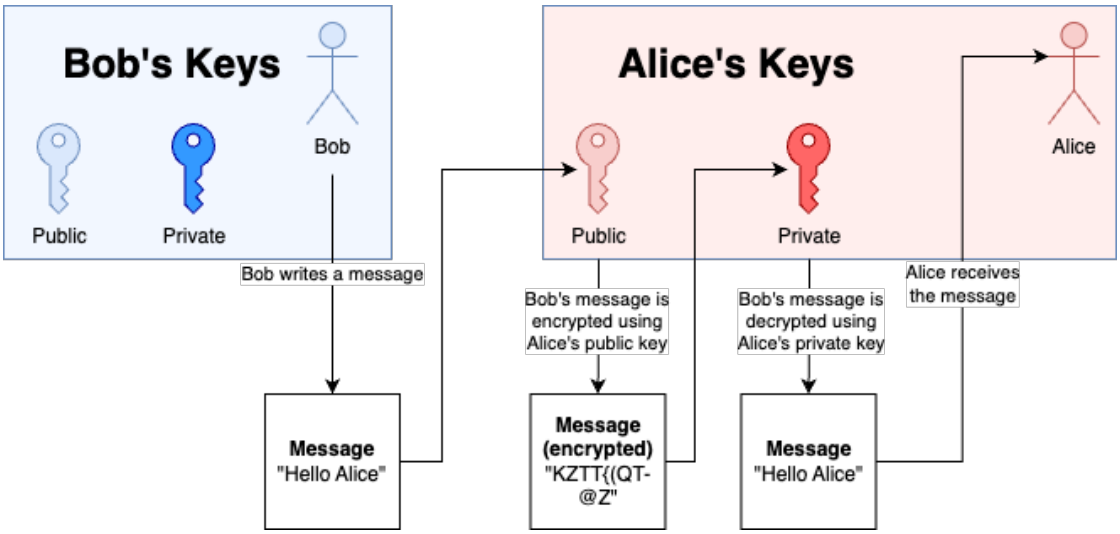

Figure 1.1: Bob sending Alice a message through public key exchange

Shor's algorithm works by exploiting the ability of quantum computers to perform certain types of calculations exponentially faster than classical computers. In particular, Shor's algorithm can factor large integers exponentially faster than the best-known classical algorithms.

The key idea behind Shor's algorithm is to use the QFT to find the period of the function $f(x) = a^x \ mod \ N$, where a is a randomly chosen integer and N is the integer to be factored. The period of this function is related to the factors of N, so by finding the period, we can find the factors of N [10].

1.1.2 Private-Key Cryptography

Private key cryptography, also known as symmetric key cryptography, uses a single key to encrypt and decrypt data. The same key is used by both the sender and the recipient of the message. [11]. DES (Data Encryption Standard) and AES (Advanced Encryption Standard) are both widely-used commonly used private-key encryption standards, with AES being the more secure of the two.

Grover's algorithm can be used to search for the secret key that was used to encrypt the data in AES. Specifically, if an attacker has access to the ciphertext

(encrypted data) and wants to recover the original plaintext, they would need to know the secret key used for encryption. The attacker could use Grover's algorithm to search for the key by iterating over all possible keys.

Grover's algorithm is a quantum algorithm that can be used to search an unsorted database in a faster way than classical algorithms. The algorithm works by iteratively applying a series of quantum operations to the database until the target item is found with high probability. It uses a technique called amplitude amplification to amplify the amplitude of the target item, making it more likely to be measured [12].

In the case of AES-128, which uses a 128-bit key, a classical brute-force attack would require trying 2^{128} possible keys, which is computationally infeasible. However, Grover's algorithm can search for the key with $O(\sqrt{(2^{128})}) = O(2^{64})$ iterations, which is still a very large number but is within the realm of possibility for some attackers.

Therefore, Grover's algorithm does not directly break AES encryption but reduces the security margin of the AES key by half. This is why AES-256 is generally recommended over AES-128, as a 256-bit key would be much more secure against quantum attacks.

1.2 Post-Quantum Cryptography

Post-quantum cryptography is a type of cryptography that is designed to be secure against attack by a quantum computer. Quantum computers are still in their early stages of development, but they have the potential to break many of the current cryptographic algorithms that are used to protect our data.

There are a number of different post-quantum cryptographic algorithms that have been proposed. CRYSTALS-Kyber is based on the hardness of solving certain mathematical problems related to module lattices [4]. Lattice-based cryptography is a type of cryptography that uses the mathematical structure of lattices to create secure cryptographic algorithms. Lattices are sets of points in a

high-dimensional space that satisfy certain conditions.

Another example is the hash-based signature scheme called SPHINCS+, which is based on the Merkle signature scheme and relies on the preimage resistance of cryptographic hash functions [13]. Hash-based cryptography is a type of cryptography that uses the mathematical structure of hash functions to create secure cryptographic algorithms. Hash functions are mathematical functions that take an input of any size and produce an output of a fixed size.

Rowhammer is a type of side-channel attack that can be used to read the contents of memory cells on a computer. Rowhammer attacks work by repeatedly accessing a memory cell in a way that causes the bits in the cell to flip. This can be used to read the contents of the cell, even if the cell is not supposed to be accessible.

Rowhammer attacks are a serious threat to the security of post-quantum cryptographic algorithms. This is because many post-quantum cryptographic algorithms rely on the assumption that the data that they are protecting cannot be read by an attacker. If an attacker can use rowhammer attacks to read the contents of memory cells, then they can break these cryptographic algorithms.

1.3 Preliminary PQC Embedded System Work

Post-quantum cryptography may have its vulnerabilities, but it is still vital to the future of cybersecurity. There will come a day when all cryptosystems will need to be overhauled. Some embedded systems like ATMs are impractical to update, which is why it is important that we start using future-proof cryptography in our embedded systems built today that will last for decades.

To simulate communication with an embedded system, myself and a partner used a MSP432, which is microcontroller with wireless communication capabilities. Figure 1.1 shows a public key exchange.

Despite the hardware limitations, the key exchange and encrypted communication between Bob and Alice worked perfectly. The two parties were success-

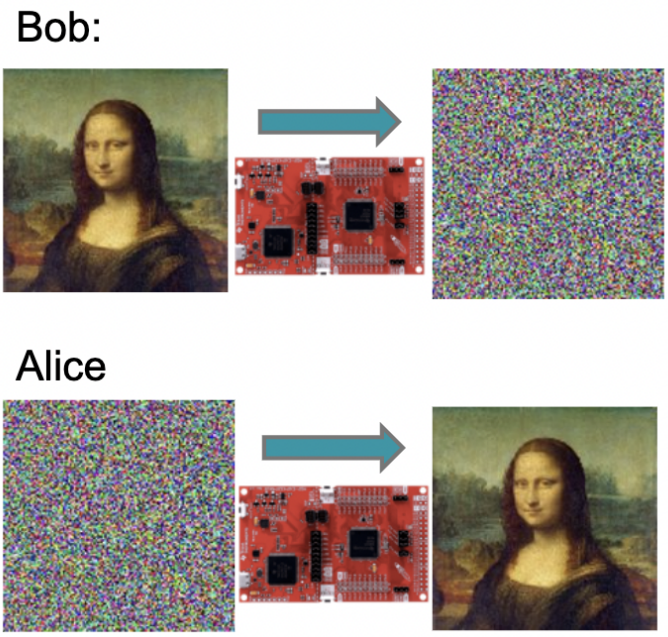

Figure 1.2: Quantum safe communication between Bob and Alice on a simulated embedded system

fully able to generate a quantum-secure shared secret. Then, the sender would encrypt their ciphertext using AES. The garbled images you see in Figure 1.2 are visual representations of the actual data after it was encrypted using AES. The two parties were then able to securely share encrypted data in a way that would be theoretically safe even against an available quantum computer.

A few years after doing this work, SIKE was broken by Castryk et al. [14]. The realization that the cryptographic algorithm I had previously studied and used in a project was no longer considered secure must have been a jarring experience. It sparked a curiosity about the underlying mechanisms of cryptographic security and a desire to explore the field further. This motivation likely led me to pursue research on attacking cryptographic schemes, with the goal of discovering vulnerabilities and weaknesses in existing systems.

1.4 Objective

The objective of this thesis is to explore the potential of rowhammer attacks to exploit post-quantum cryptography algorithms, specifically CRYSTALS-Kyber and BIKE. This thesis aims to identify the vulnerabilities of these algorithms to rowhammer attacks and investigate the effectiveness of such attacks against these cryptographic standards.

This thesis aims to highlight the importance of developing secure cryptographic algorithms that are resistant to a wide range of potential threats, including those posed by quantum computing and rowhammer attacks. The findings of this research contribute to a growing body of knowledge on the vulnerabilities of post-quantum cryptography and emphasize the need for continued research and development of secure cryptographic algorithms.

Additionally, this thesis simulates the usage of a post-quantum cryptographic algorithm, SIKE, on an embededd system to assess its practicality.

2 Background

At its core, this thesis is exploring how some post-quantum cryptographic algorithms leak information through side-channels, making them vulnerable to side-channel attacks. As such, an understanding of post-quantum cryptography and side-channel attacks is vital. Additinoally, there are several technical concepts that are important to understanding the attacks described in this paper.

2.1 Side-Channel Attacks

There are channels built into computer systems for the purpose of conveying information to the user. For instance, the screen of a phone is a channel designed with the intention of giving information to the user. Another such channel would be the vibration motor, which deliberately tells the user that they've received a notification. However, there are other channels that leak information unintentionally. For instance, someone monitoring the power consumption of a phone may be able to tell what application the user is currently running. Any third party taking advantage of these unintentional channels of information to gain some insight is performing a side-channel attack.

Backes et al. [15] performed a quite tangible attack on the acoustic side-channel of a dot-matrix printer. To perform their attack, they placed a microphone next to the printer and recorded the audio it leaked during the printing process. Their analysis of this audio was able to recover 72% of words that were printed, increasing to 95% when they made some assumptions about the context of the document being printed.

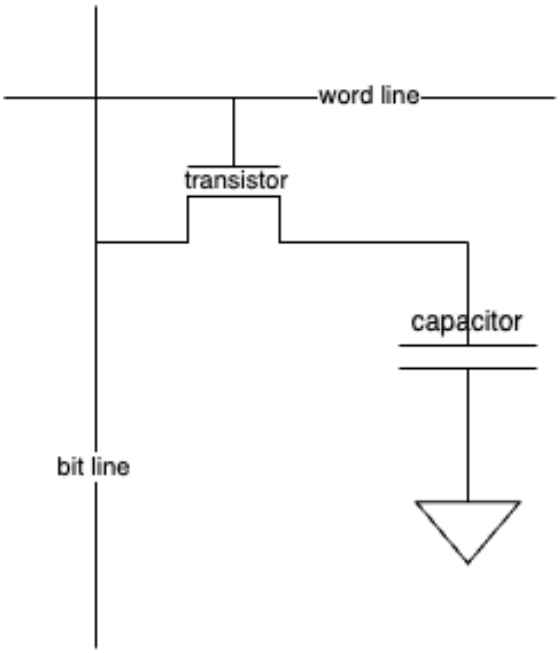

Figure 2.1: Schematic of a DRAM cell

2.2 DRAM

Understanding Dynamic Random Access Memory (DRAM) is essential to understanding rowhammer based attacks. There are two elements that make up DRAM, a transistor (similiarly to SRAM) and a capacitor (unlike SRAM). When a 1 is to be written into memory, the transistor charges the capacitor. Capacitors leak charge over time. To prevent losing data this way, DRAM "refreshes" itself every so often [16].

2.2.1 Timing Attacks and Rowhammer

Cryptographic algorithms' execution time is not necessarily independent from the input the input the algorithm receives. That is, variations in the given plaintext or ciphertext and an encryption key may cause variations in processing time. Through this leak in the timing side-channel, valuable data such as entire secret keys can be recovered [17].

It has been demonstrated that timing attacks can be used to break a number of cryptographic systems, including Diffie-Hellman, RSA, and DSS. Luckily, there are a number of techniques that can be used to mitigate the risk of timing attacks. One such technique is making sure all operations complete in the same length of time. Unfortunately, this is much easier said than done. Computers are complex machines, and unexpected timing variations and crop up for any number of reasons, including compiler optimizations and RAM cache hits.

Timing attacks are a serious threat to the security of cryptographic systems. It is important to be aware of the threat and to take steps to mitigate the risk.

Rowhammer attacks are a hardware-based security vulnerability, which may enable an attacker to flip bits in DRAM by repeatedly accessing rows of memory adjacent to a target-victim memory location. This occurs because physically-adjacent memory cells interact electrically, and when a capacitor accumulates enough charge, it can discharge into adjacent capacitors, causing their logical bits to flip, even if the attacker does not have read or write permissions to those capacitors [18].

2.3 Performance Degradation

Cache flushing can lead to performance degradation by causing cache misses, reducing cache hit rates, and inducing cache thrashing. These effects can be particularly pronounced in programs that access large amounts of data or have a high degree of spatial or temporal locality.

2.4 Frame Feng Shui

Frame Feng Shui is a technique developed by Kwong et al. [19] that is used to place a page in some specific physical location. The technique relies on the predictability of the Linux physical memory allocator. It does so without memory exhaustion, meaning the technique does not require one to allocate a significant portion of the availiable memory for this to work.

The attack involves allocating dummy pages, deallocating a selected frame, and immediately triggering the victim process to allocate its pages, with the secret-containing page landing in the desired frame. The victim process is triggered by initiating an SSH connection, served by the SSH daemon.

2.5 Related Work

A rowhammer based side-channel attack has been shown to be effective on another NIST post-quantum cryptographic algorithm. Fahr et al. [20] describes an attack to recover the secret key information from the FrodoKEM key establishment mechanism. FrodoKEM was submitted to NIST as a potential candidate algorithm. At the paper's publication, FrodoKEM was still in the running to be a new post-quantum cryptographic standard. However, as of July 2022, NIST has announced that FrodoKEM will no longer be in contention. Still, the attack described is highly relevant to similar attacks designed for candidates that have not been eliminated.

Similarly, Islam et al. [21] found an attack on another NIST post-quantum cryptography candidate called CRYSTALS-Dilithium. Like FrodoKEM, this attack was not selected to advance and is no longer in contention to be the new standard. This attack also makes use of the rowhammer technique. A simple yet effective countermeasure to dealing with timing attacks is simply to make the measured accuracy of any operation so low that gaining any valuable information is impossible. This can be done by adding in random delays into the encryption implementation [17].

Binding is a popular countermeasure used against power based side-channel attacks. It is done by binding the execution of a secret-dependent computation to a specific physical location on a device. This makes it difficult for an attacker to measure the timing, power consumption, or electromagnetic emissions of the computation, as they will be different depending on the physical location of the computation. [22]

One of the paramount advantages of discovering side-channel attacks through research is that it paves the way for further investigations aimed at thwarting such attacks, thereby enhancing the security of our systems. Messerges et al. [23] revealed some side-channel attack countermeasures in their 1999 paper. Their focus is primarily on the power consumption side-channel.

3 Implementation and Methodology

Rowhammer attacks are a multi-step process. For example, in the proposed CRYSTALS-Kyber attack the memory must be profiled, the private key data must be massaged into the target pages of DRAM, and failing ciphertexts must be generated before the private key data can be recovered.

Described in this section are the full rowhammer attacks on post-quantum cryptographic algorithms BIKE and CRYSTALS-Kyber, in addition to specific details on the implementation of some of the key components in completing these attacks.

3.1 The BIKE Attack

Attempt 1: In BIKE, we identified a mechanism to prevent the random matrix from being generated. This allows the potential to recover the secret key in a single attempt. The following block of code is where we make a modification. By changing the counter variable num_indicies to be 0, the outer loop never runs, which makes all the random indicies be equal to 0. This has the effect that the error matrix in the secret key is equivalent to a single 1 bit at the beginning of the matrix, with all 0s after.

```
ret_t sample_indices_fisher_yates(OUT idx_t *out,
                                  IN  size_t num_indices,
                                  IN  idx_t max_idx_val,
                                  IN OUT prf_state_t *prf_state) {

    for (size_t i = num_indices; i-- > 0;) {
        //This is where the randomness happens
        //We don't want this to happen
    }
    return SUCCESS;
}
```

In our testing, we discovered that this attack prevented decapsulating from ever completing because the number of possible solutions to the shortest vector problem with that error matrix approaches infinity. Therefore, Alice and Bob were highly unlikely to ever decapsulate the same shared secret.

Attempt 2: In our next attempt, we identified an instruction we might be able to modify, rather than making changes to variables. There is a MOV instruction that we identified as a potential modification point. The instruction loads data from memory and puts it into the victim register. The stack pointer is roughly `0x7FF0 0000 0000` and the value that is being added is roughly `-0x2AC0 0000 0000`. We want the sum of these to be close to 0. We can't modify the stack pointer because it is in a register, but we can modify that value that is being added to it in the instruction memory, but currently it does not quite look feasible because we would have to flip quite a few bits.

Attempt 3: Our next attempt involved patching an LEA instruction. An important property of hashing algorithms, which is what they are using to generate a long random value from a small random seed, is that they produce the same value from the input every time. Another property is that it shouldn't be obvious where the random value came from given the seed. So making the seed 0 will produce the same randomness from which the shared secret is generated. The consequence of these two is that hammering the randomness to put the seed into a different location would effectively leave the initial seed as 0, which would look like a valid random run. Our goal was to manipulate the seed into always having a value of 0, resulting in the same ciphertext being generated every time. Note that a more sophisticated attack would put a few 1's in the seed as well so that it would be virtually untraceable how we know the secret key. This attempt is currently ongoing, but for reasons that are currently unknown making such a change produces a segmentation fault.

3.2 The CRYSTALS-Kyber Attack

The following walks through the rowhammer attack on the CRYSTALS-Kyber PQC implementation in their official reference submmisison package. It has been validated in simulation.

3.2.1 Decryption Failure Attacks

CRYSTALS-Kyber occasionally fails to decrypt a valid ciphertext [24]. When this happens, information about the secret key is leaked. Ultimately, the goal of this attack is to flip some bits that so that the decryption failure rate increases and the full secret key can be recovered.

3.2.2 Determining Which Bits Need to be Hammered

An analysis was done on a simulation of many ciphertexts given additional noise added to certain locations. The goal here was to strike a balance between the number of bits necessary to generate the attack, and the failure rate. The failure rate needs to be low enough that an honest user won't notice, but high enough that we find it more often than random chance.

3.2.3 Memory Profiling

Not all DRAM is created equal. The process of creating DRAM is complex and precise, and with current manufacturing techniques the end result is DRAM with small variations in how each different cell holds a charge.

The implications of this fact for rowhammer attacks are major. Some bits will have a much lower propensity to be flipped when hammered, and some will flip easily. As such, determining which of this bits are "flippy" and which are not is vital before peforming a precise rowhammer attack such as this one. The standard technique for doing so is to attempt to hammer each bit ahead of time and record data on where the "flippy" bits are located.

3.2.4 Memory Massaging

With an idea of which virtual bits in our process need to be flipped, and which physical bits in our DRAM are susceptible to be flipped, we can find an ideal page in memory to place our victim process. We manipulate the system into placing our process in the correct page using the "Feng Shui" technique described in section 2.4.

3.2.5 Core Degradation

Rowhammering is a time-sensitive process. According to the JEDEC Standard for DDR3 ram [25], a typical refresh rate is around 64 milliseconds. In order for our rowhammer manipulation to be successful, CRYSTALS-Kyber must run in around that same length of time. Unaltered it runs a bit slower than FrodoKEM, but not slow enough. As such, it is beneficial to overload processor cores with a demanding process to slow them down and allow us more time to execute our attack.

Cache flushing can lead to performance degradation by causing cache misses, reducing cache hit rates, and inducing cache thrashing. We targeted two cache lines, one at memory location `0x6229` and one at `0x7030`. These are the memory locations of functions that get called frequently enough to cause degradation. For example, we have one function that performs a fairly simple, very quick mathematical operation. It does a multiplication operation followed by montgomery reduction. The function call is nested between three for loops, so it is called many times. By using cache flush on these locations in a separate victim process, we can force the process to fetch the code from memory every single time.

The attack was run on core 1. On our 8 core processor, we degraded cores 2-3 by flushing the virtual memory of `0x7030` from the cache by flushing the cache line that contains it. We degraded cores 4-8 by doing the same to the virtual memory of `0x6229`. This resulted in the maximum amount of time to run our attack. The median number of clock cycles per system tick for this style of degradation across

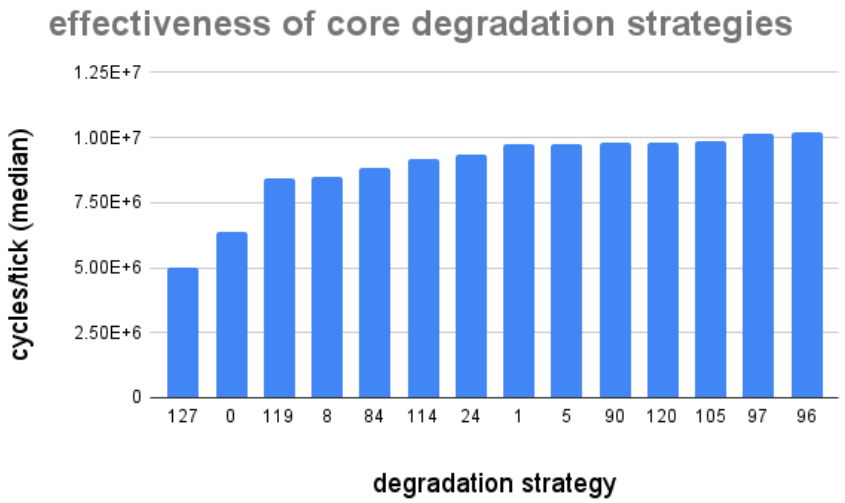

Figure 3.1: A handful of core degradation strategies displayed with their associated clock cycles per system tick. See Table 4.1 for the naming convention of these degradation strategies

1000 tests was 10192387. All 128 combinations of the two degradation processes on the 7 cores were tested, and none performed better than this.

3.2.6 Generating Failing Ciphertexts

Next, the selected bits are rowhammered and we effectively raise the rate of decryption failures to 2^{-10}. This is an improvement over the FrodoKEM attack which was only able to get the failure rate up to 2^{-27}. As such, CRYSTALS-Kyber requires orders of magnitude fewer failing ciphertexts to perform a successful attack. The FrodoKEM attack had to encrypt over 1 trillion messages to generate enough failing ciphertexts to continue with the attack.

Because this attack has a much higher decryption failure rate, the computational power needed to perform this attack is significantly less than that of the FrodoKEM attack. The FrodoKEM attack required a supercomputer to perform mass encryption at the scale to encrypt over 1 trillion messages. This attack, however, can be run on a standard laptop.

3.2.7 Secret Key Recovery

These failing ciphertexts reveal information about the secret key. With a large quantity of them, we are able to deduce the secret key entirely, effectively breaking the encryption. For a more full treatment of the details, check the approach used by Fahr et al. in the Frodo attack [20].

The process of taking these ciphertexts and extracting the secret key is complex. The cryptanalysis team working on cracking CRYSTALS-Kyber needed some example failed ciphertexts to experiment with. To do this, I modified part of the CRYSTALS-Kyber code that uses a random seed to generate a ciphertext. Instead, I incremented the seed predictably and saved off all the failing ciphertexts along with their seed.

```
//Put seed into buf from 0..KYBER_SYMBYTES
memset(buf, 0, KYBER_SYMBYTES - 8);
for (int i = KYBER_SYMBYTES - 8; i < KYBER_SYMBYTES; i++) {
  buf[i] = seed >> (KYBER_SYMBYTES-i)*8 & 0xff;
}
```

The above code shows a seed parameter being saved into a 32 byte buffer. Only the last 8 bytes are being used to represent the seeds. 8 bytes still allows representation of 2^{64} possible values. Adding noise through rowhammer, we increased the decryption failure rate to $\sim 2^{-10}$. This means that on average we will get $\sim 2^{54}$ failing ciphertexts to play with, which is sufficient to determine the secret key.

4 Results

This research project is currently in progress and making significant progress towards preventing rowhammer attacks. While there are still some pieces that need to come together, there have already been some promising preliminary findings which will contribute towards a successful prevention strategy.

4.1 Degrade Outcome

The following table shows some of the data from the work done to degrade the cores for the CRYSTALS-Kyber attack.

Table 4.1: Effectiveness of a Sample of Core Degredation Strategies

Core 2	Core 3	Core 4	Core 5	Core 6	Core 7	Core 8	Strategy	Cycles/Tick
0x6229	0x6229	0x6229	0x6229	0x6229	0x6229	0x6229	0	6347090
0x6229	0x6229	0x6229	0x6229	0x6229	0x6229	0x7030	1	9748380
0x6229	0x6229	0x6229	0x6229	0x7030	0x6229	0x7030	5	9775166
0x6229	0x6229	0x6229	0x7030	0x6229	0x6229	0x6229	8	8479307
0x6229	0x6229	0x7030	0x7030	0x6229	0x6229	0x6229	24	9344657
0x7030	0x6229	0x7030	0x6229	0x7030	0x6229	0x6229	84	8825505
0x7030	0x6229	0x7030	0x7030	0x6229	0x7030	0x6229	90	9805280
0x7030	0x7030	0x6229	0x6229	0x6229	0x6229	0x6229	96	10192387
0x7030	0x7030	0x6229	0x6229	0x6229	0x6229	0x7030	97	10134194
0x7030	0x7030	0x6229	0x7030	0x6229	0x6229	0x7030	105	9839609
0x7030	0x7030	0x7030	0x6229	0x6229	0x7030	0x6229	114	9166400
0x7030	0x7030	0x7030	0x6229	0x7030	0x7030	0x7030	119	8434309
0x7030	0x7030	0x7030	0x7030	0x6229	0x6229	0x6229	120	9810393
0x7030	0x7030	0x7030	0x7030	0x7030	0x7030	0x7030	127	4975898

The strategies shown in Table 4.1 are numbered according to the base-10 representation of their binary value where flushing the 0x6229 cache line is

represented by a 0 and 0x7030 is represented by a 1. Core 2 is considered the most significant bit and core 8 is the least significant bit. There were two cache lines to try flushing and seven cores, so in total there were $2^7 = 128$ total strategies. The table above displays a sample of these.

With no cores degraded, the median number of clock cycles per tick is a mere 7574. This means that with the optimal degrade strategy, we slowed down the process by around 134470%.

4.2 Generating and Analyzing Failing Ciphertexts

A key part of the CRYSTALS-Kyber attack is the ability to recover secret key data from failing ciphertexts. In order to do this, the cryptanalysis team needed a large number of failing ciphertexts to analyze. While the rowhammer technique is being explored as a way to increase the decryption failure rate, it is still a work in progress. As a result, we modified the CRYSTALS-Kyber code to save failing ciphertexts generated using the natural decryption failure rate of 2^{-160}.

Through this modification, we were able to save 600,000 failing ciphertexts, which provided a substantial dataset for the cryptanalysis team to analyze. The team was able to develop effective methods for recovering secret key data from the CRYSTALS-Kyber encryption scheme.

5 Conclusion

The protection of data is of utmost importance for both companies and governments, as the consequences of data breaches can be severe. The Equifax data breach serves as a reminder of the high cost of insecure data. The vulnerability of encryption standards such as RSA and AES to quantum computing further emphasizes the need for post-quantum cryptography algorithms that are resistant to attacks from quantum computers. The ongoing NIST competition to identify a new cryptographic standard is an important step towards achieving this goal.

However, it is important to note that even post-quantum cryptography algorithms such as CRYSTALS-Kyber and BIKE may be vulnerable to other types of attacks. Therefore, it is crucial to continue researching and developing new encryption methods that are resistant to a wide range of potential threats to ensure the security of our data.

The findings of this thesis showcase the potential of rowhammer attacks to target DRAM and exploit post-quantum cryptography algorithms. Specifically, this thesis provides a pathway to a successful rowhammer attack against BIKE and CRYSTALS-Kyber.

This thesis also showcases the viability of post-quantum cryptography implemented on embedded systems. It makes the case that embedded systems being produced today should be prepared for the cryptographic threats of the future.

This thesis contributes to a growing body of research that highlights the importance of developing secure cryptographic algorithms and protecting against potential vulnerabilities. Continued efforts in this area are essential to ensuring the security of our data and protecting against cyber attacks.

6 Future Work

This research is a work in progress. There are several fronts on which this work will continue in the future. Of course, the ultimate objective for both the BIKE attack and the CRYSTALS-Kyber attack is to successfully recover the secret key information. Further analysis and experimentation will be conducted to strengthen the security of cryptographic systems against these types of attacks and mitigate the risks posed by potential vulnerabilities

For the BIKE attack, one possibility we plan on exploring is altering instruction memory to entirely skip over the loop that generates the random seed. Due to automatic optimizations made by the compiler, this would still result in 8 random bytes being generated. Unfortunately, rowhammering that many bits seems unrealistic. This is a roadblock we have yet to find a solution to.

Of course, continuing with the approach described in Attempt 3 also seems like a promising avenue to breaking BIKE. We are currently getting a segmentation fault when trying to run a modified version of the source code with a specific bit flipped that should prevent randomness in the ciphertext generation. The segmentation fault is currently a mystery, but with more time the cause may reveal something that can help this attack succeed.

As for the CRYSTALS-Kyber attack, we are well on our way to launching a successful attack. Almost all the pieces are in place. Currently we are having trouble with Frame Feng Shui placing the page. Luckily we are in close contact with the team who created the technique, so some collaborative troubleshooting should hopefully solve our issues. Once that is resolved, we should be able to generate failing ciphertexts and extract the secret key information.

Those two attacks are the top priorities for the future, but we would eventually like to crack all of the NIST selected post-quantum cryptography algorithms. Islam et al. [21] has already done some work for CRYSTALS-Dilithium to show

that algorithm is potentially weak if the signing device is continually subjected to rowhammer, but Falcon [26] and Sphincs+ [13] remain as potential victims.

An introduction of a method of prevention for the kinds of attacks described in this thesis would serve a great benefit to the field of cryptography. Research to find attacks exits so that it may create research to find defenses. The security of cryptographic systems is constantly being challenged by attackers who seek to exploit vulnerabilities in these systems for malicious purposes. To ensure the security of these systems, it is crucial to understand the potential attack vectors that could be used by malicious actors. By studying different attack scenarios and developing attack models, researchers can identify potential weaknesses in cryptographic systems and develop effective countermeasures to prevent such attacks from being successful.

There are several steps that could be taken to improve the simulation of an embedded system running a post-quantum cryptographic technique, as seen in the preliminary study. For one, a more modern technique could be used. SIKE was not selected by NIST as a standard technique, and it has been proven to be insecure [14].

The preliminary study was designed to simulate an embedded system, like an ATM, communicating securely using post quantum cryptography. To simulate an embedded system we used a microcontroller, as they are similar to a lot of the computers used within embedded systems, and they possess many of the same hardware limitations. When running any kind of simulation it is important to try and keep all the variables consistent with the situation one is trying to simulate. This was our logic behind using a microcontroller in the first place. Unfortunately, there is one major variable that differed in this simulatinon: wireless communication. In theory, our method would work perfectly fine over wireless communication, but unfortunately we never got to put this theory to the test. For our simulation, we used a wired connection between the microcontroller and the PC. In the future, I'd like to see wireless communication fully simulated. Our plan for accomplishing this uses something called a Message Queuing Telemetry Transport (MQTT)

broker. MQTT is a publish-subscribe network protocol, meaning that to send a message, one publishes to a certain topic. All who are subscribed to the topic will then receive the message.

Bibliography

[1] F. T. Commission, "Equifax data breach settlement," 2019. [Online]. Available: https://www.ftc.gov/enforcement/refunds/equifax-data-breach-settlement

[2] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proceedings 35th annual symposium on foundations of computer science*. Ieee, 1994, pp. 124–134.

[3] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.

[4] R. Avanzi, J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber algorithm specifications and supporting documentation," *NIST PQC Round*, vol. 2, no. 4, pp. 1–43, 2019.

[5] National Institute of Standards and Technology (NIST), "Post-Quantum Cryptography Standardization: Round 3 Submissions," https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization/round-3-submissions, 2022, accessed: April 20, 2023.

[6] N. Aragon, P. S. Barreto, S. Bettaieb, L. Bidoux, O. Blazy, J.-C. Deneuville, P. Gaborit, S. Gueron, T. Guneysu, C. A. Melchor *et al.*, "Bike: bit flipping key encapsulation," *Submission to the NIST Post-Quantum Standardization project*, 2017.

[7] R. Azarderakhsh, M. Campagna, C. Costello, L. De Feo, B. Hess, A. Jalali, D. Jao, B. Koziel, B. LaMacchia, P. Longa *et al.*, "Supersingular isogeny key encapsulation," *Submission to the NIST Post-Quantum Standardization project*, vol. 152, pp. 154–155, 2017.

[8] S. Garfinkel, "Public key cryptography," *Computer*, vol. 29, no. 6, pp. 101–104, 1996.

[9] V. Mavroeidis, K. Vishi, M. D. Zych, and A. Jøsang, "The impact of quantum computing on present cryptography," *arXiv preprint arXiv:1804.00200*, 2018.

[10] M. A. Nielsen and I. L. Chuang, "Quantum computation and quantum information," *Phys. Today*, vol. 54, no. 2, p. 60, 2001.

[11] P. Techateerawat, "A review on quantum cryptography technology," *International Transaction Journal of Engineering, Management & Applied Sciences & Technologies*, vol. 1, pp. 35–41, 2010.

[12] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying grover's algorithm to aes: quantum resource estimates," in *Post-Quantum Cryptography: 7th International Workshop, PQCrypto 2016, Fukuoka, Japan, February 24-26, 2016, Proceedings 7*. Springer, 2016, pp. 29–43.

[13] D. J. Bernstein, A. Hülsing, S. Kölbl, R. Niederhagen, J. Rijneveld, and P. Schwabe, "The sphincs+ signature framework," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 2129–2146.

[14] W. Castryck and T. Decru, "An efficient key recovery attack on sidh," Cryptology ePrint Archive, Paper 2022/975, 2022, https://eprint.iacr.org/2022/975. [Online]. Available: https://eprint.iacr.org/2022/975

[15] M. Backes, M. Dürmuth, S. Gerling, M. Pinkal, C. Sporleder *et al.*, "Acoustic side-channel attacks on printers." in *USENIX Security symposium*, vol. 9, 2010, pp. 307–322.

[16] B. Jacob, D. Wang, and S. Ng, *Memory systems: cache, DRAM, disk*. Morgan Kaufmann, 2010.

[17] P. C. Kocher, "Timing attacks on implementations of die-hellman, rsa, dss, and other systems," in *Advances in Cryptology— Crypto*, vol. 96, 1996, p. 104113.

[18] M. J. Fahr, "The effects of side-channel attacks on post-quantum cryptography: Influencing frodokem key generation using the rowhammer exploit," 2022.

[19] A. Kwong, D. Genkin, D. Gruss, and Y. Yarom, "Rambleed: Reading bits in memory without accessing them," in *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020, pp. 695–711.

[20] M. Fahr Jr, H. Kippen, A. Kwong, T. Dang, J. Lichtinger, D. Dachman-Soled, D. Genkin, A. Nelson, R. Perlner, A. Yerukhimovich *et al.*, "When frodo flips: End-to-end key recovery on frodokem via rowhammer," in *Proceedings of the*

2022 ACM SIGSAC Conference on Computer and Communications Security, 2022, pp. 979–993.

[21] S. Islam, K. Mus, R. Singh, P. Schaumont, and B. Sunar, "Signature correction attack on dilithium signature scheme," in *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2022, pp. 647–663.

[22] J. Fan, X. Guo, E. De Mulder, P. Schaumont, B. Preneel, and I. Verbauwhede, "State-of-the-art of secure ecc implementations: a survey on known side-channel attacks and countermeasures," in *2010 IEEE International Symposium on Hardware-Oriented Security and Trust (HOST)*, 2010, pp. 76–87.

[23] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, "Investigations of power analysis attacks on smartcards." *Smartcard*, vol. 99, pp. 151–161, 1999.

[24] D. Kirkwood, B. C. Lackey, J. McVey, M. Motley, J. A. Solinas, and D. Tuller, "Failure is not an option: Standardization issues for post-quantum key agreement," in *Workshop on Cybersecurity in a Post-Quantum World*, 2015, p. 21.

[25] D. S. Specification, "Jedec standard," 2009.

[26] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, Z. Zhang *et al.*, "Falcon: Fast-fourier lattice-based compact signatures over ntru," *Submission to the NIST's post-quantum cryptography standardization process*, vol. 36, no. 5, 2018.