On Generic Side-Channel Assisted Chosen Ciphertext Attacks on Lattice-based PKE/KEMs

Towards key recovery attacks on NTRU-based PKE/KEMs

Prasanna Ravi[1,2], Martianus Frederic Ezerman[3], Shivam Bhasin[1], Anupam Chattopadhyay[1,2], and Sujoy Sinha Roy[4]

[1]Temasek Laboratories, NTU Singapore
[2]School of Computer Science and Engineering, NTU Singapore
[3]School of Physical and Mathematical Sciences, NTU Singapore
[4]Institute of Applied Information Processing and Communications, TU Graz, Austria

`{prasanna.ravi,fredezerman,sbhasin,anupam}@ntu.edu.sg`
`{sujoy.sinharoy}@iaik.tugraz.at`

In this work, we demonstrate novel side-channel assisted chosen ciphertext attack applicable to IND-CCA secure NTRU-based PKE/KEMs. In particular, we propose two types of chosen ciphertext attacks on Streamlined NTRU Prime which instantiate respectively, a plaintext-checking oracle and decryption-failure oracle to perform full key recovery. We propose efficient strategies to construct chosen ciphertexts to instantiate the aforementioned oracles to perform full key recovery. We perform experimental validation of our attacks on the optimized implementation of Streamlined NTRU Prime obtained from the *pqm4* public library, a testing and benchmarking framework for post quantum cryptographic schemes on the ARM Cortex-M4 microcontroller. We positively confirm that both the PC and DF oracle-based attacks result in full key recovery in a few thousand traces with 100% success rate. We also perform a brief survey of the various side-channel assisted chosen-ciphertext attacks on LWE/LWR-based schemes and subsequently identify critical similarities and differences between our proposed attacks as well as known attacks on the LWE/LWR-based schemes. Based on preliminary results from our proposed attacks, we do not observe any considerable increase in the attacker's effort to defeat both LWE/LWR-based schemes as well as NTRU-based schemes by a side-channel attacker in a chosen-ciphertext setting.

1. Introduction

The NIST standardization process for post-quantum cryptography is currently in the third and final round with seven finalist candidates and eight alternate candidates for Public Key Encryption

(PKE), Key Encapsulation Mechanisms (KEM) and Digital Signatures (DS) [AASA+20]. For the third round, NIST has made it clear that resistance against side-channel attacks (SCAs) and fault injection attacks (FIAs) will also be considered as an important criteria for standardization. Thus, resistance towards SCAs and FIAs as well as the cost of implementing protections against SCA and FIA have emerged as important criteria in the standardization process, especially amongst schemes with tightly matched security and efficiency [AH21]. In [AASA+20, Sections 3.4 and 2.2.3] NIST states that it

"encourages additional research regarding side-channel analysis"

of the finalist candidates and

"hopes to collect more information about the costs of implementing these algorithms in a way that provides resistance to such attacks".

Three out of the four finalist candidates for PKE/KEMs are schemes from lattice-based cryptography. They can be broadly classified into two categories. Schemes based on the Learning with Errors (LWE) [Reg09] and Learning with Rounding (LWR) [BPR] problems are in the first category. The second category collects schemes which are based on the N^{th} *order Truncated Polynomial Ring Unit* (NTRU) problem [HPS98]. On the LWE/LWR-based PKE/KEMs, there are several known chosen ciphertext attacks in a static key setting, that is, when the secret key is reused [DCQ19, QCD19, BDHD+19, BGRR19, Flu16]. These attacks mainly work by assuming the presence of an oracle that provides some information about the decrypted message.

There are at least three types of oracles that can be instantiated, depending upon the setting, when using an IND-CPA secure scheme. These are the *Plaintext-Checking* (PC) oracle, the *Decryption-Failure* (DF) oracle, and the *Full Decryption* (FD) oracle. The PC oracle, for example, typically provides a binary response, either correct or wrong, about the attacker's guess of the decrypted message (resp. shared secret key) of a PKE (resp. KEM) for a chosen ciphertext. In the presence of a decryption-failure oracle, an attacker can infer whether or not a given ciphertext results in a decryption failure. While both the PC and DF oracle only provide binary information, a full decryption oracle provides information about the complete message for chosen ciphertexts. Based on the available oracle, an attacker carefully chooses his query ciphertexts in such a way that the corresponding oracle's responses reveal the secret key.

All of the NIST candidate PKE/KEMs apply well-known CCA conversions in order to achieve (an IND-CCA2) security against adaptive chosen ciphertext attacks. An example of such a conversion is the Fujisaki-Okamoto (FO) transform [FO99]. It mainly consists of performing a re-encryption after a decryption. This is done to detect invalid or maliciously formed ciphertexts, which would return failure upon detection. Thus, malicious or invalidly chosen ciphertexts will always be rejected by the decapsulation, preventing the attacker from gaining any meaningful information from the decryption of chosen ciphertexts. This measure removes the presence of any of the aforementioned oracles. This removal is only true, however, in an ideal classical *black box* setting, since any cryptographic algorithm implemented on a real-device leaks information about some intermediate values, such as timing, power consumption or electromagnetic (EM) emanation, through side-channels.

Following this line of thought, there have been several proposed side-channel attacks on LWE/LWR-based PKE/KEMs. They utilize side-channel information to instantiate different types of oracles to provide information about the decryption output, thereby facilitating secret key recovery in several LWE/LWR-based NIST candidates for PKE/KEM [RRCB20, DTVV19, GJN20, XPRO20, BDH+21]. They include the finalists, such as Kyber [ABD+b], Saber [DKSRV], and Frodo [ABD+a]. However, a similar analysis is lacking for the schemes based on the NTRU problem, including the finalist candidates NTRU [CDH$^+$] and NTRU Prime [BBC$^+$]. The framework and underlying arithmetic of NTRU-based schemes are vastly different from those of LWE/LWR-based schemes. This raises critical questions on the susceptibility of NTRU-based schemes to such side-channel assisted chosen ciphertext attacks. Even if they are susceptible, it is unclear if there is a significant difference between the cost of such an attack on an NTRU-based PKE/KEM compared to one on an LWE/LWR-based PKE/KEM.

To address these critical questions, we in this work propose a number of different side-channel assisted chosen ciphertext attacks applicable to IND-CCA secure secure NTRU-based PKE/KEMs. We focus on the Streamlined NTRU Prime variant of NTRU Prime, which is an alternate finalist candidate in the NIST standardization process. We propose two types of chosen ciphertext attacks on Streamlined NTRU Prime. They instantiate, respectively, a PC and a DF oracle and both attacks can recover the full secret key in a few thousand chosen ciphertext queries. We believe our proposed attack can also be adapted to NTRU, the main finalist candidate, differing in the low level technical details.

Moreover, we also perform a brief survey of the various side-channel assisted chosen-ciphertext attacks on LWE/LWR-based schemes and subsequently identify critical similarities and differences between our proposed attacks as well as known attacks on the LWE/LWR-based schemes.

Contributions:

The main contributions and takeaways from our work can be summarized as follows:

1. To the best of our knowledge, we demonstrate the first side-channel assisted chosen ciphertext attack on NTRU-based PKE/KEMs, while previous related works have only focussed on the LWE/LWR-based PKE/KEMs. We adapt the chosen ciphertext attack of Jaulmes and Joux on the classical IND-CPA secure NTRU PKE scheme to the side-channel setting to propose novel chosen ciphertext attacks on the IND-CCA secure Streamlined NTRU Prime KEM.

2. We propose two types of chosen ciphertext attacks on Streamlined NTRU Prime which instantiate respectively, a PC and FD oracle to perform full key recovery. We demonstrate very efficient strategies to construct chosen ciphertexts and also efficiently utilize the EM side-channel to instantiate precise PC and DF oracles to perform full key recovery. Though our attacks are demonstrated only on the Streamlined NTRU Prime, we believe our proposed attack can also be adapted to NTRU, the main finalist candidate, albeit differing in the low level details.

3. We perform experimental validation of our attacks on the optimized implementation of Streamlined NTRU Prime obtained from the *pqm4* public library, a testing and benchmarking framework for post quantum cryptographic schemes on the ARM Cortex-M4 microcontroller [KRSS]. We positively confirm that both the PC and DF oracle-based attacks result in full key recovery in a few thousand traces with 100% success rate.

4. We also perform a brief survey of the various side-channel assisted chosen-ciphertext attacks on LWE/LWR-based schemes and subsequently identify critical similarities and differences between our proposed attacks as well as known attacks on the LWE/LWR-based schemes. Based on preliminary results from our proposed attacks, we do not observe any considerable increase in the attacker's effort to defeat both LWE/LWR-based schemes as well as NTRU-based schemes by a side-channel attacker in a chosen-ciphertext setting.

Organization of the Paper

This paper is organized as follows. Section 2 surveys prior work on side-channel assisted chosen ciphertext attacks on LWE/LWR-based schemes. Section 3 presents our proposed PC and DF oracle-based attacks on Streamlined NTRU Prime. This section also includes discussions on comparison of our proposed attacks against prior attacks proposed over LWE/LWR-based schemes. Section 4 concludes the paper. This paper also contains an appendix contains concrete details of our proposed PC oracle and DF oracle attacks on Streamlined NTRU Prime KEM.

2. Side-Channel Assisted Chosen Ciphertext Attacks on LWE/LWR-based schemes

Most side-channel assisted chosen ciphertext attacks on LWE/LWR-based schemes work in the following manner: the attacker constructs specially structured ciphertexts which when decrypted, ensure that a certain intermediate variable is limited to a very few possible values and its value also depends exlusively on a certain targeted portion of the secret key. He/She then utilizes side-channel leakage to obtain information about the targeted internal variable and such information obtained across several such chosen ciphertexts leads to full key recovery. In LWE/LWR-based PKE/KEMs, the *decrypted message* denoted as m has served as the target variable for several SCA assisted chosen-ciphertext attacks [DTVV19, RRCB20, GJN20]. We refer to the target variable as the *anchor variable* throughout the paper. We can classify these attacks that work in a chosen-ciphertext setting into three main categories:

2.1. PC Oracle-based SCA

Most LWE/LWR-based PKE/KEMs such as Kyber, Saber and Frodo (NIST finalist candidates) are built upon a generic framework which contains an IND-CPA secure PKE based on the well known LPR Encryption scheme [LPR10]. The design of the PKE scheme allows an attacker to construct chosen ciphertexts such that the *anchor variable* (i.e.) the decrypted message m can be limited to a very few values known to the attacker (i.e.) for schemes such as Kyber and Saber (based on the MLWE/MLWR problem), the message can be limited to only two values - $m = 0$ (all zeros) or $m = 1$ (all zero with 1 at LSB). Moreover, the value of the message ($m = 0/1$) for the chosen ciphertexts is also dependent upon a targeted coefficient of the secret key. An attacker who has access to an appropriate *oracle* to classify $m = 0/1$ for chosen ciphertexts can perform full key recovery one coefficient at a time.

Several works have utilized side-channels to instantiate a PC oracle which enables distinguishing between the two classes $m = 0/1$ for key recovery attacks. In this dirction, D'Anvers *et al.* [DTVV19] reported the first such attack on two post-quantum KEMs LAC and RAMSTAKE by extracting information about the message through timing side-channel information from variable-time error correcting procedures used in decryption. Subsequently, Ravi *et al.* [RRCB20] generalized the attack to constant-time implementations of multiple LWE/LWR-based PKE/KEMs by obtaining information about the decrypted message through the EM side-channel. These attacks lead to key recovery in a few thousand queries to the target device. The attack is performed in two phases:

1. **Pre-Processing Phase**: In this phase, an attacker constructs side-channel templates for all possible values of the decrypted message ($m = 0/1$) corresponding to the attacker's chosen ciphertexts. Thus, an attacker can query the decapsulation device with valid ciphertexts for

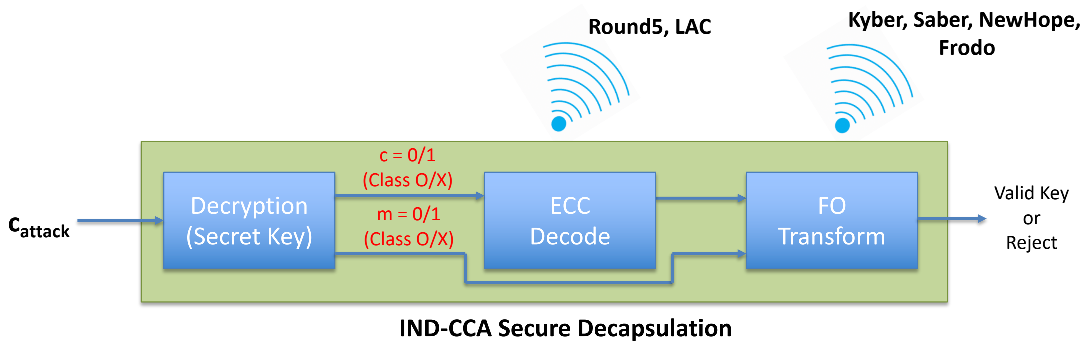

Figure 1: Pictorial representation of PC oracle-based SCA on decapsulation procedure of LWE/LWR-based schemes

the possible message values and utilize appropriate side-channel information to build templates for each class ($m = 0/1$).

2. **Key Recovery Phase**: In this phase, an attacker constructs chosen ciphertexts $\mathbf{c}_{\mathbf{attack}}$ whose decrypted message contains information about a targeted portion (coefficient) of the secret key. The attacker queries the decapsulation device with the attack ciphertexts $\mathbf{c}_{\mathbf{attack}}$ and side-channel information from decryption of these ciphertexts can be used to classify whether ($m = 0/1$) (using templates obtained from the pre-processing phase), thereby realizing a practical PC oracle. Subsequently, the side-channel oracle's responses obtained over several ciphertext queries leads to full secret key recovery. Please refer to Figure 1 for the pictorial description of the PC oracle-based SCA on LWE/LWR-based schemes targeting the IND-CCA secure decapsulation operation.

One of the main advantages of the aforementioned attack on LWE/LWR-based schemes is that the decrypted message m serves as the anchor variable. Since an attacker can construct ciphertexts corresponding to any value of m, he/she can easily build templates for different values of m. Moreover, the attacker's chosen ciphertexts always limit the value of the anchor variable (m) to just two known values, irrespective of the secret key. This ensures that the pre-processing phase is only required to be performed once for the target device.

2.2. DF Oracle-based SCA

There exists another class of attacks which works by obtaining information about decryption failures for attacker's chosen ciphertexts for key recovery. The attacker adds errors to a valid ciphertext in such a way that, the occurrence of decryption failures for these modified ciphertexts (i.e.) $m = m_{\text{valid}}/m_{\text{invalid}}$ contains information about a targeted portion of the secret key. Side-channels can be utilized to detect decryption failures for chosen-ciphertexts, thereby realizing a practical DF oracle. Such information obtained for several such modified ciphertexts leads to full key recovery. In this respect, Guo *et al.* [GJN20] exploited timing side-channel information from the non-constant time ciphertext comparison operation in the decapsulation procedure of Frodo KEM. This was used to realize a practical timing-based DF oracle which resulted in full key recovery in several thousand queries. Subsequently, Bhasin *et al.* [BDH+21] identified EM side-channel vulnerabilities in the ciphertext comparison operation to instantiate a DF oracle for full key recovery in Kyber KEM using several thousand chosen-ciphertext queries.

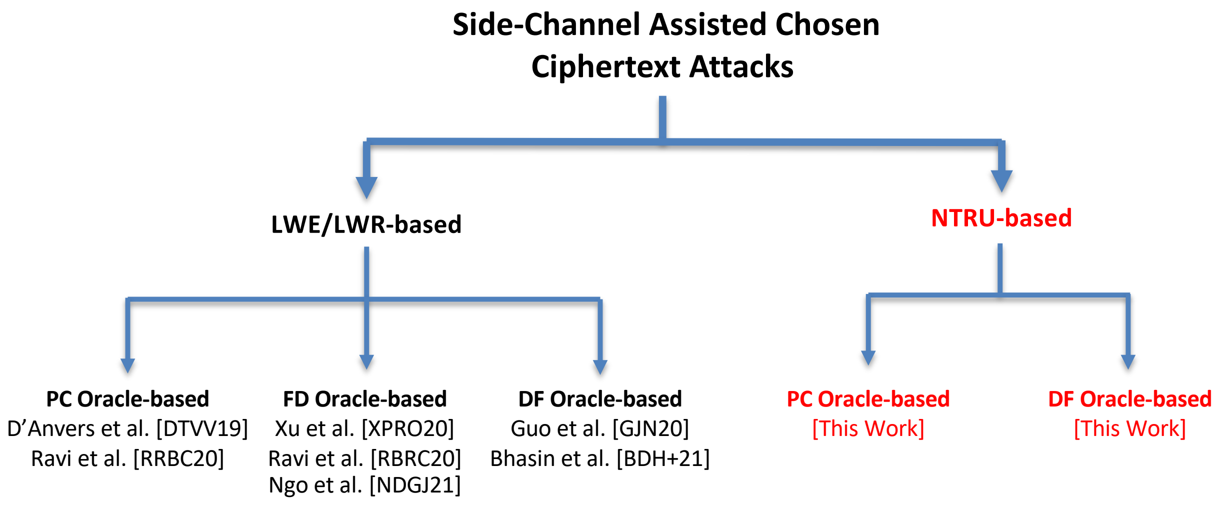

Figure 2: Classification of the various side-channel assisted chosen ciphertext Attacks on lattice-based schemes

2.3. FD Oracle-based SCA

The aforementioned attacks rely on binary information from the PC oracle or DF oracle for full key recovery in the range of thousand of queries to the target device. This raises a natural question about the possibility of more efficient attacks with a more powerful oracle which provides more than just binary information about the decrypted message. In this direction, Xu *et al.* [XPRO20] showed that an attacker who can obtain complete knowledge of the decrypted message for chosen ciphertexts can effectively *parallelize* the chosen-ciphertext attack, resulting in full key recovery in only a handful of traces/queries. They showed that vulnerabilities in the message encoding (Amiet *et al.* [ACLZ20], Sim *et al.* [SKL$^+$20]) and decoding procedure (Ravi *et al.* [RBRC20], Ngo *et al.* [NDGJ21]) which leak the complete message can be exploited to perform full key recovery only using 8 chosen ciphertext queries for Kyber (Kyber512) and the same attack can be extended to other LWE/LWR-based schemes as well.

In all the aforementioned attacks on LWE/LWR-based schemes, the decrypted message m serves as the anchor variable of interest whose information obtained through side-channels results in full key recovery. While the aforementioned attacks have only been demonstrated on LWE/LWR-based schemes, similar attacks have not yet been studied for NTRU-based schemes. In this work, we demonstrate the first SCA assisted chosen-ciphertext attack for NTRU-based schemes and in particular, Streamlined NTRU Prime KEM. Refer Figure 2 for a classification of the various SCA assisted chosen ciphertext attack on lattice-based schemes where our proposed PC oracle and DF oracle-based attacks on the Streamlined NTRU Prime scheme are highlighted in red.

3. SCA Assisted Chosen Ciphertext Attacks on Streamlined NTRU Prime KEM

In this section, we demonstrate two types of side-channel assisted chosen-ciphertext attacks on IND-CCA secure Streamlined NTRU Prime KEM. They are: 1) Plaintext-Checking Oracle-based SCA and 2) Decryption-Failure Oracle-based SCA.

3.1. Plaintext-Checking Oracle-based SCA

Refer Figure 3 for a pictorial description of our PC oracle-based SCA on the decryption procedure of Streamlined NTRU Prime KEM. Our PC oracle-based attack can be seen as a direct analogue to the generic PC oracle-based side-channel attacks in [DTVV19, RRCB20] for LWE/LWR-based schemes. Our attack is performed into two phases:

1. **Template Generation Phase**: We construct chosen ciphertexts of a very specific structure (inspired from the chosen ciphertext attack proposed by Jaulmes and Joux on the classical IND-CPA secure NTRU PKE scheme in Crypto 2000 [JJ00]) and attempt to identify a particular ciphertext denoted as $\mathbf{c}_{\text{base}}$ whose intermediate variable $\mathbf{e}$ only contains a single non-zero coefficient (i.e) $\mathbf{e} = \pm 1 \cdot x^i$ where $i \in [0, n-1]$ is the non-zero coefficient. This ciphertext $\mathbf{c}_{\mathbf{base}}$ is subsequently utilized to build attack ciphertexts to perform key recovery.

   Unlike the LWE/LWR-based schemes, where the decrypted message m is the anchor variable, the internal variable $\mathbf{e}$ within the decryption procedure, serves as the anchor variable, since its value can be controlled to be dependent on a targeted portion of the secret key. We obtain side-channel information from operations processing the anchor variable $\mathbf{e}$ (enclosed in rectangular box in Figure 3) and adopt the Welch's t-test to identify $\mathbf{c}_{\mathbf{base}}$ whose $\mathbf{e} = \pm 1 \cdot x^i$. Subsequently, we use the distinguishing features identified from the Welch's t-test to build reduced templates for two classes: $\mathbf{e} = 0$ (Class **O**) and $\mathbf{e} = \pm 1 \cdot x^i$ (Class **X**).

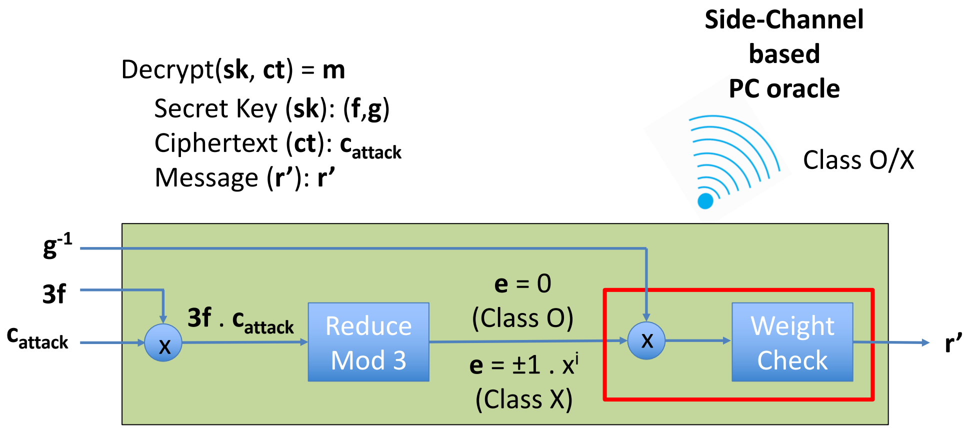

Figure 3: Pictorial Illustration of our PC oracle attack on the decryption procedure of Streamlined NTRU Prime

2. **Key Recovery Phase**: In this second phase, we use the base ciphertext $\mathbf{c}_{\mathbf{base}}$ to construct new attack ciphertexts $\mathbf{c}_{\mathbf{attack}}$. These attack ciphertexts are constructed such that the corresponding intermediate variable $\mathbf{e}$ can only belong to two exclusive classes: $\mathbf{e} = 0$ (Class O) or $\mathbf{e} = \pm 1 \cdot x^i$ (Class X). Moroever, the value of $\mathbf{e} = 0$ or $\mathbf{e} = \pm 1 \cdot x^i$ depends upon a targeted portion of the secret key. We utilize side-channel information from the decryption of the attack ciphertexts and use the side-channel templates to classify each attack ciphertext as class O/**X**, thereby realizing a practical PC oracle. We use the side-channel oracle's responses over several attack ciphertext queries to recover the full secret key. Refer Figure 4 for the flow diagram of our proposed PC oracle-based chosen ciphertext attack.

We refer the reader to appendix B for concrete details of our proposed PC oracle-based SCA on Streamlined NTRU Prime KEM.

3.1.1. Experimental Setup:

For practical experiments, we ran the optimized implementation of `sntrup761` taken from the open-source *PQM4* library [KRSS]. We used the STM32F4DISCOVERY board (DUT) housing the STM32F407, ARM Cortex-M4 microcontroller as our DUT. The implementation, compiled with `-O3 -mthumb -mcpu=cortex-m4 -mfloat-abi=hard -mfpu=fpv4-sp-d16`, was clocked at 168 MHz. The ST-LINK/v2.1 add-on board was deployed for UART communication with our DUT. EM measurements were observed from the DUT using a near-field probe and processed using a Lecroy 610Zi oscilloscope at a sampling rate of 500MSam/sec. Figure 5 shows our experimental setup to perform EM trace acquisition.

3.1.2. Experimental Results:

We implemented our attack on the optimized implementation of `sntrup761` and utilized EM side-channel information corresponding to the final weight-checking operation (denoted as Weight Check in Figure 3) within the decryption procedure to instantiate the PC oracle.

Template Generation Phase: We repeatedly query the decapsulation device with the ciphertext $\mathbf{c} = 0$ corresponding to $\mathbf{e} = 0$ (Class O) and denote the side-channel information as $\mathcal{T}_{\mathsf{O}}$ ($N = 10$ traces). We repeatedly query the decapsulation device with a malicious ciphertext $\mathbf{c}'$ and the corresponding side-channel information be denoted as $\mathcal{T}_X$ ($N = 10$ traces). We then compute the Welch's t-test between $\mathcal{T}_{\mathsf{O}}$ and $\mathcal{T}_{\mathsf{X}}$. Refer Figure 6(a) for the t-test plot for $\mathbf{c}'$ whose $\mathbf{e} = 0$. Thus, there are no peaks about the t-test threshold (± 4.5) indicating no difference between the two measurements. Refer Figure 6(b) for the t-test plot for $\mathbf{c}$ whose $\mathbf{e} = \pm 1 \cdot x^i$ which clearly shows several peaks, well above the threshold ± 4.5, thereby indicating $\mathbf{e} \neq 0$.

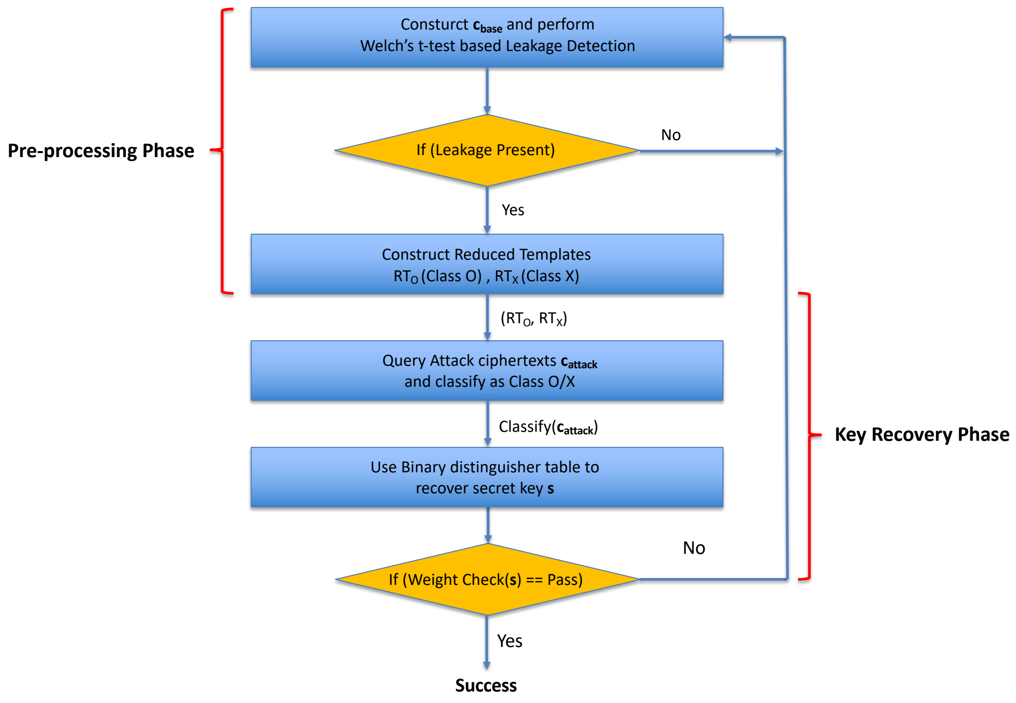
Pre-processing Phase
Consturct $\mathbf{c}_{\mathbf{base}}$ and perform Welch's t-test based Leakage Detection
If (Leakage Present)
No
Yes
Construct Reduced Templates RT$_O$ (Class O) , RT$_X$ (Class X)
(RT$_O$, RT$_X$)
Query Attack ciphertexts $\mathbf{c}_{\mathbf{attack}}$ and classify as Class O/X
Classify($\mathbf{c}_{\mathbf{attack}}$)
Use Binary distinguisher table to recover secret key **s**
No
If (Weight Check(**s**) == Pass)
Yes
Success
Key Recovery Phase

Figure 4: Attack Flow Diagram of our proposed PC Oracle-based SCA on Streamlined NTRU Prime KEM

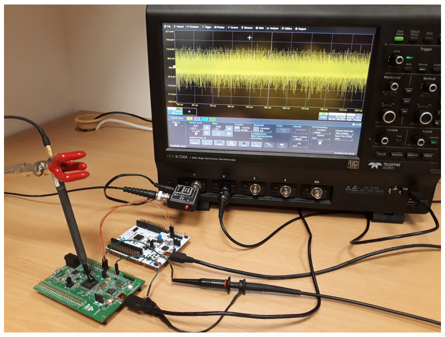

Figure 5: The experimental setup used in EM trace acquisition.

For sntrup761, we required ≈ 61 trials on average to identify the base ciphertext $\mathbf{c}_{base}$. Each trial requires $N = 10$ traces for the Welch's t-test, resulting in ≈ 610 traces to identify $\mathbf{c}_{base}$. The features in the t-test plot well above the threshold are chosen as Points of Interest (PoI) to build reduced templates for both classes $\mathbf{e} = 0$ (Class $\mathbf{O}$) and $\mathbf{e} = \pm 1 \cdot x^i$ (Class $\mathbf{X}$).

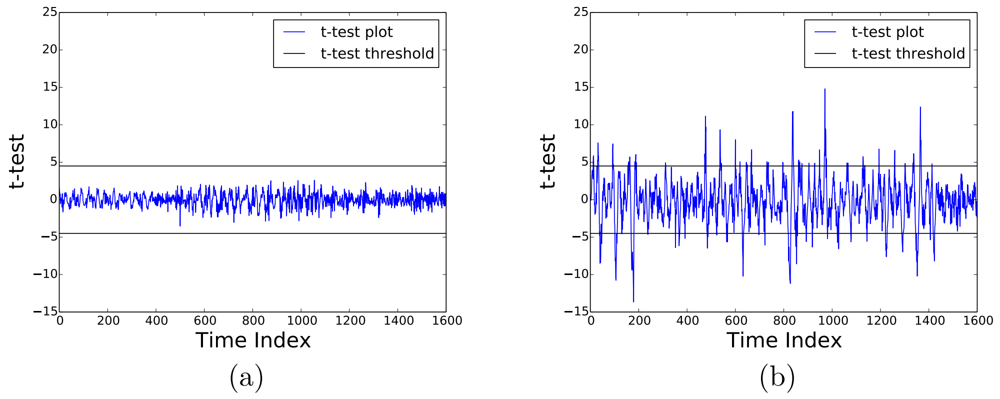

Figure 6: t-test plots between $\mathcal{T}_\mathrm{O}$ and $\mathcal{T}_\mathrm{X}$ (a) $\mathbf{e} = 0$ (b) $\mathbf{e} \neq 0$ for sntrup761 parameter set of Streamlined NTRU Prime

Key Recovery Phase: We query the decapsulation device with attack ciphertexts $\mathbf{c_{attack}}$ constructed from $\mathbf{c_{base}}$. Side-channel information from the decryption of these attack ciphertexts are matched with the side-channel templates to classify as either class $\mathbf{O}/\mathbf{X}$. Figure 7 visualizes the matching of a given attack trace with the reduced templates of both the classes. We can see that there is a clear distinguishability between the reduced templates of the two classes, leading to a classification with 100% success rate. Recovery of single targeted coefficient takes about 4 chsoen-ciphertext queries and therefore 4 side-channel traces. For sntrup761, there are 761 coefficients in the secret key, thereby yielding 3044 traces. There are a few ocassions when the retrieved secret key is not correctly retrieved which results in re-run of the attack. Altogether, complete recovery of the secret key $\mathbf{f}$ takes approximately $4.35k$ traces. Our attack works with a success rate of about 100%.

For comparison of the attacker's effort to mount the PC oracle-based SCA on LWE/LWR-based schemes, we utilize experimental results from the work of Ravi *et al.* [RRCB20] who utilized the same attack setup and target to attack LWE/LWR-based schemes. Their attack on the Kyber512 parameter set of Kyber required about $7.7k$ traces for full key recovery, but this count pertains to three attack iterations to improve the success rate. Thus, a single iteration takes about $2.56k$ traces for full key recovery assuming a single attack iteration. Though the number of traces required to attack sntrup761 is more than twice compared to Kyber512, the increase in the attacker's effort is not very significant.

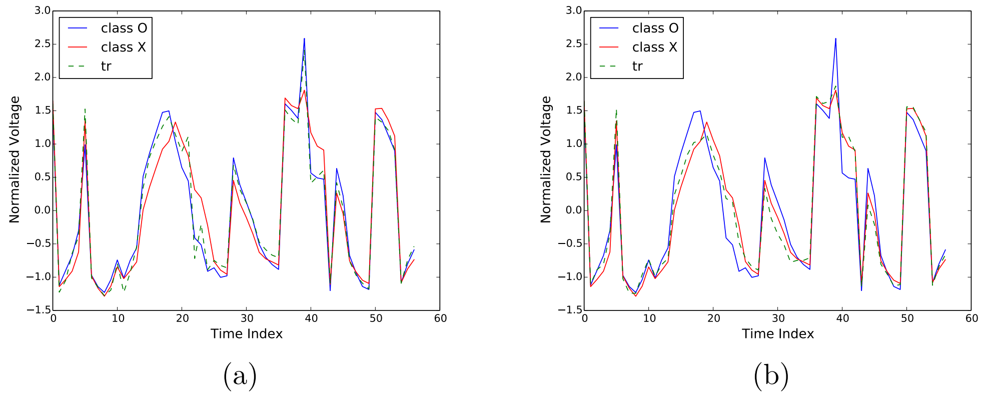

Figure 7: Matching the reduced template tr of a given attack trace with the reduced template of the two classes $\mathbf{O}$ and $\mathbf{X}$ for the PC oracle attack (a) $\mathsf{Class}(tr) = \mathbf{O}$ (b) $\mathsf{Class}(tr) = \mathbf{X}$

3.1.3. Comparison with PC Oracle-based SCA on LWE/LWR-based PKE/KEMs

While the high level approach of our proposed PC oracle-based SCA on Streamlined NTRU Prime closely resembles the attack on LWE/LWR-based schemes, we identify a few critical differences between the two attacks. The main difference arises the difference in the anchor variable (i.e.) the variable manipulated to be secret-dependent that aids key recovery.

While the attacks on LWE/LWR-based schemes use the decrypted message m as the anchor variable, an internal variable $\mathbf{e}$ within the decryption procedure serves as the anchor variable for our attack. Since $\mathbf{e}$ is an internal variable, an attacker has much lesser control over its exact value. Thus, the first phase of the attack involves a search for a special ciphertext $\mathbf{c_{base}}$ whose $\mathbf{e}$ has a single non-zero coefficient (i.e.) $\mathbf{e} = \pm 1 \cdot x^i$. This search for $\mathbf{c_{base}}$ has a minor effect on the attack's trace complexity compared to that on LWE/LWR-based schemes. However, such a search is not required for LWE/LWR-based schemes as an attacker can readily build ciphertexts for any value of the decrypted message m.

Moreover, it is not possible control the non-zero value of $\mathbf{e}$ (i.e.) $\mathbf{e} = \pm 1 \cdot x^i$, since the location of the non-zero coefficient i depends on the secret key and cannot be arbitrarily fixed by the attacker. Thus, the template generation is needed to be performed for every new secret key to be recovered. However, for LWE/LWR-based schemes, the two classes $m = 0/1$ are fixed for LWE/LWR-based schemes, irrespective of the secret key. Thus, template generation (referred to as pre-processing phase) is only required to be done once for a given target device.

3.1.4. A few observations on the PC Oracle-based SCA

We observe that the decryption procedure of Streamlined NTRU Prime KEM only outputs a valid message only when the weight of the decrypted message is w (w is a parameter depending upon the variant of Streamlined NTRU Prime). Upon failure, the decryption procedure returns a constant message with value $(1, 1, \ldots, 1, 0, 0, \ldots, 0)$. The decrypted message for the attack ciphertexts do have not have weight w (i.e.) If $\mathbf{e} = 0$ (Class O), weight$(m) = 0$ and if $\mathbf{e} = \pm \cdot x^i$ (Class X), then m contains uniformly random coefficients in the range $[-1, 1]$ and thus weight$(m) \approx 2n/3 \neq w$ with a very high probability. Thus, the attack ciphertexts always decrypt to the fixed message $(1, 1, \ldots, 1, 0, 0, \ldots, 0)$.

Thus, the effect of variable $\mathbf{e}$ ($\mathbf{O}/\mathbf{X}$) for the attack ciphertexts does not propagate beyond the decryption procedure. Thus, the PC oracle attack can only be carried out using side-channel information from operations within the decryption procedure. In particular, operations manipulating $\mathbf{e}$ within decryption (operations enclosed in rectangular box in Figure 3). This restricts an attacker from utilizing side-channel information from operations after decryption (i.e.) operations within the re-encryption procedure during decapsulation.

In the following, we improve upon the PC oracle-based attack to propose a novel DF oracle-based attack on Streamlined NTRU Prime KEM. The improved attack enables an attacker to also utilize side-channel information from many more operations within the decapsulation procedure to perform side-channel assisted key recovery.

3.2. Decryption-Failure (DF) Oracle-based SCA

We start by providing some intuition for the Decryption-Failure (DF) oracle attack. The main idea is to *perturb valid ciphertexts using ciphertexts similar to those used for the PC oracle attack, and subsequently observe the effect of perturbation on the decrypted message.* Let $\mathbf{c}_{\text{valid}}$ be a valid ciphertext whose $\mathbf{e}$ is denoted as $\mathbf{e}_{\text{valid}}$. Let $\mathbf{c}'$ be a set of specially crafted ciphertexts, which are similar to those used for the PC oracle attack. Upon decrypting $\mathbf{c}'$, $\mathbf{e}'$ can only have two possible values - (1) $\mathbf{e} = 0$ or (2) $\mathbf{e} = \pm 1 \cdot x^i$. Let the sum of perturbation ciphertext $\mathbf{c}'$ and valid ciphertext $\mathbf{c}_{\text{valid}}$ be denoted as the perturbed ciphertext $\mathbf{c}_{\text{pert}}$. Perturbing $\mathbf{c}_{\text{valid}}$ in this manner in turn perturbs $\mathbf{e}_{\text{valid}}$ in such a way that the resulting $\mathbf{e}$ for $\mathbf{c}_{\text{pert}}$ can take two possible values: (1) $\mathbf{e}_{\text{valid}}$ (Class $\mathbf{O}$) or (2) $\mathbf{e}_{\text{valid}}$ with a single coefficient error at i denoted as $\mathbf{e}_{\text{invalid}}$ (Class $\mathbf{X}$).

For the first class of ciphertexts ($\mathbf{e}_{\text{valid}}$), there is no error in decryption and thus the decryption procedure returns $\mathbf{r}'_{\text{valid}}$. However for the second class of ciphertexts, there is a single coefficient error in $\mathbf{e}$ (i.e) $\mathbf{e}_{\text{invalid}} = \mathbf{e}_{\text{valid}} \pm 1 \cdot x^i$ which causes a decryption failure and thus the constant $(1, 1, \ldots, 1, 0, 0, \ldots, 0)$ (denoted as $\mathbf{r}'_{\text{invalid}}$) is returned as the decrypted message. Thus, the perturbed

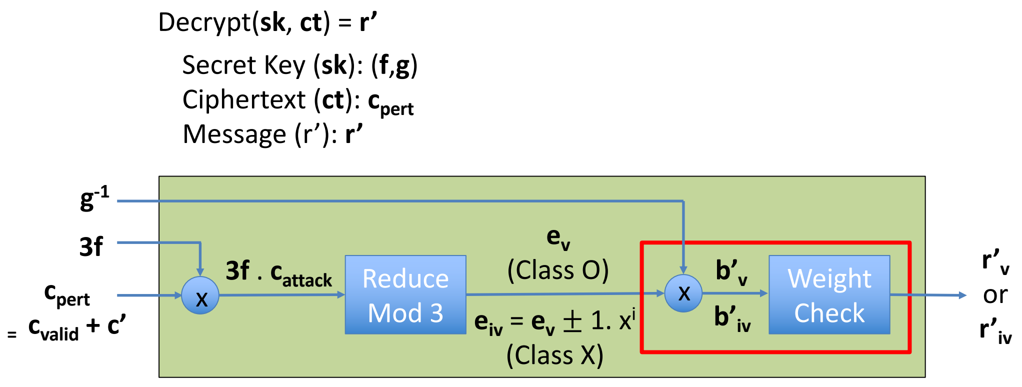

Figure 8: DF Oracle Attack on the decryption procedure of Streamlined NTRU Prime KEM. Subscript v denotes valid while subscript iv denotes invalid.

ciphertexts $\mathbf{c}_{\text{pert}}$ restrict the decrypted message $\mathbf{r}'$ to two possibilities (i.e.) $\mathbf{r}'_{\text{valid}}/\mathbf{r}'_{\text{invalid}}$. This is unlike the ciphertexts used for the PC oracle attack where the decrypted message always take a value of $\mathbf{r}'_{\text{invalid}}$.

The success/failure of decryption for the perturbed ciphertexts depends upon a targeted portion of the secret key. Thus, an attacker who can obtain information about decryption success/failure for several chosen-ciphertext queries can fully recover the secret key. This is referred to as the DF oracle attack where the adversary exploits a binary *oracle* providing information about the success/failure of decryption aiding key recovery. Figure 8 illustrates the DF oracle attack using the decryption procedure of Streamlined NTRU Prime KEM.

We observe that a decryption failure can be identified by observing either of the variables $\mathbf{e}$ or $\mathbf{r}'$. Thus, side-channel leakage from operations manipulating either of these variables can be used to instantiate a DF oracle. We can use the operations manipulating $\mathbf{e}$ within the decryption procedure (similar to the PC oracle attack). Moreover, we can also utilize operations in the re-encryption procedure since it takes $\mathbf{r}'$ as its explicit input. This widens the scope of an attacker (compared to the PC oracle attack) to obtain side-channel information from several operations in the decapsulation procedure for key recovery. Figure 9 illustrates the DF oracle attack on the decapsulation procedure of Streamlined NTRU Prime KEM. The DF oracle attack is performed in two phases.

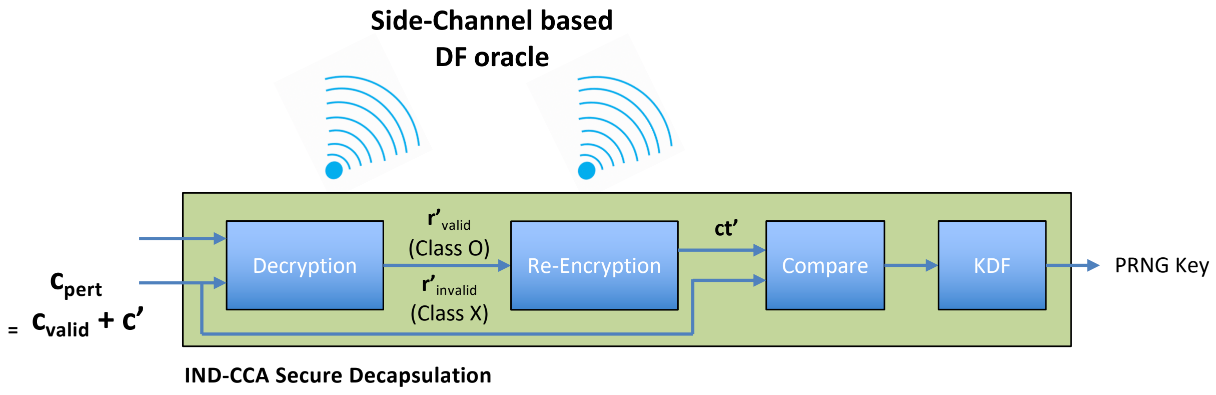

Figure 9: Illustration of the DF oracle attack on the decapsulation procedure of Streamlined NTRU Prime KEM

1. **Template Generation Phase**: The first phase involves search for a base ciphertext $\mathbf{c}_{\text{base}}$ which corresponds to a *single collision* event. This ciphertext when added to a valid ciphertext $\mathbf{c}_{\text{valid}}$ should exactly perturb a single coefficient of $\mathbf{e}_{\text{valid}}$ resulting in a decryption failure. Similar to the PC oracle-based SCA, we use the Welch's t-test to identify $\mathbf{c}_{\text{base}}$ which results in a decryption failure. Subsequently, we use $\mathbf{c}_{base}$ to construct side-channel templates for the two classes: $\mathbf{r}'_{\text{valid}}$ (Class $\mathbf{O}$) and $\mathbf{r}'_{\text{invalid}}$ (Class $\mathbf{X}$).

2. **Key Recovery Phase**: In the second phase, $\mathbf{c}_{\text{base}}$ is used to construct new perturbed attack ciphertexts $\mathbf{c}_{\text{att}}$ whose decrypted message can only have two possible values (i.e.) $\mathbf{r}'_{\text{valid}}/\mathbf{r}'_{\text{invalid}}$. Side-channel leakage from decapsulation of these attack ciphertexts is used to detect decryption failures, thereby realizing a practical DF oracle. This information obtained over several such attack ciphertexts results in full key recovery.

   We refer the reader to appendix C for concrete details of our proposed DF oracle-based SCA on Streamlined NTRU Prime KEM.

3.2.1. Experimental Results:

We implemented our attack on the optimized implementation of sntrup761 parameter set of IND-CCA secure Streamlined NTRU Prime KEM. We obtained EM side-channel measurements from the encoding operation that manipulates the decrypted message within the re-encryption procedure after decryption.

Template Generation Phase: We repeatedly query the decapsulation device with $\mathbf{c_{valid}}$ and denote the corresponding side-channel information as $\mathcal{T}_{\mathrm{O}}$ ($N = 10$ traces). We also repeatedly query the decapsulation device with a perturbed ciphertexts $\mathbf{c}_{\text{pert}}$ (perturbed using $\mathbf{c}'$) and the corresponding side-channel information be denoted as $\mathcal{T}_X$ ($N = 10$ traces). Refer Figure 10(a) which depicts the t-test plot corresponding to $\mathbf{c}_{\text{pert}}$ which does not result in any decryption failure, which is indicated by no peaks in the t-test plot. Refer Figure 10(b) for the t-test plot corresponding to $\mathbf{c}_{\text{pert}}$ whose decrypted message is $\mathbf{r}_{\text{invalid}}$ which clearly shows several peaks, well above the threshold ± 4.5, thereby indicating a decryption failure. Similar to the PC oracle-based SCA, the features in the t-test plot well above the threshold are chosen as Points of Interest (PoI) to build reduced templates for both classes $\mathbf{r}'_{\text{valid}}$ (Class $\mathbf{O}$) and $\mathbf{r}'_{\text{invalid}}$ (Class $\mathbf{X}$). The above t-test based leakage detection is repeated for different perturbation ciphertexts $\mathbf{c}'$ until we identify one which results in a decryption failure. Such a ciphertext is denoted as the base ciphertext $\mathbf{c}_{\text{base}}$. The first step to identify $\mathbf{c}_{\text{base}}$ takes ≈ 425 attempts. Each attempt requires the capture of $N = 10$ traces. Thus, it takes ≈ 4250 traces to identify $\mathbf{c}_{base}$.

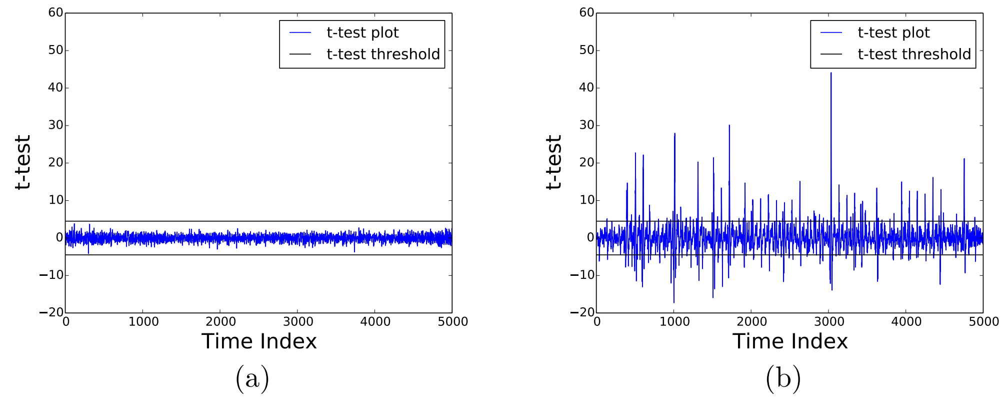

Figure 10: t-test plots between $\mathcal{T}_{\mathrm{O}}$ and $\mathcal{T}_{\mathrm{X}}$ (a) $m = m_{\text{valid}}$ (b) $m = m_{\text{invalid}}$ for sntrup761 parameter set of Streamlined NTRU Prime

Key Recovery Phase: In this phase, we query the decapsulation device with attack ciphertexts constructed using $\mathbf{c}_{\text{base}}$ which decrypt to either $\mathbf{r}'_{\text{valid}}/\mathbf{r}'_{\text{invalid}}$. We obtain side-channel information from operations within the re-encryption procedure and subsequently utilize the side-channel templates for classification. Figure 11 visualizes the matching of a given attack trace with the reduced templates of both the casses. We can see that there is a clear distinguishability between the reduced templates of the two classes, leading to a classification with 100% success rate. Recovery of single targeted coefficient takes about 4 chosen ciphertext queries and therefore 4 side-channel traces. For sntrup761, there are 761 coefficients in the secret key, thereby yielding 3044 traces. There are a few ocassions when the retrieved secret key is not correctly retrieved which results in re-run of the attack. Altogether, complete recovery of the secret key $\mathbf{f}$ takes approximately $8.1k$ traces. Our attack works with a success rate of about 100%.

For comparison with the attacker's effort to mount similar DF oracle-based SCA on LWE/LWR-based schemes, we utilize experimental results from the work of Bhasin *et al.* [BDH+21] who demonstrated DF oracle-attacks on Kyber implemented on the ARM Cortex-M4 microcontroller exploiting the EM side-channel. Their attack requires about 2^{17} decapsulation queries to reduce the security of Kyber512 to 2^{65}, but our attack only requires much fewer traces $9.6k$ to perform full key recovery. Thus, our proposed DF oracle-based attack on Streamlined NTRU Prime is much more efficient compared to the attack of Bhasin *et al.* [BDH+21] on Kyber. Moreover, the timing attack of Guo *et al.* [GJN20] also required about 2^{30} decapsulation queries for full key recovery on Frodo. Though this number includes replicated measurements for better SNR, we believe that our DF-oracle based attack methodology also performs much better than the proposed attack of Guo *et al.* [GJN20] on Frodo.

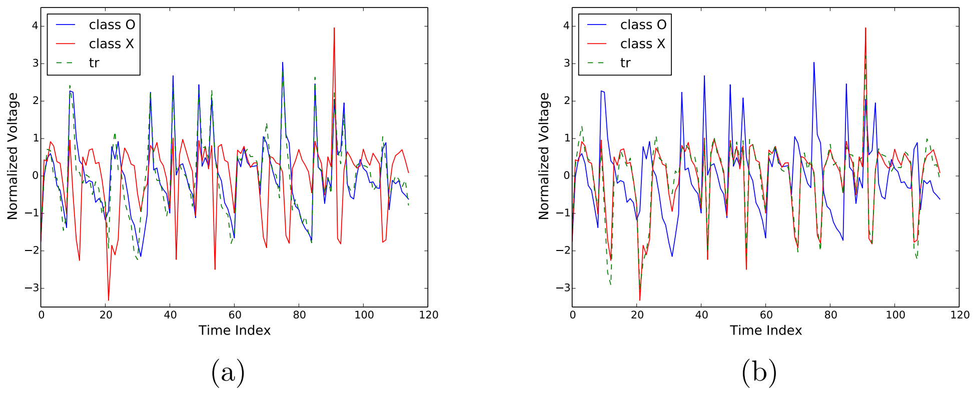

Figure 11: Matching the reduced template tr of a given attack trace with the reduced template of the two classes $\mathbf{O}$ and $\mathbf{X}$ for the DF oracle attack (a) $\mathsf{Class}(tr) = \mathbf{O}$ (b) $\mathsf{Class}(tr) = \mathbf{X}$

4. Countermeasures

In this section, we briefly address potential countermeasures for both the PC oracle and DF oracle attacks over Streamlined NTRU Prime KEM. Our attack relies on fixing targeted intermediate variables to known values and subsequently utilizing side-channel leakage to identify its value to perform key recovery. Thus, complete randomization of the internal computation through masking countermeasure thus serves as a concrete countermeasure against both our proposed attacks. For the PC oracle attack, it is sufficient to only mask the decryption procedure as side-channel leakage from other operations within the decapsulation procedure cannot be exploited. However, for the DF oracle attack, the entire IND-CCA secure decapsulation procedure needs to be masked so as to provide concrete protection.

Masking countermeasures in general are known to be costly in terms of performance. There are several works on protecting NTRU-based prmitives [LSCH10, WZW13, HCY20, SMS19] against side-channel attacks, however existing attacks as well as countermeasures only target the polynomial multiplier involving the secret key in the decryption procedure. However, our attacks have shown that other operations within the decryption and decapsulation procedure can also be targeted for key recovery. Moreover, schemes such as Streamlined NTRU Prime include non-linear operations such as the weight check operation (Figure 3) within the decryption procedure which are not trivial to mask. To the best of our knowledge, we are not aware of a concrete and complete masking scheme for NTRU-based PKE/KEMs. Thus, development of efficient and concrete masking strategies for NTRU-based PKE/KEMs is an interesting research direction that warrants immediate attention by the cryptographic research community.

5. Conclusion

In this work, we have demonstrated novel side-channel assisted PC and DF oracle-based chosen ciphertext attacks on the IND-CCA secure Streamlined NTRU Prime KEM. We perform experimental validation of our attacks on the optimized implementation of Streamlined NTRU Prime obtained from the *pqm4* public library, a testing and benchmarking framework for post quantum cryptographic schemes on the ARM Cortex-M4 microcontroller. We positively confirm that both the PC and DF oracle-based attacks result in full key recovery in a few thousand traces with 100% success rate. Based on preliminary results from our proposed attacks, we do not observe any considerable increase in the attacker's effort to defeat both LWE/LWR-based schemes as well as NTRU-based schemes in a side-channel setting. Masking serves as a concrete countermeasure against our proposed attacks and thus our work stresses on the need for concrete masking countermeasures for the NTRU-based PKE/KEMs to protect against similar chosen-ciphertext based side-channel attacks.

References

[AAB$^+$] Erdem Alkim, Roberto Avanzi, Joppe W. Bos, Leo Ducas, Antonio de la Piedra, Thomas Poppelmann, Peter Schwabe, and Douglas Stebila. NewHope (Version 1.1): Algorithm Specifications And Supporting Documentation (April 10, 2020). *Submission to the NIST post-quantum project.*

[AASA$^+$20] Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, John Kelsey, Yi-Kai Liu, Carl Miller, Dustin Moody, Rene Peralta, et al. Status report on the second round of the NIST post-quantum cryptography standardization process. *US Department of Commerce, NIST*, 2020.

[ABD$^+$a] Erdem Alkim, Joppe W. Bos, Leo Ducas, Patrick Longa, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Chris Peikert, Ananth Raghunathan, and Douglas Stebila. FrodoKEM Learning with Errors Key Encapsulation: Algorithm Specifications and Supporting Documentation (September 30, 2020). *Submission to the NIST post-quantum project.*

[ABD$^+$b] Roberto Avanzi, Joppe Bos, Leo Ducas, Eike Kiltz, Tancrede Lepoint, Vadim Lyubashevsky, John Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Kyber (version 3.0) - Algorithm Specifications And Supporting Documentation (October 1, 2020). *Submission to the NIST post-quantum project.*

[ACLZ20] Dorian Amiet, Andreas Curiger, Lukas Leuenberger, and Paul Zbinden. Defeating newhope with a single trace. In *International Conference on Post-Quantum Cryptography*, pages 189–205. Springer, 2020.

[AH21] Daniel Apon and James Howe. Attacks on NIST PQC 3rd Round Candidates, 2021. Invited talk at Real World Crypto 2021, `https://iacr.org/submit/files/slides/2021/rwc/rwc2021/22/slides.pdf`.

[BBC$^+$] Daniel J Bernstein, Billy Bob Brumley, Ming-Shing Chen, Chitchanok Chuengsatiansup, Tanja Lange, , Adrian Marotzke, Bo-Yuan Peng, Nicola Tuveri, Christine van Vredendaal, and Bo-Yin Yang. NTRU Prime: round 3 (October 7, 2020). *Submission to the NIST post-quantum project.*

[BBF$^+$] Hayo Baan, Sauvik Bhattacharya, Scott Fluhrer, Oscar Garcia-Morchon Garcia-Morchon, Thijs Laarhoven, Rachel Player, Ronald Rietman, Markku-Juhani O. Saarinen, , Ludo Tolhuizen, Jos'e Luis Torre-Arce, and Zhenfei Zhang. Round5 : Algorithm Specifications And Supporting Documentation (10th April, 2020). *Submission to the NIST post-quantum project.*

[BCLvV17] Daniel J Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. Ntru prime: reducing attack surface at low cost. In *International Conference on Selected Areas in Cryptography*, pages 235–260. Springer, 2017.

[BDH$^+$21] Shivam Bhasin, Jan-Pieter D'Anvers, Daniel Heinz, Thomas Pöppelmann, and Michiel Van Beirendonck. Attacking and defending masked polynomial comparison for lattice-based cryptography. 2021.

[BDHD$^+$19] Ciprian Băetu, F Betül Durak, Loïs Huguenin-Dumittan, Abdullah Talayhan, and Serge Vaudenay. Misuse attacks on post-quantum cryptosystems. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 747–776. Springer, 2019.

[BGRR19] Aurélie Bauer, Henri Gilbert, Guénaël Renault, and Mélissa Rossi. Assessment of the key-reuse resilience of newhope. In *Cryptographers' Track at the RSA Conference*, pages 272–292. Springer, 2019.

[BPR] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 719–737. Springer.

[CDH$^+$] Cong Chen, Oussama Danba, Jeffrey Hoffstein, Andreas Hülsing, Joost Rijneveld, John M Schanck, Peter Schwabe, William Whyte, and Zhenfei Zhang. NTRU: Algorithm Specifications And Supporting Documentation (March 20, 2019). *Submission to the NIST post-quantum project.*

[CS97] Don Coppersmith and Adi Shamir. Lattice attacks on NTRU. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 52–61. Springer, 1997.

[DCQ19] Jintai Ding, Chi Cheng, and Yue Qin. A simple key reuse attack on LWE and Ring LWE encryption schemes as key encapsulation mechanisms (KEMs). *IACR ePrint Archive*, page 271, 2019.

[DKSRV] Jan-Pieter D'Anvers, Angshuman Karmakar, Sujoy Sinha Roy, and Frederik Vercauteren. SABER: Mod-LWR based KEM (Round 3 Submission). *Submission to the NIST post-quantum project.*

[DTVV19] Jan-Pieter D'Anvers, Marcel Tiepelt, Frederik Vercauteren, and Ingrid Verbauwhede. Timing attacks on error correcting codes in post-quantum schemes. In *Proceedings of ACM Workshop on Theory of Implementation Security Workshop*, pages 2–9. ACM, 2019.

[Flu16] Scott R Fluhrer. Cryptanalysis of ring-LWE based key exchange with key share reuse. *IACR ePrint Archive*, 2016.

[FO99] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In *Annual International Cryptology Conference*, pages 537–554. Springer, 1999.

[GGJR$^+$11] Benjamin Jun Gilbert Goodwill, Josh Jaffe, Pankaj Rohatgi, et al. A testing methodology for side-channel resistance validation. In *NIST non-invasive attack testing workshop*, volume 7, pages 115–136, 2011.

[GJN20] Qian Guo, Thomas Johansson, and Alexander Nilsson. A key-recovery timing attack on post-quantum primitives using the fujisaki-okamoto transformation and its application on frodokem. In *Annual International Cryptology Conference*, pages 359–386. Springer, 2020.

[GLRP06] Benedikt Gierlichs, Kerstin Lemke-Rust, and Christof Paar. Templates vs. stochastic methods. In *International Workshop on Cryptographic Hardware and Embedded Systems*, pages 15–29. Springer, 2006.

[HCY20] Wei-Lun Huang, Jiun-Peng Chen, and Bo-Yin Yang. Power analysis on ntru prime. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 123–151, 2020.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph Silverman. NTRU: A ring-based public key cryptosystem. *Algorithmic number theory*, pages 267–288, 1998.

[JJ00] Éliane Jaulmes and Antoine Joux. A chosen-ciphertext attack against ntru. In *Annual International Cryptology Conference*, pages 20–35. Springer, 2000.

[KEF20] Paul Kirchner, Thomas Espitau, and Pierre-Alain Fouque. Fast reduction of algebraic lattices over cyclotomic fields. In Daniele Micciancio and Thomas Ristenpart, editors, *Advances in Cryptology – CRYPTO 2020*, pages 155–185. Springer, 2020.

[KRSS] Matthias J. Kannwischer, Joost Rijneveld, Peter Schwabe, and Ko Stoffelen. PQM4: Post-quantum crypto library for the ARM Cortex-M4. `https://github.com/mupq/pqm4`.

[LPR10] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. On ideal lattices and learning with errors over rings. In *EUROCRYPT*, pages 1–23, 2010.

[LSCH10] Mun-Kyu Lee, Jeong Eun Song, Dooho Choi, and Dong-Guk Han. Countermeasures against power analysis attacks for the ntru public key cryptosystem. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 93(1):153–163, 2010.

[NDGJ21] Kalle Ngo, Elena Dubrova, Qian Guo, and Thomas Johansson. A side-channel attack on a masked ind-cca secure saber kem. *IACR ePrint Archive*, 2021.

[QCD19] Yue Qin, Chi Cheng, and Jintai Ding. A complete and optimized key mismatch attack on NIST candidate NewHope. *IACR ePrint Archive*, page 435, 2019.

[RBRC20] Prasanna Ravi, Shivam Bhasin, Sujoy Sinha Roy, and Anupam Chattopadhyay. On exploiting message leakage in (few) nist pqc candidates for practical message recovery and key recovery attacks. *IACR Cryptol. ePrint Arch.*, 2020:1559, 2020.

[Reg09] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[RJJ$^+$] Prasanna Ravi, Bernhard Jungk, Dirmanto Jap, Zakaria Najm, and Shivam Bhasin. Feature Selection Methods for Non-Profiled Side-Channel Attacks on ECC. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pages 1–5. IEEE.

[RRCB20] Prasanna Ravi, Sujoy Sinha Roy, Anupam Chattopadhyay, and Shivam Bhasin. Generic side-channel attacks on cca-secure lattice-based pke and kems. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 307–335, 2020.

[SKL$^+$20] Bo-Yeon Sim, Jihoon Kwon, Joohee Lee, Il-Ju Kim, Tae-Ho Lee, Jaeseung Han, Hyojin Yoon, Jihoon Cho, and Dong-Guk Han. Single-trace attacks on message encoding in lattice-based kems. *IEEE Access*, 8:183175–183191, 2020.

[SMS19] Thomas Schamberger, Oliver Mischke, and Johanna Sepulveda. Practical evaluation of masking for ntruencrypt on arm cortex-m4. In *International Workshop on Constructive Side-Channel Analysis and Secure Design*. Springer, 2019.

[WHGH$^+$08] William Whyte, Nick Howgrave-Graham, Jeffrey Hoffstein, Jill Pipher, Joseph H Silverman, and Philip S Hirschhorn. IEEE P1363. 1 Draft 10: Draft standard for public key cryptographic techniques based on hard problems over lattices. *IACR EPrint Archive*, page 361, 2008.

[WZW13] An Wang, Xuexin Zheng, and Zongyue Wang. Power analysis attacks and countermeasures on ntru-based wireless body area networks. *KSII Transactions on Internet and Information Systems (TIIS)*, 7(5):1094–1107, 2013.

[XPRO20] Zhuang Xu, Owen Pemberton, Sujoy Sinha Roy, and David Oswald. Magnifying side-channel leakage of lattice-based cryptosystems with chosen ciphertexts: The case study of kyber. Technical report, Cryptology ePrint Archive, Report 2020/912, 2020, 2020.

A. Lattice Preliminaries

A.1. Notation

We denote by $\mathbb{Z}/q\mathbb{Z}$ the ring of integers modulo a prime q, zero-centered in the range $(-q/2, q/2] \cap \mathbb{Z}$. Given a prime number p, let R and R_q denote $\mathbb{Z}[x]/(x^p - x - 1)$ and $(\mathbb{Z}/q\mathbb{Z})[x]/(x^p - x - 1)$. Polynomials in R or R_q are written in bold lower case letters. The i^{th} coefficient of a polynomial $\mathbf{a} \in R_q$ is denoted by $\mathbf{a}[i]$. Multiplication of two polynomials $\mathbf{a}$ and $\mathbf{b}$ in R_q is denoted by $\mathbf{c} = \mathbf{a} \cdot \mathbf{b}$. A polynomial is *small* if it has coefficients in $(\mathbb{Z}/3\mathbb{Z})$, that is, in the set $\{-1, 0, 1\}$. A polynomial is said to be of *weight-w* if exactly w of its coefficients are non-zero. Any polynomial which is both *small* as well as *weight-w* polynomial is said to be a *short* polynomial. We also denote the space of all *short* polynomials by R_{sh}.

An element $\mathbf{x} \in R_q$ which is sampled from the distribution $\mathcal{D}$ with standard deviation σ is denoted by $\mathbf{x} \leftarrow \mathcal{D}(R_q)$. Byte arrays of length n are written as $\mathcal{B}^n$. The i^{th} bit in an element $x \in \mathbb{Z}_q$ is denoted by x_i. The acquisition of a side-channel trace t corresponding to a particular operation $\mathcal{X}$ on an input p is denoted by $t \Longleftarrow \mathcal{X}(p)$.

A.2. NTRU One-Way Function

Hoffstein, Pipher, and Silverman in 1998 [HPS98] proposed the NTRU cryptosystem based on the *N^{th} order Truncated Polynomial Ring Unit*. NTRU was the first practical lattice-based public key encryption scheme. It has a compact public key encryption scheme whose security is derived from a mixing system, based on polynomial algebra, but does not have any provable security guarantee. It relies only on a conjectured circular security assumption, better known as the *NTRU assumption* or the *NTRU OWF*, involving factorization of polynomials in polynomial rings [HPS98]. The original NTRU problem can be formally defined as follows.

Given $R_{NTRU} := \mathbb{Z}_q[x]/(x^N - 1)$, a small invertible polynomial $\mathbf{p} \leftarrow \mathcal{D}_\sigma(R_{NTRU})$ and another small polynomial $\mathbf{g} \leftarrow \mathcal{D}_\sigma(R_{NTRU})$, distinguish between structured samples $\mathbf{g} \cdot \mathbf{p}^{-1} \in R_{NTRU}$ from uniformly random samples in $\mathcal{U}(R_{NTRU})$.

This problem was shown to be reducible to a *shortest vector problem* (SVP) over a special class of lattices known as the *NTRU lattices* [CS97]. It is still not known if the SVP over the NTRU lattices is as hard as the SVP over general lattices. Notwithstanding, the NTRU problem gained a lot of traction since its publication in 1998. Over the years, several variants of the NTRU PKE have been proposed. One of these variants become a part of the IEEE standards under the specifications for lattice-based public-key cryptography (IEEE 1363.1-2008) [WHGH$^+$08]. It is worth noting that the NTRU cryptosystem has survived cryptanalysis for almost 24 years now. This instills a lot of confidence in its security claims despite the lack of provable security guarantee that schemes based on the LWE/LWR problem have. Two candidate PKE/KEMs in the NIST PQC standardization process - NTRU [CDH$^+$] and NTRU Prime [BBC$^+$] are based on the paradigm of the NTRU cryptosystem. NTRU is a main finalist candidate while NTRU Prime is an alternate finalist candidate.

A.3. Streamlined NTRU Prime

NTRU Prime is a suite of two IND-CCA secure KEMs, Streamlined NTRU Prime and NTRU LPRime. The former is based on the NTRU paradigm, while the latter is built upon the LPR PKE scheme and is based on the Ring-LWE problem. We focus on the Streamlined NTRU Prime, which we refer to as NTRU Prime throughout this paper for brevity. In its core is an IND-CPA secure NTRU-like PKE, which is perfectly correct, that is, without any decryption failure. It achieves IND-CCA security by using the FO transform. One of the main features of NTRU Prime is its use of a non-cyclotomic field. Most other lattice-based schemes such as Kyber [ABD$^+$b], Saber [DKSRV], NewHope [AAB$^+$], and Round5 [BBF$^+$] operate in polynomial rings with a cyclotomic structure. The choice of a non-cyclotomic field was mainly motivated by a few recent attacks that appear to gain significant speedup precisely by exploiting the cyclotomic structure [BCLvV17] as well as serious risks of potential advances in attacks exploiting cyclotomic structure in lattice-based schemes [KEF20]. In the following, we provide a brief description of the IND-CPA secure NTRU Prime PKE core, followed by a description of the IND-CCA secure NTRU Prime KEM.

A.4. Streamlined NTRU Prime PKE Core

The Streamlined NTRU Prime core is defined by three parameters (p, q, w), where p and q are prime numbers and w is a positive integer with the following restrictions: $2p \geq 3w$, $q \geq 16w + 1$ and $x^p - x - 1$ is irreducible in $(\mathbb{Z}/q\mathbb{Z})[x]$. NTRU Prime operates in the field $(\mathbb{Z}/q\mathbb{Z})[x]/(x^p - x - 1)$. The procedure $\mathsf{GenSmall}()$ takes in a small seed $\rho \in \mathcal{B}^*$ and uniformly samples for small polynomials in R_3, while $\mathsf{GenShort}$ also takes in a small seed $\rho \in \mathcal{B}^*$ and uniformly samples for short polynomials in R_{sh}. The procedure Round rounds every coefficient of a given polynomial to its nearest multiple of 3. Algorithm 1 describes the key-generation, encryption and decryption procedures.

The key generation procedure $\mathsf{PKE.KeyGen}$ produces a quotient-form NTRU instance $\mathbf{h} = \mathbf{g}/(3\mathbf{f}) \in R_q$, where $\mathbf{g} \in R_3$ and $\mathbf{f} \in R_{sh}$. Thus, $\mathbf{f}$ and $\mathbf{g}$ (or, equivalently, $\hat{\mathbf{g}} = \mathbf{g}^{-1} \in R_3$) form the secret key, while $\mathbf{h} \in R_q$ forms the public key. The hardness of retrieving the secret key from the public key comes directly from the hardness of inverting the NTRU-OWF. The encryption procedure $\mathsf{PKE.Encrypt}$ takes as input a random short polynomial $\mathbf{r} \in R_{sh}$ and generates a product-form NTRU instance $\mathbf{c} = \mathsf{Round}(\mathbf{r} \cdot \mathbf{h})$, which can also be written as $\mathbf{r} \cdot \mathbf{h} + \mathbf{m}$ with $\mathbf{m} \in R_3$ being the error introduced due to the rounding function.

The decryption procedure $\mathsf{PKE.Decrypt}$ retrieves $\mathbf{r}$ in the following manner. First, a component $\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c} \in R_q$ is computed and each coefficient of $\mathbf{a}$ is zero-centered in the range $(-q/2, q/2]$. The

Algorithm 1: Streamlined NTRU Prime PKE Core

```
1  Procedure PKE.KeyGen()
2      while g is invertible in $R_3$ do
3          $\rho \leftarrow \mathcal{U}(\mathcal{B}^*)$ $\mathbf{g} \leftarrow \mathsf{GenSmall}(\rho) \in R$
4      end
5      $\hat{\mathbf{g}} = 1/\mathbf{g} \in R_3$
6      $\rho \leftarrow \mathcal{U}(\mathcal{B}^*)$ $\mathbf{f} \leftarrow \mathsf{GenShort}(\rho) \in R_{\text{sh}}$
7      $\mathbf{h} = \mathbf{g}/(3\mathbf{f}) \in R_q$
8      return $(pk = (\mathbf{h}), sk = (\hat{\mathbf{g}}, \mathbf{f}))$
9  ________________________________________________
1  Procedure PKE.Encrypt($pk, \mathbf{r} \in R_{sh}$)
2      $\mathbf{d} = \mathbf{h} \cdot \mathbf{r} \in R_q$
3      $\mathbf{c} = \mathsf{Round}(d) \in R_q$
4      $ct = \mathsf{Encode}(\mathbf{c})$ return $(ct)$
5  ________________________________________________
1  Procedure PKE.Decrypt($ct, sk$)
2      $\mathbf{c} = \mathsf{Decode}(ct) \in R_q$
3      $\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c} \in R_q$
4      $\mathbf{e} = \mathbf{a} \bmod R_3$
5      $\mathbf{b}' = \mathbf{e} \cdot \hat{\mathbf{g}} \in R_3$
6      if $\mathsf{Weight}(\mathbf{b}') = w$ then
7          return $\mathbf{r}' = \mathbf{b}'$
8      end
9      else
10         return $\mathbf{r}' = (1, 1, \ldots, 1, 0, 0, \ldots, 0) \in R_{\text{sh}}$
11     end
```

computation of $\mathbf{a} \in R_q$ proceeds as follows:

$$\begin{aligned}\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c} \bmod q = 3\mathbf{f} \cdot (\mathbf{r} \cdot \mathbf{h} + \mathbf{m}) \bmod q = 3\mathbf{f} \cdot \mathbf{r} \cdot \mathbf{h} + 3\mathbf{f} \cdot \mathbf{m} \bmod q \\ = 3\mathbf{f} \cdot \mathbf{r} \cdot \mathbf{g}/3\mathbf{f} + 3\mathbf{f} \cdot \mathbf{m} \bmod q = 3\mathbf{f} \cdot \mathbf{m} + \mathbf{g} \cdot \mathbf{r} \bmod q.\end{aligned} \quad (1)$$

The parameters (p, q, w) are chosen to ensure that the true (non-reduced) value of every coefficient $\mathbf{a}[i]$ for $i \in [0, p-1]$ always lies in $(-q/2, q/2]$. Thus, the reduced zero-centered value of $\mathbf{a} = 3\mathbf{f}\cdot\mathbf{m}+\mathbf{g}\cdot\mathbf{r}$ modulo q is nothing but its true value. This requirement of arriving at the true value of $\mathbf{a}$ in line 4 is key to the correctness of the decryption procedure. When $\mathbf{a} \in R_q$ is reduced modulo 3, the $3\mathbf{f} \cdot \mathbf{m}$ component of $\mathbf{a}$ gets cancelled out, yielding $\mathbf{e} = \mathbf{g} \cdot \mathbf{r} \in R_3$. Then, the product $\mathbf{e} \cdot \hat{\mathbf{g}} \in R_3$ is computed. The result is $\mathbf{r}'$, which is always equal to $\mathbf{r}$ for a valid ciphertext. If $\mathbf{r}' \in R_{sh}$, then $\mathbf{r}'$ is the valid decryption output, otherwise a fixed random value of $(1, 1, \dots, 1, 0, 0, \dots, 0) \in R_3$ is the decryption output. The complete computation of $\mathbf{r}'$ in the decryption goes as follows:

$$\begin{aligned}\mathbf{r}' = ((3\mathbf{f} \cdot \mathbf{c} \in R_q) \bmod 3) \cdot \hat{\mathbf{g}} \in R_3 \\ = ((3\mathbf{f} \cdot \mathbf{m} + \mathbf{g} \cdot \mathbf{r} \in R_q) \bmod 3) \cdot \hat{\mathbf{g}} \in R_3 = \mathbf{g} \cdot \mathbf{r} \cdot \hat{\mathbf{g}} \in R_3 = \mathbf{r}.\end{aligned} \quad (2)$$

A.5. Streamlined NTRU Prime KEM

The Streamlined NTRU Prime PKE core is IND-CPA secure. The FO transform converts it into an IND-CCA secure KEM. The transform instantiates the PKE.Encrypt, PKE.Decrypt, and hash functions in the IND-CCA secure encapsulation and decapsulation procedures. Algorithm 2 provides the detail. In theory, the FO transform helps protect KEMs against chosen-ciphertext attacks since the validity of ciphertexts are checked through the re-encryption procedure during decapsulation in the if loop in KEM.Decaps. Thus, the attacker only sees decapsulation failures for invalid ciphertexts, with very high probability. Moreover, the decryption procedure is encapsulated within the decapsulation procedure, preventing the attacker from observing the output of the decryption module directly. This provides strong theoretical security guarantees against chosen-ciphertext attacks which exploit information about the decryption output in IND-CPA secure PKE/KEMs.

In this work, we show that side-channel information can be used to instantiate different types of oracles which provide varying degree of information about the decryption output, resulting in secret key recovery.

A.6. Test Vector Leakage Assessment (TVLA)

The Test Vector Leakage Assessment (TVLA) [GGJR+11] is a popular conformance-based methodology in side-channel analysis. It has been widely used in both academia and the industry to perform side-channel evaluation of cryptographic implementations. TVLA involves the computation of the univariate Welch's t-test over two given sets of side-channel measurements to identify their differentiating features. By testing for a null hypothesis that the mean of the two sets is identical, a *PASS/FAIL* decision is made. The TVLA formulation over two sets of measurements $\mathcal{T}_r$ and $\mathcal{T}_f$ is given by:

$$TVLA := \frac{\mu_r - \mu_f}{\sqrt{\frac{\sigma_r^2}{m_r} + \frac{\sigma_f^2}{m_f}}} \quad , \quad (3)$$

where μ_r, σ_r and m_r (resp. μ_r, σ_r and m_r) are the mean, standard deviation and cardinality of the trace set $\mathcal{T}_r$ (resp. $\mathcal{T}_f$).

Algorithm 2: FO transform of a IND-CPA secure PKE into IND-CCA secure KEM

```
Procedure KEM.Encaps($pk$)
    $\rho \leftarrow \mathcal{U}(\mathcal{B}^*)$
    $\mathbf{r} = \mathsf{GenShort}(\rho) \in R_{\text{sh}}$
    $c = \mathsf{PKE.Encrypt}(pk, \mathbf{r})$
    $d = \mathcal{H}(\mathbf{r}, pk)$
    $ct = (c, d)$ $K = \mathcal{G}(1, \mathbf{r}, ct)$
    return $ct, K$
________________________________________________________________
Procedure KEM.Decaps($sk, pk, ct$)
    $ct = (c, d)$ $\mathbf{r}' = \mathsf{PKE.Decrypt}(sk, c)$
    $d' = \mathcal{H}(\mathbf{r}', pk)$
    $c' = \mathsf{PKE.Encrypt}(pk, \mathbf{r}')$
    $ct' = (c', d')$
    if $ct' = ct$ then
        return $K = \mathcal{G}(1, \mathbf{r}', ct')$
    end
    else
        return $K = \mathcal{G}(1, \rho', ct')$ /* $\rho' \in \mathcal{B}^{32}$ is a random secret */
    end
```

The null hypothesis is rejected with a confidence of 99.9999% only if the absolute value of the t-test score is greater than 4.5 [GGJR$^+$11]. A rejected null hypothesis implies that the two trace/data sets are different and might leak some side-channel information and, hence, is considered a *FAIL* test. While TVLA is mainly used as a metric for side-channel evaluation, it has also been used as a tool for feature selection in multiple cryptanalytic efforts [RJJ$^+$]. Here we use TVLA as a tool for *feature selection* from side-channel measurements [GLRP06].

B. Plaintext-Checking Oracle-based SCA

This section provides a detailed description of the PC oracle attack on Streamlined NTRU Prime KEM. The reader is referred to Subsection 3.1 for high level details about the attack. Our PC oracle attack works by constructing malicious ciphertexts and, subsequently, by utilizing side-channel information from the decryption of these malicious ciphertexts to perform key recovery. It falls broadly into two main phases.

1. In the first phase, we search for a ciphertext which when decrypted, leads to what we refer to as a *single collision* event. We query the decapsulation device with several specially crafted ciphertexts and analyze their side-channel leakage in order to detect the single collision event. A chosen ciphertext that produces a single collision event is denoted as the base ciphertext $\mathbf{c}_{\text{base}}$. This ciphertext provides crucial information about the secret polynomials $\mathbf{f}$ and $\mathbf{g}$ and is also used as an importance component for key recovery in the subsequent attack phase.

2. In the attack phase, we use the base ciphertext $\mathbf{c}_{\text{base}}$ to construct new attack ciphertexts. Upon decryption, their corresponding internal variable $\mathbf{e}$ (4 in $\mathsf{PKE.Decrypt}$ of Alg.1) can only belong to two exclusive classes: $\mathbf{e} = 0$ or $\mathbf{e} \neq 0$ with a single non-zero coefficient. Moreover, the value of $\mathbf{e}$ depends on a targeted portion of the secret key. We exploit side-channel leakage from operations manipulating $\mathbf{e}$ to obtain information about the value of $\mathbf{e}$, thereby realizing

a practical PC oracle. The oracle's responses ($\mathbf{e} = 0$ or $\mathbf{e} \neq 0$) obtained for several attack ciphertexts is used to recover the full secret key.

In the following, we describe the first phase of our PC oracle-based attack which involves retrieving the base ciphertext $\mathbf{c}_{\text{base}}$.

B.1. Retrieving the Base Ciphertext $\mathbf{c}_{\text{base}}$

Our construction of chosen ciphertexts for Streamlined NTRU Prime KEM is inspired by the chosen-ciphertext attack proposed by Jaulmes and Joux on the classical IND-CPA secure NTRU PKE scheme in Crypto 2000 [JJ00]. We begin with an intuition for the approach before proposing a concrete methodology. The notations are from Algorithm 1.

B.1.1. Intuition

We first analyze the effect of decrypting the ciphertext $\mathbf{c} = k + k \cdot \mathbf{h}$, where $k \in \mathbb{Z}^+$, by looking at the computation of $\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c}$ in line 3 of PKE.Decrypt:

$$\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c} = k \cdot 3\mathbf{f} + k \cdot \mathbf{h} \cdot 3\mathbf{f} = 3k \cdot \mathbf{f} + k \cdot \mathbf{g}/3\mathbf{f} \cdot 3\mathbf{f} = 3k \cdot \mathbf{f} + k \cdot \mathbf{g}. \tag{4}$$

The coefficients of both $\mathbf{f}$ and $\mathbf{g}$ are in $[-1, 0, 1]$. Thus, the absolute maximum value of a coefficient, say $\mathbf{a}[i]$, is obtained when the corresponding coefficients $\mathbf{f}[i]$ and $\mathbf{g}[i]$ simultaneously take their maximum absolute value, that is $\mathbf{f}[i] = \mathbf{g}[i] = \pm 1$. We call the event when the corresponding coefficients of two or more polynomials attain their maximum absolute value a *collision*. Thus, $\mathbf{a}[i] = 4k$ (resp. $-4k$) when $\mathbf{f}[i] = \mathbf{g}[i] = +1$ (resp. -1). We now choose a suitable positive integer k, with $3 \mid k$, based on the conditions

$$4k > q/2 \text{ and } s \cdot k < q/2 \text{ for } s \in [0, 3]. \tag{5}$$

For the sake of explanation, let us assume that $\mathbf{f}$ and $\mathbf{g}$ only collide at the i^{th} coefficient with the value of $+1$. In that case, $\mathbf{a}$ has the coefficients

$$\mathbf{a}[j] > q/2 \text{ if } j = i \text{ and } \mathbf{a}[j] < q/2 \text{ if } j \neq i. \tag{6}$$

Since $3 \mid k$, it is clear that $3 \mid \mathbf{a}[i]$, for $i \in [0, p-1]$. But when $\mathbf{a}$ is reduced modulo q (zero-centered in $(-q/2, q/2]$)), all coefficients, except for $\mathbf{a}[i]$, retain their true value and remain a multiple of 3. This is because $\mathbf{a}[i] > q/2$ and, when reduced modulo q, is subtracted by q, which is a prime. More concretely,

$$\mathbf{a} \bmod q = \mathbf{a} - q \cdot x^i. \tag{7}$$

Subsequently, $\mathbf{e}(= \mathbf{a} \bmod 3) \in R_3$ is nothing but

$$\mathbf{e} = (-q \bmod 3) \cdot x^i. \tag{8}$$

In other words, we ensure that $\mathbf{a}[i] > q/2$ only during a collision, while $\mathbf{a}[i] < q/2$ otherwise. Thus, for a choice of k in Equation (5), $\mathbf{e}[i] \neq 0$ denotes collision at i, while all other coefficients are zero.

While the above holds for a collision with a value of $+1$, the same also applies when the collision value is -1. Subsequently, $\mathbf{a}[i] < -(q/2)$ and, thus, when q is added to $\mathbf{a}[i]$ to zero-center it in the range $[-q/2, q/2]$, the corresponding $\mathbf{e}[i] \neq 0$, This implies a collision at i. Henceforth, to avoid repetitions, we focus only on collision with the highest positive value ($+1$ in this case). The same analysis holds for the lowest negative value (-1 in this case).

In our attack, it would be ideal to have a *single collision* between $\mathbf{f}$ and $\mathbf{g}$, resulting in $\mathbf{e}$ having a single nonzero coefficient. For illustration, we use one particular parameter set of Streamlined NTRU Prime, namely sntrup761. It is defined by $(p, q, w) = (761, 4591, 286)$. We analyze the probability of a single collision, denoted by $pcoll_1$, between $\mathbf{f}$ and $\mathbf{g}$ for sntrup761. We denote by p' the probability of a collision at any given coefficient and by

$$pr' := pr_1 + pr_{-1}, \text{ where } pr_x(x = \pm 1),$$

the probability of a collision between $\mathbf{f}$ and $\mathbf{g}$ with a matching coefficient of ± 1. For $\mathbf{f} \in R_{\text{sh}}$ and $\mathbf{g} \in R_3$, we calculate $pr' := (w/3p)$ and, hence, for the parameter set sntrup761, we approximate pr' to 0.125. The probability of a single collision between $\mathbf{f}$ and $\mathbf{g}$ is

$$pcoll_1 = p * (pr' \cdot (1 - pr')^{p-1}).$$

In sntrup761 this value is impractically low at $8 \cdot 10^{-43}$. We require better choices for the ciphertexts to limit the number of collisions and, thus, the nonzero coefficients in $\mathbf{e}$.

B.1.2. Constructing Ciphertexts for Single Collision

We split the value of $\mathbf{a}$ in Equation (4) into

$$\mathbf{a} = 3k \cdot \mathbf{f} + k \cdot \mathbf{g} = 3k \cdot \mathbf{t}_1 + k \cdot \mathbf{t}_2, \tag{9}$$

where $\mathbf{t}_1 = \mathbf{f}$ and $\mathbf{t}_2 = \mathbf{g}$. To limit the number of collisions between $\mathbf{t}_1$ and $\mathbf{t}_2$ we make a generic choice for $\mathbf{c}$. This choice is

$$\mathbf{c} = k_1 \cdot (x^{i1} + x^{i2} + \ldots + x^{im}) + k_2 \cdot (x^{j1} + x^{j2} + \ldots + x^{jn}) \cdot \mathbf{h} = k_1 \cdot \mathbf{d}_1 + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h}, \tag{10}$$

where both $\mathbf{d}_1$ and $\mathbf{d}_2$ are polynomials with, respectively, m and n number of nonzero coefficients $(+1)$. The corresponding $\mathbf{a} = (3\mathbf{f} \cdot \mathbf{c})$ is given by

$$\mathbf{a} = k_1 \cdot \mathbf{d}_1 \cdot 3\mathbf{f} + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h} \cdot 3\mathbf{f} = 3k_1 \cdot \mathbf{d}_1 \cdot \mathbf{f} + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{g} = 3k_1 \cdot (\mathbf{t}_1) + k_2 \cdot (\mathbf{t}_2), \tag{11}$$

where $\mathbf{t}_1 = (\mathbf{d}_1 \cdot \mathbf{f})$ and $\mathbf{t}_2 = (\mathbf{d}_2 \cdot \mathbf{g})$. We note that the product of a polynomial $\mathbf{d}$ with x modulo $(x^p - x - 1)$ is

$$(\mathbf{d} \cdot x) \bmod (x^p - x - 1) = \mathbf{d}[p-1] + (\mathbf{d}[0] + \mathbf{d}[p-1])x + \mathbf{d}[1]x^2 + \ldots + \mathbf{d}[p-2]x^{p-1}, \tag{12}$$

with coefficients in $\{-2, -1, 0, 1, 2\}$. Multiplying by x rotates $\mathbf{d}$ by one position to the left and adds $\mathbf{d}[p-1]$ to the coefficients $d[0]$ of x^0 and $d[1]$ of x^1. We denote the resulting product by $\mathsf{Rotp}(\mathbf{d}, 1)$ and refer to it informally as the rotated variant of $\mathbf{d}$ by 1 degree. More generally, the product $(\mathbf{d} \cdot x^i)$ modulo $(x^p - x - 1)$, denoted by $\mathsf{Rotp}(\mathbf{d}, i)$, is given by

$$\begin{aligned}(\mathbf{d} \cdot x^i) \bmod (x^p - x - 1) = {} & \mathbf{d}_{p-i} + (\mathbf{d}_{p-i} + \mathbf{d}_{p-i-1})\, x + \ldots \\ & + (\mathbf{d}_{p-1} + \mathbf{d}_0)x^i + \mathbf{d}_1 x^{i+1} + \ldots + \mathbf{d}_{p-i-1}x^{p-1},\end{aligned} \tag{13}$$

with all coefficients in $\{-2, -1, 0, 1, 2\}$. Thus, $\mathbf{t}_1 = \mathbf{d}_1 \cdot \mathbf{f}$ is

$$\begin{aligned}\mathbf{t}_1 &= \mathbf{d}_1 \cdot \mathbf{f} = (x^{i1} + x^{i2} + \ldots + x^{im}) \cdot \mathbf{f} \\ &= \mathbf{f} \cdot x^{i1} + \mathbf{f} \cdot x^{i2} + \ldots + \mathbf{f} \cdot x^{im} \\ &= \mathsf{Rotp}(\mathbf{f}, i1) + \mathsf{Rotp}(\mathbf{f}, i2) + \ldots + \mathsf{Rotp}(\mathbf{f}, im),\end{aligned} \tag{14}$$

which is precisely the sum of rotations of $\mathbf{f}$ by varying degrees, governed by $\{i1, i2, \ldots, im\}$. Similarly, $\mathbf{t}_2$ is the sum of rotations of $\mathbf{g}$ by the degrees in $\{j1, j2, \ldots, jn\}$. A collision occurs at position i only if all the corresponding coefficients of $\mathsf{Rotp}(\mathbf{f}, u)$, for $u \in \{i1, i2, \ldots, im\}$, and $\mathsf{Rotp}(\mathbf{g}, v)$, for $v \in \{j1, j2, \ldots, jn\}$, are either $+2$ or -2. We observe that the probability of collisions quickly degrades as (m, n) increase.

For the choice of $\mathbf{c}$ in Equation (10), the maximum possible value for the corresponding coefficient of $\mathbf{a}$ in Equation (11) is $(3k_1 \cdot 2m + k_2 \cdot 2n)$, which is obtained upon a collision. We therefore choose values for (k_1, k_2) that satisfy three conditions:

$$3 \mid k_1, \quad 3 \mid k_2, \quad 3k_1 \cdot r + k_2 \cdot s \begin{cases} > q/2, & \text{if } r = 2m, s = 2n, \\ < q/2, & \text{otherwise,} \end{cases} \tag{15}$$

with $0 \le r \le 2m$ and $0 \le s \le 2n$. In other words, we choose (k_1, k_2) such that $\mathbf{a}[i] > q/2$ only when there is a collision at i, while $\mathbf{a}[i] < q/2$ otherwise. This ensures $\mathbf{e}[i] \neq 0$ when there is a collision at i and $\mathbf{e}[i] = 0$ otherwise.

Summarizing the above discussion, we carefully select values for (m, n) and (k_1, k_2) to form our chosen ciphertexts in Equation (10). The choice for (m, n) must be such that there is at most a single collision with very high probability. Given (m, n), we then take (k_1, k_2) which satisfies the conditions in Equation (15) such that $\mathbf{e}[i] \neq 0$ indicates a collision at the i^{th} coefficient. We also note that concrete values for both (m, n) and (k_1, k_2) can be chosen and fixed for a given parameter set of Streamlined NTRU Prime.

B.1.3. Additional Challenge: Use of Rounded Ciphertexts

Streamlined NTRU Prime uses specially structured rounded ciphertexts where all coefficients are multiples of 3. Rounding is used to deterministically generate the message component $\mathbf{m} \in R_3$. While the encryption procedures in NTRU-like PKE schemes typically take $\mathbf{m}$ as an explicit input and add it to the product $\mathbf{h} \cdot \mathbf{r} \in R_q$, Streamlined NTRU Prime simply rounds all the coefficients of $\mathbf{h} \cdot \mathbf{r}$ to their nearest multiple of 3. The *rounding noise* introduced in the product $(\mathbf{h} \cdot \mathbf{r})$ *acts as the implicit message* $\mathbf{m} \in R_3$.

Since 3 exactly divides all the coefficients of a valid ciphertext, the scheme proposes to send only the quotient of each coefficient upon division by 3, rather than the actual value. Doing so reduces the ciphertext size. Thus, every coefficient of the received ciphertext is multiplied by 3 before being used as the input to the decryption procedure. We note that the coefficients of our chosen ciphertexts according to Equation (10) are not exact multiples of 3. Our chosen ciphertexts, therefore, also need to be rounded. The rounding noise in the ciphertext is denoted by $\mathbf{m}' \in R_3$. Thus, the actual value of our chosen ciphertext used in decryption is given by

$$\begin{aligned} \mathbf{c} &= \mathsf{Round}(k_1 \cdot (x^{i1} + x^{i2} + \ldots + x^{im}) + k_2 \cdot (x^{j1} + x^{j2} + \ldots + x^{jn}) \cdot \mathbf{h}) \\ &= k_1 \cdot (x^{i1} + x^{i2} + \ldots + x^{im}) + k_2 \cdot (x^{j1} + x^{j2} + \ldots + x^{jn}) \cdot \mathbf{h} + \mathbf{m}' \\ &= (k_1 \cdot \mathbf{d}_1 + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h}) + (\mathbf{m}') \end{aligned} \tag{16}$$

The corresponding $\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c}$ is

$$\begin{aligned} \mathbf{a} &= k_1 \cdot \mathbf{d}_1 \cdot 3\mathbf{f} + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h} \cdot 3\mathbf{f} + \mathbf{m}' \cdot 3\mathbf{f} \\ &= (3k_1 \cdot \mathbf{d}_1 \cdot \mathbf{f}) + (k_2 \cdot \mathbf{d}_2 \cdot \mathbf{g}) + (3\mathbf{f} \cdot \mathbf{m}') \\ &= \mathbf{s} + \mathbf{n}, \end{aligned} \tag{17}$$

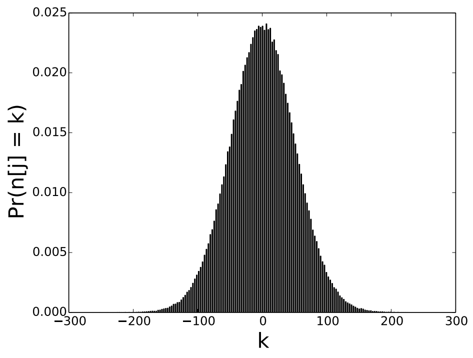

Figure 12: Distribution of coefficients of the noise $\mathbf{n} = 3\mathbf{f} \cdot \mathbf{m}'$ with mean 0 and $\sigma \approx 57$ for sntrup761 parameter set of Streamlined NTRU Prime

where $\mathbf{s} := (3k_1 \cdot \mathbf{d}_1 \cdot \mathbf{f} + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{g})$ is the signal component while $\mathbf{n} := 3\mathbf{f} \cdot \mathbf{m}'$ is the noise component. Thus, we have an additional key-dependent noise of $(3\mathbf{f} \cdot \mathbf{m}')$ in $\mathbf{a}$. But, $\mathbf{m} \in R_3$ and $\mathbf{f} \in R_{\text{sh}}$ are small polynomials, making the size of the noise $3\mathbf{f} \cdot \mathbf{m}'$ much smaller in comparison to the range q.

For sntrup761, Figure 13 shows the distribution of the coefficients $\mathbf{n}[j]$ for $j \in [0, p-1]$ of $\mathbf{n}$. It is Gaussian with mean 0 and $\sigma \approx 57$, which is much less than $q = 4591$. The noise polynomial $\mathbf{n} = 3\mathbf{f} \cdot \mathbf{m}'$ is a multiple of 3 and will get rounded to 0 when $\mathbf{a}$ is reduced modulo 3. Though $\mathbf{n}$ is small, when added to coefficients of $\mathbf{a}$ near the limit of $q/2$, the noise is capable of giving rise to a false positive or a false negative collision event. For a given choice of (m, n) and (k_1, k_2), the largest possible value of a coefficient of $\mathbf{a}$ is denoted by $m_1 := (3k_1 \cdot 2m + k_2 \cdot 2n)$. The next largest value is denoted by m_2. As stated in Equation (15), we choose values for (k_1, k_2) such that $m_1 > q/2$ and $m_2 < q/2$. Let $0 \le r \le 2m$ and $0 \le s \le 2n$. Let dm_1 (resp. dm_2) denote the distance between m_1 (resp. m_2) and $q/2$, where

$$\begin{aligned} dm_1 &= \|(3k_1 \cdot 2m + k_2 \cdot 2n) - q/2\| \\ dm_2 &= \left\| \left(\max_{\complement(r=2m, s=2n)} (3k_1 \cdot r + k_2 \cdot s) \right) - q/2 \right\|. \end{aligned} \tag{18}$$

A false positive collision occurs when $\mathbf{s}[i] = m_2$ and the corresponding $\mathbf{n}[i] > dm_2$, ensuring that $\mathbf{s}[i] + \mathbf{n}[i] > q/2$ and $\mathbf{e}[i] \neq 0$. Similarly, a false negative can occur when $\mathbf{s}[i] = m_1$ due to a valid collision. If, however, $\mathbf{n}[i] < -dm_1$, then $\mathbf{s}[i] + \mathbf{n}[i] < q/2$ and $\mathbf{e}[i] = 0$, which suppresses the collision.

Due to rounding, the noise component $\mathbf{n}$ cannot be removed. The possibility of a false positive or false negative for a collision however can be minimized by placing additional constraints in choosing the tuple (k_1, k_2). There might be multiple possible values for (k_1, k_2) which satisfy the necessary condition stated in Equation (15). We choose the tuple that maximizes the distance dm_1 (for m_1) and dm_2 (for m_2) to keep the noise coefficient $\mathbf{n}[j]$ from growing large enough to push $\mathbf{a}[j]$ to the other side of $q/2$, which is when an error occurs in the value of $\mathbf{e}$. As long as the error $\mathbf{n}[j]$ does not push $\mathbf{a}[j]$ to the other side of $q/2$, there will be no error in the value of $\mathbf{e}$. In other words, m_1 and m_2 should lie as far as possible on either side of the limit $q/2$. This additional constraint in the choice of (k_1, k_2) is simply to maximize the distance tuple (dm_1, dm_2).

Choosing Concrete Values for the Chosen-Ciphertext: We have two primary constraints. We need to select the values for (m, n) to limit the number of collisions to at most 1. For sntrup761, we empirically arrived at $(m, n) = (1, 3)$. The noise $\mathbf{n}[j]$ is normally distributed with mean 0 and $\sigma \approx 50$. Choosing $(k_1, k_2) = (102, 303)$ results in $(dm_1, dm_2) = (135, 168)$, minimizing the probability that the noise induces errors in the value of $\mathbf{e}$. We once again note that the values for (m, n) and (k_1, k_2) can be fixed and chosen beforehand for a given parameter set of Streamlined NTRU Prime.

B.1.4. Detecting Collision through Side-Channels

Given (m, n) and (k_1, k_2), we randomly select polynomials $\mathbf{d}_1$ and $\mathbf{d}_2$ in Equation (10) until we arrive at a ciphertext $\mathbf{c}$ having a single nonzero coefficient for $\mathbf{e}$. Since $\mathbf{e}$ is an internal variable, it is not possible to classically obtain information about its value. Hence, we utilize side-channel to obtain it. In particular, we want to identify $\mathbf{e} \neq 0$ through side-channel information. This leads to a classification problem with two classes, namely $\mathbf{e} = 0$ and $\mathbf{e} \neq 0$. We denote the class $\mathbf{e} = 0$ by $\mathbf{e}_\mathrm{O}$ and the class $\mathbf{e} \neq 0$ by $\mathbf{e}_\mathrm{X}$.

For $\mathbf{e}_\mathrm{O}$, $\mathbf{b}' = \mathbf{e} \cdot \hat{\mathbf{g}} = 0$ (see line 5 of $\mathsf{PKE.Decrypt}$) and, hence, $\mathsf{Weight}(\mathbf{b}') = w_{\mathbf{b}'} = 0$. However, for $\mathbf{e}_\mathrm{X}$ with a single nonzero coefficient, $\mathbf{b}' \neq 0$ with uniformly random coefficients in $\{-1, 0, 1\}$ and, hence, $w_{\mathbf{b}'} \neq 0$. Although the exact value depends on the secret polynomial $\mathbf{g}$, the average value of $w_{\mathbf{b}'} \approx 500$ for the $\mathsf{sntrup761}$ parameter set. This large difference in the weight between the two classes should be easily distinguishable through the EM side-channel. Refer Subsection 3.1.1 for complete details about the experimental setup and attack target.

Welch's *t*-test for Collision Detection: Obtaining N replicated measurements for each class $\mathbf{e}_\mathrm{O}$ and $\mathbf{e}_\mathrm{X}$ is the first step. Due to the large difference in weight, we focus on capturing EM signals from the weight calculation operation in the decryption procedure (Line 6 in $\mathsf{PKE.Decrypt}$). For $\mathbf{e}_\mathrm{O} = 0$, we simply choose an all zero ciphertext $\mathbf{c} = 0$. We repeatedly decrypt $\mathbf{c} = 0$ and obtain N replicated EM measurements. The obtained trace set, corresponding to $\mathbf{e}_\mathrm{O}$, is denoted by $\mathcal{T}_\mathrm{O}$. To test if a given ciphertext $\mathbf{c}'$ results in a collision, we obtain N replicated measurements from the decryption of $\mathbf{c}'$ and denote the obtained trace set by $\mathcal{T}_\mathrm{X}$. Let $\mathcal{T} = \mathcal{T}_\mathrm{O} \cup \mathcal{T}_\mathrm{X}$. We now perform the Welch's t-test between $\mathcal{T}_\mathrm{O}$ and $\mathcal{T}_\mathrm{X}$.

- We center (normalize) each trace $t_i \in \mathcal{T}$ by removing the mean of each trace and dividing by its standard deviation to obtain t'_i. This step is optional but can correct some environmental effects on measurements, such as DC shifts.
- We compute the Welch's t-test between the normalized traces in $\mathcal{T}_\mathrm{O}$ and $\mathcal{T}_\mathrm{X}$ (see Equation (3)). If there is at least a single peak in the t-test plot above the t-test threshold of ± 4.5, then $\mathbf{e}_\mathrm{O} \neq \mathbf{e}_\mathrm{X}$ and, hence, $\mathbf{e}_\mathrm{X} \neq 0$ with potentially a single non-zero coefficient. Otherwise, $\mathbf{e}_\mathrm{O} = \mathbf{e}_\mathrm{X} = 0$. Figure 6(a) depicts the t-test plot when $\mathbf{e} = 0$ for $\mathbf{c}'$. The plot has no peaks above the threshold. Figure 6(b) corresponds to $\mathbf{e} \neq 0$. In it one can clearly identify several peaks, well above the threshold ± 4.5.

We repeat this test for different choices of $(\mathbf{d}_1, \mathbf{d}_2)$ until we obtain one for which $\mathbf{e} \neq 0$, indicating a possible collision. There is a chance that this collision, instead of being a valid one, is a false positive. For a chosen tuple $(\mathbf{d}_1, \mathbf{d}_2)$ that corresponds to a false positive collision, however, key recovery cannot be performed correctly. Hence, we have to repeat the process until we obtain a tuple $(\mathbf{d}_1, \mathbf{d}_2)$ that corresponds to a true collision. We denote the corresponding ciphertext as $\mathbf{c}_\mathrm{base}$. It is considered to have passed the test if its t-test plot indicates $\mathbf{e} \neq 0$. We also denote the tuple $(\mathbf{d}_1, \mathbf{d}_2)$ corresponding to $\mathbf{c}_\mathrm{base}$ by $(\mathbf{d1}_\mathrm{att}, \mathbf{d2}_\mathrm{att})$. Let $\mathbf{d1}_\mathrm{att}$ and $\mathbf{d2}_\mathrm{att}$ contain m and n terms respectively.

The ciphertext $\mathbf{c}_{\text{base}}$ can then be represented as

$$\begin{aligned}\mathbf{c}_{\text{base}} &= k_1 \cdot (x^{i1} + x^{i2} + \ldots + x^{im}) + k_2 \cdot (x^{j1} + x^{j2} + \ldots + x^{jn}) \cdot \mathbf{h} \\ &= k_1 \cdot \mathbf{d1}_{\text{att}} + k_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h}. \end{aligned} \tag{19}$$

Next, we show how to use ($\mathbf{d1}_{\text{att}}$,$\mathbf{d2}_{\text{att}}$) of $\mathbf{c}_{\text{base}}$ in the attack phase to build new chosen ciphertexts for full key recovery.

B.2. PC Oracle-based Key Recovery

Attack Overview: The attack works by constructing new ciphertexts using ($\mathbf{d1}_{\text{att}}$,$\mathbf{d2}_{\text{att}}$) which collide only at i to recover single coefficients of the secret polynomial $\mathbf{f}$. These ciphertexts are constructed so that the corresponding $\mathbf{e} \in R_3$ takes only two values (1) $\mathbf{e} = 0$ and (2) $\mathbf{e} \neq 0$ with $\mathbf{e}[i] \neq 0$. The value of $\mathbf{e}$ depends on the value of a targeted coefficient of $\mathbf{f}$. This binary information obtained using side-channels over several chosen ciphertexts leads to a complete recovery of $\mathbf{f}$ one coefficient at a time.

B.2.1. Attack Methdology

We build, using ($\mathbf{d1}_{\text{att}}$,$\mathbf{d2}_{\text{att}}$), the ciphertext

$$\mathbf{c}_{\text{att}} = \ell_1 \cdot \mathbf{d1}_{\text{att}} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h} + \ell_3 = \mathbf{c}_{\text{base}} + \ell_3 \cdot x^u, \tag{20}$$

where $\ell_1, \ell_2, \ell_3 \in \mathbb{Z}^+$, $u \in [0, p-1]$ and $\mathbf{c}_{\text{base}} = \ell_1 \cdot \mathbf{d1}_{\text{att}} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h}$. Let the error introduced due to rounding be $\mathbf{m}' \in R_3$. Thus, $\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c}_{\text{att}}$ is given by

$$\begin{aligned}\mathbf{a} &= 3\mathbf{f} \cdot \mathbf{c}_{\text{att}} \\ &= 3\ell_1 \cdot \mathbf{d1}_{\text{att}} \cdot \mathbf{f} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h} \cdot 3\mathbf{f} + \ell_3 \cdot 3\mathbf{f} \cdot x^u + 3\mathbf{f} \cdot \mathbf{m}' \\ &= 3\ell_1 \cdot \mathbf{d1}_{\text{att}} \cdot \mathbf{f} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{g} + 3\ell_3 \cdot \mathbf{f} \cdot x^u + 3\mathbf{f} \cdot \mathbf{m}' \\ &= 3\ell_1 \cdot \mathbf{d1}_{\text{att}} \cdot \mathbf{f} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{g} + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u) + \mathbf{n}, \end{aligned} \tag{21}$$

where $\mathbf{n}$ is the noise term $3\mathbf{f} \cdot \mathbf{m}'$. For a given set of $(\mathbf{d1}_{\text{att}}, \mathbf{d2}_{\text{att}})$ and (ℓ_1, ℓ_2, ℓ_3), the noise polynomial $\mathbf{n}$ remains constant. For the sake of explanation, we assume that $\mathbf{d1}_{\text{att}}$ and $\mathbf{d2}_{\text{att}}$ collide at i with a value of $+2$. Thus, the coefficients of $\mathbf{a}$ can be expressed as

$$\mathbf{a}[j] = \begin{cases} 3\ell_1 \cdot 2m + \ell_2 \cdot 2n + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u)[j] + \mathbf{n}[j], \text{ if } j = i \\ 3\ell_1 \cdot r + \ell_2 \cdot s + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u)[j] + \mathbf{n}[j], \text{ if } (j \neq i), \mathsf{C}(r = 2m, s = 2n). \end{cases} \tag{22}$$

In particular, we can represent the coefficient of $\mathbf{a}$ at the colliding index i as

$$\mathbf{a}[i] = \delta + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u)[i], \tag{23}$$

where $\delta := 3\ell_1 \cdot 2m + \ell_2 \cdot 2n + \mathbf{n}[i]$ is a constant. Thus, $\mathbf{a}[i]$ is linearly dependent on $\mathsf{Rotp}(\mathbf{f}, u)[i]$.

Let β denote $\mathsf{Rotp}(\mathbf{f}, u)[i]$. Based on the rotational property of polynomial multiplication mod $(x^p - x - 1)$ (see Equation 12), we know that

$$\beta := \mathsf{Rotp}(\mathbf{f}, u)[i] = \begin{cases} \mathbf{f}[i - r], & \text{for } 0 \leq u < i, \\ \mathbf{f}[0] + \mathbf{f}[p-1], & \text{if } u = i, \\ \mathbf{f}[p - 1 + i - r] + \mathbf{f}[p + i - r], & \text{for } i < u < p. \end{cases} \tag{24}$$

By simply changing the rotation index u we can ensure dependency of $\mathbf{a}[i]$ (the colliding index i) with different coefficients of the secret polynomial $\mathbf{f}$. For a given u, the five values in $\{-2, -1, 0, 1, 2\}$ are possible candidates for β. Our task is, therefore, to select values for (ℓ_1, ℓ_2, ℓ_3) such that the occurrence of $\mathbf{a}[i] > q/2$ acts as a binary distinguisher capable of identifying every candidate for β. To distinguish $\beta = +2$, for example, we choose ℓ_1, ℓ_2, ℓ_3, each of which is an integer multiple of 3, that satisfy the condition

$$3\ell_1 \cdot r + \ell_2 \cdot s + 3\ell_3 \cdot \beta \begin{cases} > q/2, \text{ if } r = 2m, s = 2n \text{ and } \beta = 2 \\ < q/2, \text{ otherwise,} \end{cases} \tag{25}$$

with $0 \leq r \leq 2m$ and $0 \leq s \leq 2n$. This ensures that $\mathbf{a}[i] > q/2$ at the colliding index i when $\beta = +2$, while $\mathbf{a}[i] < q/2$ otherwise. For $j \neq i$, the coefficients are $\mathbf{a}[j] < q/2$, since there is no other collision than at index i. It is then certain that $\mathbf{e} \neq 0$ if and only if $\beta = +2$. Similar to identifying $\beta = 2$ using the tuple (ℓ_1, ℓ_2, ℓ_3), we can identify $\beta = -2$ by simply changing the sign of ℓ_3, that is by using $(\ell_1, \ell_2, -\ell_3)$. If, however, $\mathbf{e} = 0$ for both ciphertexts, then $\beta \in \{-1, 0, 1\}$. Let $\mathbf{O}$ denote the $\mathbf{e} = 0$ event and $\mathbf{X}$ denote the $\mathbf{e} \neq 0$ event. This binary information constitutes a distinguisher for every candidate for β.

Effect of Rounding Error: Some rounding error $\mathbf{n}$ is present on $\mathbf{a}$. Adopting a similar strategy to the one in Subsection B.1.3, we select (ℓ_1, ℓ_2, ℓ_3) that minimize the possibility of a false positive or a false negative in the collision. For distinguishing $\beta = 2$, the tuple must satisfies Equation (25). At the colliding index when $\beta = 2$, the largest possible coefficient of $\mathbf{a}$ is $m_1 := 3\ell_1 \cdot 2m + \ell_2 \cdot 2n + 3\ell_3 \cdot 2 > q/2$. Let the second largest value be $m_2 < q/2$ and the distance between m_1 (resp. m_2) and $q/2$ be dm_1 (resp. dm_2). The values for (ℓ_1, ℓ_2, ℓ_3) should be chosen so as to maximize the distance dm_1 and dm_2, where

$$\begin{aligned} dm_1 &= \|(3\ell_1 \cdot 2m + \ell_2 \cdot 2n + 3\ell_3 \cdot 2) - q/2\| \text{ and} \\ dm_2 &= \left\| \max_{\complement((r=2m, s=2n) \cap (t=2))} (3\ell_1 \cdot r + \ell_2 \cdot s + 3\ell_3 \cdot t) - q/2 \right\|, \end{aligned} \tag{26}$$

with $0 \leq r \leq 2m$, $0 \leq s \leq 2n$, and $0 \leq t \leq 2$. In other words, we should give enough leeway to ensure that the possible error $\mathbf{n}[i]$ does not push $\mathbf{a}[i]$ to tho other side of $q/2$. The same must be done for all choices of (ℓ_1, ℓ_2, ℓ_3) that are used to distinguish every candidate for β. Similar to the tuple (m, n) and (k_1, k_2) in Subsection B.2, the tuple (ℓ_1, ℓ_2, ℓ_3) can be chosen ahead and fixed for a given parameter set of Streamlined NTRU Prime.

Table 1 is the decision table for the sntrup761 parameter set. It shows unique distinguishability for every candidate for $\beta \in \{-2, -1, 0, 1, 2\}$, based on $\mathbf{O}$ or $\mathbf{X}$ for chosen ciphertexts constructed using concrete values for (ℓ_1, ℓ_2, ℓ_3). Every candidate for $\beta = \mathsf{Rotp}(\mathbf{f}, u)[i]$ can be *uniquely identified* based on the information about $\mathbf{O}$ or $\mathbf{X}$ from only *four chosen ciphertexts.*

Since $\mathbf{e}$ is an internal variable, we use side-channels information to distinguish between the classes $\mathbf{O}$ and $\mathbf{X}$. As in Subsection B.1.4, we used the Welch's t-test to identify if $\mathbf{e} \neq 0$ for a given chosen ciphertext. The peaks in the t-test plot above the pass/fail threshold of ± 4.5 in Figure 6(b) are precisely the differentiating features. In the following discussion, we demonstrate techniques to leverage the identified features in the t-test plot to build templates for the two classes $\mathbf{O}$ and $\mathbf{X}$. The templates will then be used to classify a given single trace into either of the two classes with a very high success rate.

Table 1: Unique distinguishability of every candidate for $\beta \in [-2, 2]$ depending on $\mathbf{e} = 0$ (the event $\mathbf{O}$) or $\mathbf{e} \neq 0$ (the event $\mathbf{X}$) for `sntrup761` of Streamlined NTRU Prime KEM

Secret Coeff.	Either $\mathbf{e} = 0$ (event $\mathbf{O}$) or $\mathbf{e} \neq 0$ (event $\mathbf{X}$)			
	(ℓ_1, ℓ_2, ℓ_3)			
	$(93, 276, 48)$	$(93, 276, -48)$	$(78, 237, 78)$	$(78, 237, -78)$
-2	**O**	**X**	**O**	**X**
-1	**O**	**X**	**O**	**O**
0	**O**	**O**	**O**	**O**
1	**X**	**O**	**O**	**O**
2	**X**	**O**	**X**	**O**

B.2.2. Welch's t-test based template approach for classification

We select features of the t-test plot between $\mathcal{T}_{\mathrm{O}}$ ($\mathbf{e} = 0$) and $\mathcal{T}_{\mathrm{X}}$ ($\mathbf{e} \neq 0$) whose *absolute t-test value* is greater than a certain chosen threshold Th_{sel} as our set $\mathcal{P}$ of *Points of Interest* (PoI). A reduced trace set $\mathcal{T}'_{\mathrm{O}}$ and $\mathcal{T}'_{\mathrm{X}}$ is constructed by using points in $\mathcal{P}$. The chosen threshold Th_{sel} shall preferably be slightly greater than ± 4.5 for better distinguishability. For the t-test results in Fig.6, we choose ± 7 as a convenient threshold. We subsequently calculate the respective means of $\mathcal{T}'_{\mathrm{O}}$ and $\mathcal{T}'_{\mathrm{X}}$, denoted by $m_{\mathrm{O},\mathcal{P}}$ and $m_{\mathrm{X},\mathcal{P}}$ respectively, which are the *reduced templates* for each class.

Given a single trace t for classification, it is normalized such that $t' = t - \bar{t}$ to obtain a reduced trace $t'_{\mathcal{P}}$. We then compute the sum-of-squared difference Γ_* of the trace with each reduced template:

$$\begin{aligned}\Gamma_{\mathrm{O}} &= (t'_{\mathcal{P}} - m_{\mathrm{O},\mathcal{P}})^T \cdot (t'_{\mathcal{P}} - m_{\mathrm{O},\mathcal{P}})\\ \Gamma_{\mathrm{X}} &= (t'_{\mathcal{P}} - m_{\mathrm{X},\mathcal{P}})^T \cdot (t'_{\mathcal{P}} - m_{\mathrm{X},\mathcal{P}}) \qquad (27)\end{aligned}$$

The trace t falls into the class that corresponds to the least sum-of-squared difference. Thus, a single power/EM trace of the targeted operation is sufficient to distinguish between $\mathbf{X}$ or $\mathbf{O}$. Thus, single side-channel traces from the decryption of four chosen ciphertexts constructed according to Equation (35) can recover $\beta = \mathsf{Rotp}(\mathbf{f}, u)[i]$. Figure 7 visualizes the matching of the reduced template of a given attack trace tr with the reduced templates of the respective classes $\mathbf{O}$ and $\mathbf{X}$. There is a clear distinguishability between the reduced templates of the two classes, leading to a classification with 100% success rate.

B.2.3. Recovering the Full Secret Key

We have demonstrated the recovery of a single coefficient $\beta = \mathsf{Rotp}(\mathbf{f}, u)[i]$. By simply changing the rotation index u, we can recover $\mathsf{Rotp}(\mathbf{f}, u)[i]$ for all $u \in [0, p-1]$. However, recovery of the exact secret polynomial $\mathbf{f}$ also requires knowledge about (1) the colliding index i and (2) collision value $(+2/-2)$, both of which cannot be inferred through side-channels using our technique. Thus, we need to try out all p possible values for the collision index i as well as two choices for the collision value. This amounts to a total of $2p$ choices for the secret key and for `sntrup761`, this amounts to only 1522 choices. Let us denote the recovered secret key by $\mathbf{f}'$. For each choice of $\mathbf{f}'$, we check if $\mathbf{f}' \in R_{\mathrm{sh}}$ and compute the other secret polynomial $\mathbf{g}' \in R_3$ to check if known ciphertexts are decrypted correctly.

It is possible that none of the guessed $\mathbf{f}'$ is correct, which means that the noise polynomial $\mathbf{n}$ is probably large enough for the chosen ciphertexts corresponding to $\mathbf{c}_{\text{base}}$, that is, corresponding to $(\mathbf{d1}_{\text{att}}, \mathbf{d2}_{\text{att}})$, to induce errors in $\mathbf{e}$. Another possibility is that, for a given $(\mathbf{d1}_{\text{att}}, \mathbf{d2}_{\text{att}})$, the side-channel oracle's responses do not correspond to any of the candidates for the target coefficient. This situation may arise due to errors in $\mathbf{e}$ caused by the noise. In these cases, we can simply reject the current $(\mathbf{d1}_{\text{att}}, \mathbf{d2}_{\text{att}})$ and initiate a search for a new one before repeating the attack until the correct $\mathbf{f}$ is recovered.

B.2.4. Limitations of the PC oracle-based SCA

The ciphertexts used for the PC oracle attack limit the intermediate variable $\mathbf{e}$ to only two possible values (i.e.) $\mathbf{e} = 0$ (Class $\mathbf{O}$) and $\mathbf{e} = \pm 1 \cdot x^i$ (i.e.) $\mathbf{e} \neq 0$ (Class $\mathbf{X}$). When $\mathbf{e} = 0$, $\mathbf{b}' = 0$ (Line 5 of PKE.Decrypt in Alg.1). When $\mathbf{e} = \pm 1 \cdot x^i$, $\mathbf{b}'$ has uniformly random coefficients in $\{-1, 0, 1\}$ whose exact value depends on the secret polynomial $\mathbf{g}$. In both cases, the weight of $\mathbf{b}'$ is not equal to w. Thus, the decryption procedure only returns the constant $(1, 1, \ldots, 1, 0, 0, \ldots, 0) \in R_{\text{sh}}$ (Line 10) for all the attack ciphertexts used for the PC oracle-based attack.

Thus, the effect of variable $\mathbf{e}$ $(\mathbf{O}/\mathbf{X})$ for the attack ciphertexts does not propagate beyond the decryption procedure. Thus, the PC oracle attack can only be carried out using side-channel information from operations within the decryption procedure. In particular, operations manipulating $\mathbf{e}$ within decryption (operations enclosed in rectangular box in Fig.3). This restricts an attacker from utilizing side-channel information from operations after decryption (i.e.) operations within the re-encryption procedure (line 4 of KEM.Decaps in Alg.2).

In the following, we improve upon the PC oracle-based attack to propose a novel DF oracle-based attack on Streamlined NTRU Prime. The improved attack enables an attacker to also utilize side-channel information from many more operations within the decapsulation procedure to perform side-channel assisted key recovery.

C. Decryption Failure Oracle-based SCA

This section provides a detailed description of the DF oracle attack on Streamlined NTRU Prime KEM. The reader is referred to Subsection 3.2 for high level details about the attack. Similar to the PC oracle attack, The DF oracle attack is performed in two phases.

1. The first phase involves search for a base ciphertext $\mathbf{c}_{\text{base}}$ which corresponds to a *single collision* event. This ciphertext when added to $\mathbf{c}_{\text{valid}}$ should exactly perturb a single coefficient of $\mathbf{e}_{\text{valid}}$ resulting in a decryption failure. A decryption failure detected through side-channel leakage indicates *single collision* event with a very high probability.

2. In the second phase, $\mathbf{c}_{\text{base}}$ is used to construct new perturbed attack ciphertexts $\mathbf{c}_{\text{att}}$ whose decrypted message can only have two possible values (i.e.) $\mathbf{r}_{\text{valid}}/\mathbf{r}_{\text{invalid}}$. Side-channel leakage is used to detect decryption success/failure thereby realizing a practical DF oracle. This information obtained over several such attack ciphertexts results in full key recovery.

In the following, we describe the first phase of our DF oracle attack which involves retrieving the base ciphertext $\mathbf{c}_{\text{base}}$.

C.1. Retrieving the Base Ciphertext $\mathbf{c}_{\text{base}}$

We construct a valid ciphertext $\mathbf{c}_{\text{valid}} = \mathsf{Round}(\mathbf{h} \cdot \mathbf{r})$ (Line 3 in PKE.Encrypt in Alg.1), whose corresponding $\mathbf{a} = (3\mathbf{f} \cdot \mathbf{c}_{\text{valid}})$ is given as

$$\mathbf{a}_{\text{valid}} = \mathbf{g} \cdot \mathbf{r} + 3\mathbf{f} \cdot \mathbf{m} \tag{28}$$

where $\mathbf{m}$ is the rounding error. We construct a perturbation ciphertext $\mathbf{c}'$ (similar to $\mathbf{c}_{\text{base}}$ in Equation 10 used for the PC oracle attack) as follows:

$$\mathbf{c}' = k_1 \cdot (x^{i1} + x^{i2} + \ldots + x^{im}) + k_2 \cdot (x^{j1} + x^{j2} + \ldots + x^{jn}) \cdot \mathbf{h} = k_1 \cdot \mathbf{d}_1 + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h}, \tag{29}$$

where $3 \mid k_1$, $3 \mid k_2$ and polynomials $(\mathbf{d}_1, \mathbf{d}_2)$ are respectively with, m and n number of nonzero coefficients $(+1)$. Upon decrypting $\mathbf{c}'$, the corresponding $\mathbf{a}' = (3\mathbf{f} \cdot \mathbf{c}')$ is given as

$$\mathbf{a}' = k_1 \cdot \mathbf{d}_1 \cdot 3\mathbf{f} + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{g} + 3\mathbf{f} \cdot \mathbf{m}'' \tag{30}$$

We use $\mathbf{c}'$ to perturb $\mathbf{c}_{\text{valid}}$ in the following manner:

$$\mathbf{c}_{\text{pert}} = \mathsf{Round}(\mathbf{h} \cdot \mathbf{r} + \mathbf{c}') = \mathsf{Round}(\mathbf{h} \cdot \mathbf{r} + k_1 \cdot \mathbf{d}_1 + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h}), = \mathbf{h} \cdot \mathbf{r} + k_1 \cdot \mathbf{d}_1 + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{h} + \mathbf{m}', \tag{31}$$

where $\mathbf{m}'$ is the rounding error. Upon decrypting $\mathbf{c}_{\text{pert}}$, the corresponding $\mathbf{a} = (3\mathbf{f} \cdot \mathbf{c}_{\text{pert}})$ is given by

$$\mathbf{a} = \mathbf{g} \cdot \mathbf{r} + k_1 \cdot \mathbf{d}_1 \cdot 3\mathbf{f} + k_2 \cdot \mathbf{d}_2 \cdot \mathbf{g} + 3\mathbf{f} \cdot \mathbf{m}' \tag{32}$$

where the term $3\mathbf{f} \cdot \mathbf{m}'$ is denoted as the rounding error component $\mathbf{n}$. Thus, $\mathbf{a}$ is essentially the sum of $\mathbf{a}_{\text{valid}}$ (Equation 28) and $\mathbf{a}'$ (Equation 30) along with some rounding error $\mathbf{n}''$. In order to induce a single coefficient error in $\mathbf{a}$, only a single coefficient of $\mathbf{a}$ should be pushed beyond the $q/2$ threshold with the help of the perturbation component $\mathbf{a}'$ (i.e.) $(\mathbf{a}_{\text{valid}} + \mathbf{a}' + \mathbf{n}'')[i] > q/2$.

The absolute maximum value for a coefficient of the perturbation component $\mathbf{a}'$ (Eqn.30) is $m_1 = (3k_1 \cdot 2m + k_2 \cdot 2n)$ which is obtained upon a collision. Let the next largest possible value be denoted as m_2. The tuple (m, n) for $\mathbf{c}'$ (Eqn.29) is chosen so as to have atmost a single collision with a very high probability. Upon choosing (m, n), we also need to choose concrete values for the tuple (k_1, k_2). Let us say we use the same constraints that were used for the PC oracle attack (Subsection C.1). Firstly, we choose (k_1, k_2) such that the largest possible coefficient $m_1 > q/2$ and $m_2 < q/2$ (Eqn.15). Moreover, we impose additional constraints so as to maximize the distance tuple (dm_1, dm_2) (i.e.) distance between (m_1, m_2) respectively, from the $q/2$ threshold. This is done so as to avoid accidental crossovers near $q/2$ due to the rounding error $\mathbf{n}$ (Equation 18).

However, apart from the rounding error $\mathbf{n}$, the term $\mathbf{g} \cdot \mathbf{r}$ (denoted as $\mathbf{gr}$) in Eqn.32 also contributes to crossovers near $q/2$. So, we denote $\mathbf{n}'$ as the new error component (i.e.) $\mathbf{n}' = \mathbf{g} \cdot \mathbf{r} + 3\mathbf{f} \cdot \mathbf{m}'$. More concretely, a collision at i (i.e.) $\mathbf{a}'[i] > q/2$ does not guarantee $\mathbf{a}[i] > q/2$. For instance, it is possible that $\mathbf{a}'[i] = q/2 + \gamma$ (due to a collision) where $\gamma > 0$, but $\mathbf{a}[i] < q/2$ since $\mathbf{n}'[i] < -\gamma$. This suppresses collision at i therefore also suppresses the decryption failure. This is considered to be a false negative event. Similarly, a false positive at i can also occur (i.e.) $\mathbf{a}[i] > q/2$ even if there is no collision at i. For sntrup761, Figure 13 shows the distribution of the coefficients of $\mathbf{n}'$. It is Gaussian with mean 0 and $\sigma \approx 53$. We note that the span of the noise distribution is slightly larger than that for the PC oracle attack ($\sigma \approx 50$).

A false negative only results in a retry to identify another perturbation ciphetext $\mathbf{c}'$. However, a false positive results in using a bad perturbation ciphertext $\mathbf{c}'$ for the attack phase, which does not

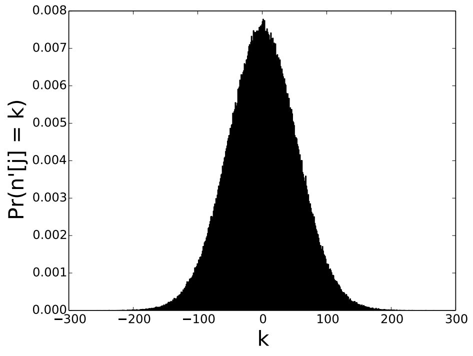

Figure 13: Distribution of coefficients of noise component $\mathbf{n}' = 3\mathbf{f} \cdot \mathbf{m}' + \mathbf{gr}$ with mean 0 and $\sigma \approx 53$ for `sntrup761` parameter set of Streamlined NTRU Prime

result in key recovery, thereby yielding unnecessary attack iterations. In order to minimize chances of a false positive, we choose (k_1, k_2) using slightly modified constraints as shown below:

$$3 \mid k_1, \quad 3 \mid k_2, \quad \mathbf{a}'[i] \begin{cases} > (q/2 - \epsilon_1) \cup < (q/2 - \epsilon_2) & \text{if } \mathbf{a}'[i] = m_1, \\ < q/2, & \text{if } \mathbf{a}'[i] < m_2, \end{cases} \tag{33}$$

with $0 \le r \le 2m$ and $0 \le s \le 2n$ and $\epsilon_1, \epsilon_2 > 0$. In other words, even if there is a collision at i (i.e.) $\mathbf{a}'[i] = m_1$, $m_1 < q/2$ and is in the range $[(q/2 - \epsilon_1)$,$(q/2 - \epsilon_2)]$ (i.e.) slighly less than $q/2$. Such a constraint for (k_1, k_2) results in several advantages. Firstly, this ensures that $\mathbf{a}[i] > q/2$ only when the following two events occur simultaneously (1) collision at $\mathbf{a}'[i]$ and (2) noise coefficient $\mathbf{n}'[i] > \epsilon_2$. This significantly increases false negatives (suppressing a collision), but also equivalently reduces chances of a false positive. Secondly, choosing m_1 to be less than $q/2$ and farther from it by atleast ϵ_2 pushes other possible values for coefficients of $\mathbf{a}'$ further away from $q/2$. This additional leeway for the other values decreases the chances of a false positive at other indices $j \neq i$ where there is no collision.

Thirdly, a collision at i is detected only when $\mathbf{n}'[i] > \epsilon_2$. In other words, we allow the noise coefficient to have a large value, which in turn also increases the chances of $\mathbf{gr}[i]$ having a large value. Thus, we not only identify a collision, but also increase the chances of achieving collision at the index where $\mathbf{gr}[i]$ has a large value. This has a positive influence in the key recovery phase of the attack in the following manner. Referring to the construction of attack ciphertexts for the PC oracle attack (Subsection B.2.1), we observe that the constraints are placed for choosing (k_1, k_2) for the attack ciphertexts such that, both (dm_1, dm_2) are maximized to avoid false positives/negatives.

As we will see in the following discussion (Subsection C.3), the $\mathbf{a}$ variable of the attack ciphertexts used for key recovery also contain the term $\mathbf{gr}$. Since $\mathbf{gr}[i]$ (colliding index) has high chances of having a large value, this allows us to relax contraints such that dm_1 need not be maximized, but can be chosen such that m_1 is closer to $q/2$. As a result, even if $\mathbf{a}'[i] = m_1$ and close to $q/2$, $\mathbf{gr}[i]$ pushes it further away from $q/2$ thereby decreasing chances of a false negative in the attack phase. Since m_1 is chosen to be closer to $q/2$, it gives additional leeway to increase dm_2 which again reduces chances of false positives at other indices $j \neq i$ where collision does not occur. Thus, choosing (k_1, k_2) according to Equation 33 for the base ciphertext has positive influence in reducing chances for a false positive/negative in both the phases of the attack.

Choosing Concrete Values for the Chosen-Ciphertext: Keeping the aforementioned constraints in mind, for `sntrup761`, we choose $(m, n) = (1, 3)$ similar to the PC oracle attack to limit the number of collisions to at most 1 with a high probability. We choose $(k_1, k_2) = (93, 279)$ whose $(dm_1, dm_2) = (63, 342)$. We note that the values for (m, n) and (k_1, k_2) can be fixed and chosen beforehand for a given parameter set of Streamlined NTRU Prime.

C.2. Detecting Decryption Failure through Side-Channels

Given (m, n) and (k_1, k_2), we randomly select polynomials $\mathbf{d}_1$ and $\mathbf{d}_2$ (Equation 29) until we arrive at a ciphertext $\mathbf{c}'$ which induces a decryption failure. To detect a decryption failure, we can utilize side-channel information from any operation within the re-encryption procedure (line 4 in KEM.Decaps of Alg.2) as well as the operations manipulating $\mathbf{e}'$ within the decryption procedure (Line 5-6 in PKE.Decruypt of Alg.1). For our experiments, we target the re-encryption procedure. In particular, we want to distinguish between $\mathbf{r}' = \mathbf{r}'_{\text{valid}}/\mathbf{r}'_{\text{invalid}}$, for which we adopt the same Welch's t-test based approach to identify features differentiating the two classes.

We obtained N replicated measurements from decapsulation of the valid ciphertext $\mathbf{c}_{\text{valid}}$ (Class $\mathbf{O}$). The obtained trace set is denoted by $\mathcal{T}_{\text{O}}$. We then build a perturbation ciphertext $\mathbf{c}'$ (Equation 29) and subsequently generate $\mathbf{c}_{\text{pert}}$. We obtain N similar replicated side-channel measurements for $\mathbf{c}_{\text{pert}}$ which is denoted as $\mathcal{T}_{\text{X}}$. Figure 10(a) depicts the t-test plot between $\mathcal{T}_{\text{O}}$ and $\mathcal{T}_{\text{X}}$ when there is no decryption failure for $\mathbf{c}_{\text{pert}}$ (i.e.) $\mathbf{r}' = \mathbf{r}'_{\text{valid}}$. The plot has no peaks above the t-test threshold. Figure 10(b) shows the t-test plot when the decryption of $\mathbf{c}_{\text{pert}}$ fails (i.e.) $\mathbf{r}' = \mathbf{r}'_{\text{invalid}}$. The plot clearly has several peaks well above the t-test threshold clearly indicating decryption failure.

We repeat the same steps until we find a perturbation ciphertext $\mathbf{c}'$ which results in a decryption failure. This ciphertext is nothing but the base ciphertext $\mathbf{c}_{\text{base}}$ whose corresponding polynomial tuple $(\mathbf{d}_1, \mathbf{d}_2)$ is denoted as $(\mathbf{d1}_{\text{att}}, \mathbf{d2}_{\text{att}})$ with m and n terms respectively. The ciphertext $\mathbf{c}_{\text{base}}$ is given as follows:

$$\begin{aligned}\mathbf{c}_{\text{base}} &= k_1 \cdot (x^{i1} + x^{i2} + \ldots + x^{im}) + k_2 \cdot (x^{j1} + x^{j2} + \ldots + x^{jn}) \cdot \mathbf{h} \\ &= k_1 \cdot \mathbf{d1}_{\text{att}} + k_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h} \end{aligned} \tag{34}$$

C.3. DF Oracle-based Key Recovery

We use the polynomials ($\mathbf{d1}_{\text{att}}$,$\mathbf{d2}_{\text{att}}$) of the base ciphertext $\mathbf{c}_{\text{base}}$ to build new perturbed attack ciphertexts. Side-channel leakage from decapsulation of these attack ciphertexts is used to identify decryption success/failure which subsequently leads to full recovery of the secret polynomial $\mathbf{f}$ one coefficient at a time.

C.3.1. Attack Methdology

We build, using ($\mathbf{d1}_{\text{att}}$,$\mathbf{d2}_{\text{att}}$), the ciphertext

$$\mathbf{c}' = \ell_1 \cdot \mathbf{d1}_{\text{att}} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h} + \ell_3 \cdot x^u \tag{35}$$

where $\ell_1, \ell_2, \ell_3 \in \mathbb{Z}^+$ and $u \in [0, p-1]$. This ciphertext is used to perturb $\mathbf{c}_{\text{valid}}$ in the same manner as in Eqn.31. Let the resulting ciphertext be denoted as $\mathbf{c}_{\text{att}}$ and the corresponding $\mathbf{a} = 3\mathbf{f} \cdot \mathbf{c}_{\text{att}}$ is given by

$$\begin{aligned}\mathbf{a} &= 3\mathbf{f} \cdot \mathbf{c}_{\text{att}} \\ &= 3\ell_1 \cdot \mathbf{d1}_{\text{att}} \cdot \mathbf{f} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{h} \cdot 3\mathbf{f} + \ell_3 \cdot 3\mathbf{f} \cdot x^u + \mathbf{gr} + 3\mathbf{f} \cdot \mathbf{m}' \\ &= 3\ell_1 \cdot \mathbf{d1}_{\text{att}} \cdot \mathbf{f} + \ell_2 \cdot \mathbf{d2}_{\text{att}} \cdot \mathbf{g} + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u) + \mathbf{gr} + 3\mathbf{f} \cdot \mathbf{m}', \end{aligned} \tag{36}$$

Table 2: Unique distinguishability of every candidate for $\beta \in [-2, 2]$ based on $\mathbf{r}' = \mathbf{r}'_{\text{valid}}$ (Class **O**) or $\mathbf{r}'_{\text{invalid}}$ (Class **X**) for `sntrup761` of Streamlined NTRU Prime KEM

Secret Coeff.	Either $\mathbf{r}'_{\text{valid}}$ (Class **O**) or $\mathbf{r}'_{\text{invalid}}$ (Class **X**)			
	(ℓ_1, ℓ_2, ℓ_3)			
	$(90, 273, 45)$	$(90, 273, -45)$	$(78, 231, 78)$	$(78, 231, -78)$
-2	**O**	**X**	**O**	**X**
-1	**O**	**X**	**O**	**O**
0	**O**	**O**	**O**	**O**
1	**X**	**O**	**O**	**O**
2	**X**	**O**	**X**	**O**

We use $\mathbf{n}$ to denote the rounding error $3\mathbf{f} \cdot \mathbf{m}'$. For the sake of explanation, we assume that $\mathbf{d1}_{\text{att}}$ and $\mathbf{d2}_{\text{att}}$ collide at i with a value of $+2$. Thus, the coefficients of $\mathbf{a}$ can be expressed as

$$\mathbf{a}[j] = \begin{cases} 3\ell_1 \cdot 2m + \ell_2 \cdot 2n + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u)[j] + \mathbf{n}[j], \text{ if } j = i \\ 3\ell_1 \cdot r + \ell_2 \cdot s + 3\ell_3 \cdot \mathsf{Rotp}(\mathbf{f}, u)[j] + \mathbf{n}[j], \text{ if } (j \neq i), \complement(r = 2m, s = 2n). \end{cases} \tag{37}$$

Similar to the PC oracle attack, we choose (ℓ_1, ℓ_2, ℓ_3) such that $\mathbf{a}[j] < (q/2)$ for $j \neq i$. Thus, when reduced modulo 3, the perturbations are reduced to 0 at indices $j \neq i$ where there is no collision. However, the occurence of whether or not $\mathbf{a}[i] > q/2$, is made to depend on a single coefficient of the rotated secret polynomial (i.e.) $\mathsf{Rotp}(\mathbf{f}, u)[i]$. Therefore, the occurence of decryption failure also depends upon this single coefficient of the rotated secret polynomial. For convenience, we use β to denote $\mathsf{Rotp}(\mathbf{f}, u)[i]$ and $\beta \in [-2, 2]$.

Thus, we choose (ℓ_1, ℓ_2, ℓ_3) so as to build a binary distinguisher that can uniquely distinguish every possible candidate for β based on the occurrence of decryption failure (Class **O**/Class **X**). We use very similar constraints, that were used for the PC oracle attack to choose concrete values for (ℓ_1, ℓ_2, ℓ_3) and build the binary distinguisher for the DF oracle attack. We thus refer the reader to Subsection B.2 for concrete details of choosing values for the tuple (ℓ_1, ℓ_2, ℓ_3).

Table 2 is the decision table for the `sntrup761` parameter set. It shows unique distinguishability for every candidate for β based on success/failure of decryption (**O**/**X**) for the attack ciphertexts constructed based on Equation 36. Every candidate can be *uniquely identified* based on the information about **O** or **X** for only *four ciphertexts*. By simply changing the rotation index u, we can recover $\mathsf{Rotp}(\mathbf{f}, u)[i]$ for all $u \in [0, p-1]$ from which the secret polynomial $\mathbf{f}$ can be recovered.

C.3.2. Welch's t-test based template approach for classification

We adopt the t-test based reduced template approach (Subsection B.2.2) to identify decryption failures through side-channel leakage. We refer to the t-test plot in Fig.10(b) which clearly identifies the features distinguishing the two classes (i.e.) $\mathbf{r}'_{\text{valid}}$ (Class **O**) or $\mathbf{r}'_{\text{invalid}}$ (Class **X**). Those features whose t-test value is greater than a certain a chosen threshold Th_{sel} are selected as the *Points of Interest* (PoI) set $\mathcal{P}$. The set $\mathcal{P}$ is used to construct reduced templates for the two classes. Given a trace tr for classification, we construct a reduced trace of tr corresponding to $\mathcal{P}$ and perform a simple LSQ test to determine the class to which tr belongs. Figure 7 visualizes the matching of the reduced template of a given attack trace tr with the reduced templates of the respective classes **O** and $\mathbf{X}$. There is a clear distinguishability between the reduced templates of the two classes, and we were able to correctly classify a given single trace with a 100% success rate.

C.3.3. Recovering the Full Secret Key

While we can use the DF oracle attack to recover coefficients of the rotated secret polynomial $\mathsf{Rotp}(\mathbf{f}, u)$ for different rotation constants $u \in [0, p-1]$, recovering the exact secret polynomial $\mathbf{f}$ also requires the following details (1) colliding index i and (2) value of collision $(+2/-2)$. Since the DF oracle does not provide these two details, we simply try out all p possibilities for i as well 2 possibilities for the colliding value which amounts to $2p$ possibilities. For the `sntrup761` parameter set, it amounts to 1522 choices. For each choice, we can check whether the recovered secret $\mathbf{f}' \in R_{\text{sh}}$ and also compute the corresponding secret polynomial $\mathbf{g}' \in R_3$ to check the correctness of decryption. This will lead to unique identification of the secret polynomials $(\mathbf{f}, \mathbf{g})$. It is also possible that the secret is not recovered correctly, due to a bad choice of the base ciphertext. In this case, we simply retry the attack until the correct key is recovered.