Correlation Power Analysis on NTRU Prime and Related Countermeasures

- A Broadly Applicable Approach -

Wei-Lun Huang[1], Jiun-Peng Chen[2] and Bo-Yin Yang[3]

[1] Academia Sinica, Taiwan, 271828182euler@gmail.com
[2] Academia Sinica, Taiwan, jpchen@ieee.org
[3] Academia Sinica, Taiwan, byyang@iis.sinica.edu.tw

Abstract. We perform correlation power analysis on ideal-lattice-based cryptosystems featuring product scanning, for example the reference implementation of NTRU Prime, a Round 2 candidate in the NIST PQC Competition. We also discuss three corresponding countermeasures in detail. The proposed approach achieves full private-key recovery in a highly efficient way with few traces. For each defensive strategy, its effectiveness is validated, and its side-channel resistance is evaluated by the TVLA *general* tests. The correlation power analysis exploits the vulnerabilities in product-scanning-based polynomial multiplications. The statistical analysis program in C++ takes time linear in the input size on average and practically less than 8 seconds on an ordinary laptop to reveal all the coefficients of each private-key polynomial. The three countermeasures together demonstrate the tradeoff between security and performance. The predictions about their effectiveness, performance, and side-channel resistance are supported by the correlation power analysis and the TVLA *general* tests based on thousands of traces.

Keywords: Correlation Power Analysis · Ideal Lattice Cryptography · NTRU Prime · SCA Countermeasures · Test Vector Leakage Assessment (TVLA)

1 Introduction

Ever since the formulation of Shor's algorithm [Sho97], quantum computing has been a potential threat to classical public-key cryptosystems, which is based on the hardness of integer factorization and discrete logarithms. These cryptographic primitives include RSA [RSA78], Diffie-Hellman key agreement [DH76], ElGamal encryption [Gam85], DSA [Bar13], ECDH [BCRS13], and ECDSA [JMV01]. Recently, quantum computers have been estimated as arriving in 10 to 15 years [Sni16, WLYZ18], and there is consequently an urgent need for the examination and standardization of post-quantum cryptosystems. This need further leads to the NIST Post-Quantum Cryptography Competition [Nat17]. The competitors are mostly based on lattices [DKL+18, BDK+18, BCD+16], error correction codes [BBC+18, McE78], multivariate quadratic equations [DS05], hash functions [BHH+15], and supersingular isogeny graphs [JF11].

Unfortunately, quantum resistance does not guarantee these submissions security in practice. There has been a vast amount of work on the implementation attacks against post-quantum cryptosystems, and [TE15] provides a comprehensive collation of the fault analyses and side-channel analyses on a variety of post-quantum cryptographic schemes. More cutting-edge side-channel analyses on digital signature schemes are introduced in [EFGT17], [KAJ17], and [PSKH18]. [EFGT17] applies electromagnetic analysis to BLISS, and achieves full key recovery from one single trace using integer linear programming.

[KAJ17] features three zero-value attacks on supersingular isogeny Diffie-Hellman using refined power analysis. [PSKH18] proposes the correlation power analysis on Rainbow and Unbalanced Oil-and-Vinegar, two signature schemes based on multivariate quadratic equations, and fully recovers the secrets in use.

More advanced side-channel analyses on lattice-based encryption are introduced in [SW07], [LSCH10], [MNY17], [PPM17], [AKJ+18], and [ATT+18]. [SW07] describes a timing attack against NTRUEncrypt, which exploits the variation in the number of hash calls during decryption. [LSCH10] not only applies both simple power analysis and correlation power analysis to a typical NTRU software implementation, but also provides the corresponding countermeasures. [MNY17] proposes an NTRUEncrypt FPGA implementation at the architecture level resistant to first-order differential power analysis. [PPM17] exploits the weaknesses in the Number Theoretic Transform, features the full private-key recovery from one single trace, and breaks the masked implementations of some lattice-based cryptographic schemes. [AKJ+18] presents a single-trace power analysis on NTRU Open Source and NTRUEncrypt. [ATT+18] mounts horizontal differential power analysis on NewHope and Frodo to achieve full private-key recovery from one single trace with >99% success rate.

NTRU Prime [BCLvV17a], a Round 2 submission to the NIST PQC Competition, is based on ideal lattices. There are two schemes in this submission: Streamlined NTRU Prime and NTRU LPRime. Though Streamlined NTRU Prime is a variant of the classic NTRU [HPS98], and NTRU LPRime shares a similar structure with NewHope [ADPS16], the reference C implementation of NTRU Prime [BCLvV17b] is not subject to the previous attacks against NTRU-like cryptosystems and NewHope. These previous attacks target the implementations with data-dependent timing differences [SW07] and the ones which employ the operand scanning method [LSCH10, KY12, WZW13, ZWW13, AKJ+18, ATT+18] or the NTT network for polynomial multiplications [PPM17]. However, the reference C implementation of NTRU Prime is constant-time and generic, realizing polynomial multiplications with the product-scanning method.

In this paper the correlation power analysis [BCO04] on NTRU Prime achieves full private-key recovery with the leakages from polynomial multiplications in decapsulation. This analysis is practically fast and asymptotically efficient, requiring few traces to reveal the sensitive information. Also, it works for other ideal-lattice-based cryptosystems whose polynomial multiplications are based on the product-scanning method [HW11]. Three countermeasures are then proposed and validated. The TVLA *general* tests [GJJR11] further evaluate their side-channel resistance. Overall, these defensive strategies display the tradeoff between performance and security.

2 Preliminaries

2.1 NTRU Prime

NTRU Prime [BCLvV17a] is a Round 2 candidate in the NIST Post-Quantum Cryptography Competition [Nat17]. It features polynomial rings distinct from those of typical Ring-LWE-based cryptosystems and NTRU to avoid potential algebraic attacks. In NTRU Prime there are two key-encapsulation mechanisms based on ideal lattices: Streamlined NTRU Prime and NTRU LPRime.

Let $\mathbb{Z}_m$ be $(-m/2, m/2] \cap \mathbb{Z}$. For a given prime p and an arbitrary $m \in \mathbb{Z}^+$, R and R_m refer to $\mathbb{Z}[x]/(x^p - x - 1)$ and $\mathbb{Z}_m[x]/(x^p - x - 1)$, respectively. A polynomial is *small* if all of its coefficients belong to $\{-1, 0, +1\}$, and *weight-w* if exactly w of its coefficients are nonzero. If not specified, the following two expressions to describe a polynomial $\sigma(x)$ of degree $n \in \mathbb{N}$ are interchangeable: σ and $\sigma_0 + \sigma_1 x + \sigma_2 x^2 + ... + \sigma_n x^n$.

Streamlined NTRU Prime has positive integer parameters p, q, and w: p and q are

primes; $2p \geq 3w$; $q \geq 16w + 1$; $x^p - x - 1$ is irreducible in $\mathbb{Z}_q[x]$. Also, it specifies a hash function producing two fixed-length strings: *confirmation* and *session key* from each *small* polynomial in R. Figure 1 shows the key generation, encapsulation, and decapsulation of this cryptographic scheme. The error detection stage in decapsulation is skipped due to its irrelevance in this paper.

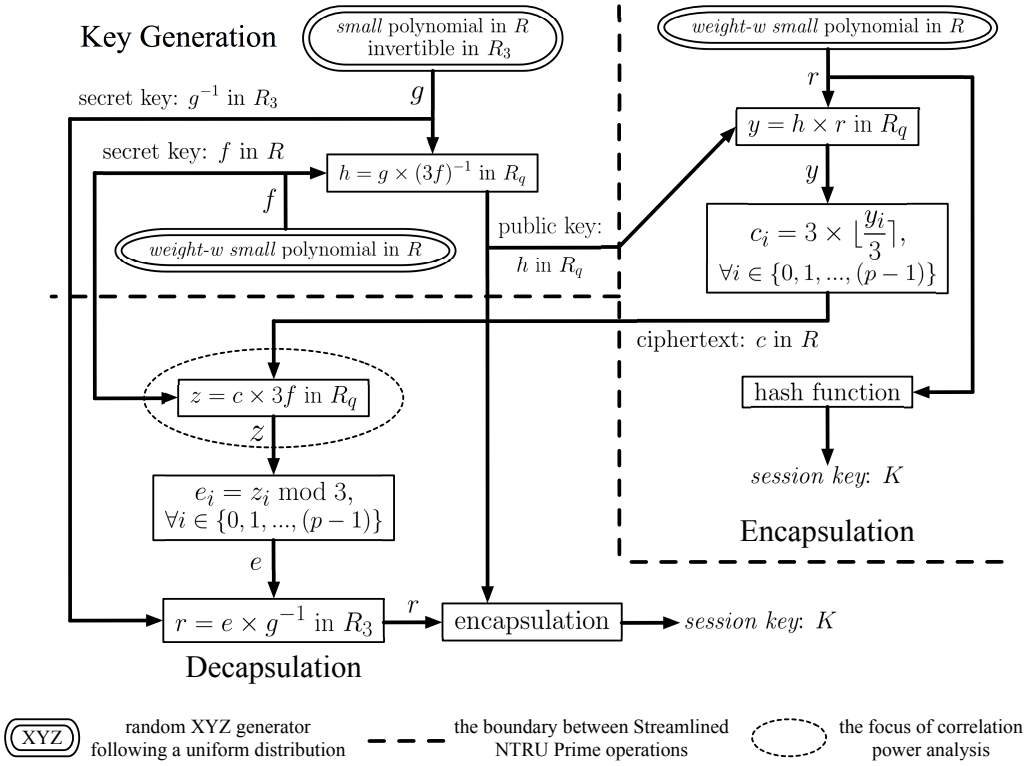


Figure 1: Streamlined NTRU Prime: key generation, encapsulation, and decapsulation

NTRU LPRime has positive integer parameters p, q, w, δ, and I: p and q are primes; $8 \mid I$; $2p \geq 3w$; $q \geq 16w + 2\delta + 3$; $p \geq I$; $x^p - x - 1$ is irreducible in $\mathbb{Z}_q[x]$. Also, it specifies a hash function producing three fixed-length strings: *cipher key*, *confirmation* and *session key* from each I-bit string. NTRU LPRime further includes four functions:

- *Generator*: producing an polynomial in R_q from each *seed* string
- *Small*: producing a *weight-w small* polynomial in R from each *cipher key*
- *Top*: mapping each element in $\mathbb{Z}_q^I$ to a fixed-length string
- *Right*: mapping each string in the image of *Top* to an element in $\mathbb{Z}_q^I$ such that $\forall C \in \mathbb{Z}_q^I$, all the coordinates of $Right(Top(C)) - C$ are in $\{0, 1, ..., \delta\}$.

Figure 2 shows the key generation, encapsulation, and decapsulation of this cryptographic scheme. The error detection stage in decapsulation is again skipped due to its irrelevance in this paper.

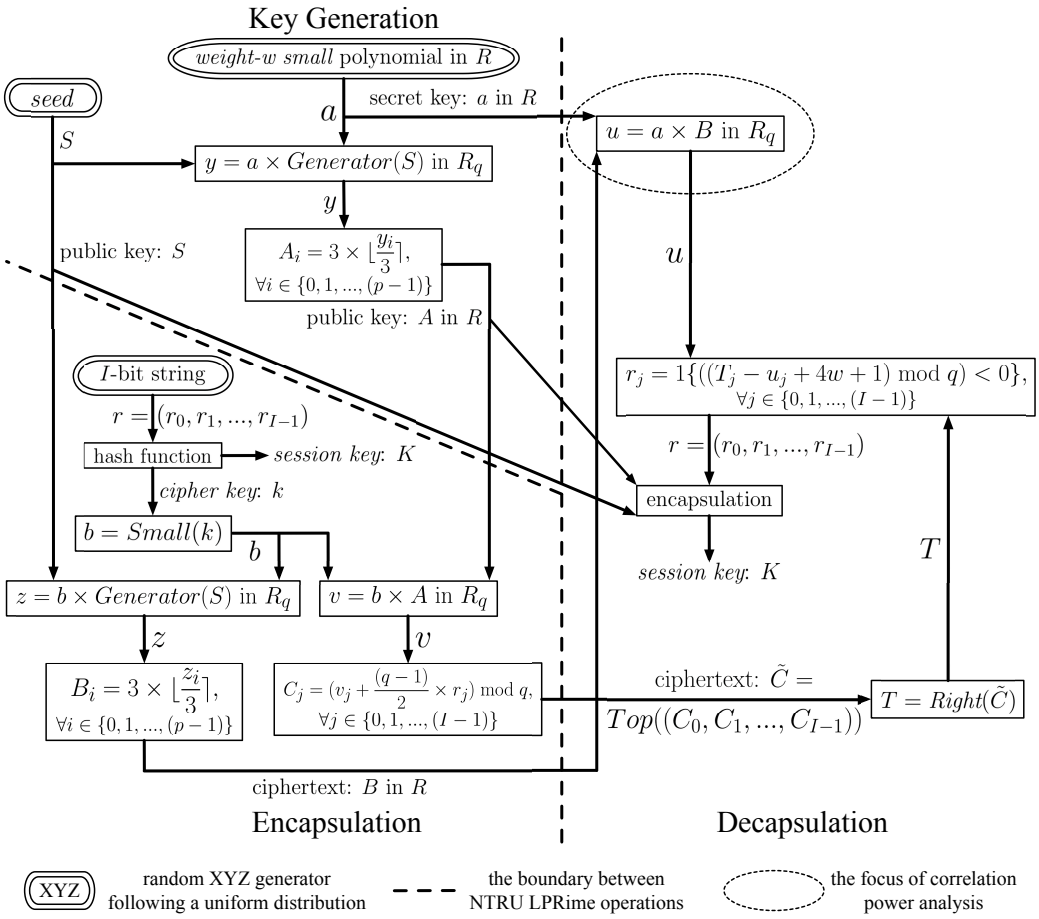


Figure 2: NTRU LPRime: key generation, encapsulation, and decapsulation

The reference C implementation of NTRU Prime [BCLvV17b] realizes polynomial multiplication in R_q in a way different from conventional NTRU implementations [LSCH10, KY12, WZW13, ZWW13, AKJ+18, ATT+18]. In decryption/decapsulation, the two input polynomials of degree $\leq (p-1)$ are a private key f and a ciphertext c. Conventional NTRU implementations view such multiplication as the superposition of $f_i \times c, \forall i \in \{0, 1, ..., (p-1)\}$:

$$f \times c = (f_0 \times c) + (f_1 \times c)x + ... + (f_{p-1} \times c)x^{p-1}$$

Since f is mostly *small*, some implementations only calculate $f_i \times c$ for nonzero f_is, and replace every scalar multiplication with a scalar addition/subtraction to accelerate the computation of $f \times c$. By contrast, the polynomial multiplication in R_q in the reference C implementation of NTRU Prime is constant-time and generic. Moreover, as illustrated in Figure 3, it calculates the coefficients of $f \times c$ one by one. Algorithm 1 describes such multiplication in a more detailed manner.

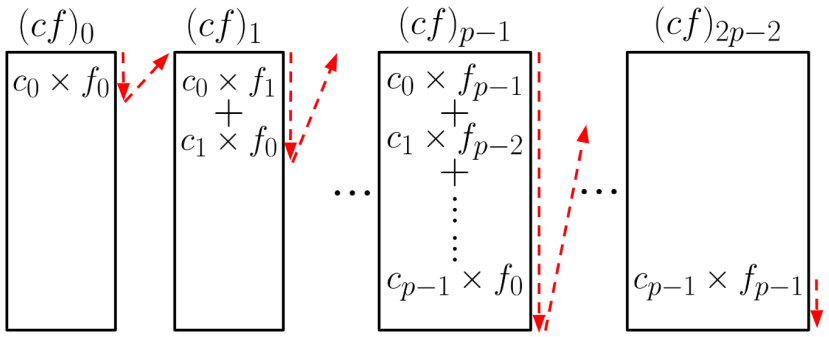


Figure 3: NTRU Prime calculates the coefficients of $f \times c$ one by one.

Algorithm 1 Polynomial Multiplication in R_q in NTRU Prime Decapsulation

Input: *small* polynomial $f \in R$ and polynomial $c \in R_q$
Output: polynomial $d = f \times c$ in R_q

for $i = 0$ to $(2p-2)$ **do** ▷ $e = (f \times c) \bmod q$
  $e_i \leftarrow 0$
end for
for $i = 0$ to $(p-1)$ **do**
  for $j = 0$ to i **do**
    $e_i \leftarrow (e_i + c_j \times f_{i-j}) \bmod q$
  end for
end for
for $i = p$ to $(2p-2)$ **do**
  for $j = (i-p+1)$ to $(p-1)$ **do**
    $e_i \leftarrow (e_i + c_j \times f_{i-j}) \bmod q$
  end for
end for
for $i = (2p-2)$ to p **do** ▷ $d = (e \bmod (x^p - x - 1)) \bmod q$
  $e_{i-p+1} \leftarrow (e_{i-p+1} + e_i) \bmod q$
  $e_{i-p} \leftarrow (e_{i-p} + e_i) \bmod q$
end for
for $i = (p-1)$ to 0 **do**
  $d_i \leftarrow e_i$
end for
return d

2.2 Correlation Power Analysis

Side-channel analysis breaks cryptosystems using implementation flaws. First, it collects side-channel leakages (execution time [Koc96], power consumption [KJJ99], electromagnetic radiation [vE85], etc.) from cryptographic devices. Then it identifies the relations between such leakages and the operations being executed or the intermediate values being processed. Finally, it employs a series of data processing, observation, and statistical analysis in order to reveal sensitive information about the cryptographic primitives in use.

Correlation power analysis (CPA) [BCO04, MOP07] is a popular branch of side-channel analysis. First, it targets specific intermediate values to decompose the entire key space into several tiny search spaces. Then it chooses suitable power models which relate these intermediate values to the power consumption of the target device. After power trace

recording, it produces an optimal guess for each search space. To be more specific, it iterates over all candidates in the search space and all time indices within the trace range, calculating the Pearson correlation coefficients between expected power consumption and its counterpart in reality. Finally, sensitive information such as private keys is derived from these optimal guesses. In the context of CPA, ciphertexts such as c in Streamlined NTRU Prime and B in NTRU LPRime are assumed accessible at low cost.

2.3 Test Vector Leakage Assessment (TVLA)

TVLA [GJJR11] is a first-order side-channel leakage assessment widely used in industry and academia. It first divides the given set of side-channel measurements into two, bringing about a significant difference in the specified sensitive information at algorithm level between the two subsets. Then it checks if there is accordingly a significant difference between the measurements in different subsets using Welch's t-test [Wel47]. If such a difference exists with high confidence, TVLA asserts the side-channel vulnerability of the implementation under examination.

Let A and B be the two subsets. Their sample mean vectors are X_A and X_B, respectively. Their sample variance vectors are V_A and V_B, respectively. Their sizes are N_A and N_B, respectively. Now the formula of the t-statistic trace T is shown below:

$$T = \frac{X_A - X_B}{\sqrt{\frac{V_A}{N_A} + \frac{V_B}{N_B}}}$$

The threshold 4.5 is a convention in TVLA: An implementation fails the test if the absolute value of any sample in its t-statistic trace exceeds 4.5. To prevent false positives, an implementation does not formally fail TVLA until it fails two independent tests at the same time index.

TVLA comprises two sorts of tests: *general* and *specific*. In the *general* tests, the given traces are partitioned according to whether their corresponding inputs are fixed or random. The *specific* tests further separate the traces of random inputs into two groups based on the difference in specified intermediate values at algorithm level. The applications of TVLA include the acceleration of side-channel analysis and the evaluation of side-channel resistance.

3 Correlation Power Analysis on NTRU Prime

3.1 Methodology

In this paper, CPA focuses on $z = c \times 3f$ in R_q in Streamlined NTRU Prime and $u = a \times B$ in R_q in NTRU LPRime, succeeding in the recovery of f and a. As for g in Streamlined NTRU Prime, the formula $h = g \times (3f)^{-1}$ in R_q and the knowledge of the public key h together enable its recovery given f. Hence, this analysis achieves full private-key recovery for both schemes in NTRU Prime.

Algorithm 2 shows the proposed approach. Because of its direct applicability to Algorithm 1, the elaboration below inherits the notations in Algorithm 1. Here the CPA concentrates on the calculation of e_{p-1} in $\mathbb{Z}_q$: $e_{p-1} = f_{p-1} \times c_0 + f_{p-2} \times c_1 + ... + f_0 \times c_{p-1} \pmod q$. It views the Hamming weight of the intermediate value being processed as the expected power consumption [KJJ99, MD99]. It also defines the optimal guess in a search space as the candidate with the most negative correlation coefficient due to the measurement setups [O'F16] in this paper. Let N be the number of trials, S be the size of a power trace, the function HW_{32} calculate the Hamming weight of a 32-bit string, and the function $corrcoef$ compute the correlation coefficient between two input arrays.

Algorithm 2 Correlation Power Analysis on NTRU Prime

Input:
a random array of polynomials $c[N] \in R_q^N$ ▷ ciphertexts
the corresponding power traces $P[S][N]$ ▷ the calculation of e_{p-1} in $\mathbb{Z}_q$
THRESHOLD ▷ to distinguish the correct guess from incorrect ones

Output: *weight-w small* polynomial f in R

for $i = 0$ to $(p-1)$ **do**
  $f_i \leftarrow 0$
end for
for $j = 1$ to $(p-w+1)$ **do** ▷ The first stage: test $\pm c_i \pm c_j$
  for $i = 0$ to $(j-1)$ **do**
    for $(x, y) \in \{1, -1\}^2$ **do**
      $e_{p-1}^{tmp}[0:N] \leftarrow (x \times c_i[0:N] + y \times c_j[0:N]) \bmod q$
      $k_{x,y} \leftarrow \underset{0 \le k < S}{\arg\min}\ corrcoef(HW_{32}(e_{p-1}^{tmp}[0:N]), p[k][0:N])$
      $\rho_{x,y} \leftarrow corrcoef(HW_{32}(e_{p-1}^{tmp}[0:N]), p[k_{x,y}][0:N])$
    end for
    $(\bar{x}, \bar{y}) \leftarrow \underset{(x,y) \in \{1,-1\}^2}{\arg\min}\ \rho_{x,y}$
    if $\rho_{\bar{x},\bar{y}} <$ THRESHOLD **then**
      $(f_{(p-1-i)}, f_{(p-1-j)}, i_{start}, k_{start}) \leftarrow (\bar{x}, \bar{y}, (j+1), (k_{\bar{x},\bar{y}} + 1))$
      $e_{p-1}[0:N] \leftarrow (\bar{x} \times c_i[0:N] + \bar{y} \times c_j[0:N]) \bmod q$
      break out of this nested **for** loop
    end if
  end for
end for
$weight = (w-2)$
for $i = i_{start}$ to $(p - weight)$ **do** ▷ The second stage: test $e_{p-1} \pm c_i$
  if $weight == 0$ **then**
    break
  end if
  for $x \in \{1, -1\}$ **do**
    $e_{p-1}^{tmp}[0:N] \leftarrow (e_{p-1}[0:N] + x \times c_i[0:N]) \bmod q$
    $k_x \leftarrow \underset{k_{start} \le k < S}{\arg\min}\ corrcoef(HW_{32}(e_{p-1}^{tmp}[0:N]), p[k][0:N])$
    $\rho_x \leftarrow corrcoef(HW_{32}(e_{p-1}^{tmp}[0:N]), p[k_x][0:N])$
  end for
  $\bar{x} \leftarrow \underset{x \in \{1,-1\}}{\arg\min}\ \rho_x$
  if $\rho_{\bar{x}} <$ THRESHOLD **then**
    $(f_{(p-1-i)}, k_{start}) \leftarrow (\bar{x}, (k_{\bar{x}} + 1))$
    $e_{p-1}[0:N] \leftarrow (e_{p-1}[0:N] + \bar{x} \times c_i[0:N]) \bmod q$
    $weight \leftarrow weight - 1$
  end if
end for
return f

The proposed approach takes probabilistically linear time in terms of p, if a $\Omega(1)$ lower bound τ for w/p exists in view of security concerns. The efficiency of this CPA is based on two reasonable assumptions: First, the correlation coefficient corresponding to the correct guess is obviously far away from those corresponding to the rest. Therefore, a guess is correct if and only if its correlation coefficient is lower than a specified negative threshold, and the CPA should find an eligible threshold with ease. Second, the relative order of operations in an implementation is the same as its counterpart in source code. Thus, the CPA should only focus on the samples of time indices greater than those corresponding to previous optimal guesses. According to the lower bound τ and the first assumption, the first stage in Algorithm 2 needs less than $(p \times (2/\tau))/2 = p/\tau$ iterations of testing $\pm c_i \pm c_j$ on average to reveal the first two nonzero coefficients of the *weight-w small* secret $f \in R$. In any case, the second stage needs less than p iterations of testing $\pm c_i$ to reveal the rest.

The proposed approach targets e_{p-1} for its calculation involves all the coefficients of f and c, and hence this term is both informative and controllable. The first stage in Algorithm 2 views the two nonzero coefficients as a pair to avoid the confusion between the samples indicating the access of c_i as the coefficient of c and those indicating e_{p-1} being c_i. To examine the candidates of high *a priori* probabilities first, The nested loop at the first stage takes the form of "**for** $j = 1$ to $(p - w + 1)$ **do for** $i = 0$ to $(j - 1)$ **do**" rather than "**for** $i = 0$ to $(p - w)$ **do for** $j = 1$ to $(p - w + 1)$ **do**". This design reduces both time consumption and the required number of traces.

3.2 Experiments and Results

To evaluate the efficacy of this method, ChipWhisperer-Lite Two-Part Version [O'F16] is employed. The target device, an STM32F303RCT7 32-bit microcontroller [STM18], runs the polynomial multiplication in R_q in NTRU Prime decapsulation written in C [BCLvV17b, KR+18]. ChipWhisperer Capture [O+18] then records and stores both power traces and input data (Ubuntu 18.04.1 LTS, VirtualBox VM on a MacBook Air). The microcontroller operates at the clock rate 7.38MHz, and the C implementation is sampled at 14.769MS/s. Each trace contains 131,092 samples, recording the power consumption during the calculation of e_{p-1}. The statistical analysis part is programmed in C++ (macOS Sierra 10.12.6, MacBook Air).

Following the recommendation in the NTRU Prime submission [BCLvV17a], the experiment sets (p, w) as (761, 286), the parameter set for Streamlined NTRU Prime. There are 10 trials in the experiment, each involving a uniformly and randomly generated secret f. The CPA adopts -0.90 as the threshold due to the stunningly high compatibility of Hamming weight power model with the target device. To fully recover each of the 10 secrets, 50 traces suffice. The statistical analysis takes less than 8 seconds to disclose every coefficient of the secret f. Figure 4, whose unit of y-axis is (V), shows the pattern in a power trace. Figure 5 and Figure 6 are screenshots of the statistical analysis.

As shown in Figure 5, the recovery of the last nonzero coefficient starts with the monomial $+x^5$ in this trial. The program then searches from the monomials of higher order to those of lower order and from the samples of smaller time indices to those of larger time indices. Updating its guess with the best known monomial-sampleId pair in terms of correlation coefficient, the CPA finally outputs $-x^2$ as its answer. The corresponding correlation coefficient -0.992438 is way lower than the threshold -0.90.

At the end of the same trial, the statistical analysis reveals both the value and the position of every nonzero coefficient of f, and therefore recovers the secret f. As shown in Figure 6, the 270th monomial is $+x^{36}$, the 271st monomial is $-x^{34}$, the 272nd monomial is $-x^{31}$, ..., the 284th monomial is $-x^9$, the 285th monomial is $+x^6$, and the 286th monomial is $-x^2$. Every optimal guess leads to a correlation coefficient lower than the threshold -0.90.

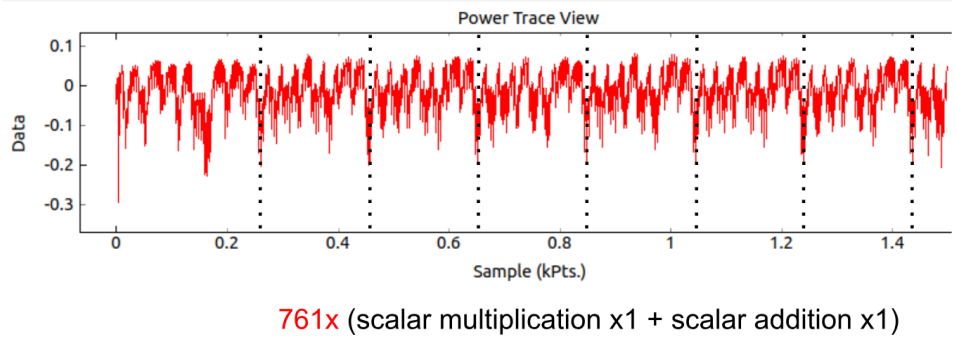


Figure 4: The Pattern of Multiplication-Addition Pairs in a Power Trace

```
maxCorr = -0.173772 | Term 286:   +5, sampleId:   924
maxCorr = -0.224462 | Term 286:   +5, sampleId:   935
maxCorr = -0.305537 | Term 286:   +5, sampleId:  1030
maxCorr = -0.327332 | Term 286:   +5, sampleId:  1620
maxCorr = -0.357381 | Term 286:   +5, sampleId:  1622
maxCorr = -0.399957 | Term 286:   +4, sampleId:  1843
maxCorr = -0.547090 | Term 286:   +3, sampleId:  1970
maxCorr = -0.743208 | Term 286:   -2, sampleId:  1264
maxCorr = -0.946183 | Term 286:   -2, sampleId:  1266
maxCorr = -0.989188 | Term 286:   -2, sampleId:  1312
maxCorr = -0.992438 | Term 286:   -2, sampleId:  1314
```

Figure 5: The Recovery of the 286th Nonzero Coefficient

```
Term 270 =  +36 | CORR: -0.991435
Term 271 =  -34 | CORR: -0.995749
Term 272 =  -31 | CORR: -0.986507
Term 273 =  +30 | CORR: -0.995044
Term 274 =  +29 | CORR: -0.982210
Term 275 =  +28 | CORR: -0.995708
Term 276 =  +24 | CORR: -0.984790
Term 277 =  +22 | CORR: -0.951976
Term 278 =  -21 | CORR: -0.994351
Term 279 =  +17 | CORR: -0.985130
Term 280 =  -15 | CORR: -0.958338
Term 281 =  -14 | CORR: -0.969362
Term 282 =  -13 | CORR: -0.966433
Term 283 =  -12 | CORR: -0.991088
Term 284 =   -9 | CORR: -0.994276
Term 285 =   +6 | CORR: -0.993532
Term 286 =   -2 | CORR: -0.992438
```

Figure 6: The Result of the Proposed CPA

4 Software Countermeasures against the Proposed Approach

4.1 Methodology

There are at least three defensive strategies available for software implementations. While their prototypes are first introduced in [LSCH10], their implementations keep evolving:

1. the random initialization of e_i
2. the randomized access to (c_j, f_{i-j}) pairs

3. a first-order masking scheme

The first countermeasure assigns a random integer $m_i \in \mathbb{Z}_q$ instead of 0 to $e_i, \forall i \in \{0, 1, ..., (2p-2)\}$ at the beginning of Algorithm 1, and removes all the m_i using modular subtractions at the end of Algorithm 1. In the second countermeasure, Algorithm 1 receives one more argument $Perm[p]$: a random permutation of $\{0, 1, \ldots, (p-1)\}$. During the calculation of e_i, the program iterates from $j = 0$ to $j = (p-1)$, adding the appropriate $c_{Perm[j]} \times f_{i-Perm[j]}$ to e_i. The third countermeasure can be briefly expressed as:

$$\begin{aligned}
&\text{Inputs: } m_f, m_c \leftarrow \text{random masks} \in R_q;\ \bar{f} = f + m_f;\ \bar{c} = c + m_c \\
&\text{Outputs: } m_d = -m_f \times m_c;\ \bar{d} = d + m_d = f \times c - m_f \times m_c \\
&\text{Algorithm: } D_1 = \bar{f} \times \bar{c};\ D_2 = \bar{f} \times m_c;\ D_3 = m_f \times \bar{c};\ \bar{d} = D_1 - D_2 - D_3
\end{aligned}$$

The above additions, subtractions, and multiplications are in R_q. Every polynomial multiplication in this design follows Algorithm 1. The three defensive strategies above are all able to protect the reference implementation of NTRU Prime from the analysis in this work.

The following comparison between these countermeasures manifests a tradeoff between security and performance. Here the discussion focuses on the execution of Algorithm 1. The initialization of e_i is the least time-consuming, since it only requires p extra subtractions in $\mathbb{Z}_q$, whose time consumption is negligible compared with that of polynomial multiplications in R_q. Yet both the timing and the trace pattern of any access to (c_j, f_{i-j}) as well as the subsequent multiplication remain the same as before, and full private-key recovery from NTRU Prime with vertical power analysis remains possible due to these highly data-dependent leakages.

The randomized access to (c_j, f_{i-j}) pairs requires $(2p^2 - p)$ extra comparisons, and the increase in time consumption is again negligible. Without trace rearrangement, the variation in each sample on the power traces of this countermeasure is random enough to conceal its dependence on f and c. However, the set of (c_j, f_{i-j}) involved in the calculation of e_i remains the same, so the entire trace is a random permutation of p (almost) fixed tiny traces. For example, the traces in the case of $c_i \neq 0, \forall i \in \{0, 1, ..., (p-1)\}$ are distinct from those in the case of $c_i = 0$, for most i. To summarize, this countermeasure has the potential to pass widely used leakage assessments, which focus on the protection against vertical power analysis. Unfortunately, it is subject to more delicate attacks involving strategic ciphertext selection and horizontal power analysis [Wal01, CFG+10].

The masking scheme is the most time-consuming and secure countermeasure. D_2 only depends on the two masks m_f, m_c and the private key f, so NTRU Prime is able to compute D_2 ahead in its key schedule. An update of D_2 is unnecessary until a regeneration of private key or mask pair. Thus, compared with the unprotected version, this countermeasure requires one extra polynomial multiplication in R_q to compute D_3 as well as $2p$ extra subtractions in $\mathbb{Z}_q$ to compute $\bar{d}$, and it takes around twice as long to complete the entire computation. In this design, the intermediate values processed by the target device are statistically independent of their counterparts at algorithm level, which renders most first-order power analyses harmless [MOP07].

4.2 Experiments, Results, and Discussion

Here the three C implementations of defensive strategies are sampled at 29.538MS/s, and every power trace contains 24,400 samples. There are two classes of traces: P1 for the validation of the protection against Algorithm 2 and P2 for the evaluation of side-channel resistance. Each trace in P1 refers to the beginning of a countermeasure. The reason behind is that the CPA is doomed to fail at the second stage if it has failed at the first stage, and if the CPA succeeds at the first stage, the countermeasure under examination

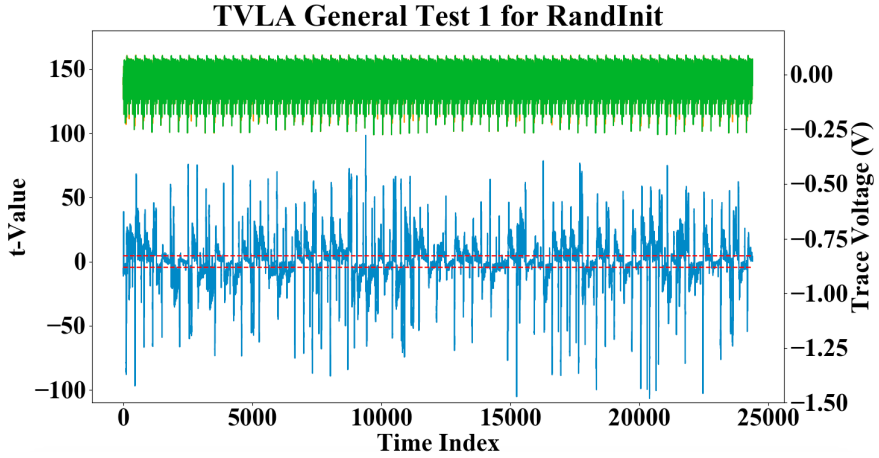
fails. Among the traces in P2, those from the first two countermeasures are centered at the midpoint of the entire computation, while those from the first-order masking scheme are centered at the first quarter, the midpoint, or the third quarter of the entire computation. The TVLA *general* test is programmed in Python 3.6.1 (macOS Sierra 10.12.6, MacBook Air). All the other settings follow the measurement setup in subsection 3.2.

As expected, Algorithm 2 works for none of the three countermeasures due to the randomized updates of e_i during polynomial multiplications in R_q. The CPA uses 5,000 traces of type P1 in each trial but in vain. In these countermeasures, the correct guess at the first stage respectively gives the correlation coefficients -0.057477, -0.068498, and -0.055104, all way higher than the threshold -0.90.

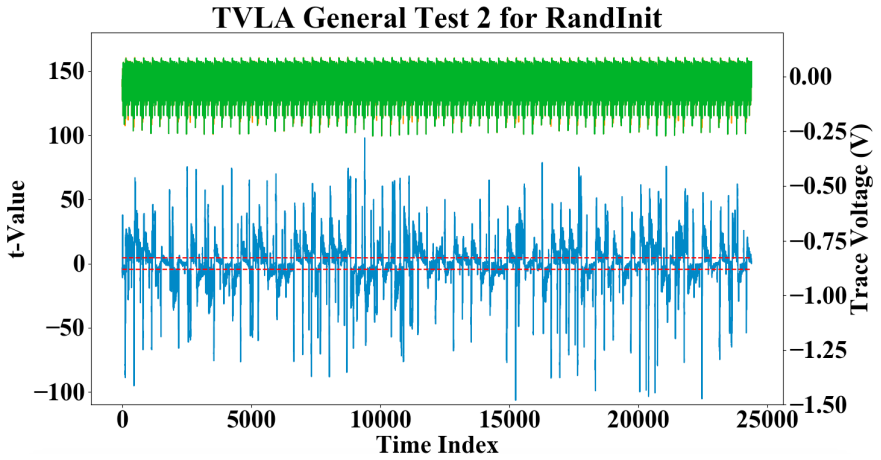
In the TVLA *general* tests below, each given set of side-channel measurements contains 20,000 power traces of type P2. These tests adopt the secret key and fixed plaintext in the *general* test for AES-128 [GJJR11] to instantiate the pseudorandom generation of the private key f_T and fixed input c_T. The test results for the three defensive strategies are shown in Figure 7–Figure 9. In each figure, the upper half is the average power trace, while the lower half is the t-statistic trace. Both halves share the same x-axis. However, the y-axis of the upper one is "Trace Voltage (V)", that of the lower one "t-Value". In Figure 7, each "t-Value" axis ranges from -100 to 150, while in Figure 8 and Figure 9, each ranges from -5.0 to 10.0. The two dotted red lines indicate the thresholds ± 4.5.

According to the two independent tests shown in Figure 7, the random initialization of e_i fails the TVLA *general* test. In both t-statistic traces, noticeable peaks appear at the beginning of every part corresponding to a multiplication-addition pair. This observation supports the vulnerability prediction in subsection 4.1.

According to Figure 8 and Figure 9, the other two defensive strategies both pass the TVLA *general* test. It is not surprising because the same time index in different power traces refers to different (c_j, f_{i-j}) pairs or a randomly masked intermediate value in these countermeasures.

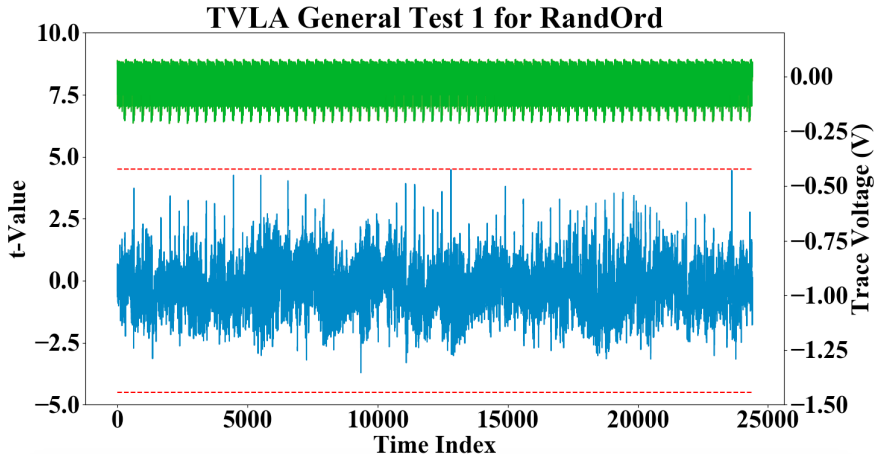


(a) The First Independent Test

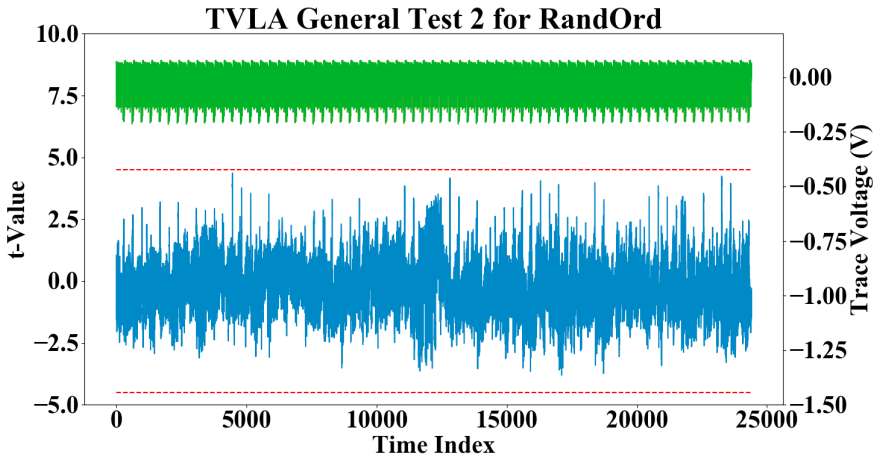


(b) The Second Independent Test

Figure 7: The TVLA General Test Results for the Random Initialization of e_i

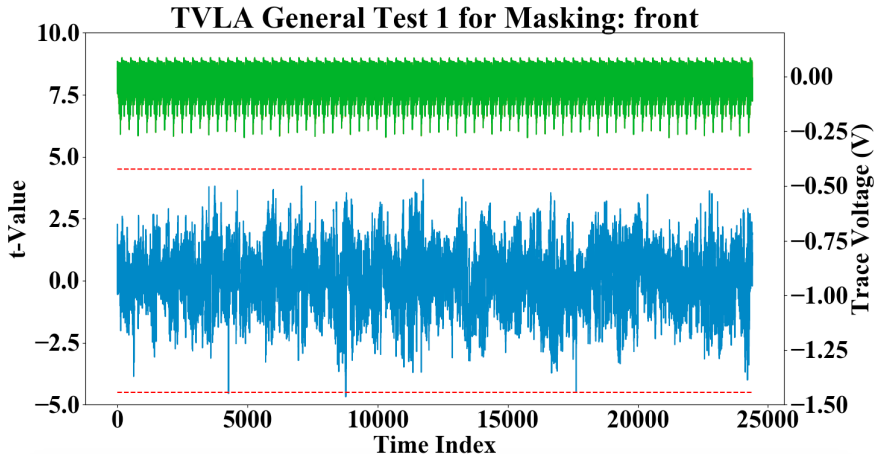


(a) The First Independent Test

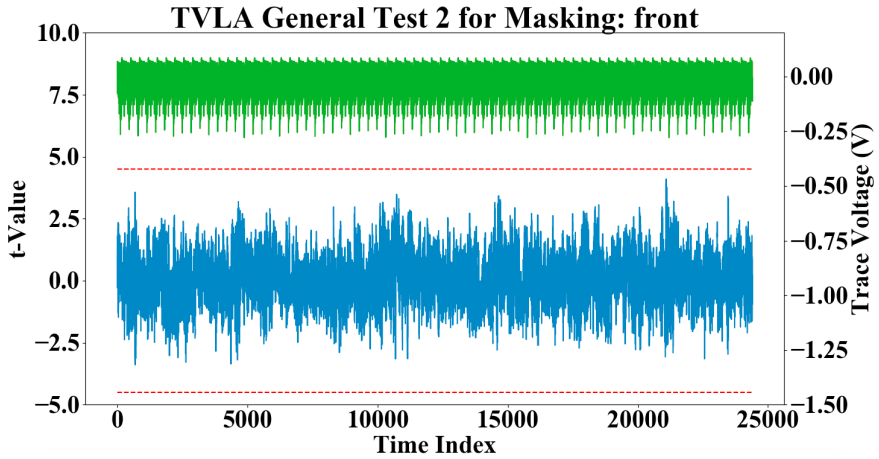


(b) The Second Independent Test

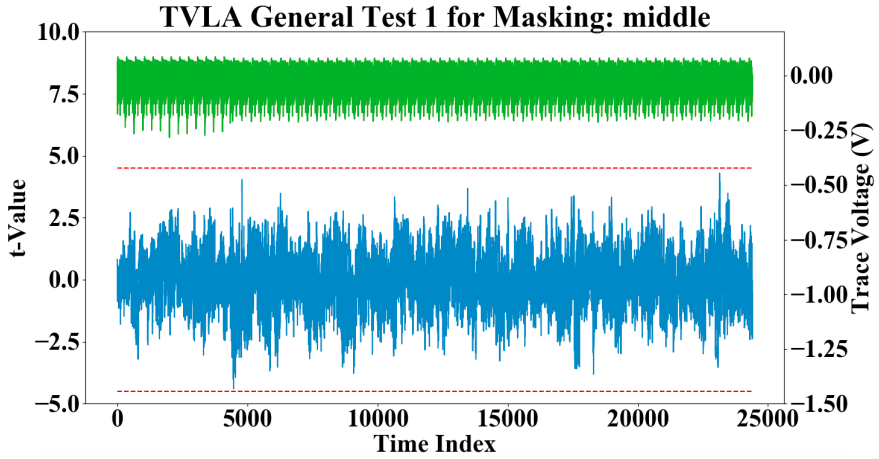
Figure 8: The TVLA General Test Results for the Randomized Access to (c_j, f_{i-j}) Pairs



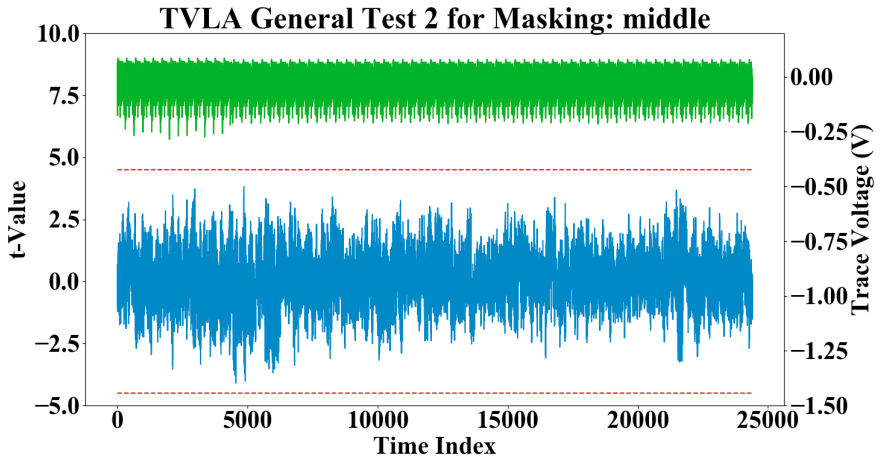
(a) The First Independent Test at the First Quarter



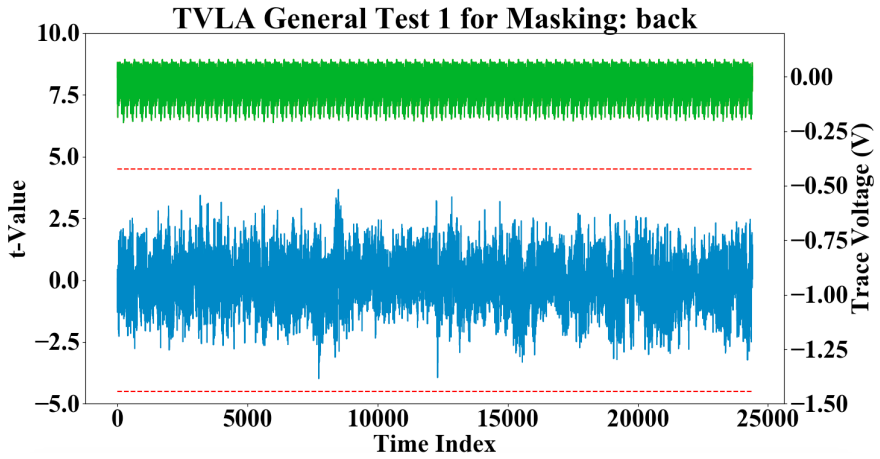
(b) The Second Independent Test at the First Quarter



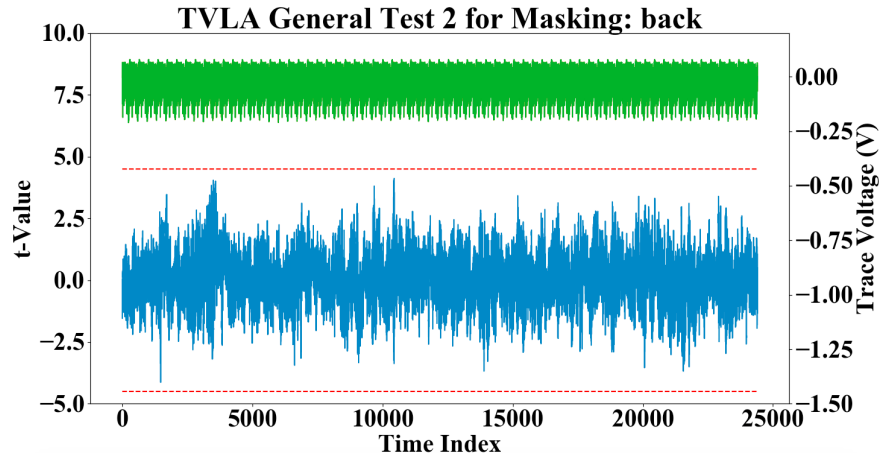
(c) The First Independent Test at the Midpoint



(d) The Second Independent Test at the Midpoint



(e) The First Independent Test at the Third Quarter



(f) The Second Independent Test at the Third Quarter

Figure 9: The TVLA General Test Results for the Masking Scheme

The unprotected implementation [BCLvV17b, KR+18] takes 65,546 clock cycles to

complete the computation. By contrast, It takes 65,596 clock cycles, 69,353 clock cycles, and 122,765 clock cycles for the random initialization of e_i, the randomized access to (c_j, f_{i-j}) pairs, and the first-order masking scheme to complete the computation, respectively. These statistics display the tradeoff referred to in subsection 4.1.

5 Conclusion

NTRU Prime, a Round 2 candidate in the NIST Post-Quantum Cryptography Competition, realizes the polynomial multiplication in R_q in a generic manner inside its constant-time reference C implementation. Despite the seeming side-channel resistance of this implementation, the correlation power analysis in this paper can recover private keys from both schemes in this submission with few power traces. This carefully designed analysis is practically fast and asymptotically efficient. Moreover, it is a generic approach applicable to other ideal-lattice-based cryptographic primitives realizing polynomial multiplications with the product-scanning method. If there were microcontroller implementations for ideal-lattice-based cryptosystems using Karatsuba [Paa96, WP06] at the high level and the product-scanning method on the low level, this CPA should also work. To protect the reference implementation, three countermeasures have been proposed. These defensive strategies manifest the tradeoff between performance and security. Their effectiveness and such tradeoff have been respectively validated by the CPA and the TVLA *general* tests based on thousands of power traces.

References

[AC18] Carlisle Adams and Jan Camenisch, editors. *Selected Areas in Cryptography - SAC 2017 - 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017, Revised Selected Papers*, volume 10719 of *Lecture Notes in Computer Science*. Springer, 2018.

[ADPS16] Erdem Alkim, Léo Ducas, Thomas Pöppelmann, and Peter Schwabe. Post-quantum key exchange - A New Hope. In Thorsten Holz and Stefan Savage, editors, *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, pages 327–343. USENIX Association, 2016.

[AKJ+18] Soojung An, Suhri Kim, Sunghyun Jin, HanBit Kim, and HeeSeok Kim. Single trace side channel analysis on NTRU implementation. *Applied Sciences*, 8(11):2014, 2018.

[ATT+18] Aydin Aysu, Youssef Tobah, Mohit Tiwari, Andreas Gerstlauer, and Michael Orshansky. Horizontal side-channel vulnerabilities of post-quantum key exchange protocols. In *2018 IEEE International Symposium on Hardware Oriented Security and Trust, HOST 2018, Washington, DC, USA, April 30 - May 4, 2018*, pages 81–88. IEEE Computer Society, 2018.

[Bar13] Elaine Barker. FIPS pub 186-4: Digital signature standard (DSS). Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States, July 2013.

[BBC+18] Marco Baldi, Alessandro Barenghi, Franco Chiaraluce, Gerardo Pelosi, and Paolo Santini. LEDAkem: A post-quantum key encapsulation mechanism based on QC-LDPC codes. In Tanja Lange and Rainer Steinwandt, editors, *Post-Quantum Cryptography - 9th International Conference, PQCrypto 2018, Fort Lauderdale, FL, USA, April 9-11, 2018, Proceedings*, volume 10786 of *Lecture Notes in Computer Science*, pages 3–24. Springer, 2018.

[BCD+16] Joppe W. Bos, Craig Costello, Léo Ducas, Ilya Mironov, Michael Naehrig, Valeria Nikolaenko, Ananth Raghunathan, and Douglas Stebila. Frodo: Take off the ring! practical, quantum-secure key exchange from LWE. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 1006–1018. ACM, 2016.

[BCLvV17a] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU Prime: Reducing attack surface at low cost. In Adams and Camenisch [AC18], pages 235–260.

[BCLvV17b] Daniel J. Bernstein, Chitchanok Chuengsatiansup, Tanja Lange, and Christine van Vredendaal. NTRU_Prime.zip, 2017.

[BCO04] Eric Brier, Christophe Clavier, and Francis Olivier. Correlation power analysis with a leakage model. In Marc Joye and Jean-Jacques Quisquater, editors, *Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings*, volume 3156 of *Lecture Notes in Computer Science*, pages 16–29. Springer, 2004.

[BCRS13] Elaine Barker, Lily Chen, Allen Roginsky, and Miles Smid. SP 800-56A: Recommendation for pair-wise key establishment schemes using discrete logarithm cryptography (revision 2). Technical report, National Institute of Standards and Technology, Gaithersburg, MD, United States, May 2013.

[BDK+18] Joppe W. Bos, Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, John M. Schanck, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM. In *2018 IEEE European Symposium on Security and Privacy, EuroS&P 2018, London, United Kingdom, April 24-26, 2018*, pages 353–367. IEEE, 2018.

[BHH+15] Daniel J. Bernstein, Daira Hopwood, Andreas Hülsing, Tanja Lange, Ruben Niederhagen, Louiza Papachristodoulou, Michael Schneider, Peter Schwabe, and Zooko Wilcox-O'Hearn. SPHINCS: practical stateless hash-based signatures. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part I*, volume 9056 of *Lecture Notes in Computer Science*, pages 368–397. Springer, 2015.

[CFG+10] Christophe Clavier, Benoit Feix, Georges Gagnerot, Mylène Roussellet, and Vincent Verneuil. Horizontal correlation analysis on exponentiation. In Miguel Soriano, Sihan Qing, and Javier López, editors, *Information and Communications Security - 12th International Conference, ICICS 2010, Barcelona, Spain, December 15-17, 2010. Proceedings*, volume 6476 of *Lecture Notes in Computer Science*, pages 46–61. Springer, 2010.

[DH76] Whitfield Diffie and Martin E. Hellman. New directions in cryptography. *IEEE Trans. Information Theory*, 22(6):644–654, 1976.

[DKL+18] Léo Ducas, Eike Kiltz, Tancrède Lepoint, Vadim Lyubashevsky, Peter Schwabe, Gregor Seiler, and Damien Stehlé. CRYSTALS-Dilithium: A lattice-based digital signature scheme. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(1):238–268, 2018.

[DS05] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariable polynomial signature scheme. In John Ioannidis, Angelos D. Keromytis, and Moti Yung, editors, *Applied Cryptography and Network Security, Third International Conference, ACNS 2005, New York, NY, USA, June 7-10, 2005, Proceedings*, volume 3531 of *Lecture Notes in Computer Science*, pages 164–175, 2005.

[EFGT17] Thomas Espitau, Pierre-Alain Fouque, Benoît Gérard, and Mehdi Tibouchi. Side-channel attacks on BLISS lattice-based signatures: Exploiting branch tracing against strongswan and electromagnetic emanations in microcontrollers. In Bhavani M. Thuraisingham, David Evans, Tal Malkin, and Dongyan Xu, editors, *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, pages 1857–1874. ACM, 2017.

[Gam85] Taher El Gamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Information Theory*, 31(4):469–472, 1985.

[GJJR11] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. A testing methodology for side-channel resistance validation. In *NIST Non-Invasive Attack Testing Workshop*, volume 7, pages 115–136, 2011.

[HPS98] Jeffrey Hoffstein, Jill Pipher, and Joseph H. Silverman. NTRU: A ring-based public key cryptosystem. In Joe Buhler, editor, *Algorithmic Number Theory, Third International Symposium, ANTS-III, Portland, Oregon, USA, June 21-25, 1998, Proceedings*, volume 1423 of *Lecture Notes in Computer Science*, pages 267–288. Springer, 1998.

[HW11] Michael Hutter and Erich Wenger. Fast multi-precision multiplication for public-key cryptography on embedded microprocessors. In Bart Preneel and Tsuyoshi Takagi, editors, *Cryptographic Hardware and Embedded Systems - CHES 2011 - 13th International Workshop, Nara, Japan, September 28 - October 1, 2011. Proceedings*, volume 6917 of *Lecture Notes in Computer Science*, pages 459–474. Springer, 2011.

[JF11] David Jao and Luca De Feo. Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies. In Bo-Yin Yang, editor, *Post-Quantum Cryptography - 4th International Workshop, PQCrypto 2011, Taipei, Taiwan, November 29 - December 2, 2011. Proceedings*, volume 7071 of *Lecture Notes in Computer Science*, pages 19–34. Springer, 2011.

[JMV01] Don Johnson, Alfred Menezes, and Scott A. Vanstone. The elliptic curve digital signature algorithm (ECDSA). *Int. J. Inf. Sec.*, 1(1):36–63, 2001.

[KAJ17] Brian Koziel, Reza Azarderakhsh, and David Jao. Side-channel attacks on quantum-resistant supersingular isogeny Diffie-Hellman. In Adams and Camenisch [AC18], pages 64–81.

[KJJ99] Paul C. Kocher, Joshua Jaffe, and Benjamin Jun. Differential power analysis. In Michael J. Wiener, editor, *Advances in Cryptology - CRYPTO '99, 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 15-19, 1999, Proceedings*, volume 1666 of *Lecture Notes in Computer Science*, pages 388–397. Springer, 1999.

[Koc96] Paul C. Kocher. Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In Neal Koblitz, editor, *Advances in Cryptology*

- *CRYPTO '96, 16th Annual International Cryptology Conference, Santa Barbara, California, USA, August 18-22, 1996, Proceedings*, volume 1109 of *Lecture Notes in Computer Science*, pages 104–113. Springer, 1996.

[KR+18] Matthias J. Kannwischer, Joost Rijneveld, et al. Post-quantum crypto library for the ARM Cortex-M4, 2018.

[KY12] Abdel Alim Kamal and Amr M. Youssef. A scan-based side channel attack on the NTRUEncrypt cryptosystem. In *Seventh International Conference on Availability, Reliability and Security, Prague, ARES 2012, Czech Republic, August 20-24, 2012*, pages 402–409. IEEE Computer Society, 2012.

[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*, 93-A(1):153–163, 2010.

[McE78] Robert J McEliece. A public-key cryptosystem based on algebraic. *Coding Thv*, 4244:114–116, 1978.

[MD99] Thomas S. Messerges and Ezzy A. Dabbish. Investigations of power analysis attacks on smartcards. In Scott B. Guthery and Peter Honeyman, editors, *Proceedings of the 1st Workshop on Smartcard Technology, Smartcard 1999, Chicago, Illinois, USA, May 10-11, 1999*. USENIX Association, 1999.

[MNY17] Moustafa Mahmoud, Mouna Nakkar, and Amr Youssef. A power analysis resistant FPGA implementation of NTRUEncrypt. In *Microelectronics (ICM), 2017 29th International Conference on*, pages 1–4. IEEE, 2017.

[MOP07] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. *Power analysis attacks - revealing the secrets of smart cards*. Springer, 2007.

[Nat17] National Institute of Standards and Technology. Post-quantum cryptography, 2017.

[O+18] Colin O'Flynn et al. ChipWhisperer - the complete open-source toolchain for side-channel power analysis and glitching attacks, 2018.

[O'F16] Colin O'Flynn. ChipWhisperer-Lite (CW1173) two-part version, 2016.

[Paa96] Christof Paar. A new architecture for a parallel finite field multiplier with low complexity based on composite fields. *IEEE Trans. Computers*, 45(7):856–861, 1996.

[PPM17] Robert Primas, Peter Pessl, and Stefan Mangard. Single-trace side-channel attacks on masked lattice-based encryption. In Wieland Fischer and Naofumi Homma, editors, *Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings*, volume 10529 of *Lecture Notes in Computer Science*, pages 513–533. Springer, 2017.

[PSKH18] Aesun Park, Kyung-Ah Shim, Namhun Koo, and Dong-Guk Han. Side-channel attacks on post-quantum signature schemes based on multivariate quadratic equations - rainbow and UOV -. *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, 2018(3):500–523, 2018.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, 1978.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997.

[Sni16] Brian Sniffen. Akamai faster forward crypto at scale, 2016.

[STM18] STMicroelectronics. *Datasheet - STM32F303xB STM32F303xC*, October 2018. Revision 14.

[SW07] Joseph H. Silverman and William Whyte. Timing attacks on NTRUEncrypt via variation in the number of hash calls. In Masayuki Abe, editor, *Topics in Cryptology - CT-RSA 2007, The Cryptographers' Track at the RSA Conference 2007, San Francisco, CA, USA, February 5-9, 2007, Proceedings*, volume 4377 of *Lecture Notes in Computer Science*, pages 208–224. Springer, 2007.

[TE15] Mostafa Taha and Thomas Eisenbarth. Implementation attacks on post-quantum cryptographic schemes. *IACR Cryptology ePrint Archive*, 2015:1083, 2015.

[vE85] Wim van Eck. Electromagnetic radiation from video display units: An eavesdropping risk? *Computers & Security*, 4(4):269–286, 1985.

[Wal01] Colin D. Walter. Sliding windows succumbs to Big Mac attack. In Çetin Kaya Koç, David Naccache, and Christof Paar, editors, *Cryptographic Hardware and Embedded Systems - CHES 2001, Third International Workshop, Paris, France, May 14-16, 2001, Proceedings*, volume 2162 of *Lecture Notes in Computer Science*, pages 286–299. Springer, 2001.

[Wel47] Bernard L. Welch. The generalization of Student's problem when several different population variances are involved. *Biometrika*, 36:293–296, January 1947.

[WLYZ18] Yuanhao Wang, Ying Li, Zhangqi Yin, and Bei Zeng. 16-qubit IBM universal quantum computer can be fully entangled. *npj Quantum Information*, 4(1):46, 2018.

[WP06] André Weimerskirch and Christof Paar. Generalizations of the Karatsuba algorithm for efficient implementations. *IACR Cryptology ePrint Archive*, 2006:224, 2006.

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

[ZWW13] Xuexin Zheng, An Wang, and Wei Wei. First-order collision attack on protected NTRU cryptosystem. *Microprocessors and Microsystems - Embedded Hardware Design*, 37(6-7):601–609, 2013.