J. Math. Cryptol. ? (????), 1–23
DOI 10.1515/jmc-????-???

(In)Security of Ring-LWE Under Partial Key Exposure

Dana Dachman-Soled, Huijing Gong, Mukul Kulkarni and Aria Shahverdi

Communicated by ???

Abstract. We initiate the study of partial key exposure in Ring-LWE (RLWE)-based cryptosystems. Specifically, we (1) Introduce the search and decision Leaky R-LWE assumptions (Leaky R-SLWE, Leaky R-DLWE), to formalize the hardness of search/decision RLWE under leakage of some fraction of coordinates of the NTT transform of the RLWE secret. (2) Present and implement an efficient key exposure attack that, given certain $1/4$-fraction of the coordinates of the NTT transform of the RLWE secret, along with samples from the RLWE distribution, recovers the full RLWE secret for standard parameter settings. (3) Present a search-to-decision reduction for Leaky R-LWE for certain types of key exposure. (4) Propose applications to the security analysis of RLWE-based cryptosystems under partial key exposure.

Keywords. Lattice-based cryptography, leakage resilience, Ring-LWE, partial key exposure.

2010 Mathematics Subject Classification. 94A60,68P25,03G10.

1 Introduction

There has been a monumental effort in the cryptographic community to develop "post-quantum" cryptosystems that remain secure even in the presence of a quantum adversary. One of the foremost avenues for viable post-quantum public key cryptography is to construct schemes from the Ring-Learning with Error (RLWE) assumption—currently 3 out of 26 of the second round NIST submissions are based on assumptions in the ring setting. RLWE is often preferred in practice over standard LWE due to its algebraic structure, which allows for smaller public keys and more efficient implementations. In the RLWE setting, we typically consider rings of the form $R_q := \mathbb{Z}_q[x]/(x^n + 1)$, where n is a power of two

This work is supported in part by NSF grants #CNS-1840893, #CNS-1453045 (CAREER), by a research partnership award from Cisco and by financial assistance award 70NANB15H328 from the U.S. Department of Commerce, National Institute of Standards and Technology.

and $q \equiv 1 \mod 2n$. The (decisional) RLWE problem is then to distinguish $(a, b = a \cdot s + e) \in R_q \times R_q$ from uniformly random pairs, where $s \in R_q$ is a random secret, $a \in R_q$ is uniformly random and the error term $e \in R$ has small norm. A critical question is whether the additional algebraic structure of the RLWE problem renders it less secure than the standard LWE problem. Interestingly, to the best of our knowledge—for the rings used in practice and practical parameter settings—the best attacks on RLWE are generic and can equally well be applied to standard LWE [28]. In this work, we ask whether improved attacks on RLWE are possible when partial information about the RLWE secret is exposed, though the secret retains high entropy.

The NTT transform. One key method for speeding up computations in the RLWE setting is usage of the *NTT transform* (similar to the discrete Fourier transform (DFT), but over finite fields) to allow for faster polynomial multiplication over the ring R_q. Specifically, applying the NTT transform to two polynomials $\mathbf{p}, \mathbf{p}' \in R_q$—resulting in two n-dimensional vectors, $\widehat{\mathbf{p}}, \widehat{\mathbf{p}}' \in \mathbb{Z}_q^n$—allows for *component-wise* multiplication and addition, which is highly efficient. In this work, we consider leakage of a fraction of NTT coordinates of the RLWE secret. Since the RLWE secret will typically be stored in NTT form (to facilitate fast computation), [4,7] leakage of coordinates of the NTT transform is a natural model for partial key exposure attacks.

This work. The goal of this work is to initiate a study of partial key exposure in RLWE based cryptosystems and explore both positive and negative results in this setting. Specifically, we (1) define search and decision versions of Leaky RLWE assumptions, where the structured leakage occurs on the coordinates of the NTT transform of the RLWE secret; (2) present partial key exposure attacks on RLWE, given $1/4$-fraction of structured leakage on the secret key; (3) present a search to decision reduction for the Leaky RLWE assumptions; and (4) propose applications of the decision version of the assumption to practical RLWE-based cryptosystems.

1.1 Leaky RLWE Assumptions–Search and Decision Versions

We next briefly introduce the search and decision versions of the Leaky RLWE assumptions. For $\mathbf{p} \in R_q := \mathbb{Z}_q[x]/(x^n + 1)$, we denote $\widehat{\mathbf{p}} := \mathsf{NTT}(\mathbf{p}) := (\mathbf{p}(\omega^1), \mathbf{p}(\omega^3), \ldots, \mathbf{p}(\omega^{2n-1}))$, where ω is a primitive $2n$-th root of unity modulo q, and is guaranteed to exist by choice of prime q, s.t. $q \equiv 1 \mod 2n$. Note that $\widehat{\mathbf{p}}$ is indexed by the set $\mathbb{Z}_{2n}^*$.

The search version of the RLWE problem with leakage, denoted Leaky R-SLWE, is parametrized by $(n' \in \{1, 2, 4, 8, \ldots n\}, \mathcal{S} \subseteq \mathbb{Z}_{2n'}^*)$. The goal is to recover the RLWE secret $\mathbf{s} = \mathsf{NTT}^{-1}(\widehat{\mathbf{s}})$, given samples from the distribution

$D^{\mathbf{s}}_{real,n',\mathcal{S}}$ which outputs $\left(\widehat{\mathbf{a}}, \widehat{\mathbf{a}} \cdot \widehat{\mathbf{s}} + \widehat{\mathbf{e}}, [\widehat{s_i}]_{i \equiv \alpha \bmod 2n' \,|\forall \alpha \in \mathcal{S}}\right)$, where $\mathbf{a}, \mathbf{s}$, and $\mathbf{e}$ are as in the standard RLWE assumption (see Appendix A.2 and [26] for the precise definition).

The decision version of the RLWE problem with leakage, denoted Leaky R-DLWE is parametrized by $(n' \in \{1, 2, 4, 8, \ldots n\}, \mathcal{S} \subseteq \mathbb{Z}^*_{2n'})$. The goal is to distinguish the distributions $D^{\mathbf{s}}_{real,n',\mathcal{S}}$ and $D^{\mathbf{s}}_{sim,n',\mathcal{S}}$, where $D^{\mathbf{s}}_{real,n',\mathcal{S}}$ is as above and $D^{\mathbf{s}}_{sim,n',\mathcal{S}}$ outputs $\left(\widehat{\mathbf{a}}, \widehat{\mathbf{u}}, [\widehat{s_i}]_{i \equiv \alpha \bmod 2n' \,|\forall \alpha \in \mathcal{S}}\right)$, where $\widehat{u_i} = \widehat{a_i} \cdot \widehat{s_i} + \widehat{e_i}$ for $i \equiv \alpha \mod 2n'$, $\alpha \in \mathcal{S}$ and $\widehat{u_i}$ is chosen uniformly at random from $\mathbb{Z}_q$, otherwise. Note that only the coordinates of $\widehat{u}$ corresponding to *unleaked* positions are required to be indistinguishable from random.

When $\mathcal{S} = \{\alpha\}$ consists of a single element, we sometimes abuse notation and write the Leaky-RLWE parameters as (n', α). Leaky-RLWE with parameters $(n', \mathcal{S})$ where $\mathcal{S} = \{\alpha_1, \alpha_2, \ldots, \alpha_t\}$, is equivalent to Leaky-RLWE with parameters $(n', \mathcal{S}')$, where $\mathcal{S}' = \alpha_1^{-1} \cdot S$ (multiply every element of $\mathcal{S}$ by α_1^{-1}). It is also not hard to see that leaky search and decision are equally hard when secret $\mathbf{s}$ is uniform random from R_q versus drawn from the error distribution (the same reduction for standard RLWE works in our case).

1.2 Our Results

Partial key exposure attacks. We present attacks on Leaky R-SLWE and test them on various practical parameter settings, such as the NewHope [7] parameter settings as well as the RLWE challenges of Crockett and Peikert [12]. Our attacks demonstrate that Leaky R-SLWE is *easy* for leakage parameters $(n' = 4, \alpha = 1)$, $(n' = 8, \mathcal{S} = \{1, 7\})$ and $(n' = 8, \mathcal{S} = \{1, 15\})$, under (1) NewHope parameter settings of $n = 1024$, $q = 12289$, and $\chi = \Psi_{16}$ (centered binomial distribution of parameter 16); (2) The same parameters above, but with $\chi = D_{\sqrt{8}}$ (discrete Gaussian with standard deviation of $\sqrt{8}$, which has the same standard deviation as Ψ_{16}), since this is the recommended setting in the case where the adversary gets to see *many* RLWE samples [3]; (3) For parameters of several of the Crockett and Peikert challenges, including those classified as "very hard." In all the above cases, we fully recover the RLWE secret with high probability, given the corresponding $1/4$-fraction of the positions in the NTT transform of the RLWE secret. See Section 3.2 for details on the experimental results.

A search-to-decision reduction. Define $T_{n'}(n)$ to be the time required to solve Leaky R-SLWE for dimension n, given positions $[\hat{s}_i]_{i \equiv \alpha \mod 2n'}$. Assuming search R-LWE *without leakage* is subexponentially $2^{\Omega(n^\epsilon)}$-hard for some constant $\epsilon \leq 1$

and polynomial modulus q, then $T_n(n) \in 2^{\Omega(n^\epsilon)}$,[1] i.e. there is a constant c such that, for sufficiently large n, $T_n(n) \geq 2^{c(n^\epsilon)}$. Also, $T_1(n) \in \mathsf{poly}(n)$, since the entire $\mathbf{s}$ is leaked. So there is some constant c' such that, for sufficiently large n, there exists $n^* = n^*(n) \in \{2, 4, 8, 16, \ldots, n\}$ such that $T_{n^*}(n) \geq 2^{c'(n^\epsilon)}$ and $\frac{T_{n^*}(n)}{T_{n^*/2}(n)} \geq n$.[2]

Theorem 1.1 (Informal). *Assume $n^* := n^*(n) > 4$, $\mathbf{s} \leftarrow R_q$, then:*

(1) $D^{\mathbf{s}}_{real,n^*,\{\alpha\}} \approx D^{\mathbf{s}}_{sim,n^*,\{\alpha\}}$ *OR*

(2) $D^{\mathbf{s}}_{real,n^*,\{\alpha,(n^*-1)\cdot\alpha\}} \approx D^{\mathbf{s}}_{sim,n^*,\{\alpha,(n^*-1)\cdot\alpha\}}$ *OR*

(3) $D^{\mathbf{s}}_{real,n^*,\{\alpha,(2n^*-1)\alpha\}} \approx D^{\mathbf{s}}_{sim,n^*,\{\alpha,(2n^*-1)\alpha\}}$.

While at first glance it may seem that the conclusions (1), (2), (3) are redundant, in fact they are incomparable; Indeed, conclusion (1) does not imply (2) (resp. (3)), since the adversary in (2) (resp. (3)) is given additional leakage. Conversely, conclusion (2) (resp. (3)) does not imply (1), since the set of NTT coordinates that are indistinguishable from random is smaller in (2).

Note that our experimental results show that for our chosen parameter settings $D^{\mathbf{s}}_{real,4,\{1\}} \not\approx D^{\mathbf{s}}_{sim,4,\{1\}}$, $D^{\mathbf{s}}_{real,8,\{1,7\}} \not\approx D^{\mathbf{s}}_{sim,8,\{1,7\}}$ and $D^{\mathbf{s}}_{real,8,\{1,15\}} \not\approx D^{\mathbf{s}}_{sim,8,\{1,15\}}$ (since we in fact fully recover the secret in all these cases). This indicates that $n^* \neq 4$ and, if $n^* = 8$ for our chosen parameter settings (as supported by our experiments), then it must be the case that $D^{\mathbf{s}}_{real,8,\{1\}} \approx D^{\mathbf{s}}_{sim,8,\{1\}}$.

Applications. The Leaky R-DLWE assumption is a useful tool for analyzing the security of RLWE-based cryptosystems subject to partial key exposure, and guaranteeing a graceful degradation in security. In particular, the Leaky R-DLWE assumption was used to analyze the NewHope protocol of [7] in the ePrint version of this paper [14]. The assumption is applicable to schemes in which the RLWE assumption is used to guarantee that a certain outcome is high-entropy (as opposed to uniform random), such as NewHope without reconciliation [6].

Practicality of our attack. We note that an attack on Leaky R-SLWE yields an attack on standard search R-LWE by guessing each possible leakage outcome, running the Leaky R-SLWE attack and checking correctness of the recovered secret. Therefore, we believe this line of research is interesting beyond the context of

[1] Search R-LWE can be solved given a subroutine that solves Leaky R-SLWE by first guessing the leakage on $\mathbf{s}$, then running the Leaky R-SLWE attack. Thus, by guessing the value of the single leaked position we obtain a $T_n(n) \cdot q$-time attack on search R-LWE *without leakage*.

[2] Otherwise for every $n_1 \in \mathbb{N}$, there exists an $n_2 \geq n_1$ such that $T_{n_2}(n_2) < 2^{c'(n_2^\epsilon)} \cdot n_2^{\log n_2} < 2^{c(n_2^\epsilon)}$.

leakage resilience, since if the attack can be made to work successfully for sufficiently low leakage rate (far lower than the $1/4$-leakage rate of our attacks), then one could potentially obtain an improved attack on standard search R-LWE.

We chose to consider partial exposure of the *NTT transform* of the R-LWE secret, since in practical schemes the secret key is often stored in the NTT domain and certain types of side-channel attacks allow recovering large portions of the secret key stored in memory. E.g., in their analysis of "cold boot attacks" on NTT cryptosystems, Albrecht et al. [4] considered bit-flip rates as low as 0.2%. However, the highly structured leakage required for our attack is unlikely to occur in a practical leakage setting such as a "cold boot attack," where one expects to recover the values of random locations in memory. We leave open the question of reducing the structure of the leakage in our attack. Specifically, as a starting point it will be interesting to see if our attack can extend to leakage patterns of $n' = 16$, $|S| = 4$ or $n' = 32$, $|S| = 8$, etc. While the leakage rate remains the same $(1/4)$ in each case, these patterns capture leakage that is less and less structured, since at the extreme, one can view leakage of a random $1/4$-fraction of the NTT coordinates as an instance of Leaky R-SLWE with parameters $n' = n$ and $|S| = n/4$.[3]

1.3 Comparison with Concurrent Work of Bolboceanu et al. [9]

One of the settings considered by [9] is sampling the RLWE secret from an ideal $I \subseteq qR$. It is straightforward to see that sampling the RLWE secret uniformly at random from R_q and then leaking the NTT coordinates i such that $i = \alpha \mod 2n'$ is *equivalent* to sampling the RLWE secret from the ideal I that contains those elements whose NTT transform is 0 in positions i such that $i = \alpha \mod 2n'$.

Nevertheless, our decisional assumption is *weaker* than the assumption of [9], since [9] require that the entire vector $\mathbf{u}$ be indistinguishable from uniform random, whereas we only require that the NTT transform of $\mathbf{u}$ is indistinguishable from uniform random at the positions i that are not leaked. Our assumption lends itself to a search-to-decision reduction while the assumption of [9] does not. While [9] do provide a direct security reduction for their decisional assumption, the required standard deviation of the error (in polynomial basis, tweaked and scaled by q) is $\omega(q^{1/n'} \cdot n^{3/2})$, which would be far higher than the noise considered in the NewHope and RLWE Challenges settings. In contrast, our assumption can be applied in practical parameter regimes and is sufficient to argue the security of several practical cryptosystems under partial key exposure.

Finally, we compare our attack to that of [9]. For fixed n, q, our attack works for noise regimes that are not covered by the attack of [9]. For example, for NewHope

[3] We thank an anonymous reviewer for bringing this research direction to our attention.

settings of $n = 1024$, $q = 12289$, the attack of [9] has success rate at most $1/1000$ when the standard deviation of noise distribution is less than 0.00562. [4] In contrast, our attack works (with success ranging from 82/200 to 2/1000) when the standard deviation of the noise is $\sqrt{8} \approx 2.83$.[5] Our attack applies only for certain leakage patterns corresponding to certain ideals I, whereas the attack of [9] works for any ideal. The techniques of the two attacks are entirely different. [9] obtain a "good" basis for the ideal via *non-uniform* advice, perform a change of basis and then use Babai's roundoff algorithm to solve the resulting BDD instance. We use the algebraic structure of the problem to convert RLWE instances over high dimension into CVP instances over constant dimension n'. We then exactly solve the CVP instances over constant dimension and determine the "high confidence" solutions that are likely to be the correct values of the RLWE error. Assuming all high confidence solutions are correct, we obtain a noiseless system of linear equations w.r.t. the RLWE secret, allowing efficient recovery of the secret.

1.4 Related Work

Leakage-resilient cryptography. The study of provably secure, leakage-resilient cryptography was introduced by the work of Dziembowski and Pietrzak in [19]. Pietrzak [29] also constructed a leakage-resilient stream-cipher. Brakerski et al. [11] showed how to construct a scheme secure against an attacker who leaks at each time period. There are other works as well considering continual leakage [17,22]. There are also work on leakage-resilient signature scheme [10,21,27].
Leakage-resilience and Lattice-based Cryptography. Goldwasser et al. [20], and subsequently [2, 16, 18] studied the leakage resilience of standard LWE based cryptosystems in the symmetric and public key settings.
Leakage Resilience of Ring-LWE. Dachman-Soled et al. [13] considered the leakage resilience of a RLWE-based public key encryption scheme for specific leakage profiles. This was followed by Albrecht et al. [4], they investigated cold boot attacks and compared the number of operations for implementing the attack when the secret key is stored as polynomial coefficients versus when encoding of the secret key using a number theoretic transform (NTT) is stored in memory. Recently, [30] showed that given multiple samples of RLWE instances such that the

[4] Note that [9] provides an upper bound of norm of error with respect to canonical basis for its attack to succeed. Using a variant of Chernoff's bound, we derive an upper bound of standard deviation of error for success rate at most 1/1000. To make the bound comparable to NewHope setting, we further convert to tweaked polynomial representation and to RLWE instance in the form of $(as + e)$ instead of $(as/q + e)$.

[5] $\sqrt{8}$ is the more conservative setting in the original NewHope specification [7]. The NIST submission uses lower standard deviation of 2, which is still not covered by the attack of [9].

public key for every instance lies in some specific subring, one can reduce the original RLWE problem to multiple independent RLWE problems over the subring. In this work we do not place any such restriction on the RLWE samples required to mount partial key exposure attack.

2 Preliminaries

For a positive integer n, we denote by $[n]$ the set $\{0, \ldots, n-1\}$. We denote vectors in boldface $\mathbf{x}$ and matrices using capital letters $\mathbf{A}$. For vector $\mathbf{x}$ over $\mathbb{R}^n$ or $\mathbb{C}^n$, define the ℓ_2 norm as $\|\mathbf{x}\|_2 = \left(\sum_i |x_i|^2\right)^{1/2}$. We write as $\|\mathbf{x}\|$ for simplicity. We use the notation $\approx_{t(n),p(n)}$ to indicate that adversaries running in time $t(n)$ can distinguish two distributions with probability at most $p(n)$.

We present the background and standard definitions related to lattices, algebraic number theory, RLWE, and NTT transform in Appendix A.

3 Partial Key Exposure Attack on Ring-LWE

3.1 Reconstructing the secret given ($\alpha \mod 8$) leakage.

Recall that for $\mathbf{p} \in \mathbb{Z}_q[x]/(x^n+1)$, the NTT transform, $\widehat{\mathbf{p}}$, is obtained by evaluating $\mathbf{p}(x) \bmod q$ at the powers ω^i for $i \in \mathbb{Z}_{2n}^*$, where ω is a $2n$-th primitive root in $\mathbb{Z}_q$. For $n' \in \{1, 2, 4, 8, \ldots, n\}$, let $u = n/n'$. For $\alpha \in \mathbb{Z}_{2n'}^*$, consider $\mathbf{p}_u^\alpha(x)$ be the degree $u-1$ polynomial that is obtained by taking $\mathbf{p}(x)$ modulo $(x^u - (\omega^\alpha)^u)$. We may assume WLOG that $\alpha = 1$. We abbreviate notation and write $\mathbf{p}_u$, instead of $\mathbf{p}_u^1$.

We consider attacks in which the adversary learns all coordinates i of $\widehat{\mathbf{s}}$ such that $i \equiv 1 \bmod 2n'$ where $n' \in \{1, 2, 4, 8, \ldots, n\}$, and aims to recover the RLWE secret $\mathbf{s}$. First, we note that in NTT transform notation the equation $\widehat{\mathbf{a}} \cdot \widehat{\mathbf{s}} + \widehat{\mathbf{e}} = \widehat{\mathbf{u}}$ holds component-wise. Therefore, given leakage on certain coordinates of $\widehat{\mathbf{s}}$, we can solve for the corresponding coordinates of $\widehat{\mathbf{e}}$. We also get to see multiple RLWE samples (which we write in matrix notation–where the $\mathbf{A}^j$ matrices are the circulant matrices corresponding to the ring element $\mathbf{a}^j$'s) as $(\mathbf{A}^1, \mathbf{A}^1\mathbf{s} + \mathbf{e}^1 = \mathbf{u}^1), \ldots, (\mathbf{A}^\ell, \mathbf{A}^\ell\mathbf{s} + \mathbf{e}^\ell = \mathbf{u}^\ell)$. Thus, for the j-th RLWE sample we learn all the coordinates $\widehat{\mathbf{e}}_i^j$, for $i \equiv 1 \mod 2n'$. Note that the leaked coordinates are the evaluation of the polynomial $\mathbf{e}_u(x)$ at the ω^i for $i \equiv 1 \mod 2n'$. We can then reconstruct the polynomial $\mathbf{e}_u(x)$ using Lagrange Interpolation.

For $i \in \{0, \ldots, u-1\}$, the $(i+1)$-st coefficient of $\mathbf{e}_u(x)$, i.e. $e_{u,i}$ is equal to

$$e_i + \omega^u \cdot e_{i+u} + \omega^{2\cdot u} \cdot e_{i+2\cdot u} + \ldots + \omega^{(n'-1)\cdot u} \cdot e_{i+(n'-1)\cdot u}$$

The coefficients of $\mathbf{e}$ can be partitioned into u groups of size n', forming independent linear systems, each with n' variables and one equation. Given only the leakage, the set of feasible secret keys is a cartesian product $\mathcal{S}_1 \times \cdots \times \mathcal{S}_u$, where for $i \in [u]$, the set $\mathcal{S}_i$ is the set of vectors $\overline{\mathbf{e}}_i := \{e_i, e_{i+u}, e_{i+2u}, \ldots, e_{i+(n'-1)u}\}$ that satisfy the i-th linear system:

$$\begin{bmatrix} 1 & \omega^u & \omega^{2\cdot u} & \cdots & \omega^{(n'-1)\cdot u} \end{bmatrix} \cdot \begin{bmatrix} e_i & e_{i+u} & e_{i+2\cdot u} & \cdots & e_{i+(n'-1)\cdot u} \end{bmatrix}^T = \begin{bmatrix} e_{u,i} \end{bmatrix}$$

Since each coordinate of $\mathbf{e}$ is drawn independently from χ and since each linear system above has small dimension n', we can use a brute-force-search to find the most likely solution and calculate its probability.

Given this information, we will carefully choose the solutions $\overline{\mathbf{e}}_i^j$ (from all possible sets of solutions $[\overline{\mathbf{e}}_i^j]_{j\in[\ell],i\in[u]}$) that have a high chance of being the correct values of the RLWE error. To obtain a full key recovery attack, we require the following: (1) In total, we must guess at least u number of n'-dimensional solutions, $\overline{\mathbf{e}}_i^j$, from all the obtained solutions $[\overline{\mathbf{e}}_i^j]_{j\in[\ell],i\in[u]}$; (2) With high probability *all* our guesses are correct. Observe that if our guess of some $\overline{\mathbf{e}}_i^j$ is correct, we learn the following linear system of n' equations and n variables ($A^{j,i} \cdot \mathbf{s} = \mathbf{u}^{j,i} - \mathbf{e}^{j,i}$), where $A^{j,i}$ is the submatrix of A^j consisting of the n' rows $i, i+u, i+2\cdot u, \ldots, i+(n'-1)\cdot u$, and $\mathbf{u}^{j,i}, \mathbf{e}^{j,i}$ are vectors consisting of the $i, i+u, i+2\cdot u, \ldots, i+(n'-1)\cdot u$ coordinates of $\mathbf{u}^j$ and $\mathbf{e}^j$. So assuming (1) and (2) hold, we learn u *noiseless* systems of n' linear equations, each with $n = u \cdot n'$ number of variables. We then construct a linear system of n variables and n equations, which can be solved to obtain the candidate $\mathbf{s}$.

In order to ensure that (2) holds, we only keep the guess for $\overline{\mathbf{e}}_i^j$ when we have "high confidence" that it is the correct solution. The probability of a particular solution $\overline{\mathbf{e}}_i^j := (e_i^j, e_{i+u}^j, \ldots, e_{i+(n'-1)u}^j)$, is the ratio of the probability of $\overline{\mathbf{e}}_i^j$ being drawn from the error distribution (which is coordinate-wise independent) over the sum of the probabilities of all solutions. For small dimension n', this can be computed via a brute-force method. In our case, we keep the highest probability solution when it has probability at least, say 0.98. The probability that *all* guesses are correct is therefore $0.98^u = 0.98^{n/n'}$.

Since computing the exact probability as above is computationally intensive, we develop a heuristic that performs nearly as well and is much faster. Note that finding the "most likely" solution is equivalent to solving a CVP problem over an appropriate n'-dimensional lattice. We then calculate the probability of the solution under the discrete Gaussian and set some threshold . If the probability of the solution is above the threshold we keep it, if not we discard it. Experimentally, we show that by setting the threshold correctly, we can still achieve high confidence.

See Figure 1 for the exact settings of the threshold for each setting of parameters. Our experiments also show that (1) also holds given a reasonable number of RLWE samples. See Section 3.2 for a presentation of our experimental results. We describe our attack in cases where the leakage is on all coordinates i such that $i \equiv \alpha_1 \mod 2n'$ or $i \equiv \alpha_2 \mod 2n'$ in Appendix B.1.

Complexity of the attack. We provide the pseudocode for the attack in Appendix D, Figure 3. While our attack works well in practice, we do not provide a formal proof that our attack is polynomial time for a given setting of parameters. Within the loop beginning on line 5, all the steps (or subroutines) shown in Figure 3 can be computed in polynomial time. Note that even step 12 (`CVP.closest_vector`), which requires solving a CVP instance, can be computed in polynomial time because for the leakage patterns we consider, the dimension of the CVP instance will always be either 4 or 8–a constant, independent of n. However, our analysis does not bound the number of iterations of the loop beginning on line 5. Specifically, we do not analyze how large the variable `RLWESamples` must be set in order to guarantee that the attack is successful with high probability. Bounding this variable corresponds to bounding the number of RLWE samples needed in order to obtain a sufficient number of "high confidence" solutions. In practice, the number of RLWE samples was always fewer than 200 for all parameter settings. In future work, we plan to compute the expected number of RLWE samples needed to obtain a sufficient number of high confidence solutions for a given parameter setting. Assuming this expected number of samples is polynomial in n, we obtain an expected polynomial time attack.

3.2 Experimental Results

We first assess the performance of our attack on the RLWE challenges published by Crockett and Peikert [12], with various parameters, ranging from "toy" to "very hard" security levels. For each parameter setting, a cut-and-choose protocol was used by [12] to prove correctness of the challenges: They committed to some number (e.g. $N = 32$) of independent RLWE instances, a random index i was chosen, and the secret key for all except the i-th instance was revealed. For each of the 31 *opened* challenges, we simulate the Leaky RLWE experiment and attempt to recover the full secret $\mathbf{s}$ using our attack. We next measure the performance of our attack on RLWE instances generated using the dimension, modulus and noise distribution proposed in the original NewHope scheme [7]. These parameters are more conservative than the ones chosen for the later submission to the NIST competition [5]. When multiple RLWE samples are released, bounded error distributions are less secure [3]. We therefore tested our attack in the *more difficult* setting of Gaussian error, in addition to the original binomial error distribution of [7].

The experiments were run using server with AMD Opteron 6274 processor, with a python script using all the cores with Sage version 8.1. We used fplll [15] library for CVP solver and the source code of all the attacks are available online at [1]. The results of our attacks are summarized in Figure 1. We report the total number of instances we broke and the average number of RLWE samples needed for those instances. To decide whether a solution is kept or discarded, its probability mass under the error distribution χ is calculated and compared to the threshold. The threshold for each parameter setting is set heuristically so that minimal weight solutions passing the threshold are correct with high confidence (see Figure 1 for the exact threshold settings). We tested leakage patterns of $(n' = 4, \mathcal{S} = \{1\})$, $(n' = 8, \mathcal{S} = \{1, 7\})$ and $(n' = 8, \mathcal{S} = \{1, 15\})$–all corresponding to $1/4$-fraction leakage—for each parameter setting and were able to break multiple Leaky RLWE instances for every parameter setting/leakage pattern shown in Figure 1. We also report the maximum time it took to break a single instance for each parameter setting in Figure 1. Overall, the maximum amount of time to break a single instance was 6 hours for the hardest instance, i.e. Challenge ID 89. We attempted to launch our attack given only $1/8$-fraction of leakage (leakage pattern $(n' = 8, \alpha = 1)$), but were only successful for the easiest case, i.e. Challenge ID 1. For, e.g. Challenge ID 89, the attack failed since for 5000 number of linear systems, the maximum confidence of any solution was 0.28, meaning that we expect to recover the secret key with probability at most $0.28^{2048/8} \approx 2^{-470}$, which is well beyond feasible.

4 Search and Decisional Ring-LWE with Leakage

Definition 4.1 (**Search RLWE (R-SLWE) with Leakage**). The search version of the R-LWE problem with leakage, denoted Leaky R-$\mathsf{SLWE}_{q,\psi,n',\mathcal{S}}$, is parameterized by $(n' \in \{1, 2, 4, 8, \ldots n\}, \mathcal{S} \subseteq \mathbb{Z}^*_{2n'})$. The experiment chooses $\mathbf{s} \leftarrow R_q$ uniformly at random, where $\mathbf{s} = \mathsf{NTT}^{-1}(\widehat{\mathbf{s}})$. The goal of the adversary is to recover $\mathbf{s}$, given independent samples from the distribution $D^{\mathbf{s}}_{real,n',\mathcal{S}}$, which outputs $\left(\widehat{\mathbf{a}}, \widehat{\mathbf{a}} \cdot \widehat{\mathbf{s}} + \widehat{\mathbf{e}}, [\widehat{s_i}]_{i \equiv \alpha \bmod 2n' \,|\forall \alpha \in \mathcal{S}}\right)$ where $\mathbf{a}, \mathbf{e}$ are obtained from $A_{\mathbf{s},\psi}$ as in standard RLWE (see Definition A.2).

Definition 4.2 (**Decision RLWE (R-DLWE) with Leakage**). The decision version of the R-LWE problem with leakage, denoted Leaky R-$\mathsf{DLWE}_{q,\psi,n',\mathcal{S}}$, is parameterized by $(n' \in \{1, 2, 4, 8, \ldots n\}, \mathcal{S} \subseteq \mathbb{Z}^*_{2n'})$. The experiment chooses $\mathbf{s} \leftarrow R_q$ uniformly at random, where $\mathbf{s} = \mathsf{NTT}^{-1}(\widehat{\mathbf{s}})$. The goal of the adversary is to distinguish between independent samples from the distributions $D^{\mathbf{s}}_{real,n',\mathcal{S}}$ and $D^{\mathbf{s}}_{sim,n',\mathcal{S}}$, where $D^{\mathbf{s}}_{real,n',\mathcal{S}}$ is the same as above, and $D^{\mathbf{s}}_{sim,n',\mathcal{S}}$ outputs

Chall ID (hardness)	n	q	χ	n'	Pattern ($\mathcal{S}$)	min-max RLWE #	Avg. RLWE #	Broken Instances	Threshold	Maximum Time (s)
1 (toy)	128	769	$D_{0.40}$	4	$\{1\}$	2-2	2	31 of 31	7e-5	2.24
				8	$\{1, 15\}$	1-2	1.93	29 of 31	7e-6	2.18
				8	$\{1, 7\}$	1-2	1.93	29 of 31	7e-6	1.23
				8	$\{1\}$	1-1	1	4 of 31	1e-8	1.3
5 (toy)	128	3329	$D_{0.80}$	4	$\{1\}$	2-3	2.38	31 of 31	7e-5	2.53
				8	$\{1, 15\}$	2-3	2.09	31 of 31	7e-6	1.99
				8	$\{1, 7\}$	2-3	2.09	31 of 31	7e-6	1.88
45 (moderate)	256	7681	$D_{0.80}$	4	$\{1\}$	2-3	2.61	31 of 31	7e-5	8.83
				8	$\{1, 15\}$	2-2	2	31 of 31	7e-8	8.78
				8	$\{1, 7\}$	2-2	2	31 of 31	7e-8	6.97
85 (very hard)	1024	59393	$D_{3.59}$	4	$\{1\}$	6-7	6.05	17 of 31	7e-5	1914
				8	$\{1, 15\}$	39-60	51.88	26 of 31	7e-9	2000
				8	$\{1, 7\}$	39-59	50.76	17 of 31	7e-9	2682
89 (very hard)	2048	86017	$D_{3.59}$	4	$\{1\}$	6-7	6.16	30 of 31	7e-5	5523
				8	$\{1, 15\}$	44-58	52.29	31 of 31	7e-9	11766
				8	$\{1, 7\}$	44-58	52.29	31 of 31	7e-9	20837
NewHope	1024	12289	$D_{\sqrt{8}}$	4	$\{1\}$	35-37	36	3 of 200	3e-4	745
				8	$\{1, 15\}$	147-220	180.85	82 of 200	7e-8	2226
				8	$\{1, 7\}$	189-204	196.5	2 of 1000	7e-8	1238
			Ψ_{16}	4	$\{1\}$	34-36	34.16	6 of 200	3e-4	796
				8	$\{1, 15\}$	149-217	183.20	94 of 200	7e-8	2039
				8	$\{1, 7\}$	177-193	184.8	5 of 1000	7e-8	975

Figure 1. Performance of attack against RLWE Challenges [12] and NewHope [7] parameter settings. For each parameter setting, we report the following: min/max and average number of RLWE samples required for successful break, total number of broken instances, and max run-time (in seconds) for successful break. Threshold is set such that the minimal weight solutions to the linear systems given in Section 3 have high confidence with sufficiently high probability.

$\left(\widehat{\mathbf{a}}, \widehat{\mathbf{u}}, [\widehat{s_i}]_{i \equiv \alpha \bmod 2n' \,|\forall \alpha \in S}\right)$, where $\mathbf{a}, \mathbf{e}$ are obtained from $A_{\mathbf{s},\psi}$ as in standard RLWE (see Definition A.2) and

$$\widehat{u_i} = \widehat{a_i} \cdot \widehat{s_i} + \widehat{e_i} \quad | \, i \equiv \alpha \bmod 2n' \; \forall \alpha \in S \qquad \text{and} \qquad \widehat{u_i} \leftarrow \mathbb{Z}_q$$

chosen uniformly random, otherwise.

5 Search to Decision Reduction With Leakage

Let the RLWE secret be denoted by $\hat{\mathbf{s}}$ and assume WLOG that there exists an adversary that obtains leakage $[\hat{s}_i]_{i \equiv 1 \mod 2n'}$ and distinguishes $\hat{\mathbf{u}} = \hat{\mathbf{a}} \cdot \hat{\mathbf{s}} + \hat{\mathbf{e}}$ from

$\hat{\mathbf{u}}'$, where $\hat{u}_i = \hat{a}_i \cdot \hat{s}_i + \hat{e}_i$ for $i \equiv 1 \mod 2n'$ and otherwise is uniform random[6]. It is not hard to see, using techniques of [23–25], that this implies an attacker that learns a single index $j \in \mathbb{Z}^*_{2n}$, $j \equiv b \mod 2n'$ of the RLWE secret, where $b \not\equiv 1 \mod 2n'$. We call this the **Basic Attack**. Due to limited space, we refer readers to Appendix C for description of **Basic Attack**.

Theorem 5.1 (Existence of Basic Attack). *If, for any $(n', \mathcal{S} \subseteq \mathbb{Z}^*_{2n'})$ adversary $\mathcal{A}$ running in time $t := t(n)$ distinguishes $D^{\mathbf{s}}_{real,n',\mathcal{S}}$ from $D^{\mathbf{s}}_{sim,n',\mathcal{S}}$ with probability $p := p(n)$, then there is some index j such that $j \neq \alpha' \mod n$ for all $\alpha' \in \mathcal{S}$ and an attack **Basic Attack** with parameters $(n', \mathcal{S}, j, t, p)$, that learns NTT coordinate $\hat{s}_j$ with probability $1 - 1/\mathsf{poly}(n)$ and takes time $\mathsf{poly}(n) \cdot t \cdot 1/p$.*

Our attack **Attack 1** uses the **Basic Attack** to learn *all* the values $[\hat{s}_i]_{i \equiv b^r \mod 2n'}$ for $r \in [n'/2]$. Let $\hat{s}^1 := \hat{s}$. The main idea of **Attack 1** is to learn all $[\hat{s}^1_i]_{i \equiv b \mod 2n'}$ in the first round, then apply an automorphism to shift the positions $i \equiv b^2 \mod n'$ into the positions $i \equiv b \mod 2n'$, resulting in a permuted RLWE secret, denoted $\hat{\mathbf{s}}^2$. Note that applying the automorphism causes the positions $\hat{s}^1_i$ such that $i \equiv b \mod n'$ to shift into the positions $i \equiv 1 \mod 2n'$. This means that we are now back where we started, and the reduction is now able to provide the required leakage (on $[\hat{s}^2_i]_{i \equiv 1 \mod 2n'}$) to the adversary and thus can learn the values of $[\hat{s}^2_i]_{i \equiv b \mod 2n'} = [\hat{s}^1_i]_{i \equiv b^2 \mod n'}$ in the second iteration, $[\hat{s}^3_i]_{i \equiv b \mod 2n'} = [\hat{s}^1_i]_{i \equiv b^3 \mod n'}$ in the third iteration, etc. We next formalize the necessary properties of the automorphisms.

For $i, j \in \mathbb{Z}^*_{2n}$, let $\phi_{i \to j}$ be the automorphism that maps $\hat{\mathbf{v}}$ to $\hat{\mathbf{v}}'$ such that $\mathbf{v}(\omega^i) = \mathbf{v}'(\omega^j)$. $\phi_{i \to j}$ induces a permutation on the elements of $\hat{\mathbf{v}}$, denoted $\rho_{i \to j}$. Specifically, $\phi_{i \to j}(\hat{\mathbf{v}})$ maps $\hat{v}_\ell$ to $\hat{v}_{\rho_{i \to j}(\ell)}$ for $i, j, \ell \in \mathbb{Z}^*_{2n}$, where $\rho_{i \to j}(\ell) = i^{-1}\ell j$.

Definition 5.2. A probability distribution $\psi : \mathbb{Z}(\zeta_m) \to \mathbb{R}$ is automorphically closed in K if for all $i, j \in \mathbb{Z}^*_m, \phi_{i \to j}(\psi) = \psi$.

We remark that RLWE error distribution χ is automorphically closed [23].

We formally define **Attack 1** in Figure 3. We next sketch how **Attack 1** can be used to complete the proof. For dimenstion n and parameter $n' \in \{1, 2, 4, 8, \ldots n\}$, let $T_{n'} := T_{n'}(n)$ be the (non-uniform) time to solve Leaky R-SLWE for dimension n and parameters $(n', \mathcal{S} = \{\alpha\} = \{1\})$, i.e. given positions $[\hat{s}^1_i]_{i \equiv 1 \mod 2n'}$, with probability $1/2$.

Assume subexponential $2^{\Omega(n^\epsilon)}$-hardness of search RLWE *without leakage* for some constant $\epsilon \leq 1$ and polynomial modulus q. Then we also have that

[6] Note that the problem is identical when the adversary obtains leakage $[\hat{s}_i]_{i \equiv \alpha \mod 2n'}$, for $\alpha \in \mathbb{Z}^*_{2n'}$ since, as we shall see next, an automorphism can be applied to shift all indices i such that $i \equiv \alpha \mod 2n'$ to positions $i \equiv 1 \mod 2n'$.

$T_n(n) \in 2^{\Omega(n^\epsilon)}$, and as discussed in the intro, there must exist a constant c' such that for sufficiently large n, there exists $n^* = n^*(n) \in \{2, 4, 8, 16, \ldots, n\}$ such that $T_{n^*}(n) \geq 2^{c'(n^\epsilon)}$ and $\frac{T_{n^*}(n)}{T_{n^*/2}(n)} \geq n$. The above implies that $T_{(n^*/2)} \in o(T_{n^*})$.

Now, if given $[\hat{s}_i^1]_{i \equiv 1 \mod 2n^*}$ leakage, there exists a $(t(n), p(n))$-distinguishing adversary (where $t(n) = \sqrt{T_{n^*}}/\mathsf{poly}(n)$ and $p(n) = 1/\sqrt{T_{n^*}}$), then we will show that there is an adversary solving the R-SLWE w.h.p. given positions $[\hat{s}_i^1]_{i \equiv 1 \mod 2n^*}$ in time *less than* T_{n^*}, leading to contradiction. We begin by running **Attack 1**, which takes time at most $o(T_{n^*})$ for our settings of $t(n)$ and $p(n)$. If $b \in \mathbb{Z}_{2n^*}^*$ is such that for some $r \in [n^*/2]$, $b^r \equiv n^* + 1 \mod 2n^*$, then we can combine the reconstructed values of $\hat{s}_i^1$ from **Attack 1** with our knowledge of $[\hat{s}_i^1]_{i \equiv 1 \mod 2n^*}$ to obtain all values $[\hat{s}_i^1]_{i \equiv 1 \mod n^*}$. This means that we can then run the search attack for $2/n^*$-fraction of leakage to recover all of $\hat{\mathbf{s}}$ in time $T_{(n^*/2)} \in o(T_{n^*})$. But then the entire attack for $(1 \mod 2n^*)$-leakage can be run in time $o(T_{n^*})$, contradicting the definition of T_{n^*}.

For $n^* > 4$, the only cases in which **Attack 1** does not recover $[\hat{s}_i]_{i \equiv n^*+1 \mod 2n^*}$, is when $b \in \{n^*-1, 2n^*-1\}$. For such b, we do not know how to rule out the possibility that given $[\hat{s}_i]_{i \equiv 1 \mod 2n^*}$, the positions $i \equiv b \mod 2n^*$ of $\hat{u}$ do not look random. In this case, however, we argue that given leakage on *both* $[\hat{s}_i]_{i \equiv 1 \mod n^*}$, and $[\hat{s}_i]_{i \equiv b \mod n^*}$, all *other* positions are indistinguishable from random, since otherwise a modified version of **Attack 1** can be run. We next state the formal theorem of this section.

Theorem 5.3. *Assume $n^* := n^*(n) > 4$, $\mathbf{s} \leftarrow R_q$, then:*

- $D^{\mathbf{s}}_{real, n^*, \{\alpha\}} \approx_{t(n), p(n)} D^{\mathbf{s}}_{sim, n^*, \{\alpha\}}$ *OR*
- $D^{\mathbf{s}}_{real, n^*, \{\alpha, (n^*-1)\cdot\alpha\}} \approx_{t(n), p(n)} D^{\mathbf{s}}_{sim, n^*, \{\alpha, (n^*-1)\cdot\alpha\}}$ *OR*
- $D^{\mathbf{s}}_{real, n^*, \{\alpha, (2n^*-1)\alpha\}} \approx_{t(n), p(n)} D^{\mathbf{s}}_{sim, n^*, \{\alpha, (2n^*-1)\alpha\}}$.

where, $t(n) = \sqrt{T_{n^*}}/\mathsf{poly}(n)$, $p(n) = 1/\sqrt{T_{n^*}}$.

Proof. We assume WLOG that $\alpha = 1$. Assume $D^{\mathbf{s}}_{real, n^*, \{1\}} \not\approx_{\sqrt{T_{n^*}}/\mathsf{poly}(n), 1/\sqrt{T_{n^*}}} D^{\mathbf{s}}_{sim, n^*, \{1\}}$. Then this means there must be an adversary A running in time $\sqrt{T_{n^*}}/\mathsf{poly}(n)$, that distinguishes on index $j \in \mathbb{Z}_{2n}^*$, where $j \equiv b \mod 2n'$ with probability at least $1/\sqrt{T_{n^*}}$.

Case 1: b is such that $b^r \equiv n^* + 1 \mod 2n^*$ for some $r \in [n^*/2]$. In this case, with appropriate setting of $\mathsf{poly}(n)$, we can use Attack 1 to recover the positions i such that $i \equiv n^* + 1 \mod 2n^*$ (w.h.p.) in time $o(T_{n^*})$. Now we can run the attack that takes as input $[\hat{s}_i]_{i \equiv 1 \mod n^*}$ and recovers all of $\hat{\mathbf{s}}$. By assumption, this

Attack 1:

Given access to $D^{\mathbf{s}}_{real,n',\mathcal{S}=\{1\}}$ (i.e. RLWE samples with leakage $[\hat{s}_i]_{i\equiv 1 \mod 2n'}$) and the distinguishing index $j \in \mathbb{Z}^*_{2n}$, where $j \equiv b \mod 2n'$, for the **Basic Attack**:

1: **for all** Leaky-RLWE samples **do**
2: Set $\hat{\mathbf{a}}^1 := \hat{\mathbf{a}}, \hat{\mathbf{u}}^1 := \hat{\mathbf{u}}, [\hat{s}^1_i := \hat{s}_i]_{i\equiv 1 \mod 2n'}$.
3: **end for**
4: **for** $r \in [1, 2, \ldots, n'/2]$ **do** ▷ $[\hat{s}^r_i]_{i\equiv 1 \mod 2n'}$ are now known.
5: **for all** j' such that $j' \equiv j \mod 2n'$ **do**
6: Run the **Basic Attack** with parameters $(n', \{1\}, j, t, p)$ on RLWE samples of the form $(\hat{\mathbf{a}} := \phi_{j'\to j}(\hat{\mathbf{a}}^r), \hat{\mathbf{u}} := \phi_{j'\to j}(\hat{\mathbf{u}}^r))$, leakage set $[\hat{s}_i := \hat{s}^r_{\rho_{j'\to j}(i)}]_{i\equiv 1 \mod 2n'}$ to recover $\hat{s}^r_{j'}$.
▷ All these values of $\hat{s}^r_{\rho_{j'\to j}(i)}$ are now known: If $i \equiv 1 \mod 2n'$ then $\rho_{j'\to j}(i) \equiv 1 \mod 2n'$, since $j \equiv j' \mod 2n'$.
7: **end for** ▷ W.h.p. all $\hat{s}^r_{j'}$ s.t. $j' \equiv b \mod 2n'$ are now known.
8: Choose an $\ell \in \mathbb{Z}^*_{2n}$ such that $\ell \equiv b^2 \mod 2n'$.
9: **for all** Leaky RLWE samples **do**
10: Set $\hat{\mathbf{a}}^{r+1} := \phi_{\ell\to j}(\hat{\mathbf{a}}^r)$ and $\hat{\mathbf{u}}^{r+1} := \phi_{\ell\to j}(\hat{\mathbf{u}}^r)$.
11: **end for**
▷ $[\hat{s}^{r+1}_i]_{i\equiv 1 \mod 2n'}$, are now known since $\hat{s}^r_{i'}$ s.t. $i' \equiv b \mod 2n'$ are now in position $\hat{s}^{r+1}_i$ s.t. $i \equiv 1 \mod 2n'$.
12: **end for** ▷ All values s_i such that $i \equiv b^r \mod 2n'$ and $r \in [n'/2]$ are now known.

Figure 2. Description of **Attack 1**.

attack runs in time $T_{(n^*/2)} \in o(T_{n^*})$. Thus, we can to recover the whole $\hat{\mathbf{s}}$ (w.h.p. greater than $1/2$) in time $o(T_{n^*})$, which is a contradiction.

By properties of the group $\mathbb{Z}^*_{2n^*}$, where n^* is a power of two, for all $b \in \mathbb{Z}^*_{2n^*} \setminus \{1, n^*-1, 2n^*-1\}$, it is the case that $b^r \equiv n^*+1 \mod 2n^*$ for some $r \in [n^*/2]$. Thus, Case 1 holds for all $b \in \mathbb{Z}^*_{2n^*} \setminus \{n^*-1, 2n^*-1\}$.

Case 2: $b = n^* - 1$**.** In this case, with appropriate setting of $\mathsf{poly}(n)$, we can use Attack 1 to recover the positions i such that $i \equiv n^* - 1 \mod 2n^*$ (w.h.p.) in time $o(T_{n^*})$. Assume $D^{\mathbf{s}}_{real,n^*,\{1,(n^*-1)\}} \approx^{\sqrt{T_{n^*}}/\mathsf{poly}(n),\sqrt{T_{n^*}}/\mathsf{poly}(n)} D^{\mathbf{s}}_{sim,n^*,\{1,(n^*-1)\}}$, then there must be some adversary $\mathcal{A}'$ that distinguishes on index $j' \in \mathbb{Z}^*_{2n}$, where $j' \equiv b' \in \mathbb{Z}^*_{2n^*} \setminus \{1, n^*-1\}$. We can combine this with the previous attack as follows:

Case 2(a): $b' \in \mathbb{Z}^*_{2n^*} \setminus \{1, n^*-1, 2n^*-1\}$**.** Due to essentially the same argument as before, by appropriately setting $\mathsf{poly}(n)$, we can (w.h.p.) learn all $[\hat{s}_i]_{i\equiv(b')^r \mod 2n^*}$ for $r \in [n^*/2]$ in time $o(T_{n^*})$ and then apply the same argument as above.

Specifically, given the initial leakage $[\hat{s}^1_i]_{i\equiv 1 \mod 2n^*}$, the attack will first learn $[\hat{s}^1_i]_{i\equiv n^*-1 \mod 2n^*}$, then learn $[\hat{s}^1_i]_{i\equiv b' \mod 2n^*}$, then, for

some (j, j') such that $j \equiv b' \mod 2n^*$ and $j' \equiv 1 \mod 2n^*$, apply automorphism $\phi_{j \to j'}$ to get $\hat{\mathbf{s}}^2$, learn $[\hat{s}_i^2]_{i \equiv n^* - 1 \mod 2n^*}$, then learn $[\hat{s}_i^2]_{i \equiv b' \mod 2n^*}$, etc. thus ultimately learning $[\hat{s}_i]_{i \equiv (b')^r \mod 2n^*}$ for $r \in [n^*/2]$. At this point, we will have $[\hat{s}_i]_{i \equiv 1 \mod n^*}$ and thus can learn all of $\hat{\mathbf{s}}$ in additional time $T_{(n^*/2)} \in o(T_{n^*})$. Thus, in total the attack takes time $o(T_{n^*})$, leading to contradiction.

Case 2(b): $b' = 2n^* - 1$**.** Due to essentially the same argument as before, with appropriate setting of $\mathsf{poly}(n)$, we can (w.h.p.) recover the positions i such that $i \equiv 2n^* - 1 \mod 2n^*$ in time $o(T_{n^*})$. The adversary now knows $[\hat{s}_i]_{i \equiv n^* - 1 \mod n^*}$. We can thus learn all of $\hat{\mathbf{s}}$ in additional time $T_{(n^*/2)} \in o(T_{n^*})$. Thus, in total the attack takes time $o(T_{n^*})$, leading to contradiction.

Case 3: $b = 2n^* - 1$**.** This essentially follows identically to Case 2.

□

Bibliography

[1] *Source Code*, 2019, `https://github.com/mathcrypt/RLWE`.

[2] Adi Akavia, Shafi Goldwasser and Vinod Vaikuntanathan, Simultaneous Hardcore Bits and Cryptography against Memory Attacks, in: *TCC 2009* (Omer Reingold, ed.), LNCS 5444, pp. 474–495, Springer, Heidelberg, March 2009.

[3] Martin Albrecht, Carlos Cid, Jean-Charles Faugere, Robert Fitzpatrick and Ludovic Perret, Algebraic algorithms for LWE problems, (2014).

[4] Martin R. Albrecht, Amit Deo and Kenneth G. Paterson, Cold Boot Attacks on Ring and Module LWE Keys Under the NTT, *IACR TCHES* **2018** (2018), 173–213, `https://tches.iacr.org/index.php/TCHES/article/view/7273`.

[5] Erdem Alkim, Roberto Avanzi, Joppe Bos, Léo Ducas, Antonio de la Piedra, Thomas Pöppelmann, Peter Schwabe and Douglas Stebila, *Newhope: Algorithm specification and supporting documentation. Submission to the NIST Post-Quantum Cryptography Standardization Project, 2017*.

[6] Erdem Alkim, Léo Ducas, Thomas Pöppelmann and Peter Schwabe, *NewHope without reconciliation*, Cryptology ePrint Archive, Report 2016/1157, 2016, `http://eprint.iacr.org/2016/1157`.

[7] Erdem Alkim, Léo Ducas, Thomas Pöppelmann and Peter Schwabe, Post-quantum Key Exchange - A New Hope, in: *USENIX Security 2016* (Thorsten Holz and Stefan Savage, eds.), pp. 327–343, USENIX Association, August 2016.

[8] Jacob Alperin-Sheriff and Chris Peikert, Practical Bootstrapping in Quasilinear Time, in: *CRYPTO 2013, Part I* (Ran Canetti and Juan A. Garay, eds.), LNCS 8042, pp. 1–20, Springer, Heidelberg, August 2013.

[9] Madalina Bolboceanu, Zvika Brakerski, Renen Perlman and Devika Sharma, *Order-LWE and the Hardness of Ring-LWE with Entropic Secrets*, Cryptology ePrint Archive, Report 2018/494, 2018, https://eprint.iacr.org/2018/494.

[10] Elette Boyle, Gil Segev and Daniel Wichs, Fully Leakage-Resilient Signatures, *Journal of Cryptology* **26** (2013), 513–558.

[11] Zvika Brakerski, Yael Tauman Kalai, Jonathan Katz and Vinod Vaikuntanathan, Overcoming the Hole in the Bucket: Public-Key Cryptography Resilient to Continual Memory Leakage, in: *51st FOCS*, pp. 501–510, IEEE Computer Society Press, October 2010.

[12] Eric Crockett and Chris Peikert, Challenges for Ring-LWE., *IACR Cryptology ePrint Archive* **2016** (2016), 782.

[13] Dana Dachman-Soled, Huijing Gong, Mukul Kulkarni and Aria Shahverdi, *On the Leakage Resilience of Ideal-Lattice Based Public Key Encryption*, Cryptology ePrint Archive, Report 2017/1127, 2017, https://eprint.iacr.org/2017/1127.

[14] Dana Dachman-Soled, Huijing Gong, Mukul Kulkarni and Aria Shahverdi, *Partial Key Exposure in Ring-LWE-Based Cryptosystems: Attacks and Resilience*, Cryptology ePrint Archive, Report 2018/1068, 2018, https://eprint.iacr.org/2018/1068.

[15] The FPLLL development team, *fplll, a lattice reduction library*, Available at https://github.com/fplll/fplll, 2016.

[16] Yevgeniy Dodis, Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert and Vinod Vaikuntanathan, Public-Key Encryption Schemes with Auxiliary Inputs, in: *TCC 2010* (Daniele Micciancio, ed.), LNCS 5978, pp. 361–381, Springer, Heidelberg, February 2010.

[17] Yevgeniy Dodis, Kristiyan Haralambiev, Adriana López-Alt and Daniel Wichs, Cryptography against Continuous Memory Attacks, in: *51st FOCS*, pp. 511–520, IEEE Computer Society Press, October 2010.

[18] Yevgeniy Dodis, Yael Tauman Kalai and Shachar Lovett, On cryptography with auxiliary input, in: *41st ACM STOC* (Michael Mitzenmacher, ed.), pp. 621–630, ACM Press, May / June 2009.

[19] Stefan Dziembowski and Krzysztof Pietrzak, Leakage-Resilient Cryptography, in: *49th FOCS*, pp. 293–302, IEEE Computer Society Press, October 2008.

[20] Shafi Goldwasser, Yael Tauman Kalai, Chris Peikert and Vinod Vaikuntanathan, Robustness of the Learning with Errors Assumption, in: *ICS 2010* (Andrew Chi-Chih Yao, ed.), pp. 230–240, Tsinghua University Press, January 2010.

[21] Jonathan Katz and Vinod Vaikuntanathan, Signature Schemes with Bounded Leakage Resilience, in: *ASIACRYPT 2009* (Mitsuru Matsui, ed.), LNCS 5912, pp. 703–720, Springer, Heidelberg, December 2009.

[22] Allison B. Lewko, Mark Lewko and Brent Waters, How to leak on key updates, in: *43rd ACM STOC* (Lance Fortnow and Salil P. Vadhan, eds.), pp. 725–734, ACM Press, June 2011.

[23] Vadim Lyubashevsky, Search to decision reduction for the learning with errors over rings problem, in: *2011 IEEE Information Theory Workshop, ITW 2011, Paraty, Brazil, October 16-20, 2011*, pp. 410–414, 2011.

[24] Vadim Lyubashevsky, Chris Peikert and Oded Regev, On Ideal Lattices and Learning with Errors over Rings, in: *EUROCRYPT 2010* (Henri Gilbert, ed.), LNCS 6110, pp. 1–23, Springer, Heidelberg, May / June 2010.

[25] Vadim Lyubashevsky, Chris Peikert and Oded Regev, On Ideal Lattices and Learning with Errors over Rings, *J. ACM* **60** (2013), 43:1–43:35.

[26] Vadim Lyubashevsky, Chris Peikert and Oded Regev, *A Toolkit for Ring-LWE Cryptography*, Cryptology ePrint Archive, Report 2013/293, 2013, `http://eprint.iacr.org/2013/293`.

[27] Tal Malkin, Isamu Teranishi, Yevgeniy Vahlis and Moti Yung, Signatures Resilient to Continual Leakage on Memory and Computation, in: *TCC 2011* (Yuval Ishai, ed.), LNCS 6597, pp. 89–106, Springer, Heidelberg, March 2011.

[28] Chris Peikert, How (Not) to Instantiate Ring-LWE, in: *SCN 16* (Vassilis Zikas and Roberto De Prisco, eds.), LNCS 9841, pp. 411–430, Springer, Heidelberg, August / September 2016.

[29] Krzysztof Pietrzak, A Leakage-Resilient Mode of Operation, in: *EUROCRYPT 2009* (Antoine Joux, ed.), LNCS 5479, pp. 462–482, Springer, Heidelberg, April 2009.

[30] Katherine E. Stange, *Algebraic aspects of solving Ring-LWE, including ring-based improvements in the Blum-Kalai-Wasserman algorithm*, Cryptology ePrint Archive, Report 2019/183, 2019, `https://eprint.iacr.org/2019/183`.

A Additional Preliminaries

A.1 Algebraic Number Theory

For a positive integer m, the m^{th} *cyclotomic number field* is a field extension $K = \mathbb{Q}(\zeta_m)$ obtained by adjoining an element ζ_m of order m (i.e. a primitive m^{th} root of unity) to the rationals.

Ring of Integers R and Its Dual $R^\vee$

Let $R \subset K$ denote the set of all algebraic integers in number field K defined above. This set forms a ring (under the usual addition and multiplication operations in K), called the *ring of integers* of K.

An (*integral*) *ideal* $\mathcal{I} \subseteq R$ is a non-trivial (i.e. $\mathcal{I} \neq \varnothing$ and $\mathcal{I} \neq \{0\}$) additive subgroup that is closed under multiplication by R, i,e., $r \cdot a \in \mathcal{I}$ for any $r \in R$ and $a \in \mathcal{I}$.

Definition A.1. For $R = \mathbb{Z}[\zeta_m]$, define $g = \prod_p (1 - \zeta_p) \in R$, where p runs over all odd primes dividing m. Also, define $t = \frac{\hat{m}}{g} \in R$, where $\hat{m} = \frac{m}{2}$ if m is even, otherwise $\hat{m} = m$.

The dual ideal $R^\vee$ of R is defined as $R^\vee = \langle t^{-1} \rangle$, satisfying $R \subseteq R^\vee \subseteq \hat{m}^{-1} R$. The quotient $R_q^\vee$ is defined as $R_q^\vee = R^\vee / qR^\vee$.

A.2 Ring-LWE

We next present the formal definition of the RLWE problem as given in [26].

Definition A.2 (**RLWE Distribution**). For a "secret" $s \in R_q^\vee$ (or just $R^\vee$) and a distribution χ over $K_\mathbb{R}$, a sample from the RLWE distribution $A_{s,\chi}$ over $R_q \times (K_\mathbb{R}/qR^\vee)$ is generated by choosing $a \leftarrow R_q$ uniformly at random, choosing $e \leftarrow \chi$, and outputting $(a, b = a \cdot s + e \bmod qR^\vee)$.

Definition A.3 (**RLWE, Average-Case Decision**). The average-case decision version of the RLWE problem, denoted R-DLWE$_{q,\chi}$, is to distinguish with non-negligible advantage between independent samples from $A_{s,\chi}$, where $s \leftarrow R_q^\vee$ is sampled uniformly at random, and the same number of uniformly random and independent samples from $R_q \times (K_\mathbb{R}/qR^\vee)$.

Theorem A.4. [26, Theorem 2.22] *Let K be the m^{th} cyclotomic number field having dimension $n = \varphi(m)$ and $R = \mathcal{O}_K$ be its ring of integers. Let $\alpha = \alpha(n) > 0$, and $q = q(n) \geq 2$, $q = 1 \bmod m$ be a $\mathsf{poly}(n)$-bounded prime such that*

$\alpha q \geq \omega(\sqrt{\log n})$. *Then there is a polynomial-time quantum reduction from* $\tilde{O}(\sqrt{n}/\alpha)$*-approximate* SIVP *(or* SVP*) on ideal lattices in* K *to the problem of solving* R*-*$\mathsf{DLWE}_{q,\chi}$ *given only* l *samples, where* χ *is the Gaussian distribution* D_ξ *for* $\xi = \alpha \cdot q \cdot (nl/\log(nl))^{1/4}$.

A Note on the Tweak. In [8], Alperin-Sheriff and Peikert show that an equivalent "tweaked" form of the Ring-LWE problem can be used in cryptographic applications without loss in security or efficiency. This is convenient since the "tweaked" version does not involve $R^\vee$. The "tweaked" ring-LWE problem can be obtained by implicitly multiplying the noisy products b by the "tweak" factor t, and, as it is explained in [8], $t \cdot R^\vee = R$. This yields new values

$$b' = t \cdot b = (t \cdot s) \cdot a + (t \cdot e) = s' \cdot a + e' \bmod qR,$$

where $a, s' = t \cdot s \in R_q$, and the errors $e' = t \cdot e$ come from the "tweaked" error distribution $t \cdot \chi$.

A.3 Number Theoretic Transform (NTT)

Let $R_q := \mathbb{Z}_q[x]/x^n + 1$ be the ring of polynomials, with $n = 2^d$ for any positive integer d. Also, let $m = 2n$ and $q = 1 \bmod m$. For, ω a m^{th} root of unity in $\mathbb{Z}_q$ the NTT of polynomial $\mathbf{p} = \sum_{i=0}^{n-1} p_i x^i \in R_q$ is defined as,

$$\widehat{\mathbf{p}} = \mathsf{NTT}(\mathbf{p}) := \sum_{i=0}^{n-1} \widehat{p}_i x^i$$

where the NTT coefficients $\widehat{p}_i$ are defined as: $\widehat{p}_i = \sum_{j=0}^{n-1} p_j \omega^{j(2i+1)}$.

The function NTT^{-1} is the inverse of function NTT, defined as

$$\mathbf{p} = \mathsf{NTT}^{-1}(\widehat{\mathbf{p}}) := \sum_{i=0}^{n-1} p_i x^i$$

where the NTT inverse coefficients p_i are defined as: $p_i = n^{-1} \sum_{j=0}^{n-1} \widehat{p}_j \omega^{i(2j+1)}$.

B Attack Algorithm for Other Leakage Patterns

B.1 Reconstructing the secret given $(\alpha_1, \alpha_2 \mod n')$ leakage

Let $\mathbf{e}_u^\alpha(x)$ be the degree $u = n/n'$ polynomial that is obtained by taking $\mathbf{e}(x)$ modulo $x^u - (\omega^\alpha)^u$. We consider two polynomials $\mathbf{e}_u^{\alpha_1}(x)$ and $\mathbf{e}_u^{\alpha_2}(x)$. We may

assume WLOG, $\alpha_1 = 1$. We therefore set $\alpha := \alpha_2$. For $i \in \{0, \ldots, u-1\}$, the $(i+1)$-st coefficient of $\mathbf{e}_u(x)$ and $\mathbf{e}_u^{\alpha}(x)$ are as follows, respectively

$$e_i + \omega^u \cdot e_{i+u} + \omega^{2\cdot u} \cdot e_{i+2\cdot u} + \ldots + \omega^{(n'-1)\cdot u} \cdot e_{i+(n'-1)\cdot u}$$

$$e_i + \omega^{\alpha\cdot u} \cdot e_{i+u} + \omega^{\alpha\cdot 2\cdot u} \cdot e_{i+2\cdot u} + \ldots + \omega^{\alpha\cdot(n'-1)\cdot u} \cdot e_{i+(n'-1)\cdot u}$$

Similar to the previous attack, we obtain the following constraints on the error, given leakage on the secret key and an RLWE sample,

$$\begin{bmatrix} 1 & \omega^u & \omega^{2\cdot u} & \cdots & \omega^{(n'-1)\cdot u} \\ 1 & \omega^{\alpha\cdot u} & \omega^{\alpha\cdot 2\cdot u} & \cdots & \omega^{\alpha\cdot(n'-1)\cdot u} \end{bmatrix} \cdot \begin{bmatrix} e_i \\ e_{i+u} \\ e_{i+2\cdot u} \\ \vdots \\ e_{i+(n'-1)\cdot u} \end{bmatrix} = \begin{bmatrix} e_{u,i} \\ e^{\alpha}_{u,i} \end{bmatrix}$$

We solve a corresponding CVP instance to find the "most likely" solution, $\overline{\mathbf{e}}_i$ for $(e_i, e_{i+u}, e_{i+2\cdot u}, \ldots, e_{i+(n'-1)\cdot u})$, since the "most likely" solution is the one with smallest norm.

Similar to our previous attack, our goal is to carefully choose the answers with "high confidence" such that (1) In total, we must guess at least u number of n'-dimensional solutions, $\overline{\mathbf{e}}_i^j$, from all the obtained solutions $[\overline{\mathbf{e}}_i^j]_{j\in[\ell],i\in[u]}$; (2) With high probability *all* our guesses are correct. We choose the candidate which has probability of at least, say, 0.95 of being correct solution. The total probability of success for this case is $0.95^u = 0.95^{n/n'}$.

Our experiments in section 3.2 again show that we can obtain enough "high" confidence solutions, without requiring too large a number of RLWE instances.

C Description of Basic Attack

In this section, we present the **Basic Attack**, following the description from [23–25] and using the fact that NTT coefficients form a CRT representation. We first recall definition of CRT representation in our setting of parameters.

Definition C.1 (CRT Representation). For $\mathbf{p} \in R_q$, and ω a m^{th} primitive root of unity in $\mathbb{Z}_q^*$, CRT representation for $\mathbf{p}$ is defined as

$$\text{CRT}(\mathbf{p}) = (\mathbf{p}(\omega^{j_1}), \ldots, \mathbf{p}(\omega^{j_n})),$$

for $j_i \in \mathbb{Z}_m^*$.

It is easy to see that $\mathrm{CRT}(\mathbf{p}) = (\widehat{p}_0, \ldots, \widehat{p}_{n-1})$.

We first introduce the following definition:

Definition C.2 (Hybrid Leaky RLWE Distribution). For $j \in \mathbb{Z}_{2n}^* = \{1, 3, \ldots, 2n-1\}$, a "secret" $s \in R_q$, and a distribution χ over R_q, a sample from the distribution $D_{real,n',\mathcal{S}}^{\mathbf{s},j}$ is generated by choosing $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}) \leftarrow D_{real,n',\mathcal{S}}^{\mathbf{s}}$ and outputting $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}} + \mathbf{u})$, where $\mathbf{u} = (u_1, u_3, \ldots, u_{2n-1}) \in \mathbb{Z}_q^n$ with u_i, $i \in \mathbb{Z}_{2n}^*$ defined as follows: u_i is chosen uniformly at random from $\mathbb{Z}_q$ if $i \neq \alpha' \mod 2n'$ for all $\alpha' \in \mathcal{S}$ and $i \leq j$, $u_i = 0$ otherwise.

Define $D_{real,n',\mathcal{S}}^{\mathbf{s},-1} := D_{real,n',\mathcal{S}}^{\mathbf{s}}$. Additionally, notice that $D_{real,n',\mathcal{S}}^{\mathbf{s},2n-1} = D_{sim,n',\mathcal{S}}^{\mathbf{s}}$. Thus if, for any $(n', \mathcal{S} \subseteq \mathbb{Z}_{2n'}^*)$ adversary $\mathcal{A}$ running in time $t := t(n)$ distinguishes $D_{real,n',\mathcal{S}}^{\mathbf{s}}$ from $D_{sim,n',\mathcal{S}}^{\mathbf{s}}$ with probability $p := p(n)$, then there is some index $j \in \mathbb{Z}_{2n}^*$ such that $j \neq \alpha' \mod n$ for all $\alpha' \in \mathcal{S}$ and a distinguisher $\mathcal{D}_j$ that is able to distinguish between the distribution $D_{real,n',\mathcal{S}}^{\mathbf{s},j-2}$ and $D_{real,n',\mathcal{S}}^{\mathbf{s},j}$ with probability at least p/n.

We now show the distinguisher $\mathcal{D}_j$ can be used to construct an algorithm that finds the value of $\hat{s}_j$. The idea of this algorithm is to try each of the possible values $\hat{s}_j$, constructing the samples on inputs from $D_{real,n',\mathcal{S}}^{\mathbf{s}}$, so that the samples are distributed according to $D_{real,n',\mathcal{S}}^{\mathbf{s},j-2}$ if $\hat{s}_j$ is guessed correctly, and the samples are distributed according to $D_{real,n',\mathcal{S}}^{\mathbf{s},j}$ otherwise. Then using the distinguisher $\mathcal{D}_j$ $\mathsf{poly}(n/p)$ times for each of the $q(= \mathsf{poly}(n))$ guesses for $\hat{s}_i$, we are able to find the correct value of $\hat{s}_j$ with probability $1 - 1/\mathsf{poly}(n)$ in time $t \cdot \mathsf{poly}(n) \cdot 1/p$.

Next we present the samples construction algorithm that takes a guess $g \in \mathbb{Z}_q$ and transform $D_{real,n',\mathcal{S}}^{\mathbf{s}}$ to either $D_{real,n',\mathcal{S}}^{\mathbf{s},j-2}$ or $D_{real,n',\mathcal{S}}^{\mathbf{s},j}$. On each sample $(\widehat{\mathbf{a}}, \widehat{\mathbf{b}}) \leftarrow D_{real,n',\mathcal{S}}^{\mathbf{s}}$, it outputs a sample

$$(\mathbf{a}', \mathbf{b}') = (\widehat{\mathbf{a}} + \mathbf{v}, \widehat{\mathbf{b}} + \mathbf{u} + g\mathbf{v}),$$

where $\mathbf{u} = (u_1, u_3, \ldots, u_{m-1}), \mathbf{v} = (v_1, v_3, \ldots, v_{m-1}) \in \mathbb{Z}_q^n$ are chosen as follows: u_k is uniform in $\mathbb{Z}_q$ if $k < j$, $k \neq \alpha' \mod 2n'$ for all $\alpha' \in \mathcal{S}$, and the rest are 0; v_k is uniform in $\mathbb{Z}_q$ if $k = j$, and the rest are 0. Note that b'_j can be written as

$$b'_j = \hat{a}_j \hat{s}_j + \hat{e}_j + u_j + g v_j = a'_j \hat{s}_j + \hat{e}_j + u_j + (g - \hat{s}_j) v_j.$$

Observe that if g is the correct guess, then $(g - \hat{s}_j)v_j = 0$. The distribution of $(\mathbf{a}', \mathbf{b}')$ is identical to $D_{real,n',\mathcal{S}}^{\mathbf{s},j-2}$. If g is a wrong guess, $(g - s_j)$ is non-zero. Since q is prime, $(g - \hat{s}_j)v_j$ is uniform in $\mathbb{Z}_q$. Thus the distribution of $(\mathbf{a}', \mathbf{b}')$ is identical to $D_{real,n',\mathcal{S}}^{\mathbf{s},j}$.

D Pseudocode of Attack from Section 3

Partial Key Exposure Attack

Given leaked coordinates on NTT version of secret key $\widehat{\mathbf{s}}$, public key a and a public value b, recover all coordinates of $\mathbf{s}$

[$\bar{\omega}$, B] = create_basis()
▷ create basis used in CVP solver and $\bar{\omega}$ being $\left[1, \omega^{u}, \omega^{2\cdot u}, \cdots, \omega^{(n'-1)\cdot u}\right]$
bTotal = [], aTotal = []
count = 0
$u = n/n'$
for $j \in [1, 2, \ldots,$ RLWESamples$]$ **do**
  $\mathbf{A}^j =$create_a$(\widehat{\mathbf{a}}^j)$ ▷ Create circulant matrix corresponding to $\widehat{\mathbf{a}}^j$
  $\widehat{\mathbf{e}}^j = \widehat{\mathbf{b}}^j - \widehat{\mathbf{a}}^j \cdot \widehat{\mathbf{s}}$
  ▷ For all leaked coordinate of $\widehat{\mathbf{s}}$ we compute the corresponding coordinates of error $\widehat{\mathbf{e}}^j$
  $\mathbf{e} =$ Lagrange polynomials$(\widehat{\mathbf{e}}^j)$
  ▷ Recover the coeffiecient of polynomial obtained by taking $\mathbf{e}(x)$ modulo $(x^u - (\omega^{\alpha})^u)$
  aMat = [], bTemp = []
  for $i \in [0, 1, 2, \ldots, u-1]$ **do**
    $X = \bar{\omega}$.solve_right$(\mathbf{e}_i)$ ▷ Solving the system of equation explained in Section 3
    $Y =$CVP.closest_vector(B, X)
    $\bar{\mathbf{e}}_i = X - Y$
    if Prob$(\bar{\mathbf{e}}_i)$ > Threshold **then**
      aMat.append$\left(\widehat{\mathbf{A}}^j[i, i+u, i+2\cdot u, \ldots, i+(n'-1)\cdot u][:]\right)$
      ▷ Select the corresponding rows from $\widehat{\mathbf{a}}^j$ and save them
      bTemp.append$\left(\widehat{\mathbf{b}}^j[i, i+u, i+2\cdot u, \ldots, i+(n'-1)\cdot u] - \bar{\mathbf{e}}_i\right)$
      ▷ Select the corresponding rows from $\widehat{\mathbf{b}}^j$, subtract $\bar{\mathbf{e}}_i$ from it to get noiseless system
      count += n'
    end if
  end for
  aTotal.append$\left($aMat$\right)$
  bTotal.append$\left($bTemp$\right)$
  if count == n **then**
    break
  end if
end for
try:
  sk = aTotal.solve_right(bTotal) ▷ solve the noiseless system to recover key
except:
  return error ▷ couldn't solve the system
return sk

Figure 3. Description of Partial Key Exposure Attack from Section 3

Received **???**.

Author information

Dana Dachman-Soled, Department of Electrical and Computer Engineering and UMIACS, University of Maryland, 8125 Paint Branch Dr., College Park, MD 20742, USA.
E-mail: danadach@ece.umd.edu

Huijing Gong, Department of Computer Science and UMIACS, University of Maryland, 8125 Paint Branch Dr., College Park, MD 20742, USA.
E-mail: gong@cs.umd.edu

Mukul Kulkarni, Department of Electrical and Computer Engineering and UMIACS, University of Maryland, 8125 Paint Branch Dr., College Park, MD 20742, USA.
E-mail: mukul@terpmail.umd.edu

Aria Shahverdi, Department of Electrical and Computer Engineering and UMIACS, University of Maryland, 8125 Paint Branch Dr., College Park, MD 20742, USA.
E-mail: ariash@umd.edu