TOWARDS A DEEPER NTRU ANALYSIS: A MULTI MODAL ANALYSIS

Chuck Easttom[1], Anas Ibrahim[2], Alexander Chefranov[3], Izzat Alsmadi[4] and Richard Hansen[5]

[1] Adjunct Georgetown University and University of Dallas
[2&3] Eastern Mediterranean University
[4]Texas A&M University
[5]Capitol Technology University

ABSTRACT

NTRU is being considered as part of the NIST quantum resistant cryptography standard. While NIST has received substantial attention in the literature, more analysis is needed. This current study uses a unique approach. The team of researchers divided into -sub-teams. Each is using a separate analysis technique on NTRU. Then the diverse -sub-teams' work was brought together into a single cohesive statistical analysis to provide well-founded conclusions regarding NTRU.

KEYWORDS

NTRU, NTRU Encrypt, Post Quantum Cryptography, Quantum Resistant Cryptography

1. INTRODUCTION

NTRU was first publicly described in 1996 by Jeffery Hoffstien, Jill Pipher, and Joseph Silverman [1]. There have been additional variations of this algorithm developed since its initial creation [2]. This algorithm is one of the most familiar and thoroughly studied cryptographic systems based on lattices [3]. NTRU is a cryptosystem which has been utilized both for encryption and digital signatures. Researchers have demonstrated NTRU to be resistant to attacks from Shor's algorithm [4], [5], unlike traditional asymmetric algorithms such as RSA.

The NIST project to identify a standard for post-quantum cryptography has been ongoing since 2017 [6]. Two versions of NTRU have made it past the second round of NIST testing and analysis [7]. Thus, NTRU is a leading candidate to become the NIST standard for quantum resistant cryptography. This indicates the importance of thorough testing of NTRU. While there are certainly existing studies of NTRU, the current study will fill specific gaps in the existing literature. One such gap is that there has been limited analysis of the randomness of the ciphertext produced by NTRU. Another such gap is that some existing published studies have not been confirmed. Repeating the experiment is a fundamental tenant of science. Furthermore, the current literature lacks a multi-modal analysis of NTRU, as is presented in this current study.

The focus of this current study will be on a comprehensive testing of NTRU. This means bringing multiple, separate testing modalities into one cohesive study. One primary focus of this study will be on lattice-basis reduction as one of the most prominent current known attacks. However, other testing modalities, such as the randomness of the output of NTRU will also be explored. A third portion of this current study will be an attempt to test the results of at least one previous study. This multi-modal approach will facilitate a broad-based understanding of the security of NTRU. The three sub-teams worked independently and did not see the other sub-

teams results until the tests were complete. This was done to mitigate issues of bias. This multi-modal testing approach provides a robust testing mechanism and demonstrates internal and external validity for the results of the overall study.

2. REVIEW OF LITERATURE

Bernstien, Chitchanok, Tanja, and Van Vredeendaal provide a solid overview of the NTRU Prime algorithm as well as the original published NTRU algorithm [8]. Their work is a good starting point for understanding NTRU. Generalized studies of cryptanalysis of various NTRU implementations have been ongoing for several years. However, the bulk of published research on cryptanalysis of NTRU focuses on specific implementations of NTRU and the weaknesses of those implementations [9], [10], [11]. These studies are useful; however, they are more relevant to those specific implementations of NTRU rather than to a generalized understanding of NTRU.

Albrecht, Bai, and Ducas explored a specific attack on NTRU [12]. Their study focused on scenarios wherein the public key was normed to a subfield to lead to an easier lattice problem. They demonstrated success with their attack modality. This indicates that specific variations in the NTRU keys can lead to weaknesses in the NTRU implementation. Kichner and Fouque [13] had similar results with a similar attack, as did Duong, Yasuda, and Takagi [14]. All of these attacks are predicated on an overstretched implementation of NTRU. While they are useful in demonstrating that certain implementations of NTRU can be compromised, these prior studies do not address the essential security of NTRU.

Bernstein, Chuengsatiansup, Lang, and van Vredendall studied modifying NTRU to what they call NTRU prime [15]. This variation of NTRU does not use the ring structures that are commonly attacked in other NTRU implementations. This study is important but does not address the fundamental security of NTRU as submitted to the NIST competition. Albrecht et al. published a general study of all LWE and NTRU based algorithms presented to the NIST Post Quantum Cryptography process [16]. Their research focused on efficiency. While important, that study also did not address the issues being discussed in this current study. Particularly the Albrecht study focused on efficiency rather than security.

Valluri specifically examined the security of the NTRU key exchange protocol proposed by Xinu et al [17]. Valluri's approach was based on a man in the middle attack specifically looking at the publicly exchanged values. Key exchange algorithms are often susceptible to man in the middle attacks, which is why many implementations also include authentication in order to mitigate such attacks.

Huerta utilized the LLL (Lenstra, Lenstra, Lovasz) algorithm for finding short vectors in a lattice [18]. Lattice based algorithms are often based on either the Closest Vector Problem (CVP) or Shortest Vector Problem (SVP). His study explored techniques that solves these problems within a factor of Cn where C is a small constant and n is the dimension of the lattice.

A review of the current literature makes it clear that while there is a substantial body of research on NTRU, there are gaps in the literature. These gaps are more significant due to the prominence of NTRU in the NIST Post Quantum Cryptography process. This current study is designed to address some of these gaps. Gaps to be addressed will include examining the randomness of the ciphertext produced, exploring lower dimension lattice attacks, examining parameter analysis, and synthesizing these different testing modalities into a single coherent conclusion.

3. EXPERIMENT ON LOWER DIMENSION LATTICE ATTACK

This experiment aims to re-examine Yang's results [19], and to find Lower Dimension Lattice (LDL) attack applicability against high dimension NTRU lattices. LDL attack shown in algorithm 1 [20] has been implemented with Block-Korkine-Zolotarev(BKZ) reduction from NTL package [21] on Intel® CoreTM 2 Duo Processor T5870 (2.00 GHz, 4Gb RAM).

Algorithm 1 lower dimension lattice attack on NTRU

Required: Fixed N, q, d_g, h and the probability $\text{prob}(f^{ls(k)} \in \mathcal{L}_I)$;
Ensure: A valid private key f';
1: $t \leftarrow 2$;
2: **while** $t < N$ **do**
3: **count** $\leftarrow$1;
4: **while** count $\Leftarrow \lceil 1/\text{prob}(f^{ls(k)} \in \mathcal{L}_I) \rceil$ **do**
5: Randomly choose a subset I of $[N]$ such that $\#I = t$;
6: Construct an $IN - Lattice\ \mathcal{L}_I$ with size t;
7: Reduce $\mathcal{L}_I$;
8: **if** the reduced basis contains a vector $\boldsymbol{v}$ which can be used to decrypt **then**
9: $f' = v$;
10: Output f', t and break;
11: **end if**
12: count = count + 1;
13: **end while**
14: $t \leftarrow t + 1$;
15: **end while**

In this studies experiments, the parameters listed in Table I are used. The value of time,t, was recorded when a valid private key,$f'$was found.The probability $\text{prob}(f^{ls(k)} \in \mathcal{L}_I)$is calculated using equation 1 as follows:

$$prob\left(f^{ls(k)} \in \mathcal{L}_I\right) = 1 - \left(1 - \prod_{i=0}^{2d_g-1}\left(1 - \frac{t}{N-i}\right)\right)^N, \quad (1)$$

Table I. The parameters used in current experiments.

N	df	dg	dr	q
19	2	2	2	41
37	4	4	2	79
57	6	6	2	113
73	6	6	2	113
83	8	8	2	151
97	9	9	2	167
107	14	14	2	257

Those results are listed in Table II.

Table II. The Results Of LDL Attack In Different NTRU Security Levels.

N	**19**	**37**	**57**	**73**	**83**	**97**	**107**
t	3	5	10	13	18	27	Not found
prob	0.999	0.999	0.987	0.997	0.649	0.125	Not found

Contrary to Yang's et. al. results, Table II shows exponential growth of parameter t when $IN-Lattice$ attack succeeds, it means that a target vector $f^{ls(k)}$ will belong to $\mathcal{L}_I$ with low probability. Thus,$IN-Lattice$ attack is infeasible for sufficiently large N (see Table 3). In this current experiment for $N=107$,no result was found after 6 hours of running the code, and no valid private key $f'$was disclosed. Figure 1 shows an exponential growth of $t$as N increases.

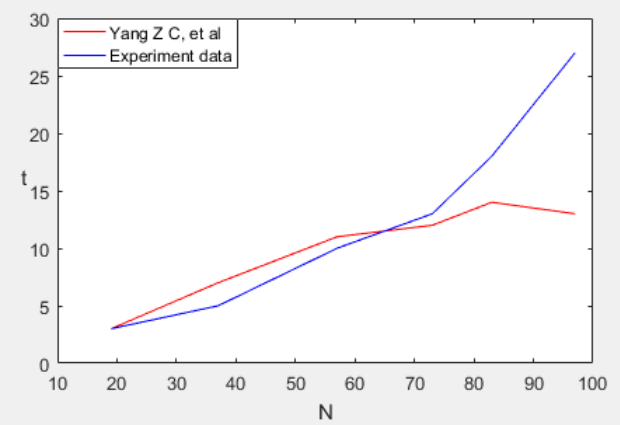

Fig. 1. Figure 1 Exponential growth of t as N increases

To determine the practicality of $IN-Lattice$ attack, this current experiment used the BKZ-NTL algorithm of NTL package inside Yang's algorithm to reduce those lattices and recorded the runtime only when wesuccessfully found a target vector $f^{ls(k)}$. Figure 2 gives the results of the experiments.

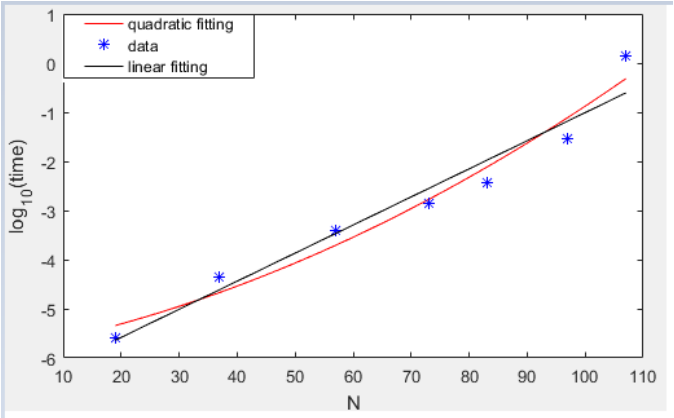

Fig. 2. Decimal logarithm of runtime in seconds of IN-Lattice Attack (blue asterisks), approximation fitting line (black), and quadratic fitting (red).

Time in this figure is given in seconds. Since those experiments were run on 2.0 GHz Core machine, the time in seconds is converted to the time in MIPS-years by first multiplying by $2.0 \cdot 1024$(to account for the 2.0 GHz machine) and then dividing by 31557600 which is the number of seconds in a year; In this case, the experimental data were approximated by linear and quadratic fitting functions respectively as follows in equation 2.

$$\log_{10}(T) \approx 0.05717 \cdot N - 6.725$$
$$\log_{10}(T) \approx 0.0002817 \cdot N^2 + 0.02158 \cdot N - 5.852 \quad (2)$$

Fitting curves and data are shown in Figure 2. The mean squared error for linear approximation is 1.063, and for quadratic approximation is 0.1077. Hence, a quadratic approximation is used for extrapolation of time for higher N values shown in Table III, which shows less time than Yang Z C, et al. extrapolation line as shown in equation 3.

$$\log_{10}(T) \approx 0.065N - 7.3. \quad (3)$$

In Table 3, the expected time (MIPS-years) to break the NTRU cryptosystem in comparison to Yang Z. C. et al. [3] is given.

Table III. Expected time (MIPS-years) to break the NTRU cryptosystem in comparison to Yang Z. C. et al. [1]

NTRU	**Our results**	**Yang, Fu, and Qu's results**
NTRU-167	$10^{5.61}$	$10^{3.55}$
NTRU-263	$10^{19.31}$	$10^{9.80}$
NTRU-503	$10^{76.28}$	$10^{25.4}$

The experimental data in Table III shows that parameter t has exponential growth as Nincreases. This means that a target vector $f^{ls(k)}$ will belong to $\mathcal{L}_I$ with low probability. In this experiment for $N = 107$, no valid private key $f'$was found in six hours for machine running. This contradicts to Yang Z. C., et al. results in the supplementary material where t shows less growth as N increases and private key f' found for $N = 107$ within 4 hours as shown in Figure B2 in the supplementary material of Yang Z C, et al. also report that private key f' was found for $N = 107$ within two hours [21].

Comparing this current study to Yang's, demonstrated statistically significant differences: the t-value is -2.09132. The p-value is .020908. The result is significant at a confidence level $p < .05$. A two-tailed Mann-Whitney U test also showed statistically significant results. The z-score is 2.01117. The p-value is .04444. The result is significant at a confidence level of $p < .05$. The current study shows that larger N values clearly do have a positive correlation with NTRU security.

This experiment supports the conclusion that larger N results in substantially increased security. using even NTRU-167 leads to a cryptographic solution that is unlikely to be broken in a practical time-frame by current systems. Using NTRU-503 it is unlikely that even a large computer cluster or super-computer would be able to break NTRU within a practical time frame. This supports the use of NTRU as a cryptographic solution both with current computing systems, as well as use against quantum computing.

4. NTRU PARAMETERS ANALYSIS

The next phase of this current study involves examining the NTRU parameters. Selection of parameters can impact the security of the cryptosystem. In comparison with classical encryption algorithms, NTRU is approaching the usage of large integers differently. In principle, NTRU is immune to the Shor algorithm because P and are Q are public information; conditions of selecting P and Q are different in NTRU from classical algorithms, RSA, etc.

Table IVbelow shows NTRU parameters. It also indicates which ones of those parameters can be considered as public information or NTRU inputs and which ones will be calculated within the process and hence are classified or private.

Table IV: A summary of NTRU parameters

Parameter	**Known**	**Input**	**Parameter**	**Known**	**Input**
N	Yes	Yes	A	No	No
P	Yes	Yes	Df	No	No
Q	Yes	Yes	Dq	No	No
D	Yes	Yes	Dr	No	No
f or f(x)	No	No	H	Possible	No
g or g(x)	No	No	R	No	No
Fp	No	No	M	Yes	Yes
Fq	No	No	E	No	No

The goal in this section is to use this underlying distribution between public and private parameters in evaluating possible attacks on NTRU encryption

We will focus experiments in this section based on the partial/possible public or known parameter, h, or the public key. H-parameter is not an input parameter, but in those experiments, we assume friendly attacks based on the knowledge of h or the public key

The public key is calculated based on the following formulas:

$$h=g * \text{inverse}(f) \quad \ldots\ldots\ldots\ldots\ldots\ldots \quad 1$$
$$h = p*f\text{-}1*g \bmod q \quad \ldots\ldots\ldots\ldots \quad 2$$

It can be generated using the public input parameters and some random intermediate parameters.For NTRU, the matrix A (that represents which private key components are aggregated to derive the signature) is a circulant matrixand decryption that depends on the decomposition of A into a product of two matrices having a special form. Together with lifting from mod q to mod, the random number r is that is inserted as the seed of the hash function is another parameter for Streamlined NTRU prime values.

In the lattice reduction step, it is observed that applying lattice reduction techniques will mostly reduce the middle vectors of the basis. We utilized in this experiment BKz 2.0 algorithm [22] to apply lattice-basis reduction. If the lattice basis was reduced sufficiently in the first phase, a collision resulting in the private key would be found by applying a rounding algorithm to the half-key guesses.

All lattice reduction algorithms known are based on Gram-Schmidt orthogonalization and decomposition (GSA). GSA assumption states that the Gram-Schmidt norms output by a lattice reduction follows a geometric sequence.

The function takes the public input parameters N, Q, and the partially public h, and returns the basis matrix A of the NTRU. Table IV below shows, for several examples of NTRU input parameters, the following two times:

- T(d): Time to decompose A into a product of two matrices having a special form (using BKZ)
- T(G): Time to Gram Schmidt reduced A (Standard GS)

All experiments are on Pentium core seven processor

Table V: T(d) and T(G) for different settings of NTRU parameters

N	P	Q	d	R	T(d)	T(G)
6	3	2048	2	23	~ 0	~ 0
7	3	41	2	23	~ 0	~ 0
7	3	401	24	23	2.33	26.31
7	29	491531	2	23	~ 0	~ 0
11	3	1024	2	23	~ 0	~ 0
16	2	379	8	23	~ 0	~ 0
16	2	379	4	23	~ 0	0.008
100	3	239	2	23	1.16	15.9
100	3	239	67	23	1.16	15.81
107	3	64	2	23	1.55	19.8
107	3	64	3	23	1.712	19.747
107	3	64	4	23	1.828	19.89
107	3	64	5	23	1.289	19.98
107	3	64	6	23	2.4	20
107	3	64	7	23	1.28	19.76
107	3	64	8	23	1.824	19.81
103	3	128	4	23	1.195	19.696
167	3	128	2	23	6.832	82.546
167	3	128	3	23	7.342	82.655
263	3	128	2	23	26.54	384.348
263	3	128	3	23	24.11	363.088
263	3	128	4	23	31.043	346.16
251	3	128	2	23	26.65	302.43
251	3	256	2	23	28.68	343.89
347	3	128	2	23	59.81	921.49
347	3	256	2	23	54.827	1001.73
503	3	64	2	23	208.61	3556.28

Comparing modulus values (Q) with T(d), the t-value is 1.01336. The p-value is .157791. The result is not significant at $p < .05$. Changes in modulus values did not have a significant correlation with increases in T(d). Similar values are found comparing changes in modulus values (Q) with T(G), the t-value is 1.10951. The p-value is .135894. The result is not significant at $p < .05$.

However, using a two-tailed Mann-Witney U Test showed that changes in N did have a significant correlation to T(G). The z-score is -2.27341. The p-value is .0232. The result is

significant at $p < .05$. Similar results were found applying the two-tailed Mann-Whitney U test to compare changes in N with T(d). The z-score is 5.03429. The p-value is < .00001. The result is significant at $p < .05$.

This analysis shows that increases in N have a significant impact on the time for various attacks, and thus increase security. Changes in Q do not appear to have a significant impact.

The time it takes to process each parameter indicates the parameter applicability to be used in actual NTRU encryption. The combination below seems to be the most realistic combination to use from the pool of valid NTRU parameters that we evaluated. Nonetheless, and since this current evaluation was performed using a standard computing machine, it is clear that such numbers can be easily processed with a short time, given a high-performance computing (HPC) machine.

Table VI: A sample of feasible NTRU parameters

N	**P**	**Q**	**d**	**R**
503	3	64	2	23

Tables V and VI show with N = 503 and Q = 64 the time to decompose A into a product of two matrices using BKZ or the time for Gram Schmide reduction of A is sufficiently long to indicate that NTRU can be used security. What this phase of the current experiment demonstrates is that again, larger N sizes improve the security of NTRU. However, the data in table IV also shows that there is minimal impact from changing the modulus size Q, given that it is over a minimal threshold size

5. NTRU RANDOMNESS TESTS

The first two phases of the current study examined specific cryptanalysis attacks on NTRU. The third phase investigated the randomness of the output of NTRU. Previous studies have examined the randomness of NTRU output [23], [24]. Any effective cryptographic algorithm should produce ciphertext that is random [25], [26], [27]. Furthermore, the United States National Institute of Standards (NIST) randomness tests are considered appropriate for testing the output of cryptographic algorithms [28],[29].

These tests were performed using the published code for NTRU Prime, the 20171130 Optimized Version. This version of NTRU Prime encrypts data 32 bytes/256 bits at a time. The resulting cipher text is 1175 bytes in length. This significant code expansion is typical of public-key algorithms.

2,528 bytes of plaintext from Scene I of Hamlet was used as the plaintext. This number was chosen because it is a multiple of 32 bytes. The resulting cipher text was a total of 92,825 bytes in 79 sections of 1,175 bytes each. The source code required modification to allow for the use of data from an external file for input to the encryption algorithm and use of the same key for all data encrypted in a single session. The following changes were made:

It was modified to process plaintext input from a text file.

It was modified to use the same public key /secret key pair for encrypting each piece of the text file, vs. the original source code's use of different keys for every 32 bytes of plaintext.

The NTRU Prime reference code produces a single output file in ASCII text format. For each run (processing of 32 bytes of ciphertext) it provides lines of data representing public and secret

keys, plaintext input, and ciphertext output. The data is printed in hex format with ASCII representation of 2 hex characters per byte of output. A label, "ct=", precedes each string representing 1,175 bytes of ciphertext.

The Linux "grep" command was used to extract the ciphertext strings. The Linux "cut" command was used to remove the label. Finally, the Linux "xxd" command was used to translate the hex representation into binary data for processing. A total of 79 binary files, one for each run, was created.

The industry standard software Cryptool V1.4.41 was used to determine the entropy of ciphertext [30], [31]. The quality was measured using the FIPS PUB-140-1 Test Battery Feature. This feature measures entropy (0-8) and other characteristics of up to the first 2,500 bytes of a file. Ciphertext was combined into 2,500-byte sections and 10 sections were selected for measurement.

The entropy in all cases was measured as 7.9 accurate to two significant digits. In all cases, the ciphertext passed the FIPS-PUB-140-1 test battery.

Public key = 1047 bytes
Secret key = 1238 bytes

Note: Can produce a separate entropy measurement for each of the 79 files of 1175 bytes each. Also note that the FIPS 140 tool provides 4 significant digits of entropy measurement vs 3 digits for the "entropy only" measurement in Cryptool. The specific entropy values for the files are given in table VI.

Table VI: Entropy Values

Sample	**Cipher Text Entropy Value (bits/character)**	**Plain Text Entropy Value (bits/character)**
1	7.914	4.378
2	7.914	4.78
3	7.925	4.465
4	7.913	4.396
5	7.916	4.354
6	7.921	4.351
7	7.918	4.434
8	7.921	4.479
9	7.931	4.354
10	7.913	4.306
11	7.914	4.315
12	7.914	4.407
13	7.925	4.429

The entropy values (shown in table VI) for the cipher text have a mean of 7.9186 and a standard deviation of 0.00564 indicating minimal variance in the data. This is relevant because it demonstrates that NTRU produces cyphertext that appears random, regardless of input. Statistically analyzing the entropy values of the ciphertext vs the plaintext demonstrated a significant improvement in entropy. The t-value is -104.24726. The p-value is < .00001. The p value result is significant at a confidence level of $p < .05$. This demonstrates that NTRU produces a statistically significant increase in entropy. This is a desirable outcome from any encryption algorithm.

The Kuskal-Wallis test also showed a statistically significant increase in entropy. The H statistic is 18.7778 (1, N = 26). The p-value is .00001. The result is significant at a confidence level of $p < .05$. The data demonstrates that NTRU produces substantial entropy in the ciphertext produced.

6. CONCLUSIONS

The current study utilized a three-pronged approach to testing NTRU. Three separate sub-teams, working independently, found NTRU met their tests security metrics. The first sub-teams study found that, contrary to Yang's studies, larger N sizes substantially impacted the success of Lower Dimension Lattice (LDL) attacks and improved the security of NTRU.

The second sub-team found that larger N sizes had a significant effect on time to decompose A into a product of two matrices having a special form (using BKZ) as well as time to Gram Schmidt reduced A (Standard GS). However, this team also found that changes in modulus had no significant effect.

The work of the first two sub-teams demonstrates that larger N sizes for NTRU produces a statistically significant improvement in NTRU security. The third-sub team demonstrated that NTRU produces cipher text with a substantial level of randomness.

Bringing these three sub-studies together, the result is clear. NTRU is a secure algorithm. Additionally, increasing the size of N has a significant impact on NTRU security. This leads the authors of this study to recommend NTRU as a secure cryptographic algorithm, and to use N sizes as large as is practical, without having a deleterious effect on performance.

REFERENCES

[1] Hoffstein, J., Pipher, J., & Silverman, J. H. An introduction to mathematical cryptography. New York City, New York: Springer. 2014.

[2] Nayak, D. R., Nanda, A. K., Awasthi, P., & Kumar, L. (2015). Multiple private keys with NTRU cryptosystem. IJRCCT, vol. 4, no. 3,pp. 250-255. 2015.

[3] Peikert, C. A decade of lattice cryptography. Foundations and Trends in Theoretical Computer Science, vol. 10, no., pp. 283-424. 2016.

[4] .Fluhrer, S.R.Quantum Cryptanalysis of NTRU. IACR cryptology ePrint Archive, 2015, p.676. 2015.

[5] Jeong, S., Park, K. and Park, Y. Quantum resistant NTRU-based key distribution scheme for SIP. In 2018 International Conference on Electronics, Information, and Communication (ICEIC) (pp. 1-2). IEEE. 2018.

[6] Kuznetsov, A., Kiian, A., Lutsenko, M., Chepurko, I., & Kavun, S. Code-based cryptosystems from NIST PQC. In 2018 IEEE 9th International Conference on Dependable Systems, Services and Technologies (DESSERT) (pp. 282-287). IEEE. 2018.

[7] Alagic, G., Alagic, G., Alperin-Sheriff, J., Apon, D., Cooper, D., Dang, Q., Liu, Y.K., Miller, C., Moody, D., Peralta, R. and Perlner, R.. Status report on the first round of the NIST post-quantum cryptography standardization process. US Department of Commerce, National Institute of Standards and Technology. 2019.

[8] Bernstein, Daniel J., Chitchanok C., Tanja L., and Van Vredendaal, C.. "NTRU Prime." IACR Cryptology ePrint Archive 2016 (2016): 461.

[9] Singh, S. and Padhye, S., 2017. Cryptanalysis of NTRU with n Public Keys. In 2017 ISEA Asia Security and Privacy (ISEASP) (pp. 1-6). IEEE

[10] Liu, Z., Pan, Y. and Zhang, Z., 2019, May. Cryptanalysis of an NTRU-Based Proxy Encryption Scheme from ASIACCS'15. In International Conference on Post-Quantum Cryptography (pp. 153-166). Springer, Cham.

[11] Huerta, C.E.V., 2019. A Description of the NTRU Cryptosystem and Its Cryptanalysis via the LL Algorithm (Doctoral dissertation, San Diego State University).

[12] Albrecht, M., Bai, S. and Ducas, L., 2016, August. A subfield lattice attack on overstretched NTRU assumptions. In Annual International Cryptology Conference (pp. 153-178). Springer, Berlin, Heidelberg.

[13] Kirchner, P. and Fouque, P.A., 2017, April. Revisiting lattice attacks on overstretched NTRU parameters. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 3-26). Springer, Cham.

[14] Duong, D.H., Yasuda, M. and Takagi, T., 2017, November. Choosing Parameters for the Subfield Lattice Attack Against Overstretched NTRU. In International Conference on Information Security (pp. 79-91). Springer, Cham.

[15] Bernstein, D.J., Chuengsatiansup, C., Lange, T. and van Vredendaal, C. NTRU Prime: reducing attack surface at low cost. In International Conference on Selected Areas in Cryptography.pp. 235-260. Springer, Cham. 2017

[16] Albrecht, M.R., Curtis, B.R., Deo, A., Davidson, A., Player, R., Postlethwaite, E.W., Virdia, F. and Wunderer, T. September. Estimate all the {LWE, NTRU} schemes!. In International Conference on Security and Cryptography for Networks (pp. 351-367). Springer, Cham. 2018.

[17] Valluri, M.R., 2018. Cryptanalysis of Xinyu et al.'s NTRU-lattice based key exchange protocol. Journal of Information and Optimization Sciences, 39(2), pp.475-479.

[18] Huerta, C.E.V., A Description of the NTRU Cryptosystem and Its Cryptanalysis via the LL Algorithm (Doctoral dissertation, San Diego State University). 2019.

[19] Yang, Z., Fu, S., Qu, L. and Li, C., 2018. A lower dimension lattice attack on NTRU. Science China Information Sciences, vol. 61, no. 5, pp. 059101.

[20] Shoup, Victor. "NTL: A library for doing number theory. http." 2019.

[21] Yang, Z., Shaojing, F., Longjiang, Q., Chao, L.. A Lower Dimension Lattice Attack on NTRU Supplementary File http://scis.scichina.com/en/2018/059101-supplementary.pdf. 2017.

[22] Chen, Y., & Nguyen, P. Q. (2011, December). BKZ 2.0: Better lattice security estimates. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 1-20). Springer, Berlin, Heidelberg.

[23] Easttom, C., 2019, January. An Analysis of Leading Lattice-Based Asymmetric Cryptographic Primitives. IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), pp. 0811-0818. 2019

[24] Easttom II, W.C., 2018. A Comparative Study of Lattice Based Algorithms for Post Quantum Computing (Doctoral dissertation, Capitol Technology University). 2018.

[25] Pironio, S., Acín, A., Massar, S., de La Giroday, A. B., Matsukevich, D. N., Maunz, P., ... & Monroe, C. (2010). Random numbers certified by Bell's theorem. Nature, 464(7291), 1021-1024.

[26] Simion, E. (2015). The relevance of statistical tests in cryptography. IEEE Security & Privacy, 13(1), 66-70.

[27] Tadaki, K., & Doi, N. (2015). Cryptography and algorithmic randomness. Theory of Computing Systems, 56(3), 544-580.

[28] Georgescu, C., Petrescu-Nita, A., Simion, E., & Toma, A. (2017). NIST Randomness Tests (in) dependence. IACR Cryptology ePrint Archive, 2017, 336.

[29] Sulak, F., UĞUZ, M., Kocak, O., & DoğanaksoY, A. (2017). On the independence of statistical randomness tests included in the NIST test suite. Turkish Journal of Electrical Engineering & Computer Sciences, 25(5), 3673-3683.

[30] Balasubramanian, K. Experiments with the Cryptool Software. Algorithmic Strategies for Solving Complex Problems in Cryptography, pp. 186-194. 2018.

[31] Duan, G., Wang, Y., Li, M., Sheng, Y., Wang, J., & Zhang, S. Research on Techniques and Methods of Developing Cryptography Virtual Laboratory. International Journal of Performability Engineering, vol. 13, no. 8, pp. 1371-1381. 2017.

AUTHORS

Chuck Easttom, D.Sc., Ph.D. IEEE Senior Member, ACM Senior Member, Distinguished Speaker of the ACM and IEEE. Adjunct lecturer at Georgetown University and Adjunct professor at University of Dallas

Anas Ibrahim, PhD Researcher at Eastern Mediterranean University

Alexander Chefranov Associate Professor at Eastern Mediterranean University

Izzat Alsmadi, Ph.D. Assistant Professor at Texas A&M University IEEE Senior Member

Richard Hansen Professor of Practice Capitol Technology University