Global Journal of Computer Science and Technology: E
Network, Web & Security
Volume 14 Issue 6 Version 1.0 Year 2014
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

LWE Encryption using LZW Compression

By M. N. M. Prasad, Mohammed Ali Hussain & C.V. Sastry

KLEF University, India

Abstract- ENCRYPTION of data has become essential, for sending confidential information from one system to another system, especially in banking sector. NTRU labs have done pioneering work using a ring of truncated polynomials which was based on the impossibility (with proper choice of parameters) of finding the polynomial with knowledge of its inverse in modular arithmetic. Recently, Learning With Errors (LWE) has been studied extensively and its hardness can be linked to the near impossibility of finding the Shortest Vector on integer lattices. In this paper we have shown that a preprocessing of input before applying the LWE algorithm greatly reduces the time of encryption and decryption.

Keywords: number theory research unit (NTRU), LWE, SVP, LZW, ring of truncated polynomials, modular arithmetic.

GJCST-E Classification : I.4.2

LWEENCRYPTIONUSING LZW COMPRESSION

Strictly as per the compliance and regulations of:

RESEARCH | DIVERSITY | ETHICS

LWE Encryption using LZW Compression

M.N.M. Prasad [α], Mohammed Ali Hussain [σ] & C.V. Sastry [ρ]

Abstract- ENCRYPTION of data has become essential, for sending confidential information from one system to another system, especially in banking sector. NTRU labs have done pioneering work using a ring of truncated polynomials which was based on the impossibility (with proper choice of parameters) of finding the polynomial with knowledge of its inverse in modular arithmetic. Recently, Learning With Errors (LWE) has been studied extensively and its hardness can be linked to the near impossibility of finding the Shortest Vector on integer lattices. In this paper we have shown that a pre-processing of input before applying the LWE algorithm greatly reduces the time of encryption and decryption.

Keywords: number theory research unit (NTRU), LWE, SVP, LZW, ring of truncated polynomials, modular arithmetic.

I. Introduction

Secure transmission of data has become the key for successful completion of all transactions. NTRU Labs have created a bench-mark in secure transmission of data using a ring of truncated polynomials [1, 2, 3, 4]. Many attempts have been made to break the crypto-systems based in NTRU technique; but no successful attempt has ever been reported. However polynomial inversions are difficult to perform in modulo-arithmetic. Moreover, polynomials are to be repeatedly chosen until they could be properly inverted.

In the last three to four years, Learning With Errors (LWE) has emerged as a versatile alternative to the NTRU cryptosystems. All cryptographic constructions based on LWE [5, 6, 7] are as secure as the assumption that SVP (Smallest Vector Problem)[8,9] is hard on integer lattices.

The LWE problem can be stated as follows:

Recover s, given $A . s \cong b$ where $s \in Z_q^n$, $b \in Z_q^n$ and A is m × n matrix with m > n and Z_q^n is set of integer vectors of size n and modulo q. In other words, we are given a set of m equations in n unknowns and the right hand side slightly perturbed with the error vector chosen from normal distribution χ with low standard deviation. More precisely we say that an solves LWE[10] if we can recover s, given that the errors are distributed according to the error distribution χ and the elements of A are chosen uniformly at random from Z_q^n [10].

The number of equations or the number of rows in the matrix is irrelevant since additional equations can be formed that are as good as new, by adding the given equations.

One way to obtain a solution to the LWE problem is to repeatedly form new equations until we get the first row of the matrix A as (1, 0, 0, 0, ..., 0) which gives a solution to the first component of s. We can repetitively apply the same procedure for the other components of s. However the probability of obtaining such a solution is almost nil, of the order of q^{-n}, and the set of equations needed are $2^{O(n \log n)}$ and with a similar running time.

The algorithm can be stated as follows:

Private Key: s, chosen uniformly at random from Z_q^n .

Public Key: m samples of (A_i, b_i).

Encryption: for each bit of the message, we chose at random a set T from the 2^m subsets of the m equations. The encryption is $(\sum_{i\in T} \mathrm{A}_i , \sum_{i\in T} \mathrm{b}_i)$, if the bit is zero and the encryption is $(\sum_{i\in T} \mathrm{A}_i , \left\lfloor\frac{q}{2}\right\rfloor + \sum_{i\in T} \mathrm{b}_i)$ if the bit is 1.

Decryption: The decryption of the pair (a, b) is 0 if b – <a, s> is closer to 0 than to $\left\lfloor\frac{q}{2}\right\rfloor$, 1 otherwise.

However, transmitting a text with bitwise encryption will be cumber-some and time-taking. We use a slightly modified version of the algorithm to encrypt 'l' bits simultaneously. We choose A, S uniformly at random from $Z_q^{m\times n}$ and $Z_q^{n\times l}$ respectively and S is the private key. We generate the error matrix $E \in Z_q^{m\times l}$ by choosing each entry according to normal distribution$\chi\alpha$, where α is a measure of standard deviation which is usually chosen as $\sqrt{\alpha q}$ and α is small. The public key is (A, B) where B= A.S + E.

Farther simplification is made by choosing the elements of A in the form of a circulant matrix. In other words we have chosen A as [11]

$$\begin{bmatrix} a_1 & a_2 & a_3 & a_4 & . & . & . & . & . & a_n \\ a_2 & a_3 & a_4 & a_5 & & . & . & . & & -a_1 \\ a_3 & a_4 & a_5 & a_6 & . & . & . & & -a_1 & -a_2 \\ . & & & & . & . & . & . & . & \\ . & & & & . & . & . & . & . & \end{bmatrix}$$

Let v be a vector belonging to message space Z_t^l . Choose a vector $a \in \{-1, 0, 1\}^m$ uniformly at random. The cipher text u corresponding to the

Author α: Research Scholar, Department of Computer Science and Engineering, KLEF University, Vaddeswaram, Guntur, Andhra Pradesh, India. e-mail: prasadmushini@gmail.com.
Author σ: Professor, Department of Electronics and Computer Engineering, KLEF University, Vaddeswaram, Guntur, Andhra Pradesh, India. e-mail: dralihussain@kluniversity.in.
Author ρ: Professor Department of Computer Science and Engineering, Regency Institute of Technology, Yanam, U.T. of Podicherry, India. e-mail: cvsastry40@yahoo.co.in.

message v is ($u = A^T a, C = B^T a + f(v)$) where f is an invertible mapping from the message space Z_t^l to Z_q^l and in this paper we have chosen the mapping as a multiplication of each co-ordinate by q/t and rounding to the nearest integer.

The original message can be recovered from the cipher text (u, C) using the private key S as $f^{-1}(C - S^T u)$ which can be seen as follows:

$$\begin{aligned} f^{-1}(C - S^T u) &= f^{-1}(B^T a + f(v) - S^T A^T a) \\ &= f^{-1}((AS+E)^T r + f(v) - S^T A^T a) \\ &= f^{-1}(E^T a + f(v)) \\ &= f^{-1}(E^T a) + v \end{aligned}$$

If a decryption error is to occur, say in the first letter, the first co-ordinate of $E^T a$ must be greater than $q/(2t)$ in absolute value the probability of which is shown to be negligible [11].

However, some pre-processing of data greatly helps to reduce the time for encryption and decryption as well as time for transmission. We choose to compress the data before encryption using LZW (Lemple-Ziv-Welch)[12,13,14] technique and encrypt the reduced text. The LZW method of compression is based on dictionary structure. It creates a dictionary of its own for each character or a string of the input text. It is known to be a lossless compression and the percentage of reduction in the text is approximately 40% [15].

Another frequently used compression algorithm is the well known Huffman Technique [16,17,18] which constructs a binary tree based on the frequency of the occurrence of the letters and the corresponding code is generated. We have also used Huffman algorithm on the same text and compared the two compression technique used with LWE.

II. Illustration of the Proposed Algorithm

The parameters of the proposed algorithm are chosen as q =2003, t =2, n =136, l =136, alpha = 0.0065 and m =2008 [11]

Original text message

str: wild animals, rocks, forest, beaches, and in general those things that have not been substantially altered by human intervention, or which persist despite human intervention.

The Compressed message using LZW is cmes=

$$\begin{matrix} 87 & 105 & 108 & 100 & . & . & . \\ & & 352 & 110 & 46 & & \end{matrix}$$

Where the integers indicate the indices to the patterns generated by the compression algorithm.

Then we convert the message vector as obtained above into a binary

v=

$$\begin{matrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 & . & . & . & 0 & 0 & \end{matrix}$$

$$f(v) = round(v \times \tfrac{q}{t}) =$$

$$\begin{matrix} 0 & 0 & 1002 & 0 & 1002 & 0 & 1002 & 1002 & 1002 \\ 0 & 0 & 1002 & 1002 & 0 & 1002 & 0 & 0 & 1002 \\ & & & & . & . & . & 0 & 0 \end{matrix}$$

$A \in Z_q^{m \times n}$ is chosen as

1591	757	1974	.	.	.	1216	1991
757	1974	892	.	.	.	1991	-1591
1974	892	1760	.	.	.	-1591	-757
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
1137	1368	375	.	.	.	-887	-1301
1368	375	449	.	.	.	-1301	-1137
375	449	154	.	.	.	-1137	-1368

$S \in Z_q^{n \times l}$ is as follows

1759	2	154	.	.	.	1044	1218
764	1434	996	.	.	.	1703	945
475	1846	462	.	.	.	956	1644
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
136	591	1728	.	.	.	1489	708
782	945	84	.	.	.	121	1215
1271	916	1500	.	.	.	1439	76

$E \in Z_q^{n \times l}$ is as follows

-3	0	-1	.	.	.	2	3
-2	0	0	.	.	.	0	0
-3	-1	-2	.	.	.	-3	0
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
-7	4	-1	.	.	.	8	1
-2	-2	0	.	.	.	2	-6
-4	1	4	.	.	.	-1	2

$$B = A \times S + E (mod\ q) =$$

$$\begin{bmatrix} 1637 & 130 & 771 & . & . & . & 671 & 453 \\ 908 & 123 & 438 & . & . & . & 1399 & 264 \\ 527 & 963 & 184 & . & . & . & 1573 & 61 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ 720 & 312 & 299 & . & . & . & 1955 & 130 \\ 403 & 389 & 357 & . & . & . & 1428 & 1659 \\ 277 & 1467 & 1094 & . & . & . & 1056 & 39 \end{bmatrix}$$

Let $\boldsymbol{a} = \{-\mathbf{1}, \mathbf{0}, \mathbf{1}\}^{\boldsymbol{m}}$ where the elements are chosen randomly.

$$\begin{bmatrix} -1 & 0 & -1 & -1 & . & . & . & 0 & 1 & 1 & 1 \end{bmatrix}$$

$$\boldsymbol{C} = \boldsymbol{B}^{\boldsymbol{T}} \times \boldsymbol{a} + \boldsymbol{f}(\boldsymbol{v})(\boldsymbol{mod}\ \boldsymbol{q}) =$$

$$\begin{bmatrix} 98 & 1396 & 408 & 1049 & . & . & . & 291 & 1356 & 193 \end{bmatrix}$$

$$\boldsymbol{u} = \boldsymbol{A}^{\boldsymbol{T}} \times \boldsymbol{a}\ (\boldsymbol{mod}\ \boldsymbol{q}) =$$

$$\begin{bmatrix} 314 & 1840 & 1588 & 148 & . & . & . & 988 & 1447 & 1125 \end{bmatrix}$$

$$\boldsymbol{D} = \boldsymbol{C} - \boldsymbol{S}^{\boldsymbol{T}} \times \boldsymbol{u}\ (\boldsymbol{mod}\ \boldsymbol{q}) =$$

$$\begin{bmatrix} 1952 & 1992 & 13 & 58 & . & . & . & 15 & 1998 & 143 \end{bmatrix}$$

D/(q/t) =

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ & 1 & 0 & 1 & 0 & 0 & 1 & . & . & . & 0 & 0 \end{bmatrix}$$

Convert binary to decimal

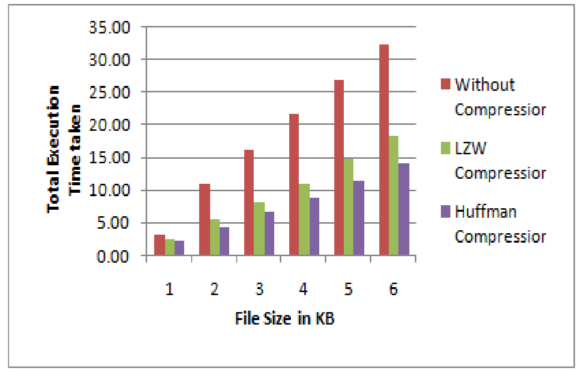
$$\begin{bmatrix} 87 & 105 & 108 & 100 & . & . & . & 87 & 105 & 108 & 100 \end{bmatrix}$$

When the compression process is reversed we get the original message:

III. Experimental Results

The following table gives the total execution time taken for a direct encryption and decryption, encryption and decryption after a LZW compression and encryption and decryption after a Huffman compression:

File Size in KB	Total Execution time without Compression	Total Execution time with LZW	Total Execution time with Huffman
1	3.13	2.4289	2.10777
2	10.84	5.3588	4.28654
3	16.26	8.1736	6.56431
4	21.63	10.9583	8.90506
5	27.00	14.7273	11.39183
6	32.43	18.2190	13.98658



IV. Conclusions

In this paper we have used ring-LWE to encrypt an input text. The text to be transmitted has been initially compressed using LZW Technique and the compressed text is encrypted using LWE. We have also used Huffman coding algorithm for compression for comparison purpose. It has been observed that compressing the input text greatly reduces the total time of transmission and Huffman coding works out to be better.

References Références Referencias

1. J. Hoffstein, J. Pipher, and J. H. Silverman. NTRU: a ring based public key cryptosystem. In Proceedings of ANTS-III, volume 1423 of LNCS, pages 267–288. Springer, June 1998.
2. J. Hoffstein, N. A. H. Graham, J. Pipher, J. H. Silverman, and W. Whyte. NTRUSIGN: Digital signatures using the NTRU lattice. In Proc. of CT-RSA, volume 2612 of Lecture Notes in Comput. Sci.,pages 122–140. Springer-Verlag, 2003.
3. J. Hoffstein, N. Howgrave-Graham, J. Pipher, and J. H. Silverman. Hybrid lattice reduction and meet in the middle resistant parameter selection for NTRUEncrypt. Submission/ contribution to ieee p1363.1, NTRU Cryptosystems, Inc., URL http://grouper.ieee.org/groups/1363/lattPK/ submissions.html#2007-02, 2007.
4. M.N.M.Prasad, Mohammed Ali Hussain, C.V.Sastry "NTRU Encryption using Huffman comprssion" , Journal of Theoretical and Applied Information Technology, Volume.59,No.2, pages 379-384,Jan 2014.
5. D. Micciancio and O. Regev. Worst-case to average-case reductions based on Gaussian measures. InProc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS), pages 372–381, 2004.
6. D. Micciancio. Improving lattice based cryptosystems using the hermite normal form. In J. Silverman,editor, Cryptography and Lattices Conference — CaLC 2001, volume 2146 of Lecture Notes in Computer Science, pages 126–145,

Providence, Rhode Island, Mar. 2001. Springer-Verlag.
7. D. Micciancio. Lattices in cryptography and cryptanalysis, 2002. Lecture notes of a course given in UC San Diego.
8. Haviv and O. Regev. Tensor-based hardness of the shortest vector problem to within almost polynomial factors. In Proc. 39th ACM Symp. on Theory of Computing (STOC), pages 469–477, 2007.
9. S. Khot. Hardness of approximating the shortest vector problem in lattices. In Proc. 45th Annual IEEE Symp. on Foundations of Computer Science (FOCS), pages 126–135, 2004.
10. O. Regev. On lattices, learning with errors, random linear codes, and cryptography. J.ACM, 56(6),2009.
11. D. Micciancio and O. Regev. Lattice-based cryptography. In D. J. Bernstein and J. Buch-mann, editors, Post-quantum Cryprography. Springer, 2008.
12. Parvinder Singh,ManojDuhan and Priyanka (2006)" Enhancing LZW Algorithm to Increase Overall Performance", Annual IEEE Indian Conference,pp1-4.
13. Ming-Bo Lin, Jang-Feng Lee, G. E. Jan,(2006)"A Lossless Data Compression and Decompression Algorithm and Its Hardware Architecture" VLSI IEEE Transactions ,Vol.14,pp925-936.
14. Mateosian, R, "Introduction to Data Compression" (1996), vol.16.
15. V.Sulochana Verma, Design and Implementation of LZW Data Compression Algorithm, in IJIST vol2, No.4, July 2012.
16. Salomon D. Huffman Coding available at http://www.springer.com/978-1-84800-071-1, ISBN:978-1-84800-071-1,Softcover,2008
17. J. S. Vitter: Algorithm 673 Dynamic Huffman Coding. ACM Transactions on Mathematical Software, 15(2), June 1989, 158-167 (1989).
18. Lecture-15: Huffman Coding (CLRS-16.3) available at www.cse.ust.hk/~dekai/271/ *notes/L15/L15.pdf*

Global Journals Inc. (US) Guidelines Handbook 2014

www.GlobalJournals.org