An Enhanced RSA Encryption Scheme based on Mixed Radix Conversion with Data Compression

Salifu Abdul-Mumin
Department of Computer Science
University for Development Studies
Navrongo, Ghana

Alhassan Abdul-Barik
Department of Computer Science
University for Development Studies
Navrongo, Ghana

ABSTRACT

In this paper, an enhanced Riveset Shamir Adleman (RSA) encryption scheme with data compression features based on Residue Number Systems (RNS) has been proposed and implemented. This scheme is implemented using the moduli set, $(2^n - 1, 2^n)$. The scheme has two level of encryption and a two level of decryption to form a double layer public key encryption scheme. The first level comprises the classical RSA encryption algorithm and in the second layer, RNS number representation is utilised. Mixed Radix Conversion is employed in the first level of decryption and in the second level; the classical RSA decryption process is used. This will ensure a secured data transmission of messages of varying lengths. The private key length is further enhanced as the moduli set are part of the private key for decryption processes. The residues which form the basis of the transmitted cipher text are smaller in terms of number of bits than the cipher text generated from the classical RSA encryption scheme. This will increase the rate of data transmission in the communication channel.

Keywords

Riveset Shamir Adleman (RSA), Residue Number Systems (RNS), Encryption, Decryption, Security, Data Compression

1. INTRODUCTION

One of the factors for improving the quality of service in data transmission is to increase the speed of the transmission channel. The speed of the transmission channel can be increased by increasing the transmission bandwidth; the range of frequencies available for data transmission and or by having fewer number of bits transmitted across a communication channel.

Cypher text generated from most public key encryption schemes especially the classical RSA cryptosystem have number of bits far more the plaintext. This increases the traffic of the communication channel and thus slows down the rate of data transfer between communicating parties. Data compression is the major technology used for reducing the size of a transmitted data. Usage of data compression in data transmission nodes can significantly reduce the volume of data to be transmitted. This reduces the transmission power thereby increase the lifetime of the network. It is shown in [1] that the use of data compression leads to a significant reduction in the number of housekeeping data and thus increases the total bandwidth in Wireless Sensor Networks.

For the last ten years or more, the world has witnessed a revolution transformation in the way we communicate. This transformation includes the proliferation of the internet; the explosive growth of the IP network and mobile communications; and the ever-increasing importance of video communication. Data compression is one of the enabling technologies for each of these aspects of the multimedia revolution [2]. It is not possible to put images, let alone audio and video, on the internet if it were not for data compression algorithms. Cellular phones would not be able to provide communication with increasing clarity without data compression. The advent of digital TV would not be possible without compression. Data compression initially was for the domain of a relatively small group of engineers and scientists. In recent times, everyone in one or the other uses data compression. Make a long-distance call and you are using compression. Use your modem, or your fax machine, and you will benefit from compression. Listen to music on your *mp3* player or watch a DVD and you are being entertained courtesy of compression [2].

Given the explosive growth of data that needs to be transmitted and stored, calls for the need to focus on the design and implementation of better transmission and storage technologies. This is happening, but it is not enough [2]. There have been significant advances that permit larger and larger volumes of information to be stored and transmitted without using compression, including CD-ROMs, optical fibers, Asymmetric Digital Subscriber Lines (ADSL), and cable modems. However, while it is true that both storage and transmission capacities are steadily increasing with new technological innovations, as a corollary to Parkinson’s First Law [3], it seems that the need for mass storage and transmission increases at least twice as fast as storage and transmission capacities improve. Then there are situations in which capacity has not increased significantly [2].

2. DATA COMPRESSION

Data compression is the technology used for the reduction of the amount of information to be transmitted or stored by removing excess or irrelevant information without the loss of the ability to reconstruct the original data. Data compression algorithms are employed in different file formats including sound, video, text, and image, and can be classified as either lossless or, lossy dictionary or non-dictionary based [4], [5], [6].

Compression technique or compression algorithm refers to two algorithms. There is the compression algorithm that takes an input X and generates a representation X_C that requires fewer bits, and there is a reconstruction algorithm that operates on the compressed representation X_C to generate the reconstruction Y.

Lossless compression techniques, as their name implies, involve no loss of information. In this type of compression, the original data can be reconstructed exactly from the compressed data. Lossless compression is mostly used for applications that cannot tolerate any difference between the original and reconstructed data. [2].Text compression is an important application area for lossless compression. It is very essential that the reconstruction is same as the original text, as little differences can result in statements with entirely different meanings. Consider the sentences “Do not send money” and “Do now send money.”

Lossy compression techniques involve some loss of information. In this compression technique, the original dat cannot be recovered or reconstructed exactly. However, for accepting this distortion in the reconstruction, we can generally obtain much higher compression ratios than is possible with lossless compression [2]. In many applications, this lack of exact reconstruction is not a problem. For instance, when transmitting or storing speech, the exact value of each sample of speech is not necessary.

In many applications, the output of the source consists of recurring patterns. A classic example is a text source in which certain patterns or words recur constantly. Also, there are certain patterns that simply do not occur, or if they do, occur with great rarity. A very reasonable approach to encoding such sources is to keep a list, or dictionary, of frequently occurring patterns. When these patterns appear in the source output, they are encoded with a reference to the dictionary. If the pattern does not appear in the dictionary, then it can be encoded using some other, less efficient, method. In effect we are splitting the input into two classes, frequently occurring patterns and infrequently occurring patterns. Non-dictionary based techniques are those that do not keep a list or a dictionary.

3. RELATED WORKS

Morse code is one an early example of data compression, developed by Samuel Morse in the mid-19th century. In this example, letters sent by telegraph are encoded with dots and dashes. Morse noticed that certain letters occurred more often than others. In order to reduce the average time required to send a message, he assigned shorter sequences to letters that occur more frequently, such as e (.) and a (·−), and longer sequences to letters that occur less frequently, such as q (− −·−) and j (· − − −) [3].

In recent days, the most widely used data compression algorithms are based on the work of Ziv and Lempel [7]. These are dynamic algorithms that build a dictionary representative of the input text and code dictionary entries using fixed-length code words. The compression algorithm typically reduces a file to 40-50% of its original size. It is very fast but has a large memory requirements (450 kbytes) [8].

A research on improving the traditional Huffman’s data encoding algorithm was carried out by Alhassan et al, using RNS [9]. His work demonstrated an enhanced data compression scheme with the traditional Huffman’s encoding where the frequency of occurrences of each character is used to generate binary codes.

Abdul-Barik et al also applied RNS to the LZW algorithm using the traditional moduli set for efficient and secured data encoding and decoding [10]. They modified the LZW algorithm by applying RNS resulting in new efficient data encryption and decryption schemes, new encoder and decoder pairs which also allows for conversion from one number representation to the other. The output of the LZW algorithm has been further modified to allow for secrete order bit stream or channel or residual archiving or transmission of data through networks so that network intruders cannot make meaning of intercepted data.

4. ENCRYPTION-COMPRESSION SCHEME (FORWARD CONVERTER) FOR THE MODULI SET, $(2^n - 1, 2^n)$

Given the moduli set above, let $m_1 = 2^n - 1,\ m_2 = 2^n$, where $m_i, i = 1, 2$ are the moduli representing the channel-order of the moduli set.

Now, in order to convert a number X from binary/decimal to its residues equivalent (forward conversion) using the moduli set; we compute the $r_{i,\{i=1,2\}}$, the residue set by performing $r_i = |X|_{m_i}$ which is the modulus operation on X with respect to each modulus. $M = \prod_{i=1}^{2} m_i = [2^{2n} - 2^n]$ (1)

Which indicates that X is a $(2n)$-bit number and represented in binary as:

$$X = x_{2n-1}x_{2n-2} \dots x_1x_0 \qquad (2)$$

It follows therefore that the r_i’s can be computed as follows;

- r_2 is the n least significant bit (LSB) of X in binary.
- For r_1, we partition X into two n-bit blocks B_1, B_2, where;

$$\left.\begin{aligned} B_2 &= \sum_{j=n}^{n+1} x_j 2^{j-n} \\ B_2 &= \sum_{j=0}^{n-1} x_j 2^j \end{aligned}\right\} \qquad (3)$$

This implies

$X = B_1 + 2^n B_2$ (4)

Therefore,

$r_1 = |X|_{2^n-1} = |B_1 + 2^n B_2|_{2^n-1}$

$=||B_1|_{2^n-1} + |B_2 2^n|_{2^n-1}|_{2^n-1}$

$= |B_1 + B_2|_{2^n-1}$

Example1:

Given the moduli set $\{2^n - 1, 2^n\}$, take $n = 4$ and a number, $X = 50$. Then the conversion process is as follows;

$50 = 110010 = 00110010$ (8-bits, since X is a $(2n) - bit$ number)

Since $n = 4$, we partition X into two 4-bits blocks

Thus, $B_1 = 0010,\ and\ B_2 = 0011$

Therefore;

$r_1 = |B_1 + B_2|_{2^n-1}$

$\Rightarrow |50|_{2^4-1} = |50|_{15} = |2 + 3|_{15} = 5$, and

Since r_2 is the LSB of X in binary, then we have $B_1 = 0010 = 2$.

This implies that $|50|_{14,16} = \{5, 2\}$

The hardware realisation of the forward conversion is achieved by using simple fast adder like carry propagate adder (CPA) for two bits addition.

The hardware requirements of this architecture and the delay imposed in computing the residues will be as follows:

$$Area(A) = A_{CPA}$$
$$= (n-1)A$$

And

$$Delay(D) = D_{CPA}$$
$$= 2(n-1)D = (2n-2)D$$

5. THE TRANSMISSION SYSTEM

The residue channels are the medium for the data transmission. The cipher text generated from the RSA encryption scheme passed through a residue channel to produce the residues with respect to each modulus in the moduli set. These residues are transmitted and received by a reverse converter where the original cipher text is reconstructed and the decryption is carried out by the RSA encryption scheme.

6. FIRST LEVEL OF DECRYPTION (REVERSE CONVERTER) FOR THE MODULI SET, $(2^n-1, 2^n)$

The Mixed Radix Conversion (MRC) is utilised to convert the number X in RNS representation to its binary/decimal equivalent.

The general form of the MRC is as follows;

$$X = d_1 + d_2m_1 + d_3m_1m_2 + \cdots + d_nm_1m_2m_3\ldots m_{n-1} \quad (5)$$

Where $d_i, i = 1,2,\ldots,n$ are the Mixed Radix Digits (MRDs) and are calculated as follows:

$$d_1 = x_1$$

$$d_2 = \left|(x_2 - d_1)|m_1^{-1}|_{m_2}\right|_{m_2}$$

$$d_3 = \left|\left((x_3 - d_1)|m_1^{-1}|_{m_3} - d_2\right)|m_2^{-1}|_{m_3}\right|_{m_3}$$

$$\vdots$$

$$d_n = \left|\left(\ldots\left((x_3 - d_1)|m_1^{-1}|_{m_n} - d_2\right)|m_2^{-1}|_{m_n} - \cdots - d_{n-1}\right)|m_{n-1}^{-1}|_{m_n}\right|_{m_n} \quad (6)$$

That is, X in the interval $[0, M]$ can be uniquely represented.

Theorem1: Given the moduli set, $(2^n-1, 2^n)$ where $m_1 = 2^n - 1$, $m_2 = 2^n$, for every integer $n > 1$, following holds true;

$$|m_1^{-1}|_{m_2} = 2^n - 1 \quad (7)$$

Proof*:* If it can be demonstrated that $\left|m_i^{-1} \times m_i\right|_{m_i} = 1$, then m_i^{-1} is the multiplicative inverse of m_i with respect to m_i. Thus;

$$|(2^n-1)(2^n-1)|_{2^n}$$

$$|2^{2n} - 2^n - 2^n + 1|_{2^n} = 1 \; as \; requred$$

Therefore (6) can be re-written as;

$$d_1 = x_1$$

$$d_2 = |(x_2 - d_1)(2^n-1)|_{2^n} = |2^nx_2 - 2^nd_1 - x_2 + d_1|_{2^n} \quad (8)$$

And equation (5) becomes

$$X = x_1 + d_2(2^n-1) = x_1 + 2^nd_2 - d_2 \quad (9)$$

7. HARDWARE REALISATION

Equation (8) and equation (9) are simplified as follows;

$$d_1 = \underbrace{x_{1,n-2}\,\underbrace{x_{1,n-3}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{1,1}}_{}\,\underbrace{x_{1,0}}_{}}_{n-1} \quad (10)$$

$$d_2 = |2^n\,\underbrace{(x_{2,n-1}\,\underbrace{x_{2,n-2}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{2,1}}_{}\,\underbrace{x_{2,0}}_{})}_{n}$$
$$-\,2^n\underbrace{(x_{1,n-2}\,\underbrace{x_{1,n-3}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{1,1}}_{}\,\underbrace{x_{1,0}}_{})}_{n-1}$$
$$-\,\underbrace{(x_{2,n-1}\,\underbrace{x_{2,n-2}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{2,1}}_{}\,\underbrace{x_{2,0}}_{})}_{n}$$
$$+\,\underbrace{x_{1,n-2}\,\underbrace{x_{1,n-3}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{1,1}}_{}\,\underbrace{x_{1,0}}_{}}_{n-1}|_{2^n}$$

$$= |\underbrace{\underbrace{x_{2,n-1}\,\underbrace{x_{2,n-2}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{2,1}}_{}\,\underbrace{x_{2,0}}_{}}_{n}\overbrace{00.\ldots.00}^{n}}_{2n}$$
$$+\,\underbrace{\underbrace{\bar{x}_{1,n-2}\,\underbrace{\bar{x}_{1,n-3}}_{}\cdots\cdots\cdots\cdots\underbrace{\bar{x}_{1,1}}_{}\,\underbrace{\bar{x}_{1,0}}_{}}_{n-1}\overbrace{11.\ldots.11}^{n}}_{2n-1}$$
$$+\,\underbrace{(\bar{x}_{2,n-1}\,\underbrace{\bar{x}_{2,n-2}}_{}\cdots\cdots\cdots\cdots\underbrace{\bar{x}_{2,1}}_{}\,\underbrace{\bar{x}_{2,0}}_{})}_{n}$$
$$+\,\underbrace{x_{1,n-2}\,\underbrace{x_{1,n-3}}_{}\cdots\cdots\cdots\cdots\underbrace{x_{1,1}}_{}\,\underbrace{x_{1,0}}_{}}_{n-1}|_{2^n}$$

$$d_2 = \underbrace{d_{2,n-1}d_{2,n-2}\ldots\ldots\ldots.d_{2,1}d_{2,0}}_{n} \quad (11)$$

From equation (9), let $A = x_1 + 2^nd_2$

$$A = \underbrace{x_{1,n-2}x_{1,n-3}\ldots x_{1,1}x_{1,0}}_{n-1} \bowtie \underbrace{d_{2,n-1}d_{2,n-2}\ldots d_{2,1}d_{2,0}}_{n}\overbrace{00\ldots0}^{n}$$
$$= \underbrace{x_{1,n-2}x_{1,n-3}\ldots x_{1,1}x_{1,0}d_{2,n-1}d_{2,n-2}\ldots d_{2,1}d_{2,0}}_{2n-1}$$
$$= A_{2n-2}A_{2n-3}\ldots A_1A_0$$

Therefore

$$X = A_{2n-2}A_{2n-3}\ldots A_1A_0 - \underbrace{d_{2,n-1}d_{2,n-2}\ldots\ldots\ldots.d_{2,1}d_{2,0}}_{n}$$

$$A_{2n-2}A_{2n-3}\ldots A_1A_0 + \underbrace{\bar{d}_{2,n-1}\bar{d}_{2,n-2}\ldots\ldots\ldots.\bar{d}_{2,1}\bar{d}_{2,0}}_{n}$$

8. PERFORMANCE EVALUATION

Given the moduli set $\{2^n-1, 2^n\}$, take $n = 7$ for encryption and transmission of 30, 25 and 20 and $n = 6$ for the encryption and transmission of 15, 10 and 5. Let p = 101, q = 113, e = 3 (e is an odd public exponent between 3 and n – 1 that is relatively prime to p – 1 and q – 1). Then $N = pq = 101*113 = 11413, \Phi(N) = (p-1)(q-1) = 11200$ which implies $e.d = 1 \bmod \Phi(N).$ Therefore d = 7467. The table shows the cipher text of some messages, m produced by the RSA encryption scheme.

Table I: RSA Cipher Text Bits Evaluation

Plaintext (m)	m^e	Cipher Text (c)	No of Transmitted Bits
30	27000	4174	13
25	15625	4212	13
20	8000	8000	13
15	3375	3375	12
10	1000	1000	10
5	125	125	7

In the above table, transmitting messages, 30 and 25 is secured. However, transmitting the messages, 20, 15, and 5 are not secured since $m^e < n$. Taking the e^{th} root of the cipher text will easily reveal the message m.

Table II: Proposed Scheme Cipher Text Bits Evaluation

Plaintext (m)	m^e	Cipher Text (C_1)	Cipher Text (C_2)	No of Transmitted Bits	Percentage Gain
30	27000	4174	$(110, 78)$	7	46.2
25	15625	4212	$(21, 116)$	7	46.153
20	8000	8000	$(126, 64)$	7	46.153
15	3375	3375	$(36, 47)$	6	50
10	1000	1000	$(55, 40)$	6	40
5	125	125	$(62, 61)$	6	14.3

In the table above, transmitting all the messages are secured. The second level of encryption will transform the cipher text generated from the classical RSA encryption scheme into residues which will be transmitted by the residue channel. Here, more information is needed to reveal the message as the moduli set are part of the private key of the RSA encryption scheme.

In this scheme, the number of bits to be transmitted is less than the number bits generated in the classical RSA encryption scheme. From table II, transmitting the message 30 yielded a 46.2% compression gain. As the number of bits in the transmitted message decreases, the percentage gain in compression also decreases to a 14.3% when transmitting the message 5. This will reduce the data traffic and hence increase the data transmission speed.

9. CONCLUSION

A new data compression scheme based RSA cryptosystem have been designed and implemented using residue number system. The new system has two level of encryption and two level of decryption. A forward converter is designed using the moduli set $(2^n - 1, 2^n)$ for purposes of the second level of encryption. The residues generated from this level are transmitted using a residue channel. A reverse converter is designed for the first level of decryption. The new scheme is evaluated with the classical RSA encryption scheme and proposed scheme outperformed the classical RSA in terms of both security and data compressioin.

10. REFERENCES

[1] Yatskiv V.: Nonlinear data coding in wireless sensor networks/ Vasyl Yatskiv, Su Jun,Nataliya Yatskiv, Anatoly Sachenko // International Journal of Computing. – Vol. 10,Issue 4, 2011, p. 383-390.

[2] Khalid Sayood: Introduction to Data Compression 3rd ed., Morgan Kaufmann Publishers, 2006

[3] Bell T.C., Cleary J.G., and Witten I.H.. Text Compression. Advanced Reference Series. Prentice Hall, Englewood Cliffs, NJ, 1990.

[4] Amit J., “Comparative Study of Dictionary Based Compression Algorithms on Text Data”. International Journal of Computer Engineering and Applications, Vol I, Issue II, Pg.1-11, India, 2014

[5] Jane H., Trivedi J., “A Survey on Different Compression Techniques Algorithm for Data Compression”, International Journal of Advanced Research in Computer Science and Technology, Vol II, Issue III, Pg1-5, 2014.

[6] Welch T. A., “A Technique for High Performance Data Compression”, IEEE, Sperry, Research Centre, Pp 8-19, 1984.

[7] Ziv, J. and Lempel, A. Compression of individual sequences via variable rate coding. IEEE Trans. Inf. Theory 24, 5 (sept, 1978), pp: 530-536

[8] Lelewer Debra A. and Hirschberg Daniel S.: An Order-2 Context Model for Data Compression with Reduced Time and Space Requirements, Technical Report No. 90-33

[9] A. Alhassan, I. Saeed, and P.A. Agbedemnab, “The Huffman’s Method of Secured Data Encoding and Error Correction using Residue Number System (RNS)”, Communication on Applied Electronics (CAE) Journal, Foundation of Computer Science (FCS), New York, USA, 2015.

[10] Abdul-Barik Alhassan, Kazeem A. Gbolagade, and Edem K. Bankas: A Novel and Efficient LZW-RNS Scheme for Enhanced Information Compression and Security, International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 11, November 2015