Jurnal Teknologi Full Paper USE OF NEW EFFICIENT LOSSLESS DATA COMPRESSION METHOD IN TRANSMITTING ENCRYPTED BAPTISTA SYMMETRIC CHAOTIC CRYPTOSYSTEM DATA

Article *in* Jurnal Teknologi · June 2016

CITATIONS
0

READS
93

10 authors, including:

Muhamad Azlan Daud
Universiti Malaysia Sabah (UMS)
13 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Kamel Ariffin Mohd Atan
Universiti Putra Malaysia
56 PUBLICATIONS **138** CITATIONS
SEE PROFILE

Che Haziqah Che Hussin
Universiti Malaysia Sabah (UMS)
22 PUBLICATIONS **55** CITATIONS
SEE PROFILE

Nurliyana Juhan
Universiti Malaysia Sabah (UMS)
12 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Jurnal Teknologi

Full Paper

USE OF NEW EFFICIENT LOSSLESS DATA COMPRESSION METHOD IN TRANSMITTING ENCRYPTED BAPTISTA SYMMETRIC CHAOTIC CRYPTOSYSTEM DATA

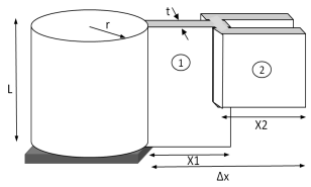
Muhamad Azlan Daud[a*], Muhammad Rezal Kamel Ariffin[b], S. Kularajasingam[c], Che Haziqah Che Hussin[a], Nurliyana Juhan[a], Mohd Mughti Hasni[d]

[a]Preparatory Center for Science and Technology, Universiti Malaysia Sabah, Jalan UMS, 88400 Kota Kinabalu, Sabah, Malaysia
[b]Department of Mathematics, Faculty of Sciences, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
[c]Sunway College Johor Bahru, No.3, Jalan Austin Heights Utama, Taman Mount Austin, 81100 Johor Bahru, Johor, Malaysia
[d]Department of Fundamental and Applied Sciences, Faculty of Science and Information Technology, Universiti Teknologi PETRONAS, 36210 Bandar Seri Iskandar, Perak, Malaysia

Article history
Received
24 July 2015
Received in revised form
16 December 2015
Accepted
24 January 2016

*Corresponding author
azlan.daud@ums.edu.my

Graphical abstract



Abstract

A new compression algorithm used to ensure a modified Baptista symmetric cryptosystem which is based on a chaotic dynamical system to be applicable is proposed. The Baptista symmetric cryptosystem able to produce various ciphers responding to the same message input. This modified Baptista type cryptosystem suffers from message expansion that goes against the conventional methodology of a symmetric cryptosystem. A new lossless data compression algorithm based on theideas from the Huffman coding for data transmission is proposed.This new compression mechanism does not face the problem of mapping elements from a domain which is much larger than its range.Our new algorithm circumvent this problem via a pre-defined codeword list. The purposed algorithm has fast encoding and decoding mechanism and proven analytically to be a lossless data compression technique.

Keywords: Lossless, lossy, baptista cryptosystem, Huffman, coding, encoding

1.0 INTRODUCTION

The Baptista type cryptosystem suffers from message expansion that goes against the conventional methodology of a symmetric cryptosystem [3]. However its polyaphabetic cipher structure allures the continuance of research into enabling this application.

Data compression is a process of reducing the size of a file by doing some alteration to the structure. In real world applications, compression is very useful because it helps to reduce the consumption of expensive resources such as memory space, total time for data transfer over network and communication costs by using available bandwidth effectively. There are 2 types of compression: lossy and lossless.

The first category is lossy data compression techniques. Through this technique the decompression process of compressed data produces results with loss of some information. This

compression technique is called irreversible compression since it is not possible to reconstruct 100% the original message during the decompression process. As lossy cannot generate the original message perfectly, the difference between the original and after message decompressing, cannot be tolerated.

This paper will solely focus on one "lossless data compression technique". This technique compresses data without effectively losing detail. Therefore data can be perfectly reconstructed. Thus, the information after being decompressed does not change from its original structure before compression. It is also known as reversible compression since the original data is reconstructed by decompression process. An example is the ZIP file mechanism. Since the original data becomes smaller, it is easy to be transmitted through today's public bandwidth.

Prior to transmission we propose a novel lossless data compression method on the ciphertext. This strategy has facilitated a possible practical deployment of the Baptista cryptosystem.

2.0 THE ENCRYPTION AND COMPRESSION ALGORITHM

The modified Baptista cryptosystem [7] become more secure against attacks similar to the one-time pad attack that occurs in year 2003[1]. The strong characteristics from the original Baptista cryptosystem were sustained. In this subsection, we will go through Baptista cryptosystem via matrix secret key based on IFS [7].

IFS consisting of the maps,

$$w_i(x,y) = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}, i = 1,2,\dots,N \quad (1)$$

for $i = 1$. That is,

$$w_1 = \begin{pmatrix} x_{i+1} \\ y_{i+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}\begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} (2),$$

and let the matrix

$$A = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$$

consist of only elements within set $\{0,1\}$.

Next, the 2×1 matrix

$$B = \begin{pmatrix} x_i \\ y_i \end{pmatrix}$$

will consist of Baptistaciphertext values, and the matrix

$$C = \begin{pmatrix} e \\ f \end{pmatrix}$$

will be equal to zero (i.e. $C = 0$).

2.1 Encryption Algorithm

Preparing a chaotic map.

i. Assume that we construct a look-up table consisting of j ε-intervals.
ii. Represent each site with $S_1, S_2, S_3, \dots, S_j$.
iii. The minimum value of the first interval is 0, and the upper bound of the interval is 1.
iv. Choose a one-dimensional chaotic map. The logistic map;

$$x_{i+1} = bx_i(1 - x_i) \text{for } b = 4.$$

Preparing the matrix secret key.

i. Generate a $k \times k$ matrix ($[A]_{k\times k}$) such that its inverse ($[A]^{-1}_{k\times k}$) exists.

$$A = \begin{pmatrix} M_{11} & M_{12} & \dots & \dots & M_{1k} \\ M_{21} & \dots & \dots & \dots & M_{2k} \\ \vdots & \dots & & \dots & \vdots \\ \vdots & \dots & \dots & \dots & \vdots \\ M_{k1} & M_{k2} & \dots & \dots & M_{kk} \end{pmatrix},$$

The matrix will consist elements only from the set $\{0,1\}$. This matrix will be the secret key.

Preparing distorted plaintext

i. Encrypt each plaintext via Baptista method.
ii. The iteration numbers are denoted as C_1.
iii. Group each element of C_1 into matrix of dimension $k \times 1$. Then do the following matrix multiplication:

$$[C_2]_{k\times 1} = [A]_{k\times k} \times [C_1]_{k\times 1} (4),$$

iv. $[C_2]_{k\times 1}$is the ciphertext to be transmitted to the recipient.

2.2 Compression Algorithm

The algorithm will continue with observing the following codeword. The right column (binary code) is the code word for its counterpart residing within the same row in the left column (number).

Table 1 Binary codes to represent the integers

Number	Binary Code
1	1
2	10
3	100
4	1000
5	10000
6	100000
⋮	⋮
$n-1$	$1[(n-1)-1]0's$
n	$1[(n-1)]0's$

Prior to the encoding process, to ensure correct decoding the size of the original data, *n* should be known to both the encoder and decoder. We denote $\|b\|$ to be the length of the corresponding data string $b = \{0,1\}^n$for $[C_2]_{k\times 1}$. For $j = 1,2,3,\dots$ we define the j-th data string as $b_j = \left(2^{\|b_{j-1}\|} - 1\right) - b_{j-1}$. Given a data string input b_0, we will do the following;

i. Convert b_0 to its decimal value.
ii. Compute, $b_1 = \left(2^{\|b_0\|} - 1\right) - b_0$
iii. Code the difference between the length of $\|b_0\|$ and $\|b_1\|$ as w_1 (refer Table 1)
iv. Continue the loop $b_j = \left(2^{\|b_{j-1}\|} - 1\right) - b_{j-1}$ for $j = 1,2,3,\dots,k$until $0 \le b_j \le 3$ (observe that $\|b_j\| = 2$). In each loop a codeword w_j will be produced based on the difference between the length of $\|b_{j-1}\|$and $\|b_j\|$. Observe that the values of $b_j$are strictly decreasing, and as soon as it reaches$0 \le b_j \le 3$ the algorithm will terminate.
v. From the codeword list $\{w_1, w_2, \dots, w_{k-1}, w_k\}$ we will append b_k at the end of the codeword to gain $[C_2]_{k\times 1} = \{w_1, w_2, \dots, w_{k-1}, w_k, b_k\}$. Once again observe that $\|[C_2]_{k\times 1}\| = n$. Then, focus on the last codeword $w_k b_k$ will be shifted to the left according to the number of zeros in w_k. The result is compressed data denoted by $[C_2]_{k\times 1_c}$.
vi. The encoder will then send the compressed data $[C_2]_{k\times 1_c}$. Notice that the zero's within w_k is excluded in the corresponding sequence which constructs $[C_2]_{k\times 1_c}$. Hence, $\|[C_2]_{k\times 1_c}\| \le \|[C_2]_{k\times 1}\|$.

3.0 UNIQUENESS OF THE DECOMPRESSION PROCESSAND DECRYPTION ALGORITHM

3.1 Uniqueness of The Decompression process

Proposition 1 (Decompression Algorithm)

The following decoding process of an encoded information by section2.2 is unique.

1- Expand $[C_2]_{k\times 1_c}$ to the original size $\|[C_2]_{k\times 1}\|$ by shifting back b_k to the right by padding in zero's until we have $\|[C_2]_{k\times 1_c}\| = \|[C_2]_{k\times 1}\|$. To decode we have to decide where each code begins and ends, since they do not have the same length. During the encoding process we utilized the codeword list as given by Table 1. As a result, we only need to scan through the input string of m_c from right to left until we recognize the first codeword. Then, we are able to determine the corresponding value and start looking for the next codeword. Observe that from Table 1 all cases will begin with 0 from the right and stop with 1 on the left.
2- Excluding b_k, start by extracting the codeword from the LSB of m_c. Translate the codeword from Table 1.
3- Compute, $b_{k-1} = \left(2^{\|p_0\|} - 1\right) - b_k$where $\|p_0\| = \|w_k\| + \|b_k\|$.
4- Next, compute, $b_{k-2} = \left(2^{\|p_1\|} - 1\right) - b_{k-1}$where $\|p_1\| = \|w_{k-1}\| + \|p_0\|$.
5- Continue until $b_{k-i} = \left(2^{\|p_{i-1}\|} - 1\right) - b_{k-i+1}$, where $i = 1,2,3,\dots,k$. The original data is b_0.

Proof

Let b_{k-i+1} be parameter that is used to input into the decoding procedure prior to the procedure giving output b_{k-i} (i.e. $b_{k-i} = \left(2^{\|p_{i-1}\|} - 1\right) - b_{k-i+1}$). The compression algorithm consists of a sequence of subtractions. Assume that the decoding process is not unique, then for a pair $(b_{k-i}, \|p_{i-1}\|)$, we have the following relations;

$$b_{k-i+1} = \left(2^{\|p_{i-1}\|} - 1\right) - b_{k-i}$$

and

$$b_{k-i+1} = \left(2^{\|p_{i-1}\|} - 1\right) - b'_{k-i}$$

where $b_{k-i} \neq b'_{k-i}$.

Following through we will have:

$$\left(2^{\|p_{i-1}\|} - 1\right) - b'_{k-i} = \left(2^{\|p_{i-1}\|} - 1\right) - b_{k-i}$$

This would imply that $b_{k-i} - b'_{k-i} = 0$. Thus, $b_{k-i} = b'_{k-i}$. This is a contradiction. Hence, assumption is false and the decoding process provides a unique output.■

3.2 Decryption Algorithm

Multiply $[A]_{k\times k}^{-1}$ with the following ciphertexts($[C_2]_{k\times 1}$).

i. Do the following matrix multiplication:
$$[C_1]_{k\times 1} = [A]_{k\times k}^{-1} \times [C_2]_{k\times 1}(5),$$
ii. This would result in a list of integer.
iii. Use each integer to iterate the logistic map. Start iterating the logistic maps until it falls in the corresponding phase space of the first character and continue iterating until the final character to get the original plaintext.

4.0 RESULTS AND DISCUSSIONS

Example 1

Let us use a 26-alphabets source, $S = \{a, b, \dots, z\}$. For illustrative purposes the key X_0=0.232323 and parameter $b = 4$. The text message is given by $P = attackatdawn$. Table 2 represents the Phase Space for $S = \{a, b, \dots, z\}$ while Table 3 shows the Ciphertext appears after Baptista Cryptosystem (a).

Table 2 Phase Space for $S = \{a, b, \ldots, z\}$.

Site	Associated interval (phase space)
a	$[0, 0.038462)$
b	$(0.038462, 0.076923)$
c	$(0.076923, 0.115385)$
d	$(0.115385, 0.153846)$
e	$(0.153846, 0.192308)$
f	$(0.192308, 0.230769)$
g	$(0.230769, 0.269231)$
h	$(0.269231, 0.307692)$
i	$(0.307692, 0.346154)$
j	$(0.346154, 0.384615)$
k	$(0.384615, 0.423077)$
l	$(0.423077, 0.461538)$
m	$(0.461538, 0.5)$
n	$(0.5, 0.538462)$
o	$(0.538462, 0.576923)$
p	$(0.576923, 0.615385)$
q	$(0.615385, 0.653846)$
r	$(0.653846, 0.692308)$
s	$(0.692308, 0.730769)$
t	$(0.730769, 0.769231)$
u	$(0.769231, 0.807692)$
v	$(0.807692, 0.846154)$
w	$(0.846154, 0.884615)$
x	$(0.884615, 0.923077)$
y	$(0.923077, 0.961538)$
z	$(0.961538, 1]$

1. Encryption.
   i. Choose $k = 2$.
   ii. Preparing matrix key, let $A = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$,
   iii. Each character P, was encrypted via Baptista cryptosystem.
   iv. The following Plaintexts:

Table 3 Ciphertext appears after Baptista Cryptosystem (a)

Plaintext, P	Ciphertext, C_1
a	8
t	63
t	25
a	19
c	1
k	1
a	55
t	4
d	33
a	3
w	4
n	134

   v. Next, group each integer of C_1 into matrix of dimension $k \times 1$. Then do the following matrix multiplication:
$$[C_2]_{k\times 1} = [A]_{k\times k} \times [C_1]_{k\times 1} (6),$$
   vi. From the following Cihertexts, C_1. Do the matrix multiplication procedure.
   vii.
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 8 \\ 63 \end{pmatrix} = \begin{pmatrix} 71 \\ 63 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 25 \\ 19 \end{pmatrix} = \begin{pmatrix} 44 \\ 19 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 55 \\ 4 \end{pmatrix} = \begin{pmatrix} 59 \\ 4 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 33 \\ 3 \end{pmatrix} = \begin{pmatrix} 36 \\ 3 \end{pmatrix},$$
$$\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} 4 \\ 134 \end{pmatrix} = \begin{pmatrix} 138 \\ 134 \end{pmatrix}.$$
   viii. The following Ciphertexts, C_2: 71, 63, 44, 19, 2, 1, 59, 4, 36, 3, 138, 134.
   ix. Apply the compression algorithm from 2.2. Consider that data transmission with the ability to transfer 1 bit data per second and transmit the data by text in each transmission. The Table 4 below shows that the data size after transmission.

Table 4 Comparison Data Bit Size after Transmission (a)

Ciphertext	Ciphertext Original Size	Ciphertext Compressed Size
71	7	7
63	6	3
44	6	5
19	5	4
2	2	2
1	1	1
59	6	6
4	3	3
36	6	4
3	2	2
4	3	3
134	8	8

Example 2

Let us use a 26-alphabets source, $S = \{a, b, \ldots, z\}$, refer Table 5. For illustrative purposes, we assume the key X_0=0.383838 and parameter $b = 4$. The text message is given by $P = attackatdawn$.

2. Encryption.
   i. Choose k = 2.
   ii. Preparing matrix key, let A = $\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}$,
   iii. Each character P, was encrypted via Baptista cryptosystem.
   iv. The following Plaintexts:

Table 5 Ciphertext appears after Baptista Cryptosystem (b)

Plaintext, P	Ciphertext, C_1
a	30
t	127
t	1
a	15
c	12
k	89
a	16
t	172
d	44
a	3
w	45
n	9

   v. Next, group each integer of C_1 into matrix of dimension $k \times 1$. Then do the following matrix multiplication:

$$[C_2]_{k\times 1} = [A]_{k\times k} \times [C_1]_{k\times 1} (7),$$

   vi. From the following CiphertextsC_1. Do the matrix multiplication procedure.
   vii.

$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}30\\ 127\end{pmatrix} = \begin{pmatrix}157\\ 127\end{pmatrix},$$
$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}1\\ 15\end{pmatrix} = \begin{pmatrix}16\\ 15\end{pmatrix},$$
$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}12\\ 89\end{pmatrix} = \begin{pmatrix}101\\ 89\end{pmatrix},$$
$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}16\\ 172\end{pmatrix} = \begin{pmatrix}188\\ 172\end{pmatrix},$$
$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}44\\ 3\end{pmatrix} = \begin{pmatrix}47\\ 3\end{pmatrix},$$
$$\begin{pmatrix}1 & 1\\ 0 & 1\end{pmatrix}\begin{pmatrix}45\\ 9\end{pmatrix} = \begin{pmatrix}54\\ 9\end{pmatrix}.$$

   viii. The following Ciphertexts,

$$C_2: 157, 127, 16, 15, 101, 89, 188, 172, 47, 3, 54, 9.$$

   ix. Apply the compression algorithm from section 2.2. Consider that data transmission with the ability to transfer 1 bit data per second and transmit the data by text in each transmission. Table 6 below shows the data size after transmission.

Table 6 Comparison Data Bit Size after Transmission (b)

Ciphertext	Ciphertext Original Size	Ciphertext Compressed Size
157	7	7
127	6	3
16	6	5
15	5	4
101	2	2
89	1	1
188	6	6
172	3	3
47	6	4
3	2	2
54	3	3
9	8	8

5.0 COMPRESSION RATIO AND TRANSMISSION SPEED

Compression ratio is defined as,

$$\text{Compression ratio, } CR = \frac{\text{Uncompressed size} - \text{Compressed Size}}{\text{Uncompressed size}}$$

Remark 1

From section 2.2, it implies that if $C_R \to 1$ (i.e. Compressed size $\to 0$), the algorithm has an excellent compression rate.

For data transmission speed, we consider that data transmission with the ability to transfer 1 bit data per second. See the following table (Table 6 and 7).

6.0 CONCLUSION

In this paper we have applied our new proposed compression algorithm on the Baptista cryptosystem. The result proves that our compression algorithm works on Baptista cryptosystem.

Table 7 Conclusion of Experiment 1

Data length before compression	50 bits
Compression Ratio	0.120
Speed (without compression)	50
Speed (with compression)	44

Table 8 Conclusion of Experiment 2

Data length before compression	72 bits
Compression Ratio	0.181
Speed (without compression)	72
Speed (with compression)	59

From Tables 7 and 8 the speed of the compression are different. The compression ratio for experimental

example 1 is 0.120 and in example 2 is 0.181. This is because we changed the initial value (i.e. seed) in example 2 were changed. Also take note that experimental example 2 suffers from the ciphertext (data length before compression) expansion, but the new compression ratio is better than before. That is, to achieve better speed by making a small change in a Baptista cryptosystem in order to achieve better compression. Hence, from the above table (Table 8), it is clear that in some cases, by making a small change in the Baptista cryptosystem, the data will actually produce better compression ratio and speed.

Through this work, the scheme can easily be visualized on current transmission technology and would be efficient for live data streaming. This newly developed algorithm has facilitated practical deployment of the Baptista cryptosystem. Future research can continue to be conducted, mainly to facilitate another algorithm to make it more efficient in terms of compression ratio and speed for the deployment of the Baptista cryptosystem.

Acknowledgement

We would like to thank University Malaysia Sabah for supporting our participation and scholarship.

References

[1] Alvarez, Montoya, G., Romera, F. M. and Pastor. 2000. *Cryptanalysis of a chaotic encryption system from Phys. Lett. A*. 276: 191-196.
[2] Ferreira, A. J., Oliveira, A. L. and Figueiredo, M. A. T. 2009. On the Suitability of Suffix Arrays for Lempel-Ziv Data Compression. *DCC*. 2009: 444.
[3] Baptista, M. S. 1998. *Cryptography with Chaos from Phys. Lett. A*. 240: 50-54.
[4] Daud, M. A. and Ariffin M. R. K. 2013. A New Efficient Analytically Proven Lossless Data Compression for Data Transmission Technique. *Malaysian Journal of Mathematical Sciences*. 7(S): 117-129.
[5] Burrows, M. and Wheeler, D. J. 1994. A Block-sorting Lossless Data Compression Algorithm. *SRC Research Report*. 124: 1-18.
[6] Ward, M. D. 2005. Exploring Data Compression via Binary Trees. 143-150.
[7] Ariffin, M. R. K., Al-Saidi, N. M. G., Said, M. R. M., Mahad, Z. and Daud, M. A. 2012. A New Direction in Utilization of Chaotic Fractal Functions for Cryptosystems. Book Chapter in *Applications of Chaos and Nonlinear Dynamics in Science and Engineering – Vol 2. Understanding Complex System*. Berlin Heidelberg: Springer-Verlag. 233-248.
[8] Ahmadi, O. and Menezes. 2005. Irreducible Polynomials of Maximum Weight. *CACR Technical Reports*.
[9] Hellebrand, S. and Wurtenberger, A. 2002. Alternating Run-Length Coding- A Technique for Improved Test Data Compression. *Handout IEEE International Workshop on Test Resource Partitioning*. USA: Baltimore, MD.
[10] Pathak, S., Singh S., Singh S., Jain M. and Sharma A. 2011. Data Compression Scheme of Dynamic Huffman Code for Different Languages. *2011 International Conference i=on Information and Network Technology*. 4: 201-205.
[11] Roman S. 1997. *Introduction to Coding and Information Theory*. Springer.