A New Symmetric Cryptosystem Based on of Permutation Matrices

M. Shafriezal, Md Kasim, A. Sofiyyullah Razalai, Ghassan Khaleel, Sherzod Turaev
Department of Computer Science, International Islamic University Malaysia, Gombak, Malaysia

Abstract— A new symmetric cryptosystem based on permutation matrices is proposed. In this paper, the encryption and decryption algorithms were built upon random multiple selection of the elements in the permutation matrix. This technique promotes the confusion, diffusion, and increases the complexity of proposed algorithms. The security and performance of the proposed cryptosystem is evaluated.

Keywords— Permutation Matrices, Key, Ciphertext, Dömösi's cryptosystem, Modified Dömösi's cryptosystem.

I. Introduction

Cryptography is a method of storing and transmitting information in a particular form so that only the intended recipient able to read and process it. The cryptosystem is composed of two different modes: encryption and decryption. In the encryption mode the message is converted to the unreadable message (ciphertext) using an encryption key and to decrypt the ciphertext, a decryption key is used. For encryption and decryption, various cryptographic algorithms are used that can be broadly classified into two main types: symmetric and asymmetric cryptosystems. In the symmetric cryptosystems, sender and recipient share a common key for encryption and decryption. While, in the asymmetric cryptosystems, where two different keys are used: a public key for encryption and private key for decryption. The symmetric cryptosystems can be distinguished into two types by the type of input data: block ciphers and stream ciphers. In the stream ciphers, the encryption is done one bit or byte at a time, can be run very quickly and usually uses low hardware complexity. Whereas, the block ciphers encrypt fixed – length groups of information for instance 64-bit, 128-bit blocks or more. This system depends on many rounds of permutations and substitutions (confusions and diffusions) in order to generate the ciphertext.

In this study, the researcher focused on designing a new symmetric stream cipher based on permutation matrix. The process of this new cryptosystem is different from the most of the implementation of matrix based cryptosystems. The production of the ciphertext does not consider the multiplication, exclusive or operation (XOR). Instead, the encryption and decryption process depends on the randomly multiple selections of the elements in the permutation matrix.

II. RELATED WORKS

Many cryptosystems have been designed based on automata theory. The first cryptosystem based on cellular automata was introduced by [1]. This system consists of a simple one dimensional cellular automaton, each having value 0 or 1. These values are updated in discrete time based on a rule. Many systems depend on this idea, for instance, linear feedback shift register (LFSR). On the other hand, the security of the system relies on the difficulty of finding the seed of the key. Many other cryptosystems based on cellular automata all share the common drawbacks of technical realization, for instance [2,3,4].

In 1985, Tao and Chen [5] proposed a public key cryptosystem based on Mealy machine (FAPKC). The idea of this system is as follows: the private key consists of two finite automata, which are constructed so that their inverses are easily computed. By applying the inverses of automata to the ciphertext, the plaintext can be recovered. For encryption, the public key consists of another automaton which is the combination of the two finite automata. After 10 years, in 1995, Bao and Igarashi [6] found some security weakness in the systems: FAPKC (0, 1 and 2) against chosen plaintext attacks. So, to prevent attacks, a refinement of these systems, called FAPKC3, was developed [7]. But this modification was also broken by Meskanen [8]. Finally, Tao Renji, Chen Shihua, and Chen Xuemei introduced FAPKC4 [9]. Other cryptosystems based on Mealy machines can be found [10,11]. In 2008, Pal Dömösi [12] introduced cryptosystem based on finite automata without output. This cryptosystem similar to the Mealy machine based cryptosystems in that encrypting and decrypting is performed using a key automaton. Unlike Mealy machine in generating the ciphertext. This system has many advantages over some cryptosystems based on finite automata. Firstly, cannot attack the system with the method used for defeating cryptosystems based on Mealy machines, for instance, FAPKC. Secondly, the relation between the random number generator and the key is independent. On the other hand, Dömösi's cryptosystem has a big drawback in the encryption algorithm, which affects entire performance of the system.

In order to overcome the drawbacks and improve the performance of Dömösi's cryptosystem to a better linear time without backtracking, Khaleel et.al [13] proposed an additional control system used together with the Dömösi's encryption algorithm. This control system prevents backtracking in the encryption algorithm by generating two vectors according to the current state, input signals and final states. The control system consists of the initialization stage and the operation stage. In the initialization stage, the control system generates all the control vectors V_1 and V_2, where V_1 consists of all input signals that take the automaton from the current state to any non-final state, whereas V_2 consists of all input signals that take the automaton from any state to one of the target final states. In the operation mode. First, the algorithm constructs a prefix of ciphertext of length $t-1$ by randomly selecting signals from vectors V_1, and second, it selects a random signal from V_2 finalizing the construction of ciphertext. Since the modification overcomes the backtracking, the ciphertext is constructed in linear time proportional to the maximum length of the ciphertext blocks. The other recent results on finite automata based cryptosystems can be found in [14-16].

III. Proposed Work

In this research, motivated from the Dömösi's cryptosystem, we propose a new symmetric cryptosystem based on a permutation matrix. The main idea of this system depends on a huge size permutation matrix of ciphertext characters, the size of this matrix is $2^{10} \times 2^{10}$, such that each raw contains a permutation of the 2^{10} bits string, and each character consists of 10 bits. Basically, permutation matrix consists of four partitions as shown in the Fig.1.

In the permutation matrix, the ciphertext is calculated from partitions 1 and 3, while from partition 2, we can recover the original plaintext. In this partition, we assign a number of columns to each character. For instance, if the set of plaintext contain on 128 characters, we can assign four columns for each character.

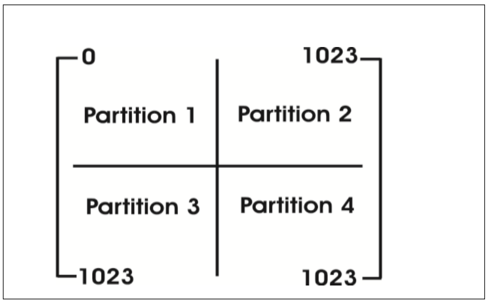

Fig.1 Permutation Matrix.

IV. Encryption and decryption

The encryption process is simple and the time complexity is linear and depends on the length of the ciphertext blocks. The message can be encrypted through a round of operations. A number of rounds depends on the length of the ciphertext blocks. The encryption algorithm reads the secret key k (row number), where $k \in$ [0, number of columns in the partition 1]. For each character in the plaintext set, the algorithm randomly generates the length of the ciphertext r, the random number generator generates column number t, where $t \in$ [0, number of columns in the partition 1] and the first ciphertext will be the element in the kth row and the tth column of the partition 1. Thus, the second ciphertext will be the element in the tth row, and the current column number will generate from the random number generator. This process will be repeated $r-1$ times to generate a prefix of the ciphertext. Finally, in the r step, the current row r_1 is the previous column number, while the algorithm randomly selects any column $c_1 \in [511,1023]$ from partition 2 corresponding to plaintext character, and the ciphertext is the element in r_1th row and c_1th column of the partition 2.

Algorithm 1. Encryption algorithm

```
Procedure Encryption
Input: $b_0 b_1 \cdots b_n \in \Pi^+$
Input : Permutation matrix PM
Input : $k$ (secret key)
Output: $c_0 c_1 \cdots c_k \in \Sigma^+$
$w \leftarrow \lambda$
$i \leftarrow 1, j \leftarrow 1$
 while $i \leq n$ do
        read $b_i$
        Select a random $r$
         while $j \leq r-1$ do
            Select a random $t \in [0,511]$
            $c_j \leftarrow PM[k,t]$
            $w \leftarrow w \cdot c_j$
            $k \leftarrow t$
            $j \leftarrow j+1$
        end while
        Select a random $t$ with column that
              assciated with $b_i$ in the partition 3
        $c_j \leftarrow PM[k,t]$
        $w \leftarrow w \cdot c_j$
        $i \leftarrow i+1$
    end while
return $w$
```

The decryption algorithm also is simple and the time complexity is linear and depends on the length of the ciphertext blocks and the type of search algorithm. The procedure reads the secret key (row number). After reading the first ciphertext character, it uses a search function to find the column(s) corresponding to the ciphertext character and row number. If the column(s) is in the partition 2, then we can retrieve the plaintext character that associated with this column(s). Else, the new row number will be the current column, and repeat same operations. The encryption and decryption algorithms are illustrated in the algorithms 1 and 2 respectively.

Algorithm 2. Decryption algorithm

```
$Procedure\ Decryption$
$Input: c_0c_1 \cdots c_n \in \Sigma^+$
$Input:\ Permutation\ matrix\ PM$
$Input:\ k\ (secret\ key)$
$Output: b_0b_1 \cdots b_k \in \Pi^+$
$P \leftarrow \lambda$
$i \leftarrow 1$
 $while\ i \leq n\ \ do$
        $read\ c_i$
        $t = Search(c_i, k)$
        $if\,(t \in columns\ that\ associated\ with\ b_i)\ \{$
            $P = P \cdot b_i\}$
             $k \leftarrow t$
         $i \leftarrow i + 1$
        $end\ while$
$return\ P$
```

V. Performance Analyses

The practical test of encrypting and decrypting algorithms was held in Lenovo Notebook E430 having Intel(R) Core(TM) i5-3230M CPU 2.6 GHz with 4 GB RAM under 64-bit Operating System Windows 10. The simulation program we used was written in visual C++. The results of performance tests of encryption and decryption algorithms can be seen in Table I.

TABLE II
Results of speed test.

Plaintext (MB)	Encrypting time (Sec.)	Decrypting time (Sec.)
1	0.031	0.4
2	0.063	0.9
4	0.11	1.9
8	0.2	4.1
16	0.39	7.8
32	0.84	13.2
64	1.54	25.6

Table III shows high performance of the encryption algorithm; the encrypting time is more than 40 megabytes per second. While the decryption algorithm has low performance (comparing with the performance of the encryption algorithm), the decrypting time reaches to 2 megabytes per second. In this practical test, we consider the length of the ciphertext block is five characters long.

VI. Confusion and Diffusion Tests

In order to test the security of the proposed cryptosystem, we examine the confusion and diffusion of the system by calculating the avalanche rate. We chose random plaintext blocks and let the length of the ciphertext blocks is 5. To test the confusion, encrypted the plaintext blocks by a key, then we changed one bit in the key, encrypted again. We calculated the number of different bytes in the ciphertext blocks. Similarly, to test the diffusion, we encrypted the plaintext blocks, then we changed one bit in the plaintext blocks, encrypted again, then we computed the number of different bytes in the ciphertext blocks. Tables IV and IV show the avalanche rate of proposed cryptosystem under the effect of changing one bit in the keys and plaintext blocks respectively.

TABLE VI
Confusion test

Key	Avalanche rate
01100100	0.4985
01100101	
01100101	0.4949
01100110	
01100110	0.4952
01100111	

TABLE IVII
Diffusion test

Plaintext	Avalanche rate
01100011	0.52
01100010	
00101000	0.54
00101001	
00101010	0.52
00111010	

Tables VIII and IIX show that avalanche rate reaches to about 0.5. Hence, the proposed cryptosystem has reasonable avalanche effect.

VII. Randomness Tests

This section describes the randomness tests of the proposed cryptosystem. To perform these tests, an ENT 2008 pseudorandom number sequence test program [17] is used. Table V shows the statistical factors of the stream bytes of ciphertext, where the size of the output ciphertext file is 1MB. The minimal and maximal length of the ciphertext block are between 5 and 10 characters long. Moreover, Fig 2. Simulates the character distribution of the ciphertext.

TABLE V
RANDOMNESS TEST

Statistical factors	Output results
Optimum compression	0%
Entropy (bits/byte)	7.999827
Chi-square distribution	244.82
Arithmetic mean value	127.4075
Monte Carlo value (Error)	0.19%
Serial correlation	0.000226

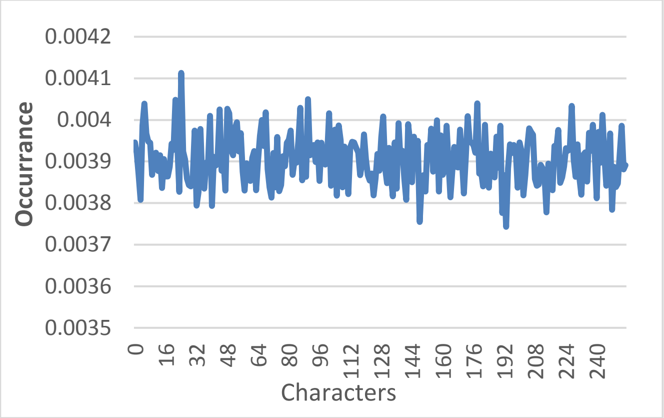

Fig. 2 Character distribution of stream bytes

VIII. Conclusion

We proposed a new symmetric cryptosystem based on permutation matrix. The practical tests illustrate that the system has good avalanche rate. Further, the randomness tests show that the sequence of stream bytes of the generated ciphertext is essentially random. In the performance part, the system has an appropriate performance of encryption algorithm. In the future work, security analysis should be necessary. Moreover, there is a need to improve the performance of decryption algorithm.

Acknowledgements

This research has been supported by FRGS grant **FRGS13-066-0307** Ministry of Education, Malaysia through International Islamic University Malaysia.

References

[1] S. Wolfram, "Cryptography with cellular automata", In: Williams, H.C. (ed.) Advances in Cryptology CRYPTO'85 Proceedings. LNCS, vol. 218, pp. 429–432. Springer (1986).

[2] P. Guan, "Cellular automaton public-key cryptosystem", Complex Systems 1, 51–56 (1987).

[3] J. Kari, "Cryptosystems based on reversible cellular automata". Personal communication (1992).

[4] H. Gutowitz, "Cryptography with dynamical systems", In: Goles, E., Boccara, N. (eds.) Automata and Cooperative Phenomena. pp. 237–273. Kluwer (1993).

[5] R. Tao, R., S. Chen, "Two varieties of finite automaton public key cryptosystem and digital signatures", J. of Compt. Sci. and Tech. 1, pp. 9–18 (1986).

[6] F. Bao, Y. Igarashi, "Break finite automata public key cryptosystem", In: International Congress of Mathematicians. pp. 147–158 (1995).

[7] R. Tao, S. Chen, "FAPKC3: a new finite automaton public key cryptosystem", Journal of Computer Science and Technology 12(4), pp. 289–305 (1997).

[8] T. Meskanen, "On finite automaton public key cryptosystems", TUGS Technical Report 408, Turku Centre for Computer Science, Turku (2001).

[9] R. Tao, S. Chen, "The generalization of public key cryptosystem FAPKC4", Chinese Science Bulletin 44(9), pp. 784–790 (1999).

[10] M. Gysin, "A one –key cryptosystem based on finite nonlinear automata", Cryptography. Policy and algorithms. LNCS 1029, Springer Verlag, pp. 165-173 (1996).

[11] B. Gandhi, A. Sekhar, S. Srilakshmi, "Cryptographic Scheme for Digital Signals using Finite State Machines". In: International journal of computer applications. (0975-8887). Vol. 29. (2011).

[12] P. Dömösi, "A novel stream cipher based on finite automata", In: IntelliSec – The 1st International Workshop on Intelligent Security Systems. Bucharest, Romania (November 11–14, 2009).

[13] Khaleel, G., Turaev, S., Mohd, I., and F. Imad, "A Performance Improvement of Dömösi's Cryptosystem", AIP Conference Proceedings 1705, 020007, 2016.

[14] Khaleel, G., Turaev, S., M.I. Mohd Tamrin, F. Imad, "Performance and Security Improvements of Dömösi's Cryptosystem", International Journal of Applied Mathematics and Statistics 55(2), pp. 32– 45 (2016).

[15] Khaleel G., Turaev, S., Zhukabayeva T, "A novel stream cipher based on nondeterministic finite automata", In: Information Technologies in Science, Management, Social Sphere and Medicine (ITSMSSM 2016), Tomsk, Russia, Atlantis Press, pp. 110-191 (May 23-26, 2016).

[16] Khaleel G., Turaev S., Mohd, I., and F. Imad. "A New Block Cipher Based on Finite Automata Systems", International Journal on Perceptive and Cognitive Computing 2(1), pp. 23–26, (2016).

[17] (ENT 2008)A pseudorandom number sequence test program http://www.fourmilab.ch/random