New Cryptosystem Based-on Permutation Matrix

Rooya Karimnia, Ghassan Khaleel, Sherzod Turaev

roya.k440@gmail.com, ghassankhaleel@yahoo.co.nz, sherzod@iium.edu.my

Department of Computer Science, International Islamic University Malaysia, Gombak, Malaysia

Abstract— Cryptography is defined as a technique of transmitting information in a secretive manner so that only authorized people are able to read and process it. The main aim of cryptography is to make sure the message has been received by the receiver in a secure manner, and not being understood by anyone else but the receiver. Although there exist several different cryptosystems, the choice of the best algorithm is the main concern of the researches. In this paper, a new cryptography system based on matrix permutation has been introduced. A permutation matrix is an $\boldsymbol{n}\times\boldsymbol{n}$ matrix which is obtained by permuting its rows and columns according to some permutations of the numbers $\mathbf{0}$ to $\boldsymbol{n}$. In this paper, encryption and decryption methodologies have been proposed based on permutation matrices. The algorithms are based on the random selection of the permutation matrix - known as key matrix - entries. The main objective of this research is to evaluate the security and the performance of the proposed cryptosystem as well as to promote the confusion and diffusion.

Keywords— Put your keywords here, keywords are separated by comma.

I. INTRODUCTION

Cryptography is defined as a technique of changing the message in a secretive manner so that only authorized people can read and process it. In other words, it is the study of hiding the messages from unauthorized people. Therefore, it is crucial to secure information from any internal or external attacks. The objective of cryptography is to ensure confidentiality, integrity, non-repudiation and authenticity of the transmitted information [1]. Each cryptosystem is composed of encryption and decryption. Encryption is the process of converting the message (plaintext) into a meaningless and unreadable text (ciphertext) using encryption key, while decryption is the reverse process of encryption - meaning the process of gaining back the plaintext from ciphertext.

Moreover, there are two major branches of cryptography, namely symmetric-key (private-key) and asymmetric-key (public-key) cryptography. Symmetric-key cryptography refers to an encryption system which sender and receiver use a shared common key to perform encryption and decryption. In asymmetric cryptography, two different keys are used: public key for encryption and private key for decryption [2,3]. Private-key cryptosystem divides into two categories: block and stream cipher. Block cipher algorithms produce n-bit size ciphertext from an n-bit size plaintext. Whereas, stream cipher algorithms mainly consist of two components: a function that performs permutation and key stream generator that generates key permutations. The function is usually the exclusive or (XOR) operation. On the other hand, two main features of stream cipher encryption are the fact that the length of the ciphertext does not depend on the length of the plaintext as well as it can run very quickly. Using stream cipher encryption, also, does not require excessive hardware complexity [4]. The taxonomy of cryptographic techniques is shown in Fig. 1.

The main focus of this work is to design a new symmetric stream cipher based on permutation matrix. The strength point of the proposed cryptosystem is in the mixing function. The ciphertext generation does not rely on the XOR operation, whereas the encryption and decryption process depend on the randomly multiple selections of the entries in the permutation matrix.

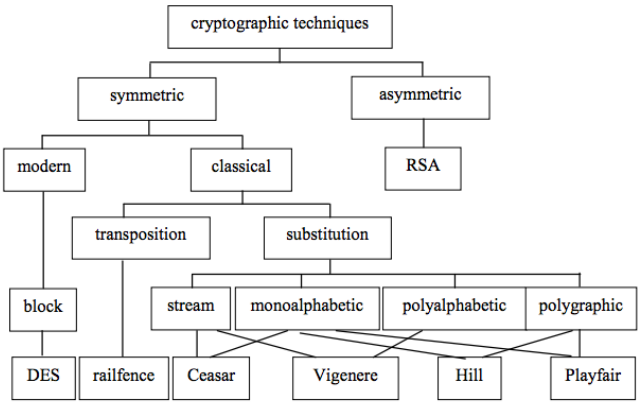

Fig. 1 Taxonomy of cryptographic techniques

II. RELATED WORKS

There have been many cryptosystems based on linear algebra. The first cryptosystem is Hill Cipher introduced by Lester S. Hill in 1929 [5]. In this cryptosystem, each letter has been represented by a number. The encryption processes the multiplication of each block of n size plaintext, into an invertible $n\times n$ key matrix. The result $n\times 1$

module 26 will be the ciphertext of the selected plaintext. The decryption process is the multiplication of the cipher matrix into the inverse of the key matrix. Nowadays, Hill Cipher is not being used because it is vulnerable to known-plaintext attack, however, it is used as the base work of so many other new researches.

In 2002, S. Saeednia introduced a new way to make Hill Cipher more secure [6]. He suggested to secure the cipher by adding the feature of random permutation of columns and rows of the key matrix, however, it is proved that its vulnerable to known-plaintext as well. The next attempt to increase the security of the Hill Cipher has been done by Lin Chu-Hsing [7]. They proposed two encryption parameters which one is random and the other one is generated using a one-way hash function. Another modification of the Hill Cipher was introduced by [8]. In this research, the author proposed the use of an initial vector that multiplies successively by some orders of the key matrix to produce the corresponding key of each block. However, [9] proved that this algorithm is having the same problem and it is not secure.

New variants of stream and block ciphers based on sequential and parallel finite automata systems, which are closely related to permutation matrix-based cryptosystems, have been thoroughly studied in [11-17]. Not surprisingly, these systems use transition matrices of finite automata as secrete keys in encryption and decryption algorithms.

Recently, a new symmetric cryptosystem based on permutation matrix has been introduced by [18]. They proposed the encryption as a random selection of entries in key matrix to generate the ciphertext, and a sequential search in the matrix to retrieve the plaintext. The performance speed of the introduced algorithm is fast however, there is no parallelism and the decrypting time is slow.

In this research, the proposed algorithm is similar to [18], however, by using parallelism and an improvement on the decryption algorithm, the decrypting time is less, and the performance is better and faster.

III. PARALLELISM

Parallel processing is introduced as processing a program by dividing its instructions among several processors to reduce the running time. At early ages of computers, only one program ran at a time, therefore, the processing time was a lot. There are different types of parallel computing, namely, bit-level, instruction, data and task. As nowadays the main concern is power usage of the computers, therefore, the parallelism became a dominant in computer architecture, mainly in the form of multi-core processors. Parallel computers can be classified in to two according to the hardware level supports parallelism. Multi-core and multi-processor computers are having multiple processing elements in a single machine, whereas, clusters, MPPs and grids use multiple computers to perform a specific task [10].

In this research, the focus is to parallelize the data processing functions. The parallelism will be implemented in such a way that first the plaintext will be divided into the number of cores or processing elements using the fork function, hence, all the plaintext blocks will be processed simultaneously. Then, using a join function, the result of each block will be combined and the final ciphertext will be generated. Later at the receiver side, the same process of forking and joining will be implemented on ciphertext to retrieve the plaintext. The purpose of using parallelism is to improve the performance speed significantly.

IV. PROPOSED WORK

The main idea of this cryptosystem is based on a permutation matrix. As it was discussed earlier, A permutation matrix is an $n \times n$ matrix which is obtained by permuting its rows and columns according to some permutations of the numbers 0 to n.

A. Key Matrix Generation

In this research, there are four proposed key matrices which each of them is having different permutations. They must be constructed in big size to eliminate the brute force attack. Considering permutation matrices with the size of $2^{10} \times 2^{10}$ such that each row consists of 2^{10} bits' string characters which each character consists of 10 bits.

In addition, some of the columns of each matrix, selected randomly, will be labelled with plaintext characters. Each character will be assigned to more than one columns. Considering the set of all possible plaintext characters include 128 characters, therefore, the number of columns assigned to each character can be four. A visualization of permutation matrix is demonstrated in Fig 2.

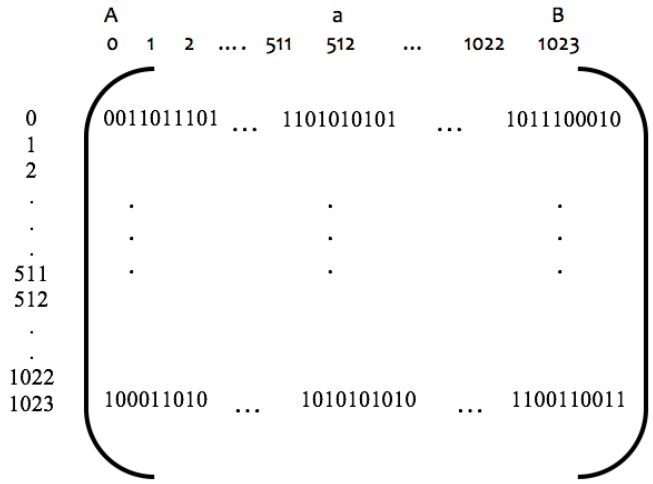

Fig. 2 Permutation matrix

B. Encryption Algorithm

The encryption process is simple and fast. It is remarkable that the main strength of this cryptosystem is the difference between the length of the plaintext and ciphertext. Meaning, for each plaintext's character (P_1), there exists t number of ciphertext characters ($c1\ c2\ \dots\ t$). The value t will be selected randomly random and it will vary for each plaintext character. Therefore, the length of the ciphertext is variable and unpredictable to any attacker.

The first step of the algorithm is to divide the plaintext into four blocks. Each block will be given a key matrix. The next step to select a row $r \in [0, 2^{10}]$ from the key permutation matrix using a random number generator. Then, another random number, $t \in [0, 50]$, will be generated which t specifies the number of ciphertext characters for each plaintext characters. Next step is to find the corresponding column for the row $r, m \in [0, number\ of\ unlabelled\ columns]$. As the row and column are found, the matrix entry corresponding to r and m, c_1, will be recorded as the first ciphertext character. Furthermore, the second ciphertext character will be selected from m^{th} row and another randomly selected column from unlabelled columns. This process will be repeated $t-1$ times to generate a prefix of the ciphertext. At last, on the t^{th} round, the current row r_t, is the previous column number and the column m_t will be randomly chosen from columns which are labelled as the plaintext character ($P1$). Thus, the last entry, $C_t\,[r_t, m_t]$ is the ciphertext representing $P1$. The encryption algorithm has been illustrated in algorithm 1.

ALGORITHM I
THE ENCRYPTION ALGORITHM

Encryption (P)
Input: P // plaintext, $P \in \Pi^+$
Input: permutation matrices $PM_0, PM_1, \dots, PM_{k-1}$
/* k permutation matrices */

$PM_i\,[0..n, 0..n], n \geq 2^{10}, i = \overline{0, k-1}$
Input: r (first row) // selected randomly
Output: C // ciphertext: $C = Encryption\ (P)$

$P = P_0 . P_1 . \dots . P_{k-1}$ /* divide p into k blocks, each block has a number of letters */

$P_i = P_{i0}, P_{i1}, \dots, P_{is_i}, i = \overline{0, k-1}$
$Permu(P) = P_{\Pi(0)}, P_{\Pi(1)}, \dots, P_{\Pi(k-1)}$ // Permute the blocks // of P

For all i in parallel do
  Let $C \leftarrow \lambda, w \leftarrow \lambda$
  $i \leftarrow 0, j \leftarrow 0, s \leftarrow 0$
  $k \leftarrow 0$
  while $i < k-1$ do // plaintext block
    while $j \leq |P_i|$ do // plaintext letters of each block
      read P_{ij} // read the plaintext letter
      select a random $t \in [0, 50]$
      while $s \leq t-1$ do // producing the prefix // for ciphertext
        select a random m from unlabelled columns
        $w \leftarrow PM_i\,[r, m]$
        $C_i \leftarrow C_i . w$ // concatenate the // ciphertext with
        $r \leftarrow m$
        $s \leftarrow s + 1$ // increment to the // next prefix letter
      end while
        // now we want to produce the ciphertext

      select a random m from columns labelled with P_{ij}
      $w \leftarrow PM_i\,[r, m]$
      $C_i \leftarrow C_i . w$ // concatenate the ciphertext with
      $j \leftarrow j + 1$ // increment j for next plaintext // letter
    end while
    $i \leftarrow i + 1$ // increment i for next plaintext block
    $C = C . C_i$ // to join all the ciphertext blocks
  end while

  return $\boldsymbol{C}$

C. Decryption Algorithm

The decryption process is also easy and fast. Unlike most cryptosystems which the decryption is the exact reverse of the encryption process, the proposed decryption method is a bit different. At the receiver side, new key matrices will be generated and they will be used to decrypt the messages. These new matrices will be an alteration of the main key matrices.

Considering PM is one of the permutation matrices. The PM' will be constructed in such a way that a new matrix will be introduced with the columns labelled as the sorted values of PM entries. Then, the entries of new matrix PM' will be set as the column number corresponding to the entry in PM. For example, $PM\,[0,3] = a$, therefore $PM'\,[0, a] = 3$. An example on PM' construction is shown in Fig. 3.

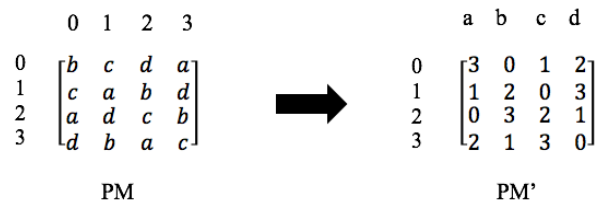

Fig. 3 New Matrix Construction

Now, to find the specific value in a row, only indexing with a linear time complexity is needed, and to find the next column, the procedure will refer to $PM'[row, column]$ to get the next row's value. If the column is one of the labelled columns, then the plaintext character will be

retrieved from the label of the column. The decryption algorithm has been illustrated in algorithm 2.

ALGORITHM III
THE DECRYPTION ALGORITHM

$\boldsymbol{Decryption(C)}$
Input: C //ciphertext
Input: permutation matrices $PM_0, PM_1, \dots, PM_{k-1}$
// k permutation matrices
$PM_i\ [0..n, 0..n], n \geq 2^{10}, i = \overline{0, k\text{-}1}$
Input: r (first row)
Output: P // plaintext: $P = Decryption\ (C)$

$Permu^{-1}(C) = C_{\Pi(0)}, C_{\Pi(1)}, \dots, C_{\Pi(k-1)}$ // to get the original
// order
Let $w \leftarrow \lambda, P \leftarrow \lambda$
$i \leftarrow 0, j \leftarrow 0, s \leftarrow 0, v \leftarrow 0, k \leftarrow 0,$

For all i in parallel do
  while $i < k - 1$ do // ciphertext block
    while $j \leq |C_i|$ do // ciphertext letters of each block
      read C_{ij} // read the ciphertext letter
      $s \leftarrow PM_i'[r, C_{ij}]$
      $v \leftarrow column\ number\ from\ s$
      while (v is not labelled)
        $j \leftarrow j + 1$ // move to the next ciphertext letter
        $s \leftarrow PM'[C_{ij}, v]$
      end while
      if (v is labelled)
        $P_i \leftarrow P_i.value\ of\ s$
      end if
    end while
    $i \leftarrow i + 1$
    $P \leftarrow P\ .Pi$
  end while
return $\boldsymbol{P}$

V. SECURITY

In this section, the avalanche effect of the proposed cryptosystem is estimated. Where, an avalanche rate is one of the most common techniques used to estimate the confusion and diffusion of the cryptosystems. Moreover, the property of avalanche effect is very important of the block ciphers. The cryptosystem has avalanche effect when one-bit change in the plaintext or in the key leads to a significant change in the corresponding ciphertext block, and also one-bit change in the ciphertext results a significant change in the corresponding plaintext block and vice versa.

Here, only the confusion of the proposed cryptosystem will be estimated. Therefore, a random plaintext is chosen and then it is being divided into 32 bits blocks. Then, each block has been encoded and then, one bit in the key (row value) for each matrix has been changed and then encoded again. Then, the number of different bits in the ciphertext blocks of each plaintext block is calculated where,

$$Avalanche\ rate = \frac{\text{number of flipped bits in ciphertext}}{\text{number of bits in ciphertext}}$$

Fig. 4 shows the avalanche rate of the proposed block cipher, where x-axis is the ciphertext blocks number. While y-axis is the avalanche rate corresponding to each ciphertext block. It also illustrates that the avalanche rate of most of the ciphertext blocks of the presented new block cipher is ranging from 0.4 to 0.6. Therefore, the proposed cryptosystem has enough avalanche effect, which makes the cryptosystem secure against the adaptive chosen plaintext and ciphertext attacks.

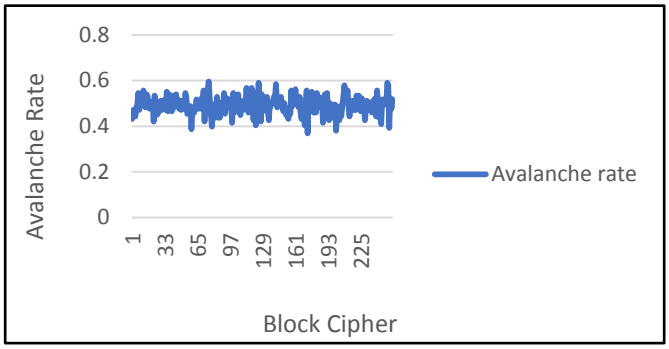

Fig. 4 Avalanche rate of the proposed block cipher

VI. PERFORMANCE ANALYSIS

The practical test of encrypting and decrypting algorithms 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 the tables I and II respectively. In this implementation, we use four distinct permutation matrices, where the plaintext is varying from 1 MB to 64 MB, with the length of the ciphertext blocks varies from 4 to 6 characters long.

TABLE III
PERFORMANCE TEST OF ENCRYPTION ALGORITHM

Plaintext(MB)	Encrypting time(Sec.)
1	0.007
2	0.013
4	0.024
8	0.046
16	0.09
32	0.17
64	0.33

From above tables, one can easily identify the high performance of the encryption and decryption algorithms based on parallel computations. The throughput of encryption and decryption algorithms reach to 186 MB/S.

TABLE IVI
PERFORMANCE TEST OF DECRYPTION ALGORITHM

Ciphertext(MB)	Decrypting time(Sec.)
1	0.007
2	0.013
4	0.023
8	0.045
16	0.0865
32	0.15
64	0.295

VII. COMPARISON

To compare this work with the previous research done by [11], the best way to compare is through the comparison between the performance and the security analysis. As a reference, character distribution and performance test of encryption and decryption algorithms of [11] have been copied here.

As all the tests and avalanche rates has been conducted and taken from the same computer device for both researches, hence, the comparison is fair enough. The device used is 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. From the results gained from [11] and this research, we can see that the newly introduced cryptosystem is having less encryption and decryption time due to the use of the parallelism. It also can be identified that the decryption time is significantly less because the use of sequential search has been eliminated and as a result, new key matrices have been constructed.

Moreover, from Fig. 5 and table III it can be concluded that this research is having more stable character distribution rate compare to the previous research. Therefore, it is concluded that the security that it provides is higher because it provides higher rate of randomness.

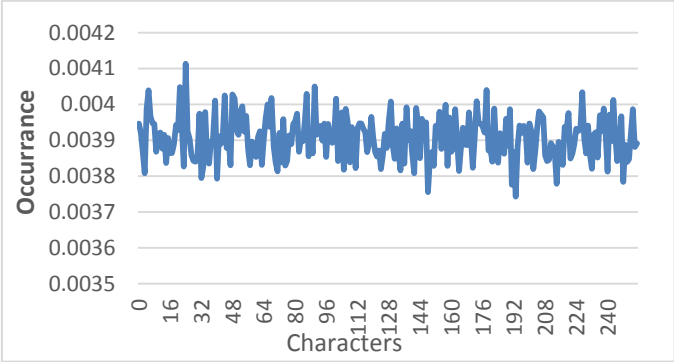

Fig. 5 Character distribution of previous research

TABLE VII
PERFORMANCE TEST OF PREVIOUS RESEARCH

Ciphertext(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

VIII. CONCLUSION

In the given work, the proposed cryptosystem is a symmetric cryptography, therefore, both sender and receiver sides use the same keys which are the introduced permuted matrices. To make sure of the security, after few times of encrypting, the permutation of the matrices can be changed and hence the new keys will be generated. The encryption process generates the ciphertext with an unpredictable length. The decryption process is to construct the new key matrices based on the main key matrices and then to decrypt easily with an indexing function.

Moreover, since the selection of the rows and columns are random, the cryptosystem provides an ultimately high security. In addition, due to the use of parallelism and to process of the blocks of data simultaneously, the encryption and decryption process are fast. In short, the result shows that the new proposed cryptosystem has a remarkable low time complexity, low memory requirement and high security.

IX. ACKNOWLEDGEMENTS

This work has been supported through International Islamic University Malaysia Research Initiative Grant Scheme **RIGS16-368-0532**.

REFERENCES

[1] S. Chandra, S. Paira, S. Alam, and G. Sanyal, "A comparative survey of symmetric and asymmetric key cryptography", 2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE), pp 1, IEEE.
[2] http://www.webopedia.com/TERM/S/symmetric_key_cryptography.html
[3] I. Sharma, C. P. Gupta, "Fully Homomorphic Encryption Scheme with Symmetric Keys", pp 2.
[4] S.O. Sharif, S.P. Mansoor, "Performance analysis of Stream and Block cipher algorithms", 2010 3rd International Conference on Advanced Computer Theory and Engineering(ICACTE), pp 1.
[5] https://en.wikipedia.org/wiki/Hill_cipher.
[6] S. Saeednia, "How to Make the Hill Cipher Secure," Cryptologia Journal, Vol.24, No.4, pp.353-360, Oct. 2000.

[7] C.H. Lin, C.Y. Lee, and C.Y. Lee, "Comments on Saeednia's improved scheme for the Hill cipher," Journal of the Chinese institute of engineers, Vol.27, No.5, pp.743-746, 2004.
[8] I.A. Ismail, M. Amin, and H. Diab, "How to repair the Hill cipher," Journal of Zhejiang University-Science A, Vol.7, No.12, pp.2022-2030, Dec. 2006.
[9] D. Zhang, and G. Chen, Cryptanalysis of an image encryption scheme based on the Hill cipher, Journal of Zhejiang University - Science A 9(8) (2008) 1118- 1123.
[10] https://en.wikipedia.org/wiki/Parallel_computing.
[11] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, M.I.M. Tamrin, and T. Zhukabayeva, "A Novel Stream Cipher Based on Nondeterministic Finite Automata. Information Technologies in Science, Management, Social Sphere and Medicine, 23-26 May 2016, Tomsk, Russia, Atlantis Press, pp. 110-191, 2016.
[12] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, and M.I.M. Tamrin, "Performance and Security Improvements of Domosi's Cryptosystem". International Journal of Applied Mathematics and Statistics, 55(2), pp. 32-45, 2016.
[13] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, and M.I.M. Tamrin, "A New Block Cipher Based on Finite Automata Systems". International Journal on Perceptive and Cognitive Computing, 2(1), pp. 23-26, 2016.
[14] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, and M.I.M. Tamrin, "An Overview of Cryptosystems Based on Finite Automata. Journal of Advanced Review on Scientific Research 27(1), pp. 1-7, 2016.
[15] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, and M.I.M. Tamrin, "A Comparative Performance Analysis of Modified Domosi's Cryptosystem and Data Encryption Standard". International Journal on Perceptive and Cognitive Computing 3(1), pp. 11-14, 2017.
[16] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, M.I.M. Tamrin, and T. Zhukabayeva, "A Symmetric Cryptosystem Based on Nondeterministic Finite Automata". Journal of Theoretical and Applied Information Technology 95(10), pp. 1489-1498, 2017.
[17] G. Khaleel, S. Turaev, I.F. Al-Shaikhli, and M.I.M. Tamrin, "Symmetric Cryptosystems Based on Finite Automata". IIUM Press, 2017, ISBN: 978-967-418-678-4.
[18] M.S. Md Kasim, A.S. Razali, G. Khaleel, and S. Turaev, "A New Cryptosystem Based on Permutation Matrices". International Journal on Perceptive and Cognitive Computing Vol. 2, No. 2, pp. 36-40, 2016.