A New Block Cipher Based on Finite Automata Systems

Gh. Khaleel, S. Turaev, M.I.M. Tamrin and I.F. Al-Shaikhli

Kulliyyah of Information and Communication Technology, International Islamic University Malaysia
53100 Kuala Lumpur, Malaysia, ghassan.khaleel@live.iium.edu.my, {sherzod, izzuddin, imadf}@iium.edu.my

Abstract— The performance and security have central importance of cryptography field. Therefore, the need to use block ciphers are become very important. This paper presents a new block cipher based on finite automata system. The proposed cryptosystem is executed based on parallel computations to reduce the delay time. Moreover, to achieve high security, we use different machines (variant non-deterministic automata accepters) as keys for encryption and decryption.

Keywords— Dömösi's cryptosystem, parallel computing, nondeterministic finite automata, control vectors.

I. Introduction

With the rapid spread of digital communication networks, there is a great need for security of transmitted information. Hence, the methods of securing information are becoming a major issue. Therefore, many cryptosystems have been implemented to improve the security of information. Where, the cryptosystem is a method of storing and transmitting information in a particular form, so that only the intended recipient can read and process it. A cryptosystem is composed of two different modes: encryption and decryption. In the encryption mode the message is converted to unreadable message (ciphertext) using encryption key and to decrypt the ciphertext decryption key is used. For encryption and decryption, various cryptographic algorithms are used that can be broadly classified into two main parts: symmetric or private key algorithms and asymmetric or public key algorithms.

Cryptographic algorithms are usually implemented sequentially such that each instruction of the program is executed one after other, starting from the beginning towards the end on the single processor. These sequential operations may slow down the performance of encryption - decryption algorithms. In addition to this, also the resources of CPUs may effect on the performance of algorithms [1]. By using the parallel processing, the resources of CPUs can be reduced and high performance can be achieved in term of time. In this work, we exploited the idea of parallel computations in order to design a new block cipher based on finite automata systems to achieve high security as well high performance.

II. Related Works

Pál Dömösi [2-4] proposed a novel symmetric cryptosystem apparatus based on deterministic finite automata (Rabin-Scott automata). The author used the deterministic finite automaton as a key automaton for encryption and decryption. In this part, the proposed cryptosystem is similar to Mealy machine in which the encoding and decoding are performed using the same key automaton, but it is unlike Mealy machine in generating ciphertext. On the other hand, Dömösi's cryptosystem is similar to cellular automata based cryptosystems in that the key automaton is an automaton without outputs. This cryptosystem has many advantages over many others stream ciphers. Firstly, the random number generator is independent from the key. Secondly, this system cannot be attacked with methods used for defeating FAPKC cryptosystems [4-8]. Thirdly, Dömösi's cryptosystem overcomes some complicated mechanisms in broadcasting/datacasting systems. However, Dömösi's cryptosystem suffers from performance and security weaknesses: the encryption algorithms of Dömösi's cryptosystem are not efficient; the generation of ciphertext blocks involves frequent backtracking steps, resulting in a slower construction of the ciphertext. While, the resistance against crypto attacks depends on the construction of large size automata and relatively large minimal and maximal block lengths of ciphertexts, which results in producing much longer ciphertexts than given plaintexts. Moreover, the security of Dömösi's cryptosystem depends on the selection of special finite automata, which must be irreversible.

To overcome the drawbacks and improve the performance of Dömösi's cryptosystem to a better linear time without backtracking, G. Khaleel at al. [9] introduced a modified Dömösi's cryptosystem. The authors proposed an additional control system used together with the Dömösi's encryption algorithm. This control system prevents the backtracking search in the encryption algorithm by generating two vectors according to the current state, input signals and final states. The elements of the first vector consists of all input signals that take the automaton from the current state to any non-final state, whereas the elements of the second vector consists of all input signals that take the automaton from any state to one of the target final states.

The performance test showed that the performance has been improved significantly. While, the security analysis proved that the modified Dömösi's cryptosystem has at least same security level of Dömösi's cryptosystem.

Thereafter, Khaleel et.al [10] developed a novel stream cipher based on non-deterministic finite automata to reduce the dependency of the key automata on the size of ciphertext blocks as well as on irreversibility. The main model of this stream cipher depends on using a nondeterministic automaton accepter as a key for encryption and decryption. Moreover, this system also used an additional control vectors to prevent the backtracking search in the encryption algorithm and to enhance the performance. The security level of this cryptosystem depends on generating giant numbers of ciphertext corresponding to each plaintext character. Hence, neither statistical attacks nor analytical attacks can be attack this system. On the other hand, the performance of this system is very fast due to the time complexity of encryption and decryption algorithms are linear and depends on control vectors approach.

III. Proposed Work

In this paper, in order to improve the earlier work developed by Khaleel et.al [10], we construct a new block cipher based on finite automata system. The idea of this block cipher depends on using several different machines (automata) for encryption and decryption concurrently based on parallel computations wise. Thus, we propose two reasonable models for parallel encryption and decryption algorithms. The basic design of these models rely on concept of data parallelism. Let Π, Σ be a plaintext alphabet and a ciphertext alphabet, respectively. The procedure involves dividing a given plaintext into $|N|$ blocks $i_0, i_1, i_2, \cdots, i_{|N|}$, where $|N|$ is the number of cores in the CPU, and sending these blocks to the permutation box to assign each one of them to the corresponding core (as shown in Fig. 1) In the parallel encryption algorithm, each core P_k executes an encryption algorithm (NFAEA) on different plaintext part i_j independently to generates the corresponding output ciphertext block w_{ij}, $w_{ij} \in \Sigma$. The final ciphertext block will be concatenated of all sub ciphertext parts, i.e., $W = w_{i_0}.w_{i_1} \cdots w_{i_{|N|}}$. On the other side, to recover the plaintext, the ciphertext block is sliced into $|N|$ parts, such

that the length of each part $|w_{ij}| = t_j$, where $t_j \in [s_{min}, s_{max}]$, then sending these parts to decryption mode through permutation box. In the parallel decryption, each core P_k executes a decryption algorithm (NFADA) independently to recover the original plaintext. To achieve high security, we consider each core executes based on a different automaton, i.e., different transition matrix, different final states and different length of the ciphertext block. The parallel encryption and decryption algorithms are illustrated in the Algorithms 1 and 2.

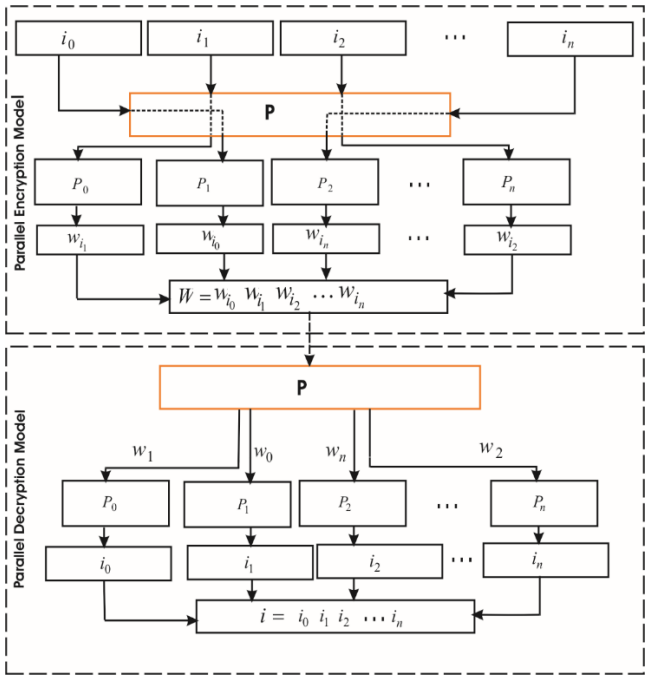

Fig. 1 Parallel encryption and decryption structure

Algorithm 1: Parallel of NFAEA

Procedure PARALLELENCRYPTION
Input: $i_0 i_1 \cdots i_k \in \Pi^+$
Output: $W_0 W_1 \cdots W_k \in \Sigma^*$
$r \leftarrow n$ // number of cores
$j \leftarrow 0$
$while(j < k)$
$Read\ i_0, i_1, \cdots i_r$
$For\ all\ cores\ P_r\ in\ the\ CPU\ do\ in\ parallel$
$if\ P_r = 0\ then\ w_{i_0} = NFAEA\ (i_0)$
$if\ P_r = 1\ then\ w_{i_1} = NFAEA\ (i_1)$
$\vdots$
$if\ P_r = r\ then\ w_{i_r} = NFAEA\ (i_r)$

$Wait\ for\ all\ operations\ to\ complete\ then$
$compute\ the\ ciphertext$
$W_j = w_{i_0} w_{i_1} \cdots w_{i_r}$
$End\ parallel$
$j \leftarrow j + r$
$return\ W_0 W_1 \cdots W_j;$

IV. Parallel Implementation

In implementation part, both the approaches, parallel approach or the proposed block cipher based on finite automata (parallel of NFAEA and NFADA algorithms) and sequential approach of NFAEA and NFADA algorithms have been implemented in quad core (Core i5) by using standard platform interface OpenMP, under 64-bit Operating System Windows 10. The simulation programs are compiled using Visual Studio C++ 2013. In this implementation, we consider four different nondeterministic key automaton, one for each

Algorithm 2: Parallel of NFADA

Procedure PARALLELDECRYPTION
Input: $W_0W_1 \cdots W_k \in \Sigma^*$
Output: $i_0 i_1 \cdots i_k \in \Pi^+$
$r \leftarrow n$ // number of cores
$j \leftarrow 0$
$while(j < k)$
$read\ W_j, W_{j+1}, \cdots W_{j+r}$ with $|W_j| = t_0$, $|W_{j+1}| = t_1, \cdots |W_{j+r}| = t_r$
$for\ all\ cores\ P_r\ in\ the\ CPU\ Do\ in\ Parallel$
$if\ P_r = 0\ then\ i_0 = NFADA\ (w_{i_0})$
$if\ P_r = 1\ then\ i_1 = NFADA\ (w_{i_1})$
$\vdots$
$if\ P_r = r\ then\ i_r = NFADA\ (w_{i_r})$
$Wait\ for\ all\ operations\ to\ complete\ then$
$compute\ the\ plaintext$
$i_j = i_0 i_1 \cdots i_r$
$End\ parallel$
$j \leftarrow j + r$
$return\ i_0 i_2 \cdots i_k;$

core, $\mathcal{M}_1 = (Q, \Sigma, \delta_1, q_0, F_1)$, $\mathcal{M}_2 = (A, \Sigma, \delta_2, a_0, F_2)$, $\mathcal{M}_3 = (S, \Sigma, \delta_3, s_0, F_3)$ and $\mathcal{M}_4 = (B, \Sigma, \delta_4, b_0, F_4)$, four different ciphetext lengths $t_1 = 10, t_2 = 9, t_3 = 10, t_4 = 9$. The plaintext alphabet and the set of final states of the key automata were the same consisting of 16 elements. We also applied 256 states and 256 signals for all machines, and for each signal take the automaton to non-final state in the key automaton, there are four additional signals ($n = 5$). The plaintext P sliced into four parts (based on quad cores) i_0, i_1, i_2, i_3, and size of each part depends on the number of final states $|F|$. For instance, if the plaintext string bits is 0111001111110000 and number of final states $|F| = 16$, then $i_0 = 0111$, $i_1 = 0011$, $i_2 = 1111$ and $i_3 = 0000$.

V. PERFORMANCE ANALYSIS

To estimate the performance of the software implementation of the two approaches, parallel (proposed block cipher) and sequential of NFAEA and NFADA, we conduct the comparison analysis in terms of executing time of the two versions. Fig. 2 shows the graphical simulation of sequential and parallel of NFAEA in term of encryption time for different plaintext sizes from 1 MB to 10 MB. It is obvious that the encryption time of parallel NFAEA based on four cores is approximately two times faster than the encryption time of sequential NFAEA.

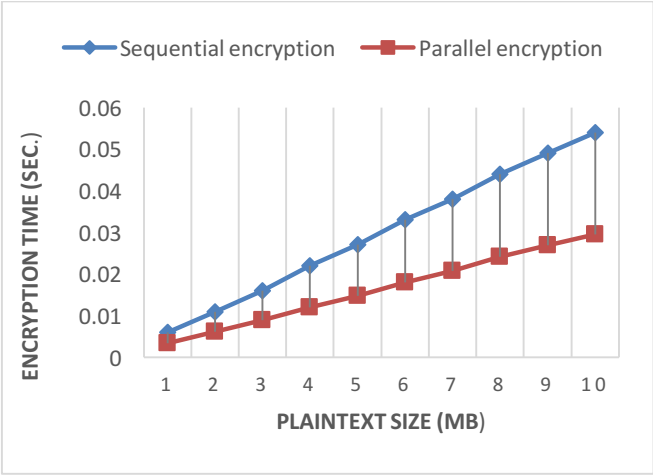

Fig.2 Execution time of parallel and sequential encryption algorithms

Similarly, Fig. 3 illustrates the decryption time of parallel and sequential NFADA for different ciphertext sizes from 1 MB to 10 MB on quad – cores processor. It not difficult to see the performance improvement of parallel NFADA.

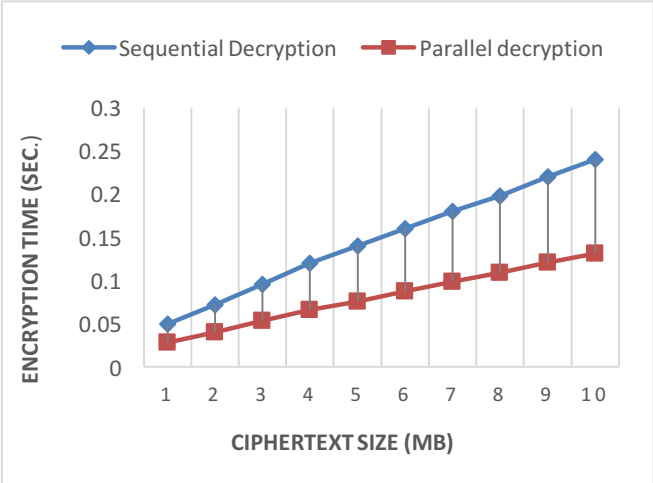

Fig.3 Execution time of parallel and sequential decryption algorithms

VI. STATISTICAL PROPERTIES

To design high quality block cipher, the randomness of the stream bytes of the output ciphertext should be high. Therefore, to test the randomness of proposed cryptosystem, we test the statistical properties of the output ciphertext by using ENT 2004 program. To perform these tests, we apply four machines as defined in section IV, we also use a random sample of plaintext, let the minimal length and maximal length of ciphertext block between 2 and 8. TABLE 1 shows that the entropy test of the output ciphertext of different lengths. Obviously, the ciphertext has high entropy reach to 0.7999. So, the information is essentially random. In addition, the arithmetic mean value of the proposed cryptosystem reaches to 127.5, thus the information is close to random. Chi-square distribution test shows that the byte sequences of the ciphertext are random. Moreover, the serial correlation coefficient is close to the zero, which means that the byte sequence of the ciphertext is uncorrelated.

TABLE I
ENTROPY TEST

Ciphertext Block length	*Entropy*	*Chi square*	*Mean value*	*Monte Carlo*	*Serial correlation*
2	7.996542	32.65%	127.48	3.1362	0.00024
3	7.996164	69.17%	127.14	3.1517	0.0026
4	7.997742	43.88%	127.52	3.1496	-0.0027
5	7.997937	60.21%	127.27	3.1453	0.0012
6	7.998134	57.28%	127.33	3.1444	0.0061
7	7.998233	89.37%	127.41	3.1463	-0.0001
8	7.998873	37.45%	127.71	3.1458	0.0042

Moreover, Fig.3 demonstrates the character distribution of the ciphertext.

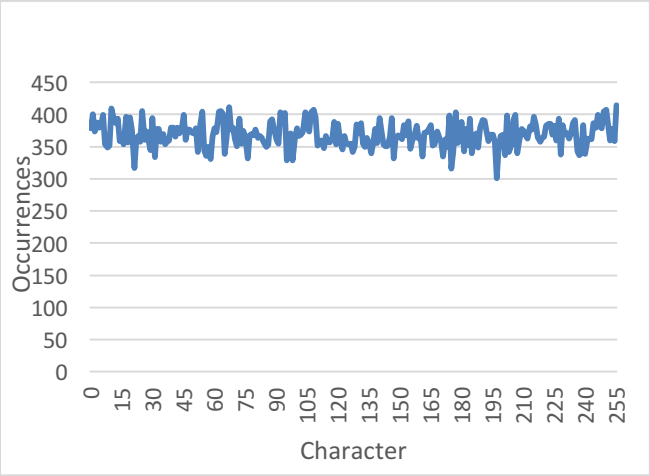

Fig.3 Character distribution

From the above Figure, it is not difficult to see that the probability of occurrences of the characters in the ciphertext are approximately between 0.003 to 0.004, where the size of the ciphertext is 94410 characters.

VII. CONCLUSIONS

In this paper, a new block cipher based on finite automata system was presented. By comparing experimental results, it shows that the performance of proposed cryptosystem (new block cipher version) has the priority over sequential version (stream cipher based on non – deterministic finite automata). Moreover, we evaluated the randomization of the proposed cipher system by calculating the entropy of the output ciphertext.

REFERENCES

[1] S. Saxena and B. Kapoor, "State of the art parallel approaches for RSA public key based cryptosystem", In: International Journal on Computational Sciences & Applications (IJCSA) Vol.5, No.1, February 2015.

[2] P. Dömösi, "A novel cryptosystem based on finite automata without outputs", In: M. Ito, Y. Koboyashi, and K. Shoji (eds.), Automata, Formal Languages and Algebraic Systems, World Scientific, p. 23-32, 2008.

[3] 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).

[4] P. Dömösi, P.: US. Pub. No. US 2009/0092251 A1.

[5] R. Tao, S. Chen, "A finite automaton public key cryptosystem and digital signature", Chinese Journal of Computers 8(6), pp. 401–409, 1985.

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

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

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

[9] G. Khaleel, S. Turaev, M.I. Mohd Tamrin and I.F. Al-Shaikhli, "A Performance Improvement of Dömösi's Cryptosystem", AIP Conference Proceedings 1705, 020007, 2016.

[10] G. Khaleel, S. Turaev, and T. Zhukabayeva, "A Novel Stream Cipher Based on Nondeterministic Finite Automata", Russia Conference 2016