A New Iterative Secret Key Cryptosystem Based on Reversible and Irreversible Cellular Automata

Said Bouchkaren, Saiida Lazaar
(Corresponding author: Said Bouchkaren)

Department of Mathematics and Computer Science/LTI, ENSA of TANGIER, AbdelMalek Essaadi University
P.O. Box 1818 Principal Tangier, Tangier, Morocco
(Email: saidbouchkaren1@hotmail.com)

(Received Jan. 27, 2015; revised and accepted May 5 & July 13, 2015)

Abstract

Many cryptosystems have been released to secure electronic data on internet. Some data are very critical to be transmitted as plaintext. Thus, to ensure the data confidentiality and integrity, a list of cryptosystems have been elaborated. The most important ones are divided into two categories: symmetric algorithms encrypting and decrypting data in blocks using a single secret key; and asymmetric algorithms using public keys to cipher texts and secret keys to reconstruct plaintexts. The of the present work is the design and implement a new secret key cryptosystem encrypting and decrypting data in blocks according to a number of iterations. Each plaintext block is encrypted using cellular automata and a list of sub keys deduced from a secret key through cellular automata. To demonstrate the feasibility, the proposed scheme is compared with AES algorithm, the well-known symmetric block cipher. We prove that our algorithm resists against statistical attacks and it is faster than AES-256 achieving good confusion and diffusion tests.

Keywords: Block ciphers, cellular automata, reversible, irreversible, secret key

1 Introduction

In the modern world everything is handled by smart devices which are in general connected to each other and communicate via a network. Each network is connected to other networks in order to simplify and to improve the relationship between distant communities. Most internet applications send and receive critical data such as logins and passwords, credit card number and PIN, bank account details, personal identity, etc. These data can be intercepted by malicious people and can be used for passive or active attacks.

In this context, a number of researches have been carried out in the field of cryptography, and leads to a number of methods to guaranty confidentiality and integrity of data, and to ensure authentication and non-repudiation. The researches focus on four components: Confidentiality: only authorized may access data. Integrity: to ensure that transmitted data were not altered. Authentication: to identify correctly the two parts of a communication. And the forth component is non-repudiation which validates the signature.

The new generation of cryptography methods are divided into three types: asymmetric cryptosystems in which the processes of encryption and decryption use a pair of keys: the public key used to perform the cipher text and the secret one used to reconstruct the plaintext, these systems are used in general to exchange secret keys and to sign documents; we remind that RSA is the most popular asymmetric algorithm in cryptography world [16]. The second type of modern cryptography concerns symmetric cryptosystems which uses single secret keys as for instance block algorithms which encrypt and decrypt data in blocks within a number of iterations or rounds; the well-known algorithms are AES, DES, 3DES, RC4 [3, 11, 16]. The third type is signature systems used to sign documents and to guaranty the integrity of data. The three systems types complete each other to achieve the four components of data security.

However, in cryptanalysis field, many attacks were carried out on these algorithms making them sometimes vulnerable. This vulnerability increases with technological advances and changing profiles pirates; these principal reasons motivate researchers to build robust and reliable cryptosystems.

To contribute to this research, we propose a new iterative symmetric cryptosystem based on reversible and irreversible cellular automata. First, the plaintext is divided into blocks, the principal secret key is given randomly by a first irreversible cellular automaton, and then a number of sub keys are generated and used for each iteration to cipher each block with a reversible cellular automata. the proposed cryptosystem is compared with AES algorithm, the well-known iterative symmetric block cipher, the computational results prove that it resists against statistical attacks and it is faster than AES-256.

The remainder of this paper is organized as follows: The second section presents some contributions in the domain, the third section describes the proposed system, and the forth part explains in details the sub keys scheduling process. To test the reliability of the present algorithm, various numerical tests are presented on the fifth section, and in the last section a conclusion and perspectives are given.

2 Brief Presentation of Cellular Automata and Related Works

Cellular automata (CA) are discrete dynamical systems widely used to simulate complex phenomena in several areas including physics, biology, chemistry, computer science and cryptography without defining analytical solutions of the problems. More precisely, on a given grid, CA are an evolution of a collection of cells on discrete time steps according to some rules based on the state of the neighbors of cells.

Mathematically, a cellular automaton A of dimension d is defined by $A = \{S, \mathbb{Z}^d, f, V\}$ where S is the set of states, $\mathbb{Z}^d$ is the space of the CA and $f : S^n \mapsto S$ is the transition rule; $n = card(V)$ where V is the set of neighborhoods. More details on CA supported by many illustrations can be found in [14, 21].

In cryptography field, CA allows ciphering texts and generating secret keys starting from a chaotic and complex state. The first algorithm based on CA belongs to S.Wolfram; the work presented in [20] gives interesting explanations about CA concept and since this first work, numerous contributions on the field were released.

In [6], a public-key cryptosystem is constructed with inhomogeneous cellular automata and according to the author the time to break the algorithm grows exponentially with the length of message blocks. Reversible cellular automata (RCA) was proposed in [8] with some efficiency due to parallelism property and this kind of CA was used to construct public and secret key cryptosystems. In [17], a novel secret key cryptosystem using RCA was developed.

To secure medical data sent over the internet, a block encryption method based on hybrid additive cellular automata was implemented in [1] where results demonstrate the power of CA encryption. In [13], an encryption method was built upon layered cellular automata, and used a number of layered grids applying a list of reversible transition rules [9, 10] to produce the cipher text. In [5], a generic strategy to design new block encryption methods based on CA is presented with an evolutionary computation mechanism to create new, fast and secure cryptosystems using non-uniform second-order CA. In [2] a description of a new and fast private key cryptosystem using two-dimensional reversible CA based on Margolus neighborhoods is presented, this algorithm can be used to encrypt any kind of data as for instance image data. also the paper [19] present a novel lightweight block cipher algorithm based on cellular automata.

This non-exhaustive overview on CA is closed by other works related to image encryption [4, 7, 12, 22].

3 Proposed Algorithm

In this proposed algorithm, which we call Cellular Automata Encryption System (CAES), two reversible CA are used to encrypt and decrypt plaintext and one irreversible CA to generate sub keys starting with the secret key. The concept of reversibility is well explained in [9, 10].

3.1 Algorithm Specifications

The proposed algorithm encrypts and decrypts data in blocks according to a number of iterations (rounds) introducing for each round the corresponding sub key; each block cipher and each sub key are generated by cellular automata. The specifications are the following: Data are divided into blocks of 256-bits, the size of the principal key is equal to 256-bits and the round number to encrypt or to decrypt each block corresponds to 12.

3.2 Encryption and Decryption Processes

Encryption process starts by dividing the plaintext into blocks of 256-bits and by copying the data into a matrix M of size 4x8 (4x8 bytes=256-bits) then M passes through a number of transformations named $Shift()$, $IMix()$, $PMix()$ and $Addkey()$, the pseudo-code for the encryption is shown in Algorithm 1.

Algorithm 1 Encryption algorithm

```
procedure ENCRYPT($M, Key$)   ▷ $M$ is the plaintext message block and $Key$ is the encryption key
    $SKeys[12] \leftarrow SubKeys(K)$;   ▷ Generating 12 sub keys
    for $i$ from 0 to 11 do
        $M = Shift(M)$
        $M = IMix(M)$
        $M = PMix(M)$
        $M = AddKey(M, SKeys[i])$
    end for
    return $M$   ▷ $M$ contains the encrypted message
end procedure
```

For the decryption process, the inverse transformations: $invShift()$, $invIMix()$, $invPMix()$ and $AddKey()$ are applied. The decryption process can be written in Algorithm 2.

Algorithm 2 Decryption algorithm

```
procedure DECRYPT($Mc, Key$)    ▷ Mc is the encrypted message block and Key is the encryption key
    $SKeys[12] \leftarrow SubKeys(K)$;    ▷ Generating 12 sub keys
    for $i$ from 11 downto 0 do
        $Mc = AddKey(Mc, SKeys[i])$
        $Mc = invPMix(Mc)$
        $Mc = invIMix(Mc)$
        $Mc = invShift(Mc)$
    end for
    return $Mc$    ▷ $Mc$ contains the plaintext
end procedure
```

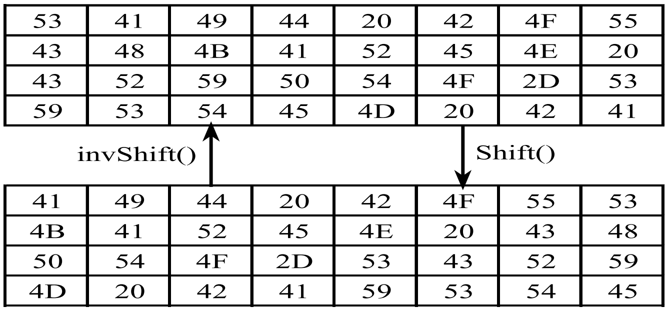
In the following, we describe the transformations used in encryption and decryption algorithms: $ENCRYPT()$ and $DECRYPT()$.

3.3 $Shift()$ and $invShift()$ Transformations

The $Shift()$ transformation acts on the bytes of data for each row, it is implemented using reversible cellular automaton defined as:

- States are the bytes of the row L.
- Transition rule is: each byte $B[i]$ becomes $B[(i + L)\%8]$.

The $invShift()$ is the reverse transformation of $Shift()$. The cellular automaton used in $Shift()$ (respectively) in $invShift()$ is a byte left (respectively) right rotation of a row L by 8 bits. Figure 1 demonstrates these transformations.



53	41	49	44	20	42	4F	55
43	48	4B	41	52	45	4E	20
43	52	59	50	54	4F	2D	53
59	53	54	45	4D	20	42	41

41	49	44	20	42	4F	55	53
4B	41	52	45	4E	20	43	48
50	54	4F	2D	53	43	52	59
4D	20	42	41	59	53	54	45

Figure 1: $Shift()/invShift()$ illustration

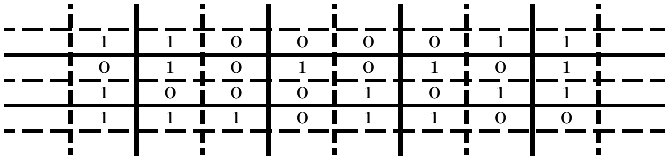
3.4 $IMix()$, $PMix()$, $invIMix()$ and $invPMix()$ Transformations

These transformations act on the entire block of 256-bits. They use a reversible cellular automaton of two dimensions which is built using MARGOLUS neighborhoods [18] and defined as follow:

- Convert the entire data block of 256-bits to binary, and fit this bits into a matrix $Mb[4][64]$.
- Partition Mb to blocks B of 4-bits (2x2).
- Look up $Y = f(X)$ where f is the transition rule with $X = B_{00}B_{01}B_{11}B_{10}$.
- Put Y into the block B.
- The transition rule f is: $\{15, 2, 3, 5, 7, 11, 13, 4, 6, 8, 10, 12, 14, 9, 1, 0\}$ for $PMix()$ and $\{0, 1, 9, 14, 12, 10, 8, 6, 4, 13, 11, 7, 5, 3, 2, 15\}$ for $IMix()$.
- Use periodic conditions on the edges of the matrix Mb.

For further details on this process we can refer to [2].

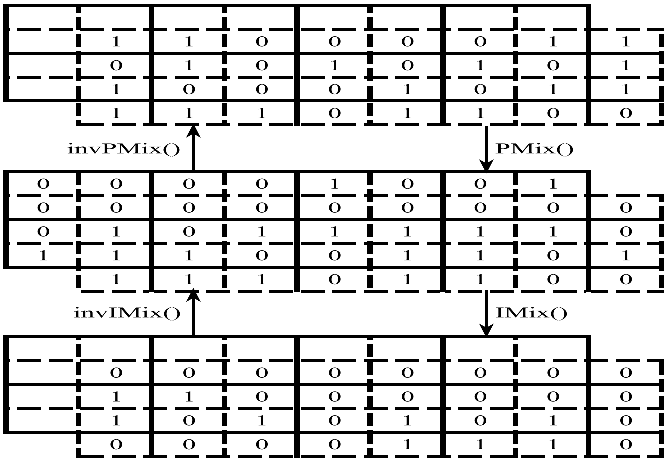
Figure 2 illustrates the effects of $PMix()$ and $IMix()$.



1	1	0	0	0	0	1	1
0	1	0	1	0	1	0	1
1	0	0	0	1	0	1	1
1	1	1	0	1	1	0	0

Figure 2: Acts of $PMix()$ and $IMix()$

The $PMix()$ transformation is applied on the dashed line blocks (Figure 2), however the $IMix()$ is applied on the solid line blocks. For example the first dashed line block is 1110 in binary which is equal to 14 in decimal representation and using the transition rule of $PMix()$, we get the value 1 in decimal or 0001 in binary representation so we replace 1110 with 0001. After applying $PMix()$ and $IMix()$ on the data in (Figure 2), we get the data represented in Figure 3.



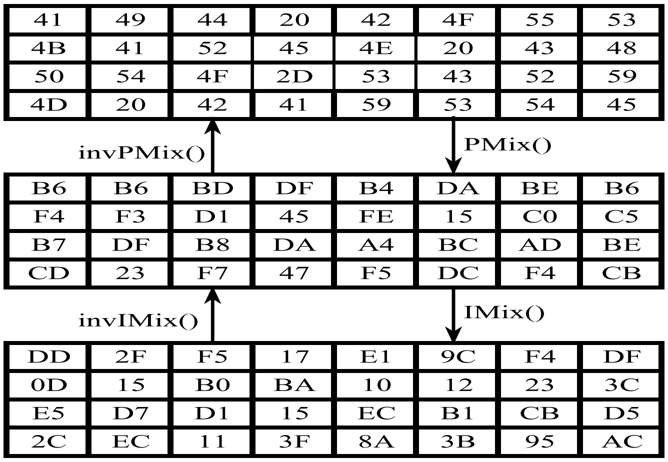
	1	1	0	0	0	0	1	1
	0	1	0	1	0	1	0	1
	1	0	0	0	1	0	1	1
	1	1	1	0	1	1	0	0

0	0	0	0	1	0	0	1	
0	0	0	0	0	0	0	0	0
0	1	0	1	1	1	1	1	0
1	1	1	0	0	1	1	0	1
	1	1	1	0	1	1	0	0

	0	0	0	0	0	0	0	0
	1	1	0	0	0	0	0	0
	1	0	1	0	1	0	1	0
	0	0	0	0	1	1	1	0

Figure 3: $PMix()$ and $IMix()$ acts illustration

The $invPMix()$ is the reciprocal transformation of $PMix()$ and it uses the same cellular automaton used in $PMix()$ but it uses the transition rule expressed as $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$. $invIMix()$ is the reciprocal transformation of $IMix()$. it follows the same logic as $invPMix()$ it uses the cellular automaton used in $IMix()$ except it uses the transition rules defined as $\{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\}$. instead of $\{0,1,9,14,12,10,8,6,4,13,11,7,5,3,2,15\}$. Figure 4 illustrates the act of $IMix()$, $PMix()$ and their inverses



41	49	44	20	42	4F	55	53
4B	41	52	45	4E	20	43	48
50	54	4F	2D	53	43	52	59
4D	20	42	41	59	53	54	45

B6	B6	BD	DF	B4	DA	BE	B6
F4	F3	D1	45	FE	15	C0	C5
B7	DF	B8	DA	A4	BC	AD	BE
CD	23	F7	47	F5	DC	F4	CB

DD	2F	F5	17	E1	9C	F4	DF
0D	15	B0	BA	10	12	23	3C
E5	D7	D1	15	EC	B1	CB	D5
2C	EC	11	3F	8A	3B	95	AC

Figure 4: Example of $PMix()$ and $IMix()$ and their inverses

3.5 AddKey Transformation

This transformation takes two parameters, 32 bytes (256-bits) data block M and a sub key K_i of 32 bytes; then it calculates $M' = M \oplus K_i$ where $\oplus$ represents the XOR operator. This transformation can be written in Algorithm 3.

Algorithm 3 AddKey() procedure

```
procedure AddKey($M, K_i$)
    for $i$ from 0 to 31 do
        $M[i] = M[i] \oplus K_i$   ▷ $M[i]$ is the $i^{th}$ byte of $M$
    end for
    return $M$
end procedure
```

4 Key Scheduling

As mentioned above, the $AddKey()$ transformation needs sub keys to be applied on the given data. These sub keys are generated from the encryption key using a function called $SubKeys()$. This function aims to generate sub keys to be used by the transformation $AddKey()$. These sub keys are derived recursively from the global encryption key K as follow:

$$\begin{cases} K_i &= next(K_{i-1}) / i \in \{1,2,\ldots,11\} \\ K_0 &= K. \end{cases}$$

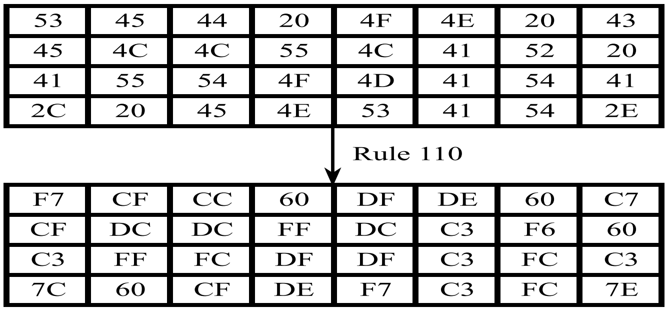
The function $next()$ is executed in three steps.

4.1 Step 1: Irreversible Cellular Automaton

In this step, we apply an irreversible cellular automaton of one dimension defined as:

- Neighbors of a cell i are $i-1$, i, $i+1$;
- A state for a given cell is 0 or 1;
- Periodic conditions on edges;
- Transition rule is 110.

The rule 110 has been chosen because it is classified as fourth class of cellular automata [15] and it produce a chaotic behavior. Figure 5 gives an example for this step.



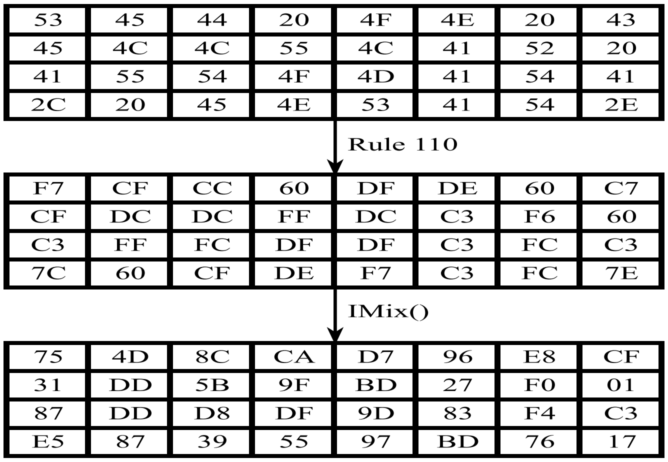
53	45	44	20	4F	4E	20	43
45	4C	4C	55	4C	41	52	20
41	55	54	4F	4D	41	54	41
2C	20	45	4E	53	41	54	2E

F7	CF	CC	60	DF	DE	60	C7
CF	DC	DC	FF	DC	C3	F6	60
C3	FF	FC	DF	DF	C3	FC	C3
7C	60	CF	DE	F7	C3	FC	7E

Figure 5: Rule 110 effect example

4.2 Step 2: Applying $IMix()$

In this step the resulting data from the step 1 is taken and then the $IMix()$ transformation described above is applied. For illustration purpose we get the results shown in Figure 6.



53	45	44	20	4F	4E	20	43
45	4C	4C	55	4C	41	52	20
41	55	54	4F	4D	41	54	41
2C	20	45	4E	53	41	54	2E

F7	CF	CC	60	DF	DE	60	C7
CF	DC	DC	FF	DC	C3	F6	60
C3	FF	FC	DF	DF	C3	FC	C3
7C	60	CF	DE	F7	C3	FC	7E

75	4D	8C	CA	D7	96	E8	CF
31	DD	5B	9F	BD	27	F0	01
87	DD	D8	DF	9D	83	F4	C3
E5	87	39	55	97	BD	76	17

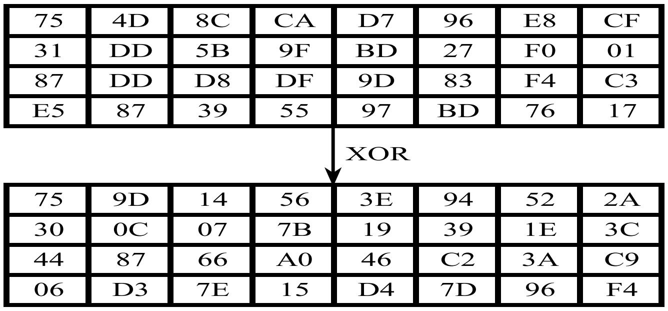
Figure 6: Rule 110 and IMix() illustration

4.3 Step 3: Bytes XORing

Let $E2$ be the data from Step 2.

$$E2[i] = E2[i-1] \oplus \sim E2[32-i] \quad 1 \le i \le 31$$

where $E2[i]$ denotes the i^{th} byte of $E2$ and $\sim$ denotes the binary negation. The Figure 7 an example presentation.



75	4D	8C	CA	D7	96	E8	CF
31	DD	5B	9F	BD	27	F0	01
87	DD	D8	DF	9D	83	F4	C3
E5	87	39	55	97	BD	76	17

75	9D	14	56	3E	94	52	2A
30	0C	07	7B	19	39	1E	3C
44	87	66	A0	46	C2	3A	C9
06	D3	7E	15	D4	7D	96	F4

Figure 7: Data XORing example

5 Numerical Tests

To prove the algorithm reliability, three tests are proposed: a test of bits change rates, and confusion and diffusion tests. All these tests are compared to AES-256.

5.1 Bits Change Rate

This test measures the bits changing rate between the clear message and the encrypted message. To carry out this test a random key of 256 bits is chosen, and then a list of plaintext message, which have variable length, is randomly generated, finally each message is encrypted using the same key. Figure 8 shows the results of this test using the proposed algorithm CAES and the AES-256 algorithm. We observe that the proposed algorithm gives almost the same rates compared to AES with marginal improvement.

5.2 Confusion Test

In this test, we measure the confusion property which make the relationship between the encryption key and the cryptogram as complex as possible, it measures the number of changed bits in an encrypted message by changing some bits in the encryption key. To achieve this test, a plaintext message and an encryption key are randomly selected, the message is kept unchanged while some key bits are flipped, and then the encryption algorithms (CAES and AES-256) are performed. Figure 9 illustrates the results of the test.

It is clear that the proposed algorithm is better than AES-256, the changing rate is between 48.44 and 52.15 for CAES and between 48.44 and 51.96 for AES-256.

5.3 Diffusion Test

In cryptography, the diffusion property makes the relationship between plaintext and encrypted message. It evaluates the impact of changing some bits in a plaintext message on the resulting cipher text while keeping the encryption key unchanged. To accomplish this test,a plaintext message and a key are randomly chosen. Figure 10 shows that the proposed system improves the diffusion property compared to AES-256.

5.4 Performance Test

This test evaluates the proposed algorithm performance regarding the CPU time consumption. It is performed as follows: a random key of size 256 bits is given and a list of messages of different sizes are generated. For each message the encryption process is ran and the time to complete the operation is calculated. For this test, the same keys are used for the proposed system CAES and AES-256 algorithm.

Figure 11 shows the results carried out on a PC of Intel CPU i5/2.5MHz, and 4GB of RAM. The CPU time of the proposed algorithm is compared to that of AES-256.

According to the numerical simulation, we can conclude that the proposed system consumes much lower time than AES-256 to accomplish encryption and decryption processes.

5.5 Key Scheduling Example

In the following example we consider the encryption key:

$$53414944420424F5543484B4152454E20$$

$$414E442053414944441204C415A414152$$

And we consider the plaintext message:

$$43525950544F53595354454D20424153$$

$$4544204F4E2043454C4C554C4152204$$

The key and message are written in hexadecimal representation. Table 1 shows the encryption process and shows data for each round.

5.6 CAES Robustness

The proposed algorithm uses 256-bits keys, it implies 2^{256} usable keys. Suppose that we have a sophisticated machine that can test a validity of a key in 10^{-20} seconds, this machine will take approximately $10^{-20} * 2^{256} \cong 1.15 * 10^{57}8$ seconds which means more than $3 * 10^{49}$ years, we deduce then that a brute force attack with an exhaustive key search is impossible.

According to confusion and diffusion test we can assume that statistical attacks can not lead to any positive results.

6 Conclusion and Perspectives

This paper presented a new secret key cryptographic algorithm based on three cellular automata (CA); two reversible CA of 2-dimension , and one irreversible CA of

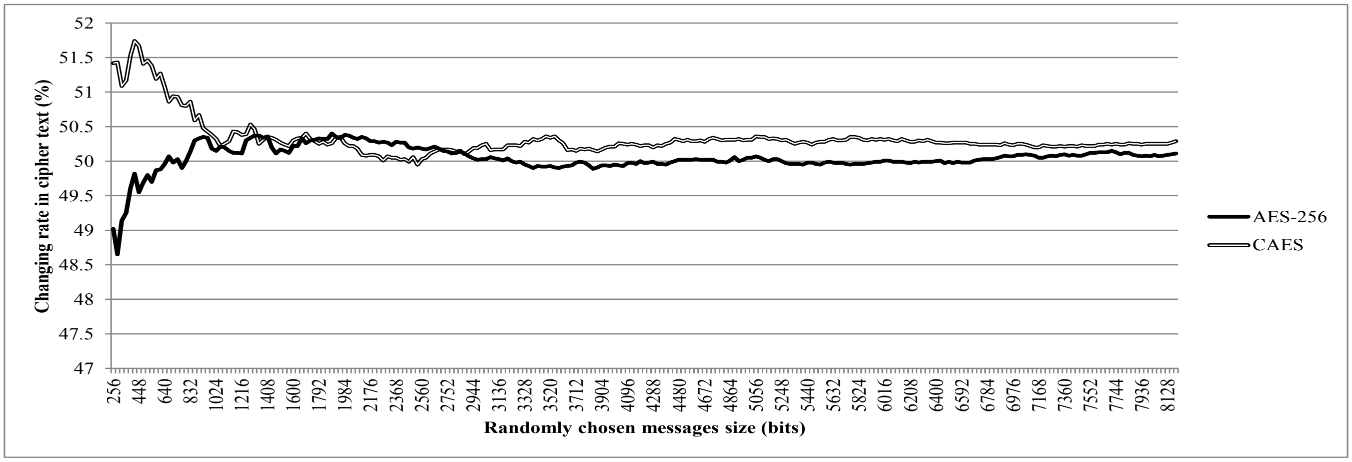


Figure 8: Proposed system CAES and AES-256 changing bits rate

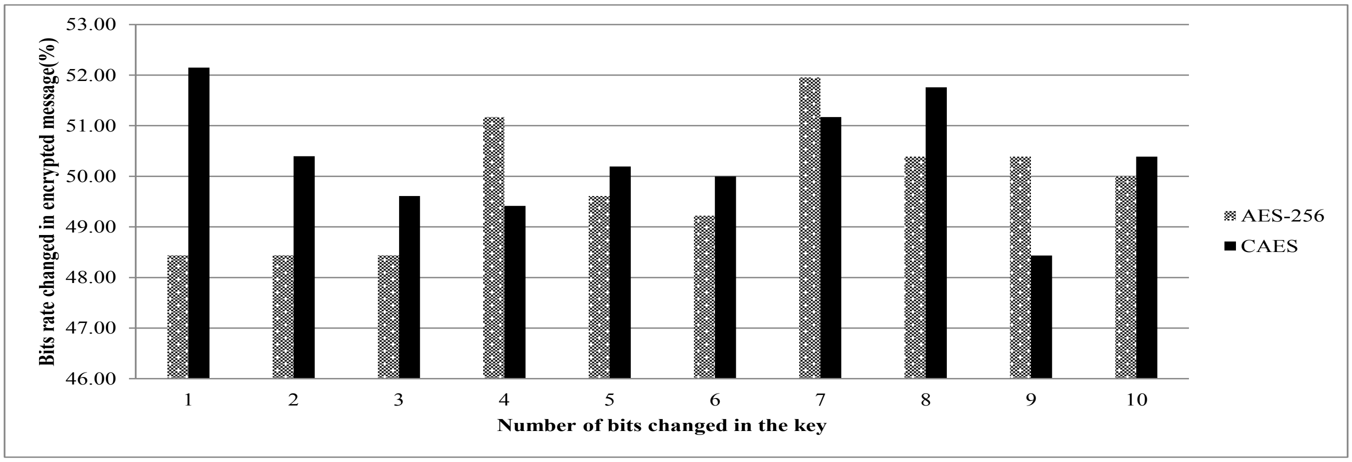


Figure 9: Confusion results of CAES and AES-256

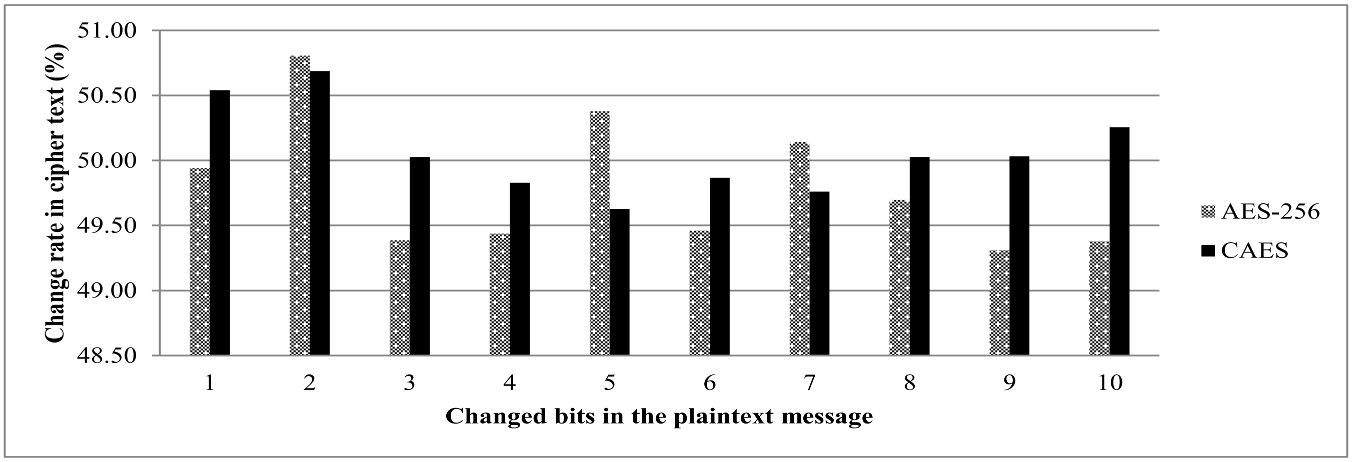


Figure 10: Diffusion results of CAES and AES-256

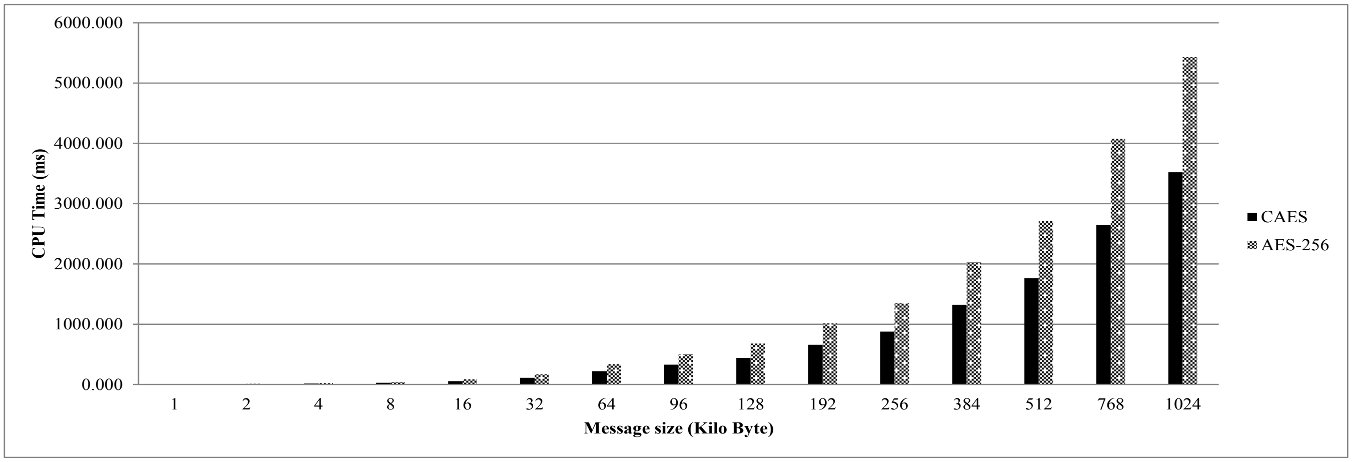


Figure 11: CPU time comparison between CAES and AES-256

Table 1: Encryption example

Round number	Sub key	Cipher message on current round
0	5341494420424F5543484B4152454E20 414E44205341494441204C415A414152	0E264E9879636A0C5B05689B07C8C951 121D03F6461511191C572B97E61B3FEF
1	CFC7055ED408CC328035552D350B680B 738710E42EFC569CE32E1DEAC1609AA2	C507745F4F1F9C0CE2F01D5502219689 9246620B8B8B3466F6CACEA240966452
2	D74701A78383BB4222E0E240C043CF0F 3DCDFD417EC1DCC31EA3E79BE7BF41F9	A952F8334BB0486712BBEFE840CD3A67 6C9A04484427CB8FF84B3F33F93D3B57
3	7F14361E360EA0D9F9280A7258242C06 8E77A47FD855A077715708F930D118F3	9BD1F8FA1959D017C439FCBA0FF7885A 3B413D6DA1F7CEA60149166674B2EEA3
4	F7F957FDF51717F1FADA4D4F4840E361 21BFA31CAB1BA98C89876F878D8F2721	69F6558838FE4FB5269F7F029E89BCAA 4B410B368CCB9C0C112C6AF624A698B8
5	DF4F27C6C7CFCFC1C38B83A98B291919 896F895F2B7D0175497747774F76AE1E	8E153003FA2E8DEF460F227AC80BACC4 21BCE1BEC61615E658A952FFC4DC69BC
6	DF91F191C9B1117052D226A707076F4F 279707FF075F86AB068967291F717F11	043E35306AED77B3EAA749F8321FE6C9 E387EC9F8EE93D977C5F12D67065BE65
7	5FEC1203C53D2B535B7B3B9B6B6A3A33 BB77B227B3D71397339F4B89B34FA2B1	D671F7716C1157AC54716734701BDF80 5A16BDEE3BEB4BB4C5823C3F147BA19D
8	7765E54435DF0721A92129282020A9A9 A9FFA976A97EA87620FE0626EC574DD7	E9E96619233DCCAF7D210183B3CDD2FB 2BB40CB2B039F5FA0202D2968D699CA3
9	75FB62E36107071F9F9F99139F1D9911 15FB9D7F1FF395F595758D75EBF76A6E	46208C7529D5BB23649E129543DBD91E FD56D86239B7D22DFEDE935F1D47D737
10	FF9F269F9098909030B8F870B0387878 B83FB87F30BFB8FF305F3057385881E1	156EBDD30EBA87C1586FA904D075F608 C4710AE8045C89F1A6A0CDF9F957506B
11	750674926C8C9C7D757771F36D676062 ED70EF77E5E967EF65E784F76409827B	F778A842E791633BAAF78F73DFD5DCB7 E02F3AF0C1B78A370C9A606B01CF87FC

1-dimension. The cryptosystem (named CAES) considered 256-bits for both key and message block and it was executed into 12 rounds evolving 12 sub keys. To prove the reliability of the proposed algorithm, various computational results were presented including confusion, diffusion and CPU times comparison. The most advantageous features of the algorithm include fastness and robustness against a brute force attack. Further work is now to implement CAES in a smart card and to realize side channel attacks.

References

[1] P. Anghelescu, S. Ionita, and E. Sofron, "Block encryption using hybrid additive cellular automata," in *IEEE 7th International Conference on Hybrid Intelligent Systems (HIS'07)*, pp. 132–137, 2007.

[2] S. Bouchkaren and S. Lazaar, "A fast cryptosystem using reversible cellular automata," *International Journal of Advanced Computer Science and Applications*, vol. 5, no. 5, pp. 207–210, 2014.

[3] R. H. Brown, M. L. Good, and A. Prabhakar, "Data encryption standard (DES)," *Federal Information Processing Standards (FIPS) Publication 46*, vol. 2, 1993.

[4] K. M. Faraoun, "Fast encryption of RGB color digital images using a tweakable cellular automaton based schema," *Optics & Laser Technology*, vol. 64, pp. 145–155, 2014.

[5] K. M. Faraoun, "A genetic strategy to design cellular automata based block ciphers," *Expert Systems with Applications*, vol. 41, no. 17, pp. 7958–7967, 2014.

[6] P. Guan, "Cellular automaton public-key cryptosystem," *Complex Systems*, vol. 1, no. 1, pp. 51–56, 1987.

[7] J. Jin, "An image encryption based on elementary cellular automata," *Optics and Lasers in Engineering*, vol. 50, no. 12, pp. 1836–1843, 2012.

[8] J. Kari, "Cryptosystems based on reversible cellular automata," *Manuscript*, Apr. 16, 1992. (`http://users.utu.fi/jkari/CACryptoScanned.pdf`)

[9] J. Kari, "Reversibility and surjectivity problems of cellular automata," *Journal of Computer and System Sciences*, vol. 48, no. 1, pp. 149–182, 1994.

[10] J. Kari, "Reversible cellular automata," in *Developments in Language Theory*, pp. 57–68, Springer, 2005.

[11] NIST AES, "Advanced encryption standard," *Federal Information Processing Standard, FIPS-197*, vol. 12, 2001.

[12] P. Ping, F. Xu, and Z. J. Wang, "Image encryption based on non-affine and balanced cellular automata," *Signal Processing*, vol. 105, pp. 419–429, 2014.

[13] J. N. Rao, A. C. Singh, "A novel encryption system using layered cellular automata," *International Journal of Engineering Research and Applications*, vol. 2, no. 6, pp. 912–917, 2012.

[14] P. Sarkar, "A brief history of cellular automata," *ACM Computing Surveys*, vol. 32, no. 1, pp. 80–107, 2000.

[15] J. L. Schiff, *Cellular Automata: A Discrete View of the World*, John Wiley & Sons, 2011.

[16] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition*, John Wiley & Sons, Inc, 1996.

[17] M. Seredynski and P. Bouvry, "Block cipher based on reversible cellular automata," *New Generation Computing*, vol. 23, no. 3, pp. 245–258, 2005.

[18] T. Toffoli and N. H. Margolus, "Invertible cellular automata: A review," *Physica D: Nonlinear Phenomena*, vol. 45, no. 1, pp. 229–253, 1990.

[19] S. Tripathy and S. Nandi, "LCASE: Lightweight cellular automata-based symmetric-key encryption.," *International Journal of Network Security*, vol. 8, no. 3, pp. 243–252, 2009.

[20] S. Wolfram, "Cryptography with cellular automata," in *Advances in Cryptology (CRYPTO'85)*, LNCS 218, pp. 429–432, Springer, 1986.

[21] S. Wolfram, *A New Kind of Science*, Wolfram media Champaign, 2002.

[22] X. Wu, D. Ou, Q. Liang, and W. Sun, "A user-friendly secret image sharing scheme with reversible steganography based on cellular automata," *Journal of Systems and Software*, vol. 85, no. 8, pp. 1852–1863, 2012.

Said Bouchkaren obtained his state engineer diploma in software engineering from AbdelMalek Essaadi University, Morocco, in 2010. Actually, his is PhD student at Natinal School of Applied Sciences of Tangier. In 2011, He joined the department of Computer sciences and mathematics as a professor. His research focuses on cryptography and information security.

Saiida Lazaar started her scientific career with a research contract funded by the CNRS in France with which she prepared a Ph.D. in applied mathematics developing fast algorithms based on wavelets to solve some numerical problems. After her Ph.D., she has held various positions as a researcher with IFP (Institute Franais du Pétrole) and ONDRAF (Office National des Dchets Radioactifs et des matires Fissiles) in Belgium. In 2001, she joined AbdelMalek Essaadi University in Morocco as a Research-Professor. Her research area focuses on wavelets, cryptography, numerical analysis, mathematical and numerical modeling of environment and technological problems. She has a patent assigned at IFP with Dr. Dominique Gurillot. She published various works and special issues in international journals. She participated to national and international conferences, she organized international conferences and workshop and she was member of various scientific committees and scientific projects. She is currently Professor at the National School of Applied Sciences of Tangier; she teaches and supervises projects on mathematics, cryptography and computer networks security. She is also president of the Association ”la Colombe pour la promotion du progiciel libre”.