CA-PCS: A Cellular Automata based Partition Ciphering System

Fatima Ezzahra Ziani[1], Anas Sadak[2], Charifa Hanin[3], Bouchra Echandouri[4], Fouzia Omary[5]
Faculty of Sciences, University Mohammed V
Computer Science Department
Rabat, Morocco

***Abstract*—In this paper, the authors present a modified version of the Partition Ciphering System (PCS) encryption system previously proposed. The previously developed encryption system PCS uses the partition problem to encrypt a message. The goals of newly developed system are avoiding statistical and frequency attacks, by providing a balance between 0s and 1s, ensuring a good level of entropy and achieving confidentiality through encryption. One of the novelties of the new design compared to its predecessor is the use of cellular automata (CAs) during the encryption. The use of CAs is justified by their good cryptographic properties that provide a level of security against attacks, and better confusion and diffusion properties. The new design is first presented with details of the encryption and decryption mechanisms. Then, the results of the DIEHARDER battery of tests, the results of the avalanche test, a security analysis and the performance of the system are outlined. Finally, a comparison between CA-PCS and PCS as well as the AES encryption system is provided. The paper shows that the modified version of PCS displays a better performance as well as a good level of security against attacks.**

Keywords*—Partition ciphering system; partition problem; frequency analysis; cellular automata; avalanche effect; confusion; diffusion; statistical properties; cryptographic properties

I. Introduction

One of the five pillars of cryptography is achieving confidentiality. This latter comprises two principles: data confidentiality and privacy. Data confidentiality ensures that no data is accessed or revealed to unauthorized parties. Privacy controls the access to data and storage of data by concerned parties [1]. This paper presents a modified version of the Partition Ciphering System (PCS), which was previously developed by the authors [2]. It is a symmetrical encryption system based on the partition problem, more precisely the Card-Partition version. The use of the partition problem in PCS was motivated by the fact that it changes the frequency of the appearance of characters between the plaintext and the ciphertext. Consequently, PCS is robust against frequency cryptanalysis; an adversary cannot learn any information about the plaintext from the ciphertext. However, PCS has some limitations to check the diffusion property and resistance to some attacks like linear and differential attacks. A cellular automaton (CA) is a suitable candidate to provide better confusion and diffusion. Also, the CA cryptographic properties could be studied to verify the security level. These later are nonlinearity, algebraic degree, balancedness, resiliency, and correlation immunity. A CA is a dynamic system involving a network of cells. CAs are widely used in cryptography and other fields to benefit from their simplicity, parallelism, and unpredictability. Besides, CAs make the hardware and software implementations easier [3]. In this paper, a new design called CA-PCS (Cellular Automata based Partition Ciphering System) is proposed. It consists of a hybrid CA, with satisfying cryptographic properties, that evolves multiple iterations to increase resistance to linear and differential attacks, followed by the insertions of necessary blocks so that the frequency of all the blocks is the same. In addition to a random permutation is applied to the results of the second step. Each layer produces better confusion and diffusion, and consequently, better resistance to linear and differential cryptanalysis. Also, the cryptographic properties of the CA ruleset are studied and display good results. A high nonlinearity, high algebraic degree, and balancedness are satisfied. CA-PCS was compared to AES and PCS in terms of randomness, security, and performance. Thus, the CA-PCS results are satisfying.

The rest of this article is organized as follows: In Section 2, a brief background on cellular automata is presented. Next, in Section 3, the related works are included. Then, CA-PCS is detailed in Section 4. Section 5 provides a brief description of the PCS and AES encryption systems. Finally, Section 6 presents results and security analysis.

II. Background on Cellular Automata

The history of cellular automata goes back to the 1940s when Stanislaw Ulam [4] initiated their study by taking interest in self-replicating automata. Then in the 1960s, John von Neumann used them in Biology for modeling self-reproduction [5]. They were later on popularized by John Conway's game of life in the 1970s [6]. They were first use in cryptography by Stephen Wolfram in the 1980s [7]. Simply put, a cellular automaton is a network of cells, each of which has a state that changes from a time step t to a time step t+1 according to a defined local rule and depending on its neighbors. The interest of the scientific community in cellular automata stems from the fact that simple local calculations at the cells scale produce a complex behavior at the automaton scale. Another interesting aspect of using cellular automata is that both uniformity and non-uniformity can be modeled. A cellular automaton is defined as [3] (d, L, S, N, f), where d represents the cellular space dimension, L represents the cellular space, S is the finite set of states, N is the neighborhood vector and f or (f_1,f_2,...) is the local rule or ruleset respectively. The global rule of the cellular automata is designated by Φ.

By modifying the tuple (d, L, S, N, f), different kinds of cellular automata can be obtained. One interesting type of cellular automata was introduced by Wolfram in [8]. This kind

TABLE I. AN EXAMPLE OF A LINEAR AND NONLINEAR ECA RULE

Rule	105		135	
Linear?	Yes		No	
Algebraic Normal Form	$1 \oplus x_{i-1} \oplus x_i \oplus x_{i+1}$		$1 \oplus x_i.x_{i+1} \oplus x_{i-1}$	
Truth table	111	0	111	1
	110	1	110	0
	101	1	101	0
	100	0	100	0
	011	1	011	0
	010	0	010	1
	001	0	001	1
	000	1	000	1

of CAs is called Elementary Cellular Automata (ECAs). They are one-dimensional, two-state (0 or 1), 3-neighborhood CAs. They are of particular interest in cryptography as their simple implementation, both in hardware and software, their good cryptographic properties and the small number of possible rules ($2^{2^3} = 256$) are well suited in this field as they can be thoroughly studied. The local rules can be either linear (only XOR operator $\oplus$ in their Boolean expression) or nonlinear (AND($\cdot$)/OR($+$) operators as well in their Boolean expression). Table I shows an example of a linear and nonlinear rule.

III. RELATED WORK

The partition problem or Equal Piles Problem, which is the source of inspiration for this work, was first studied by Jones and Beltramo in [9], where they defined a challenging instance. They tried nine standard genetic algorithms, but without finding an optimal solution. To solve this instance of the problem, Falkenauer [10] and William [11] proposed particular types of genetic algorithms. Concretely, Falkenauer [10] tried to adjust the grouping genetic algorithm that he designed previously using specific crossover and mutation operators suited for similar problems. William [11]used a particular approach in the design of the Eager Breeder genetic algorithm, which makes the manipulation of genetic materials easier and produces better results compared to the previous algorithms. However, their results are not that good for this article's proposed design.More recently, evolutionist algorithms were also used to come up with a solution to the partition problem as in the works of Trichni [12], Bougrine [13] and Kaddouri [14].

The first use of cellular automata in cryptography goes back to Wolfram in [7]. He applied rule 30 to design a pseudorandom number generator (PRNG) and a stream cipher. A more recent example of the use of CAs in an encryption algorithm is the design of Das et al. [15] who proposed a block cipher using one dimensional programmable CAs. Other works using one dimensional uniform CAs include Bhaumik [16] and Roy [17]. Non uniform one-dimensional CAs were studied by Mehta [18] and Bouchkaren [19]. Two dimensional uniform CAs were used by Bouchkaren [20] and Faraoun [21]. CAs were also used for image encryption by Li in [22], who made use of two dimensional non-uniform CAs. Other image encryption schemes can be found in [23] and [24].

IV. CA-PCS DESIGN

A. CA-PCS Encryption Algorithm

The CA-PCS encryption scheme goes through three steps:

1) CA Evolution: The first step includes the hybrid CA evolving of the binary message using the rules $\{90, 150, 30, 180, 45, 90, 150, 30\}$. Linear rules 90 and 150 provide better diffusion property and high cycle length [25]. While nonlinear rules 30, 45, and 180 provide better confusion property [26]. Moreover, these rules provide resistance to linear attacks and differential attacks. Because of the high nonlinearity met after a few iterations and the significant algebraic degree.

2) Blocks Insertion: The second step consists of representing the first step's result as a partition and add some blocks at random positions to get the same appearance frequency for all blocks. At first, the CA output is split into blocks of a randomly chosen size $2 \leq k \leq 16$. Then the ideal cardinality IC is computed $IC = max\{Card(L_1), Card(L_2), ..., Card(L_m)\}$. Next, for each block B_i, the cardinality of the corresponding L_i, representing the positions of Bi in the CA output, is compared to the IC. Accordingly, if $Card(L_i) < IC$, then B_i is inserted in a random position $1 < P_{ij} < size(CAoutput)$ where $0 < j < IC - Card(L_i)$. Next, the P_{ij} is inserted in the ListOfInsertedBlocksPositions.

3) Permutation: Finally, a random permutation is applied to the set $\{L_1, L_2, ..., L_m\}$. This permutation is useful to change the blocks' occurrence lists L_is. It is denoted formally by $\pi : S \rightarrow S$ where S is a set of m elements. m! permutation of $\{L_1, L_2, ..., L_m\}$ are possible. A possible example of a random permutation for m=10, $\pi : \{L_1, L_2, L_3, L_4, L_5, L_6, L_7, L_8, L_9, L_{10}\} \rightarrow \{L_2, L_4, L_1, L_6, L_3, L_9, L_7, L_{10}, L_8, L_5\}$. Following this example, $L_1 \rightarrow L_2$, $L_2 \rightarrow L_4$, $L_3 \rightarrow L_1$, $L_4 \rightarrow L_6$, $L_5 \rightarrow L_3$, $L_6 \rightarrow L_9$, $L_7 \rightarrow L_7$, $L_8 \rightarrow L_{10}$, $L_9 \rightarrow L_8$, $L_{10} \rightarrow L_5$. Accordingly, B_1 will appear in the positions of B_2, B_2 will appear in those of B_4, and so on.

4) Key generation: The secret key comprises four elements:
$SK = \{k, CASeq, ListOfInsertedBlocksPositions, PSeq\}$
The random integer k is the blocks size. The CASeq binary sequence where $CASeq = M \oplus M'$ where M is the plaintext, and M' is the output of the CA evolution step. The ListOfInsertedBlocksPositions which comprises the positions where blocks are inserted. the PSeq binary sequence $PSeq = M" \oplus C$ where M" is the output of the blocks insertion step and C is the ciphertext. Fig. 1 summarizes the encryption process of CA-PCS.

B. CA-PCS Decryption Algorithm

The CA-PCS decryption process, as Fig. 2 displays, is as follows, given the ciphertext C and the secret key $SK = \{k, CASeq, ListOfInsertedBlocksPositions, PSeq\}$: At first, the PSeq sequence is XORed with the ciphertext to get M". Then, M" is split into blocks of size k. Next, inserted blocks are removed from M" using the ListOfInsertedBlocksPositions to get M'. Then M' is XORed with the CASeq to get the plaintext.

Algorithm 1 CA-PCS Encryption Algorithm

Input: The message M
Output: The ciphertext C and the secret key K
Begin
$it \leftarrow 64$
$size \leftarrow sizeOf(M)$
$k \leftarrow random(2,16)$ ▷ random integer$2 < k \leq 16$
$ruleSet \leftarrow \{30, 90, 150, 30, 180, 45, 90, 150\}$
for $0 < j \leq it$ **do**
  for $0 < i \leq size$ **do**
    $x \leftarrow i - 1 \bmod sizeOf(ruleSet)$
    if $(x == 0)||(x == 3)$ **then** ▷ 30
      $M'[i] \leftarrow M[i-1] \oplus (M[i] + M[i+1])$
    else if $(x == 1)||(x == 6)$ **then** ▷ 90
      $M'[i] \leftarrow M[i-1] \oplus M[i+1]$
    else if $(x == 2)||(x == 7)$ **then** ▷ 150
      $M'[i] \leftarrow M[i-1] \oplus M[i] \oplus M[i+1]$
    else if $x == 4$ **then** ▷ 180
      $M'[i] \leftarrow M[i-1] \oplus (M[i].(1 \oplus M[i+1]))$
    else ▷ 45
      $M'[i] \leftarrow M[i-1] \oplus (M[i] + (1 \oplus M[i+1]))$
    end if
  end for
end for
$CASeq \leftarrow M \oplus M'$
$M" \leftarrow DivideIntoBlocks(M', k)$
$n \leftarrow sizeOf(M")$
$m \leftarrow NumberOfDifferentBlocks(M")$
$Partition \leftarrow ToPartition(M")$
$ListOfBlocks \leftarrow DifferentBlocks(M")$ ▷ $\{B_1, ..., B_m\}$
$IC \leftarrow ComputeIdealCardinality(PlaintextPartition)$
for $1 \leq i \leq m$ **do**
  while $Card(L_i) < IC$ **do**
    $M" \leftarrow insert(B_i, M", randomPosition)$
    $Insert(ListOfInsertedBlocksPositions, randomPosition)$
    $Insert(L_i, randomPosition)$
  end while
end for
$permutation \leftarrow generateRandomPermutation(\{1, 2, ..., m\})$
$Ciphertext \leftarrow applyPermutation(M", permutation)$
$PSec \leftarrow Ciphertext \oplus M"$
$secretK \leftarrow \{k, CASeq, ListOfInsertedBlocksPositions, PSec\}$
End

Algorithm 2 Decryption algorithm

Input:The secret key SK and the ciphertext C
Output: The message M
Begin
$M" \leftarrow C \oplus PSec$
$M^{(3)} \leftarrow DivideIntoBlocks(M", k)$
for i from $sizeOf(ListOfInsertedBlocksPositions)$ to 1 **do**
  $M' \leftarrow Remove(M^{(3)}, ListOfInsertedBlocksPositions[i])$
end for
$M \leftarrow M' \oplus CASeq$
End

V. The PCS and AES description

This section presents a brief description of a previously developed scheme Partition Ciphering System (PCS) and the Advanced Encryption Standard (AES).

A. Partition Ciphering System (PCS)

The Partition Ciphering System PCS [2] is a symmetric enryption schemme that encrypts a plaintext in three steps. the first step consists of the construction of a partition from the plaintext, which is initially split into blocks of size k>2. Each block is associated with a list of occurrences. This partition undergoes some transformations in a way to make the ciphertext resistant to frequency cryptanalysis. Next, the ideal cardinality IC is computed : let $c = \frac{n}{m}$, where n is the number of blocks in the plaintext, and m is the number of different blocks in

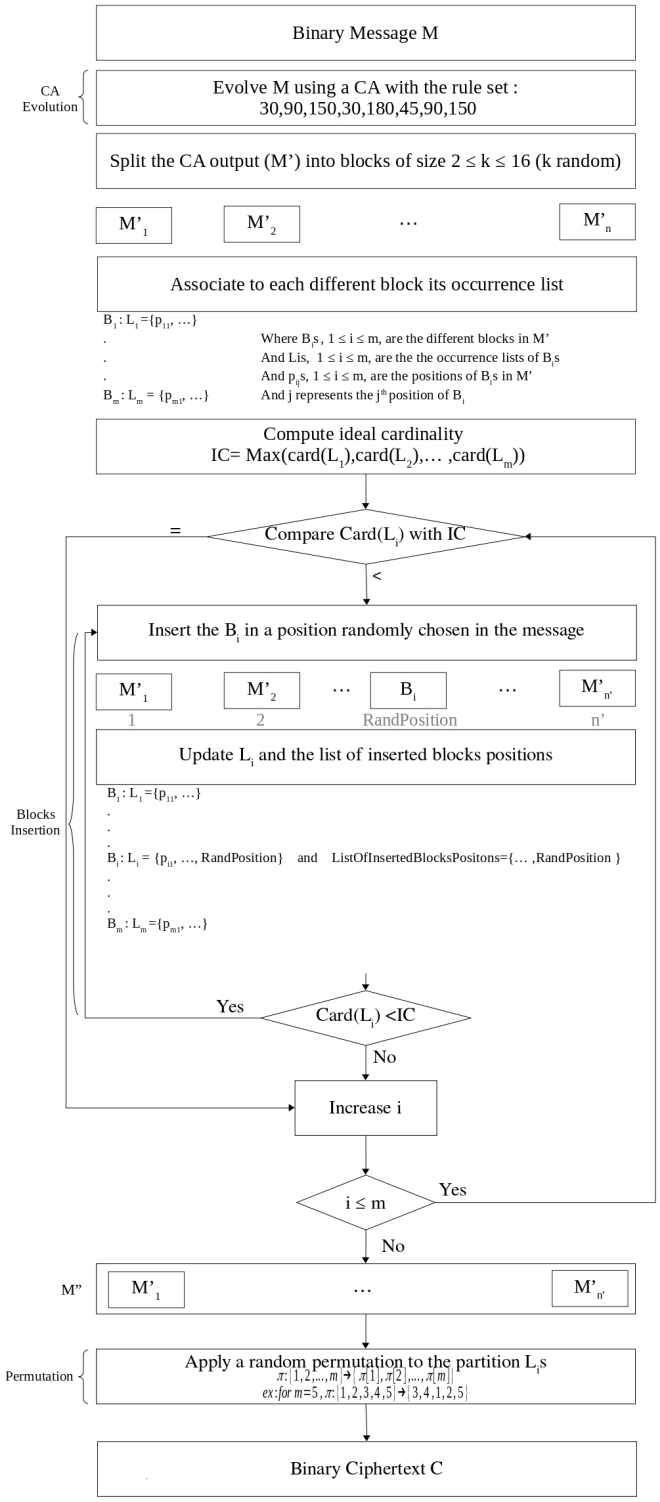

Fig. 1. CA-PCS Encryption

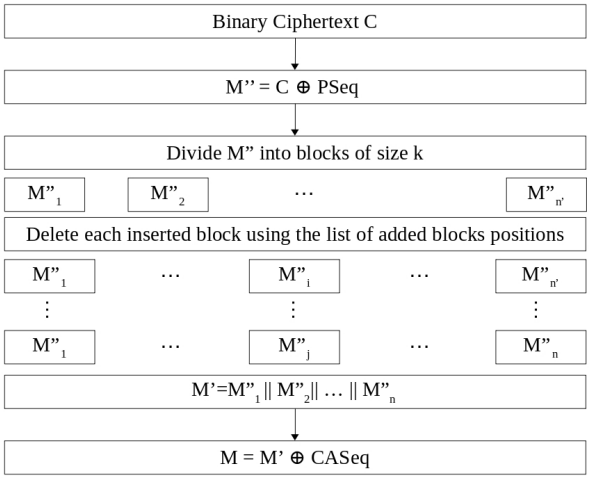

Fig. 2. CA-PCS Decryption

the plaintext. If $c \in \mathbf{N}$ then $IC = c$ else $IC = \lceil c \rceil$. This cardinality defines the number of occurrences of each block in the ciphertext. In the last step, the blocks B_is are inserted or deleted according to the cardinal of the corresponding list of appearances L_i. When the $Card(L_i) < IC$, then the block B_i is appended to the message. When the $Card(L_i) > IC$, then the block B_i is deleted from a random position.

B. Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) [27] is a symmetric cipher that encrypts 128-bit blocks using keys of size 128 bit, 192 bit, or 256 bit. It comprises N rounds, where N changes according to the length of the key: 10 for a 128-bit key, 12 for a 256-bit key, and 14 for a 192-bit key. In the first step, the plaintext is XORed by the first 128 bit of the key. Next, for N-1 iteration, four operations are performed: SubBytes, ShiftRows, MixColumns, and AddRoundKey. [27] provides a detailed description of these operations. Finally, the last round consists of only SubBytes, ShiftRows, and AddRoundKey operations.

VI. Results and Security Analysis

This section displays the statistical tests and the confusion and diffusion properties of CA-PCS compared to the AES.

A. Dieharder Test

The battery of tests Dieharder was designed by Robert G. Brown to check out the behavior of PRNGs and cryptographic primitives like encryption systems, hash functions, and MACs. It involves tests from diehard, some NIST tests, and other tests developed by Brown and Bauer [28]. The authors generated three files of 10 Mb using PCS, CA-PCS, and AES ciphers. Then, they run the battery overs these files. Table II displays the results. The P-values are the probability that the generated sequences are random. If 0.005<P-value<0.995, then the systems pass the test. Since 0.10<P-values(PCS)<0.91, 0.2<P-values(CA-PCS)<0.92, and 0.005<P-values(AES)<0.95, then all the systems pass all the tests. Also, the P-values of the ciphers are uniformly distributed in the range [0, 1], to conclude, CA-PCS displays good results regarding the statistical tests compared to PCS and AES.

TABLE II. Dieharder Results of CA-PCS, AES, and PCS

Tests names	P-values CA-PCS	P-values AES	P-values PCS
Diehard birthdays	0.5357	0.0836	0.8625
Diehard operm5	0.4946	0.0967	0.8971
Diehard rank 32x32	0.5887	0.7711	0.1402
Diehard rank 6x8	0.7192	0.6936	0.3240
Diehard bitstream	0.4615	0.6593	0.4530
Diehard opso	0.5559	0.7204	0.3559
Diehard oqso	0.5092	0.6363	0.1898
Diehard dna	0.5686	0.3142	0.2811
Diehard count 1s str	0.4114	0.8797	0.8988
Diehard count 1s byt	0.6995	0.8451	0.7611
Diehard parking lot	0.2622	0.8514	0.773
Diehard 2dsphere	0.4555	0.5370	0.7910
Diehard 3dsphere	0.6735	0.3863	0.2487
Diehard squeeze	0.6888	0.8732	0.7991
Diehard sums	0.9130	0.0058	0.1779
Diehard runs	0.2342	0.3810	0.7702
Diehard craps	0.7063	0.8630	0.9093
Marsaglia tsang gcd	0.6682	0.7107	0.4046
Sts monobit	0.5815	0.6915	0.54319
Sts runs	0.4394	0.4656	0.1070
Sts serial	0.6616	0.5643	0.6388
Rgb bitdist	0.6689	0.5724	0.4844
Rgb minimum distance	0.5515	0.3475	0.4441
Rgb permutations	0.6639	0.6588	0.4145
Rgb lagged sum	0.5074	0.5363	0.6067
Rgb kstest test	0.2840	0.4934	0.1025
dab bytedistrib	0.5920	0.4758	0.2636
dab dct	0.8842	0.9448	0.8735
dab filltree	0.4757	0.4721	0.5212
dab filltree2	0.8987	0.7090	0.3727
dab monobit2	0.8994	0.0507	0.6055

B. Confusion and Diffusion Tests

This section presents the confusion and diffusion properties of the CA-PCS system in comparison with AES. A secure encryption system from statistical analysis, as stated by Shannon [29], has good confusion and diffusion properties (e.g., AES is a secure system). If the relation between the ciphertext and the secret key is hidden, then the confusion property is verified. In other terms, replacing one bit in the secret key has an impact on most of the bits in the ciphertext. If the relation between the plaintext and the ciphertext is masked, then the diffusion property is checked. In other words, changing one bit in the plaintext affect almost all the bits of the ciphertext. Fig. 3 shows the confusion property for CA-PCS compared to the AES. According to Fig. 3, the percentage of the changed bits in the ciphertext is approximately 50% for CA-PCS and AES. Concretely, the values for CA-PCS are between 0.40% and 0.61%, while the values for AES are between 0.36% and 0.61%. These values confirm that CA-PCS has better confusion property. Fig. 4 illustrates the diffusion property of CA-PCS and AES. The mean value of the percentages of changed bits in the ciphertext is nearly 50%. The values for CA-PCS are

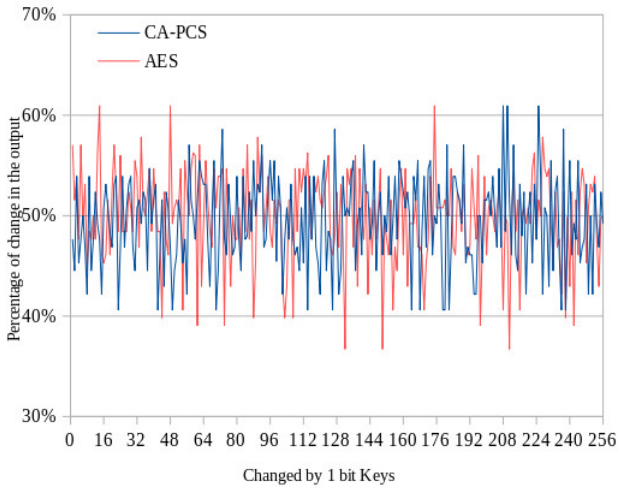

Fig. 3. Confusion Test of CA-PCS and AES

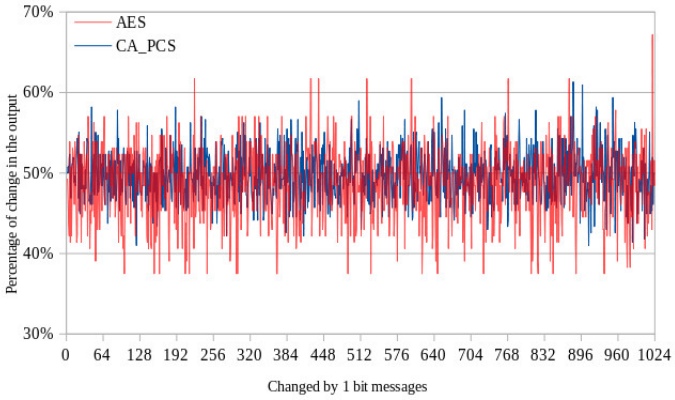

Fig. 4. Diffusion Test of CA-PCS and AES

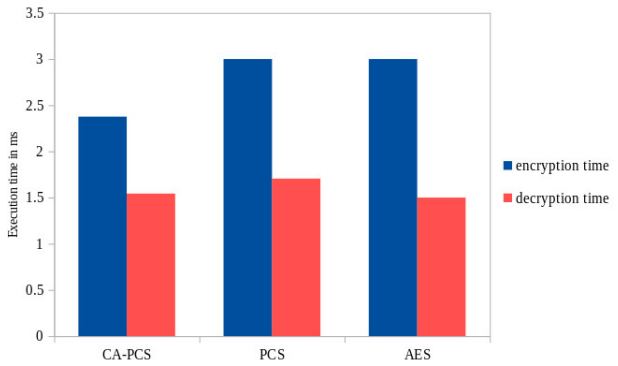

Fig. 5. Encryption and decryption time of CA-PCS, PCS and AES

between 41% and 61%, and the values for AES are between 37% and 67%. Consequently, CA-PCS has better diffusion.

C. Encryption and Decryption Time of CA-PCS, AES and PCS

This part (Fig. 5) compares the encryption and decryption time of CA-PCS with the previously developed scheme PCS and AES. Fig. 5 shows that CA-PCS requires less time in the encryption process compared to PCS and AES. While the PCS and AES take the same time to encrypt. The time of decryption is approximately the same for CA-PCS, PCS, and AES. To conclude CA-PCS displays good results.

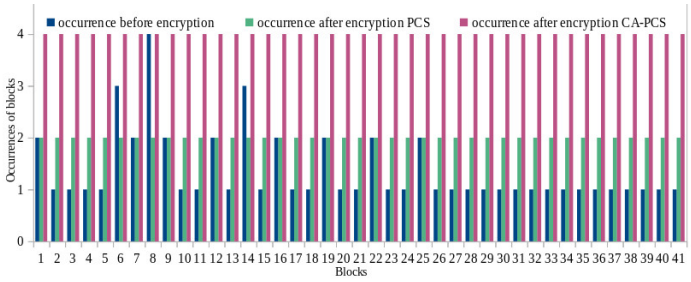

Fig. 6. Frequency of blocks before and after encryption for CA-PCS and PCS

TABLE III. NONLINEARITY

Iterations	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	2	0	0	2	2	2	0	0
2	8	8	8	8	8	8	8	8
3	32	48	48	48	28	44	48	48

D. Frequency Analysis

This part presents the frequency analysis of the outputs of CA-PCS and PCS. As mentioned in [2], the purpose was to have a ciphertext with blocks appearing with the same frequency, so that frequency analysis does not reveal any information about the plaintext. As CA-PCS is an improved version of PCS, the same objective persists. CA-PCS is different from PCS in all steps. The CA evolution is the first step of CA-PCS. Next, the ideal cardinality computation. Later, the insertion of blocks follows. The resulting intermediate output undergoes a permutation. While in PCS, the ideal cardinality is computed in a way to have blocks to add or remove. The objective of CA-PCS design is to provide better confusion and diffusion, in addition to resistance to some attacks like linear and differential attacks. Fig. 6 represents the frequency analysis performed on the outputs of CA-PCS and PCS for the same plaintext. Fig. 6 shows that frequency analysis will never divulge any information. As a result, frequency cryptanalysis is impossible.

E. Cryptographic Properties of the Ruleset Used in the CA Evolution

This section presents the cryptographic properties, namely, nonlinearity, algebraic degree, correlation immunity, resiliency, and balancedness, of the CA ruleset $\{30, 90, 150, 30, 180, 45, 90, 150\}$. It is applied alternately on the CA cells in the evolution step. In this section, to study the ruleset, an example of 8 cells is considered. Tables III to VII shows the variation of the cryptographic properties with iterations.

Nonlinearity and algebraic degree increase significantly within iterations. Also, balancedness persists. The resiliency and the correlation immunity decrease with iterations because high nonlinearity affects the level of resiliency and correlation immunity. Most of the cryptographic systems require high

TABLE IV. ALGEBRAIC DEGREE

Iterations	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	2	1	1	2	2	2	1	1
2	3	2	2	3	3	3	2	2
3	4	3	3	4	5	4	3	3

TABLE V. RESILIENCY

Iterations	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	0	1	2	0	0	0	1	2
2	0	2	2	0	1	0	2	2
3	0	0	0	0	0	0	0	1

TABLE VI. CORRELATION IMMUNITY

Iterations	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	0	1	2	0	0	0	1	2
2	0	2	2	0	1	0	2	2
3	0	0	0	0	0	0	0	1

TABLE VII. BALANCEDNESS

Iterations	x_0	x_1	x_2	x_3	x_4	x_5	x_6	x_7
1	✓	✓	✓	✓	✓	✓	✓	✓
2	✓	✓	✓	✓	✓	✓	✓	✓
3	✓	✓	✓	✓	✓	✓	✓	✓

TABLE VIII. BRUTE FORCE ATTACK OF AES, PCS AND CA-PCS

Encryption schemes	AES-128	AES-192	AES-256	PCS	CA-PCS
Key length	128 bit	192 bit	256 bit	$\geq$ 256 bit	$\geq$ 256 bit
# possible keys	2^{128}	2^{192}	2^{256}	$\geq 2^{256}$	$\geq 2^{256}$
Security level	near term	near term	long term	long term	long term

nonlinearity, and algebraic degree as well as balancedness. These cryptographic properties are important to avoid attacks, particularly linear attacks, differential attacks, and statistical cryptanalysis.

F. Brute-Force Attack

In a brute-force attack, the attacker tests each possible key to get a comprehensible plaintext from the transformation of the ciphertext [1]. The key length is considered the security parameter that provides the security level of the studied system. This attack needs more time and resources to get the right key when the key length is high. It can be impossible unless an attacker has a quantum computer. If the level of security desired is for the near term, then a symmetric key of at least 128 bit is used. The key should be of at least 256 bit to reach long term security. Since the AES has three versions, AES-128, AES-192, and AES-256, both security levels can be satisfied. PCS, from [2], has a secret key of size greater than 256 bit. Also, CA-PCS has a secret key of at least 256 bit. Unless an attacker has a quantum computer, he cannot get the secret key to decrypt to an intelligible plaintext. Table VIII summarises the security level of AES, PCS, and CA-PCS.

G. Linear and Differential Attacks

Linear attack analyzes the linear approximations of the plaintext, the ciphertext, and the secret key [30]. It is a known-plaintext attack, while differential attack studies the differences between plaintexts and ciphertexts [31]. It is a chosen-plaintext attack. A cipher should be robust against the linear and differential attacks. The confusion property, which is satisfied using the nonlinear parts of the system, is necessary to resist these types of attacks. In general, S-Boxes are responsible for this purpose. But, other primitives, like nonlinear cellular automata, can lead to the same results. In CA-PCS, the ruleset used to evolve the CA has high nonlinearity, and maintain the balancedness. These features make these attacks difficult for a cryptanalyst.

VII. CONCLUSION

In this article, an enhanced version of PCS, a previously developed encryption scheme, is proposed. The proposed system, called CA-PCS, makes use of cellular automata to increase the security level of the design. Precisely, the ruleset used provides satisfying results in terms of cryptographic properties, randomness tests, confusion, and diffusion properties. Linear and differential attacks are difficult to achieve because of the high non-linearity and the high algebraic degree provided by the ruleset. Also, the balancedness and the randomness produce resistance to statistical cryptanalysis. Moreover, CA-PCS is robust against brute force attacks. Besides, the performance of CA-PCS is better than PCS and AES. In future work, the authors will extend the proposed scheme to ensure authentication.

REFERENCES

[1] W. Stallings, Cryptography and network security: principles and practice. Pearson Prentice Hall, 2017.

[2] F. E. Ziani and F. Omary, "Partition Ciphering System: A Difficult Problem Based Encryption Scheme," International Journal of Advanced Computer Science and Applications, vol. 10, no. 11, 2019

[3] K. Bhattacharjee, N. Naskar, S. Roy, and S. Das, "A survey of cellular automata: types, dynamics, non-uniformity and applications," Natural Computing, 2018.

[4] S. Ulam, "Random processes and transformations," in Proceedings of the International Congress on Mathematics, vol. 2, pp. 264-275, 1952.

[5] J. T. Schwartz, J. V. Neumann, and A. W. Burks, "Theory of Self-Reproducing Automata," Mathematics of Computation, vol. 21, no. 100, p. 745, 1967.

[6] M. Gardner, "On cellular automata self-reproduction, the garden of eden and the game of Life," Scientific American, vol. 224, no. 2, pp. 112 - 118, 1971.

[7] S. Wolfram, "Cryptography with Cellular Automata," Lecture Notes in Computer Science Advances in Cryptology - CRYPTO-85 Proceedings, pp. 429 - 432.

[8] S. Wolfram, A new kind of science. Champaign, IL: Wolfram Media, 2002.

[9] D. R. Jones and M. A. Beltramo, " Solving Partitionning Problems with Genetic Algorithms," Proceedings of the Fourth International Conference on Genetic Algorithms, 1991.

[10] F. Emannuel, "Solving Equal Piles with the Grouping Genetic Algorithm," Proceedings of the Sixth Intenational Conference on Genetic Algorithms, 1995.

[11] W. A. Greene, "Genetic Algorithms For Partitioning Sets," International Journal on Artificial Intelligence Tools, vol. 10, no. 01n02, pp. 225 - 241, 2001.

[12] S. Trichni, F. Omary, B. Boulahiat, and M. Bougrine, "A new approach of mutation's operator applied to the ciphering system SEC," 6th IC-CIT: International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2011), 2011.

[13] M. Bougrine, F. Omaiy, S. Trichni, and B. Boulahiat, "New evolutionary tools for a new ciphering system SEC version," 2012 IEEE International Carnahan Conference on Security Technology (ICCST), 2012.

[14] Z. Kaddouri, F. Omary, A. Abouchouar , and M. Daari, "Balancing Process to the Ciphering System SEC," Journal of Theoretical and Applied Information Technology, 2013.

[15] A. Ray and D. Das, "Encryption Algorithm for Block Ciphers Based on Programmable Cellular Automata," Communications in Computer and Information Science Information Processing and Management, pp. 269 - 275, 2010.

[16] J. Bhaumik and D. R. Chowdhury, "Design and implementation of Cellular Automata based diffusion layer for SPN-type block cipher," 2012 International Conference on Informatics, Electronics & Vision (ICIEV), 2012.

[17] S. Roy, S. Nandi, J. Dansana, and P. K. Pattnaik, "Application of cellular automata in symmetric key cryptography," 2014 International Conference on Communication and Signal Processing, 2014.

[18] R. K. Mehta and R. Rani, "Pattern generation and symmetric key block ciphering using cellular automata," 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2016.

[19] S. Bouchkaren and S. Lazaar, "A New Cryptographic Scheme Based on Cellular Automata," Lecture Notes in Electrical Engineering Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015, pp. 663 - 668, 2016.

[20] S. Bouchkaren and S. Lazaar, "A fast cryptosystem using reversible cellular automata," International Journal of Advanced Computer Science and Applications, vol. 5, no. 5, 2014.

[21] 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.

[22] K. Li, M. Sun, L. Li, and J. Chen, "Image Encryption Algorithms Based on Non-uniform Second-Order Reversible Cellular Automata with Balanced Rules," Intelligent Computing Theories and Application Lecture Notes in Computer Science, pp. 445 - 455, 2017.

[23] A. Y. Niyat, M. H. Moattar, and M. N. Torshiz, "Color image encryption based on hybrid hyper-chaotic system and cellular automata," Optics and Lasers in Engineering, vol. 90, pp. 225 - 237, 2017.

[24] Y. Wang, Y. Zhao, Q. Zhou, and Z. Lin, "Image encryption using partitioned cellular automata," Neurocomputing, vol. 275, pp. 1318 - 1332, 2018.

[25] K. Chakraborty and D. R. Chowdhury, "CSHR: Selection of Cryptographically Suitable Hybrid Cellular Automata Rule," Lecture Notes in Computer Science Cellular Automata, pp. 591-600, 2012.

[26] L. Mariot, "Cellular Automata, Boolean Functions and Combinatorial Designs," dissertation, 2018.

[27] J. Daemen and V. Rijmen, "The Advanced Encryption Standard Process," Information Security and Cryptography The Design of Rijndael, pp. 1 - 8, 2002.

[28] Robert G. Brown's General Tools Page. [Online]. Available: https://phy.duke.edu/ rgb/General/dieharder.php.

[29] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, no. 4, pp. 623 - 656, 1948.

[30] A. Biryukov and C. Canniere, "Linear Cryptanalysis for Block Ciphers," Encyclopedia of Cryptography and Security, pp. 351 - 354, 2011.

[31] E. Biham, "Differential Cryptanalysis," Encyclopedia of Cryptography and Security, pp. 147 - 152.