Security of a New Hybrid Ciphering System

Mohammed BOUGRINE¹, Fouzia OMARY², Salima TRICHI²
Faculty of Sciences, Mohammed V University in Rabat
Department of Computer Science
Rabat, Morocco

Abstract—The protection of privacy is a very sensitive subject and comes into force in all areas. They represent the first priority in the development of new technologies. In fact, opt for a new Big data or IOT technology is a very difficult decision for organizations and calls into question the confidentiality, integrity, authenticity and non-repudiation of their data. Convincing these organizations to adhere to technological intelligence is tantamount to providing them with powerful tools and mechanisms of security that are resistant to new types of vulnerability. However, the problem today is that most security tools are based on old cryptographic primitives. Certainly; they have proved their resistance until today but the need to have others becomes crucial in order to meet the new technological requirements. In this paper, we propose a new hybrid encryption alternative based on two encryption systems, the first one is an evolutionary encryption system and the second one is based on an asymmetric encryption system. To present this work we begin with a description of our evolutionary cipher system. Then, we present the principle of proposed hybridization and its contribution compared to other existing systems. Finally, we perform a detailed study on the safety of this system and its long-term resistance.

Keywords—Security; confidentiality; hybrid encryption; evolutionary algorithms; symmetrical encryption; cryptography

I. INTRODUCTION

Symmetrical encryption systems, although invented long before the asymmetric encryption systems, are still the most commonly used type of cryptosystems used in applications and information systems [1].

The widespread use of symmetrical encryption systems is mainly due to their simplicity, speed and security strength compared to asymmetric encryption systems [2]. This is the case as long as an attacker cannot discover the secret key, which represent a critical criterion for the application. Therefore, the main difficulty lies in the distribution and the agreement over the keys to enable the entities concerned by this communication to share the same initial secret without any potential attacker intercepting it [1][2]. The delivery of the secret key must take advantage of all possible means of protection to ensure the authentication, the integrity and the confidentiality of all the information exchanged.

Whitfield Diffie and Martin Hellman in [3] were able to put an end to this problem and avoid the pitfalls of symmetrical systems using a new mechanism based on two keys, one public and one private [3]. The emergence of asymmetric cryptosystems, or public key cryptosystems, provide an indubitable answer to the key exchange problem. The robustness of this type of algorithms is based on the difficulty and complexity of resolution of certain mathematical problems [4]. However, these algorithms lack speed and are practically unusable especially for an online exchange with large volumes of data. However, this kind of cryptosystems is used in hybrid cryptosystems. Hybrid cryptosystems are a new approach that consists of a combination of symmetric and asymmetric algorithms in order to take advantage of the benefits of each of them and make them complementary.

In this paper, we took inspiration from the hybrid cryptosystems approach [5][6][7][8], to design a new Hybrid Evolutionary Cryptosystem. The goal of the present work is to describe the process of this system and then to show his strength against other existing hybrid cryptosystems which are in widespread use.

II. RELATED WORK

Philip Zimmermann was the first to introduce hybrid cryptographic systems. He managed to combine the IDEA symmetric encryption system with the RSA asymmetric encryption system. His work gave birth to the Pretty Good Privacy (PGP) cryptosystem [5]. PGP was the first hybrid encryption system created. Since then, PGP has incorporated other cryptographic concepts to cover not only the data confidentiality but also the different security requirements for the exchange, storage and disclosure of data for private use (signing, compression, etc.).

III. BACKGROUND

A. Description of the Advanced Symmetrical Evolutionary Ciphering (ASEC)

Brief History:

In 2006, the Symmetrical Evolutionary Ciphering (SEC) was created. It was one of the first systems introduces evolutionary algorithms [22][24] as an encryption process in [9][10] and [11]. It is based on a simple principle. First, the plaintext is encoded and each character is linked to its positions’ list. Then, a search through the different iterations of the genetic algorithm is done in order to find the most powerful combination of these lists to realize a well secured encryption [11].

In each step of this algorithm, a set of mathematical mechanisms and methods is applied in order to find the solution that meets the need of confidentiality. In 2011, in order to respond to new security requirements, we developed an advanced version of this system called “Advanced Symmetrical Evolutionary Ciphering” (ASEC), and we
introduced the partition problem in the stage of mutation [12] and also at the level of the evaluation function [13].

Ciphering Algorithm:
To explain the ciphering, let’s T be the plaintext.
T is an input of our system.

Step 1: Encoding
This is the stage of coding the plaintext as a chromosome.
T contains the following characters: c1, c2, c3, ..., cm.

Each character occurs at least in one position in the text, then we define for each character his list of positions in this text called Li (0 ≤ i ≤ m+1). So, the plaintext is represented by the vector T = {(c1, L1) ... (cm, Lm)} which will be the initial chromosome of all the populations.

Also, with this representation, they are two important properties of this population, which are:

1. Li ∩ Lj = Ø, for i, j ∈ [1, m], with i ≠ j.
2. Li, L2, ..., Lm is a partition of the set {1, 2 ..., n}

Step 2: Generating the initial population
Let:
• q be the population size
• CHj (j ∈ [1, q]) be the representation of each chromosome
• and P1 be the representation of the initial population

Then, the first population will be represented by: P1 = {CH11, CH12, CH13, ..., CH1q}

The second one is: P2 = {CH21, CH22, CH23, ..., CH2q}, and so on. Each chromosome CHj (j ∈ [1, q]) is defined as a new combination between the ci and Li.

The first generated population must not follow a well-defined function but it must rely on random events to generate it because more the initial population generation is random more the algorithm is efficient.

Step 3: Evaluation
In this step, we evaluate a random partition Ej constructed from each chromosome Xj such as:
Ej = {ej1, ej2, ..., ejm}.

And then you have to assign a value to each chromosomal partition in order to evaluate its effectiveness using the fluid formula [13]:

\[ F(Xj) = \sum_{i=1}^{m} [\text{Card} (eji) - \lfloor n/m \rfloor] \]

Through this function, we try to find the partition that all his elements has a similar cardinality.

Step 4: Selection
Using a selection method the roulette wheel selection. As its name suggests, the principle of this function is based on the casino roulette performance [14]. It can transform the performance of each parent to a probability that will be distributed later on the roulette of the game. We randomly choose the value of the parameter “r” that can be considered as the ball to be cast on the wheel in order to choose the elected chromosome [15].

Step 5: Genetic operators
There are two steps: a crossover and a mutation.

MPX Crossover method:
In this case, we must use the crossover that maintains the characteristics of the original population which are:

1. Li ∩ Lj = Ø, for i, j ∈ [1, m], with i ≠ j.
2. Li, L2, ..., Lm is a partition of the set {1, 2 ..., n}

This is why the MPX crossover is used [16] where the coding of the child has a strong analogy with that of the parents.

This crossover was developed specifically for the TSP problem by Gorges-Schleuter and Mühlenbein [17] in 1988. The MPX operator is illustrated in the example below:

Mutation:
Contrary to the old SEC encryption system, in ASEC the mutation step is the most important step in the algorithm.
In fact, at this level we try to create the new partition of the positions lists [12].

It is noted that the new lists must absolutely respect the properties of the original text. That said:
✓ they must be independent: L'j ∩ L'i = Ø
✓ they must be ordered

Construction of the new generation:
The new generation is built keeping the same chromosomes of the population of the crossover except that this time it is based on the new lists.
In other words, instead of having the child: L3-L1 - L5 - L10-L7-L2-L4-L9-L6-L8
We will have L’3-L’1 - L’5 - L’10-L’7-L’2-L’4-L’9-L’6-L’8 and so on.

Discussion:
From the new design of the lists of positions, we can see that a character can replace 1, 2 or even more characters as it can be replaced by several other characters instead of a single...
character. The relationship between the initial character and the replacement character becomes more complex.

**Encryption Key:**

Finally, to encrypt plaintext, the key is represented as follow:

- The sequence of numbers with the permutation of the elected child.
- The sequence of numbers with the permutation of the elected child in the new partition of lists.
- The sequence of numbers with the cardinals of the elected child lists.
- And ultimately, the final permutation of encryption.

**Decryption:**

To Decrypt message, we applied the same Key in inverse order.

### IV. HYBRID EVOLUTIONARY CRYPTOSYSTEM

**Problematic:**

ASEC can be considered as a symmetric encryption system because it uses the same key for encryption and decryption. The only difference is that the evolutionary encryption resembles the disposable mask encryption mechanism. In fact, the encryption key is not exchanged once but changes from an execution to another. It is then considered a session key. The problem that arises in this case is that this key must absolutely be secured whenever we wish to establish a communication using this system of encryption.

**Solution:**

To address this problem, we propose to use the principle of hybrid cryptosystems using the symmetric ciphering ASEC that allows to include the session keys generation step. In this new cryptosystem, the keys are generated implicitly by the system of encryption.

The principle is simple and can be illustrated by the following diagram (Fig. 1).

The question that arises is: what is the benefit of ASEC for a user compared with other symmetrical systems used in the PGP cryptosystem?

For this, we propose to begin with the security study of this system because it represents the first factor to choose it. Then we give a comparative study with the others symmetrical systems used in the PGP cryptosystem.

To answer this question, we need to specify the selection criteria for these symmetrical systems and see how our system can meet these criteria.

We can then distinguish the following criteria:

- Security degree
- Time: Fast or slow
- Material: the capacity of the equipment used to encrypt and decrypt messages.
- Setting: the size of the key and the blocks, allowing to increase the strength of the algorithm against brute-force attacks.
- Power: ability to resist the different possible attacks according to the setup of this algorithm in the PGP system.
- Reputation: it is related mostly to the seniority of the algorithm in the field.
- Patented.

In fact, in this study, we can't rely on all criteria to demonstrate the effectiveness of this system because obviously ASEC has not yet been released to talk about his reputation and accessibility. However, we're going to focus on security study of this system because it represents one of the most interesting factors in this phase and then we will do a comparison study based on the other criteria such as the execution time, the setting and the Material.

### V. SECURITY STUDY

#### A. System Setting and Brute Force Attack

Symmetric ciphers systems setting is a prerequisite essential for his security [17]. The setting comprises the key size and the block size needed to ensure the resistance of the system to brute Force attacks [18][19].

Two major factors increase the resistance to this type of attack:

- The size of the key, which must be as high as possible [27].
- The representative sequence of the key which must be undistinguishable from a true random output by a third person [19].

**Key Length:**

The size of this key depends on the number of different characters of the plaintext to be encrypted using the following relation: 

$$(8*n)^*4$$

with $n$ being the number of different characters of the plaintext.

An experimental study is performed on several plaintexts of different sizes and from different sources. Table I and Fig. 2
and 3 show the progression of this key depending on the size of the text to be encrypted.

### TABLE I. DEPENDENCE BETWEEN THE KEY AND THE SIZE OF THE TEXT

<table>
<thead>
<tr>
<th>Size of the message (characters)</th>
<th>Size of the message (bits)</th>
<th>Different characters</th>
<th>the key size</th>
</tr>
</thead>
<tbody>
<tr>
<td>642</td>
<td>5136</td>
<td>40</td>
<td>1280</td>
</tr>
<tr>
<td>864</td>
<td>6912</td>
<td>31</td>
<td>992</td>
</tr>
<tr>
<td>1204</td>
<td>9632</td>
<td>52</td>
<td>1664</td>
</tr>
<tr>
<td>1516</td>
<td>12128</td>
<td>41</td>
<td>1312</td>
</tr>
<tr>
<td>2893</td>
<td>23144</td>
<td>55</td>
<td>1760</td>
</tr>
<tr>
<td>4543</td>
<td>36344</td>
<td>66</td>
<td>2112</td>
</tr>
<tr>
<td>5514</td>
<td>44112</td>
<td>72</td>
<td>2304</td>
</tr>
<tr>
<td>6097</td>
<td>48776</td>
<td>80</td>
<td>2560</td>
</tr>
<tr>
<td>6181</td>
<td>49448</td>
<td>110</td>
<td>3520</td>
</tr>
<tr>
<td>5514</td>
<td>44112</td>
<td>72</td>
<td>2304</td>
</tr>
<tr>
<td>9162</td>
<td>73296</td>
<td>82</td>
<td>2624</td>
</tr>
<tr>
<td>14250</td>
<td>114000</td>
<td>83</td>
<td>2656</td>
</tr>
<tr>
<td>20531</td>
<td>164248</td>
<td>85</td>
<td>2720</td>
</tr>
<tr>
<td>23396</td>
<td>187168</td>
<td>100</td>
<td>3200</td>
</tr>
<tr>
<td>24280</td>
<td>194240</td>
<td>91</td>
<td>2912</td>
</tr>
</tbody>
</table>

### Key generation:

Following this experimental study, we can see that the increase in the size of the key is very small (constant for the larger plaintexts) relative to the size of the message to be encrypted. These makes sense because having a larger text does not imply necessarily that it contains more characters than a small text. As a result, the average size of the key in our case is 2231 bits. However, in theory, the maximum size that can be reached is 8192 bits if we consider that the text to be encrypted contains all 256 possible characters. We can then say that the ASEC key size ensures resistance against brute force attacks and offers long term security.

#### Key generation:

The security of ASEC is not only related to the size of its key but to other strong points which lie primarily in the way in which it is generated, namely:

- It does not use any key generation system.
- The key is automatically generated by the system.
- It is built through a non-deterministic algorithm [23].
- It uses several probabilistic mechanisms that rely on random choices to decide the optimal solution [25].
- Its size is variant.

### Session key:

Each plaintext encrypted by the ASEC system has one and only one key which depends on its structure, its size and its nature. A change in one of these criteria gives birth to a new key. As a result, the same plaintext can lead to two different ciphertexts and this is achieved by changing the initial population based on the evolutionary algorithm [26].

Having a session key in our system allows extending authentication across the communication medium and preventing different attacks seeking to know the key [19]. Indeed, finding the key won't be very useful because it will be only used in the current transaction.

### B. Algorithm Performance

#### Complexity:

The principle of evolutionary cryptographic algorithm is based on the idea of creating equiprobable partitions whose size is almost the same. This introduces the partition problem which is a difficult problem to solve. Normally, this kind of design is used in asymmetric ciphers. It Increases the complexity of solving this encryption exponentially. This makes the cryptosystem much more resistant to different types of attacks.

#### Avalanche Test:

To test the randomness of ASEC ciphering result, we are applied hamming distance between the input and output messages as in [20][21]. For each message Mi, We execute the ASEC Encrypting Algorithm with different Key bits changed. Then, we calculate the average of Hamming distance value between the message and all his Cipher text.

In fact, the Hamming distance of the cipher obtained should be a half of the output size (see Fig. 4).
We conclude that the Hamming distance of the cipher obtained converge to the half of the output size average (Hamming(H(x),H(y)) ≈ n/2).

This result prove the randomly of output cipher.

**Statistical test: Diehard**

In order to bypass statistical attacks, we applied various statistical tests included in the DIEHARD package [28]. This platform offers all verifying tools of the different statically tests to demonstrate the strongest and efficacy of bit sequence ciphering generated by our system. It’s also checking the randomness up to an extreme level.

Table II shows the result given by execution of DIEHARD tests on file that contains all ASEC ciphers message:

**TABLE II. EXECUTION OF DIEHARD TESTS**

<table>
<thead>
<tr>
<th>Test</th>
<th>P-value</th>
<th>Interpretation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Diehard birthdays</td>
<td>0.50646335</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard operm5</td>
<td>0.50241355</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_rank_32x32</td>
<td>0.65910598</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_rank_6x8</td>
<td>0.73626155</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_bitstream</td>
<td>0.75444424</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_opso</td>
<td>0.84942117</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_opso</td>
<td>0.59592027</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_dna</td>
<td>0.09103884</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard_count_1s_str</td>
<td>0.94765782</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard count1s byt</td>
<td>0.77998540</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard parking lot</td>
<td>0.6971967</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard 2d sphere</td>
<td>0.03413893</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard 3d sphere</td>
<td>0.09723242</td>
<td>PASSED</td>
</tr>
<tr>
<td>Diehard squeeze</td>
<td>0.28015448</td>
<td>PASSED</td>
</tr>
</tbody>
</table>

**VI. CONCLUSION AND PERSPECTIVES**

The most common obstacles for the exchange of the keys is their generation and their transmission. In this work, we are designed a new Hybrid Cryptosystems that we called Hybrid Evolutionary Cryptosystems because it uses the Symmetric Evolutionary Ciphering ASEC. As we are shows and experiment it in this paper, the robustness of this system is lies to several factors that can be reduced as following:

- **Key size**: the key is so large and is sufficiently secure.
- **No blocks**: No need to split the message into blocks, the encryption and decryption are not based on the entire message. It can be likened to encryption algorithms block, where the block has a large dimension equal to its size, which increases its level of security and it allows also to avoid the propagation of errors likely by sending block by block.
- **Random secret key generation**.
- **ASCII coding can be used and offer the compatibility with ASCII systems**.
- **The statistical attacks tests are conclusive because of the randomness of bit sequence ciphering generated by this system**.

**REFERENCES**


[19] Agence nationale de la sécurité des systèmes d’information, Référentiel Général de Sécurité, version 2.0 : Choix et dimensionnement des mécanismes cryptographiques


