A Robust Scheme to Improving Security of Data using Graph Theory

Khalid Bekkaoui[1], Soumia Ziti[2], Fouzia Omary[3]
Intelligent Processing and Security of Systems (IPSS)
Faculty of Sciences, Mohammed V University
Rabat, Morocco

***Abstract*—With the incredible growth of using internet and other new telecommunication technologies, cryptography has become an absolute necessity for securing communications between two or more entities, particularly in the case of transferring confidential data. In the literature, many encryption systems have been proposed against attack threats. These schemes should normally overcome the concerns by ensuring confidentiality, integrity and authenticity of transmitted data. However, several of them have shown weaknesses in terms of security and complexity. Hence the need for a robust and powerful non-standard encryption algorithm to prevent any traditional opportunity to sniff data. In this work, we propose a new encryption system that perfectly meets the security requirements. The scheme is based essentially on the principles of graph theory which are very promising at plain text representations. Our approach proposes another use of the concept of Hamiltonian circuit and adjacency matrix using a shared key and a pseudo-random generator. After analysis of the experimental results, which were very promising, the technique was found to be both efficient and robust.**

***Keywords*—Cryptography; encryption; security; graph theory; Hamiltonian circuit; adjacency matrix**

I. Introduction

Cryptography is a branch of cryptology that relies on a set of techniques and methods that transform a clear or readable message into a completely unintelligible one. This discipline deals with several security issues such as the confidentiality of transmissions through unsecured channels, the privacy of individuals, the archiving of data on unsecured media, and so on. Cryptography thus allows the study and analysis of data encryption systems aimed at minimizing the reach of hackers and limiting, as much as possible, their unauthorized access to such data, while preserving the key concepts of information security that are confidentiality, integrity, authentication and finally non-repudiation [1]. The purpose of cryptography is therefore the construction of protection schemas that provide ironclad assurances of confidentiality or authenticity of transmitted messages when dealing with malicious attempts to access them.

Confidentiality is an essential aspect of security. It can be guaranteed via an encryption mechanism, through which data becomes unintelligible to any unauthorized party attempting to access it. The role of encryption algorithms is to transform a clear message into an encrypted one so that only authorized people can retrieve the message in its original, clear form by performing the reverse-process to encryption, namely decryption. By design, decryption should be made as difficult as possible to any unqualified, unauthorized party attempting to carry it out.

Over the years, cryptography has steadily evolved and gradually become indispensable to modern society. Any and all contributions to this field of work have always been of great interest. According to the literature, there are three types of cryptography. The first is symmetric-key cryptography, which involves using a secret key, such as DES [2], IDES [3], AES [4], or others for the purposes of encryption. The second is asymmetric cryptography, which is based on the use of two different keys (one private and the other public) [1], such as RSA [5], ElGamal [6], Diffie-Hellman [7], and so on. The third and last type is what's called hybrid cryptography, and it combines the previous two encryption methods.

Today, modern cryptology is able to make use of a considerable set of mathematical tools. This has led to greater gains in performance and efficiency. Graph theory in particular is an area that is seen as being potentially promising in this respect, as it introduces concepts that might help solve problems in all areas related to networks.

Graph theory in mathematics and computer science is the study of graphs. Generally, a graph can be used to represent the structures and connections that form a complex set while using clear representations of the elements and expressing the relationships between them in a more tangible way, that is by defining communication networks for instance, as well as road networks, electrical circuits, and so on. Graphs therefore offer a way of thinking that allows for the modeling of a wide variety of problems using edges and vertices to represent them. The Seven Bridges of Königsberg (1736) [8] is a mathematical problem known for having laid the foundations for graph theory. Since the beginning of the 20th century, it has developed into a full-fledged branch of mathematics, thanks in part to the work of König, Menger, Cayley, Berge, and Erdös. (References must be added)

Graph theory has become a key element in many applications within computer science. It's a relatively recent concept that has been successfully integrated and has allowed for the development of more powerful encryption algorithms that have proven difficult to crack even for modern software solutions. This is essentially a matter of modeling encryption problems by representing them in graph form so that they become problems in graph theory to which solutions are generally known or more accessible. Solutions to graph problems can be relatively easy and efficient (the time it takes to process them computationally can be fairly reasonable given their polynomial dependence on the number of vertices in the graph). The solutions can also be quite difficult (where processing time increases exponentially) in which case a heuristic a practical

problem-solving approach is used to find the optimal solution.

As a relatively new yet quite powerful tool, graph theory is recognized by government agencies and organizations that have a vested interest in security as having made considerable contributions. It is used in the development of various encryption techniques as well as in sophisticated data communications. This has led to the application of the concepts introduced in graph theory in cryptography on a broad scale, seeing as many NP-hard problems stem from this theory.

Given all of the above, there seemed to be a great need for a new encryption system based on graph theory to be developed, that would ensure a high degree of security while requiring relatively simple resource processing. In this paper, an application of the principles related to this theory in the field of cryptography is presented; its aim being the development of a communications scheme that is both efficient and secure. This proposal makes use of disjoint Hamiltonian circuits for the presentation of data, and of the divide-and-conquer paradigm to simplify processing and facilitate encryption.

The remainder of the paper is arranged as follows. Section 2 presents preliminary knowledge. A literature review of related works is explained in Section 3. Section 4 describes the proposed scheme. Experimental results and analyses are detailed in Section 5, and finally, the conclusion is given in Section 6.

II. Preliminary Knowledge

- **Graph** : Graph theory is a branch of applied mathematics. Fundamentally, a graph consists of a set of vertices, and a set of edges, where an edge is something that connects two vertices in the graph. A graph is a pair (V, E), where V is a finite set and E is a binary relation on V. V is called a vertex set whose elements are called vertices. E is a collection of edges, where an edge is a pair (u,v) with u,v in V. Graph $G = (V, E)$ is a collection of V nodes connected by E links [1].
- **Simple graph** : Undirected graph that has no loops (edges connected at both ends to the same vertex) and no more than one edge between any two different vertices.
- **Path** : A path is a simple graph whose vertices can be ordered so that two vertices are adjacent if and only if they are consecutive in the list [1].
- **Undirected Graph** : A graph in which each edge symbolizes an unordered, transitive relationship between two nodes. Such edges are rendered as plain lines or arcs [1].
- **Cycle** : Refers to a chain where the initial and terminal node is the same and that does not use the same link more than once is a cycle.
- **Hamiltonian Path** : A path that visits each vertex exactly once in an undirected graph.
- **Hamiltonian Circuit** : A Hamiltonian cycle (or Hamiltonian circuit) is a Hamiltonian Path such that there is an edge (in graph) from the last vertex to the first vertex of the Hamiltonian Path.
- **Adjacency Matrix** : Given a graph G with n vertices (ordered from v_1 to v_n). The Adjacency Matrix M of size $n \times n$ related to G can be defined by:

$$\begin{cases} M_{ij} = p & \text{if There exists a path from } v_i \text{ to } v_j \\ M_{ij} = 0 & \text{otherwise.} \end{cases} \quad (1)$$

  Where p is the weight of the edge (v_i, v_j).

- **Divide and Conquer** : An algorithmic strategy which is mainly based on dividing an initial problem into several roughly equal sub-problems, and then solve the sub-problems separately before combining their results. This strategy is able to considerably reduce the complexity of mathematical problems that require a lot of processing.
- **Blum Blum Shub (BBS)** : [9] A simple unpredictable pseudo-random number generator that was proposed in 1986, and whose mathematical equation is described as follows:

$$x_{n+1} = x^2 \mod M \quad (2)$$

  where $M = p.q$ is the product of two prime numbers p and q. The security of this generator fully depends on the complexity of factoring M, which means that the two primes must be properly chosen to ensure a certain robustness. BBS is a good choice for many applications, especially those related to cryptography as it can generate unpredictable sequences.

III. Related Works

Nowadays, graph theory has contributed greatly to the development of various encryption techniques. A review of the relevant literature reveals several methods that have been put forward for such purposes.

Yamuna [10] proposed an encryption mechanism using Hamiltonian paths. The data is represented using a Hamiltonian path, and the complete graph is constructed by weighting the remaining vertices to increase the level of security.

Al Etaiwi [11] put forward a new encryption algorithm based on graph theory. His paper presents a new symmetric encryption algorithm that uses the concepts of complete graph and minimum spanning tree to strengthen security.

Yamuna [12] showed that Hamiltonian circuits could be used to represent multiple messages through a single graph and that encryption could be done using time-dependent functions.

Yamuna and al. [13] used musical notation to represent the secret key (musical note) and graph theory properties to generate keys. This approach is based on the Propagating Cipher Block Chaining (PCBC) mode for encrypting binary messages. In 2014, the same authors proposed a PIN-code encryption method in the form of a digraph [14].

In [15], the authors have proposed a graph based modified DES (GMDES) algorithm which is more secure than the classical DES algorithm [2]. The proposed graph is not fully depended on secret key, and for the same plain text it produces

different cipher text using the same secret key which reduces the probability of various attacks.

Agarwal and Uniyal [16] proposed an encryption scheme based on transforming ASCII values into prime numbers using an encryption key of similar size to the clear message. The authors then randomly generated a prime weighted graph by taking into account the prime number weights assigned to each of the edges.

The system that Amounas [17] put forward handles the original data using graph theory and some of its properties. The main concept being the generation of the complete weighted graph. More specifically, this approach offers a new way of labeling the edges of a graph. It subsequently applies the matrix approach based on ECC operations to generate strong encrypted text.

Recent work in the literature includes the technique proposed in [18], where each character of the data has been encrypted into an Euler Graph. they used the Hamiltonian Circuit as key to secure the data.

Selim G. Akl designed in [19] an encryption process to transmit a secure message. The author employed three distinct graphs constructed successively, and based on an unconventional mapping, conjectured to be a trapdoor one-way function, and which is conceived especially for graph structures.

In [20], two graph based public key cryptosystems have been proposed for protecting valuable information. The first method is purely based on matrix properties, and the second is based on graphical codes.

IV. The Proposed Approach

The system put forward in this paper uses the fundamental concepts of graph theory to facilitate the handling of raw data. The basic idea is to generate weighted graphs by using Hamiltonian circuits in a novel way.

The results that have been achieved in this work appear to be promising, especially in terms of complexity and speed when it comes to processing the clear message. The proposed mode of operation is essentially based on the divide-and-conquer design paradigm which involves breaking down an initial problem into smaller sub-problems and then dealing with each sub-problem separately.

We conceive this approach primarily to address complexity concerns, as most existing works represent a clear message using a graph of similar size. Which, in turn, becomes an adjacency matrix used to process and handle data. It automatically follows that in such cases, the larger the matrix, the more complex the linear operations will be.

With that in mind, using the divide-and-conquer design paradigm has allowed us to reduce complexity by dividing the message into smaller blocks instead of processing it in its entirety. Each block is represented by disjoint Hamiltonian circuits to reduce the size of the graph associated with the block.

To describe fully how the new encryption technique actually works, we will illustrate it in two main algorithms (Encryption in Algorithm 1 and Decryption in Algorithm 2). Each algorithm includes some functions that we will define and explain their functioning.

The input of our Graph Encryption algorithm is a clear message with n characters. The function **ASCII_Transfomation** converts each character of message into its ASCII value. It returns an array of integer belonging to the interval [0,255].

The second function **Decomposition_Block** decomposes the array into several k' blocks $BlockSet_{k'}$ using the following formula:

$$n = 25k + r \tag{3}$$

Such that n is the size of the message, r (belonging to the interval $[0, 24)]$) is the remainder of the division of the message by 25, and k its quotient. If the division is accurate then $k' = k$, otherwise $k' = k + 1$.

The third function **Decomposition_Key** generates from KEK (Key Encryption Key) master key of size m = 256, K_i sub-keys as square matrices of order 13 (where $i = 0 \dots k'-1$) which will be used to encrypt each block (Fig. 1 depicts this process).

The generation of sub-keys is carried out in two steps. First, a size m = $13k'$ Key is generated from a parameter m = 256 size KEK master key. Then, k' other sub-keys are generated from that Key as square matrices of order 13. For each $Block_i$, the ASCII value of the first character is used to specify a digit of the KEK master key (using the position that is supposed to fill in the range [0,255]). This digit is normally used to generate the seed S_i of size 13. Then we use the vector S_i to generate the sub-key K_i in the form of square matrices of order 13.

Additionally, each size-25 block (minimum block size) is partitioned into two size 13 and 12 sub blocks respectively. Given that in a complete graph with n vertices, there is (n-1)/2 disjoint Hamiltonian circuits if n is an odd number strictly greater than 3 [21].

A graph of size 13 can contain 6 disjoint Hamiltonian circuits. It follows that we represent each block by a graph of size 13 containing two disjoint Hamiltonian circuits. Moreover, we convert the odd size sub block into an Eulerian cycle using the ASCII values of its characters, thus representing the weights of the edges. Thereafter, we represent the second sub block by using one of the other five Hamiltonian circuits by filling the missing values with the ASCII code of the null character (**Block_Graph**). Then we use the adjacency matrix to represent the resulting graph (**Graph_Matrix**). This representation is very advantageous, not only in terms of the complexity of the processing but also compared to the traditional representation of the message, which would normally take place within a single Eulerian cycle.

A. Encryption Mode

We encrypt each block using CBC (Cipher Block Chaining). In this mode, an 'exclusive OR' (XOR) operation is applied to each $Block_i$ using the preceding block$'$s encryption before encrypting the current block itself using the same

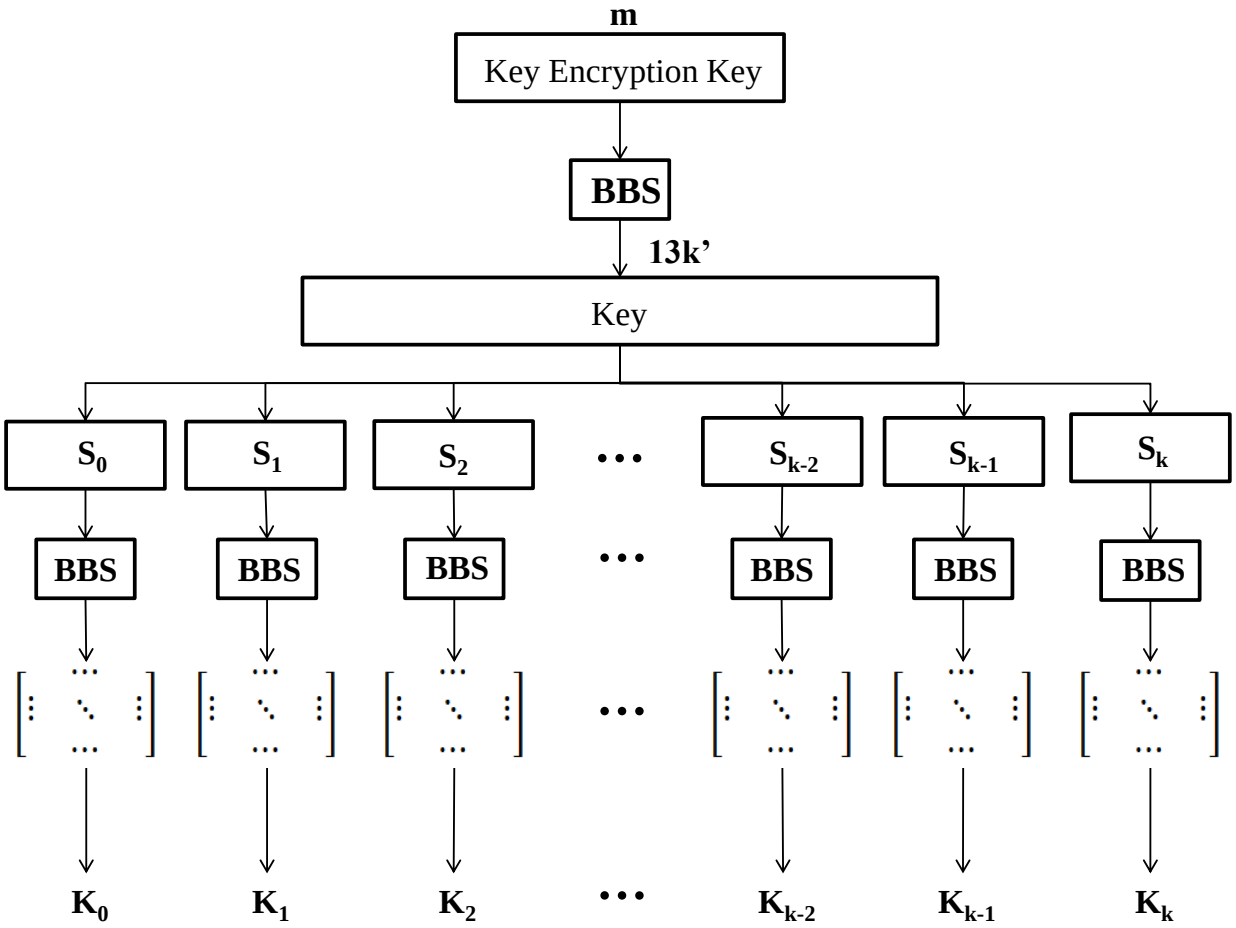

Fig. 1. Key Generator

process. The binary operator XOR concerns in our case the M_i and the C_{i-1} matrices as follows:

$$M'_i = C_{i-1} \oplus M_i \tag{4}$$

The result is then used in another XOR operation with the K_i sub-key produced by the BBS pseudo-random generator to obtain the C_i of the current block,

$$C_i = M'_i \oplus K_i \tag{5}$$

The encryption of the first block (M_0) is performed after passing a randomly generated initialization matrix (IM) through an XOR gate. Each encrypted block will be represented by a vector V_i of size 13^2 by concatenating the lines of the adjacency matrix using the function **ConcatenateLines**. Finally, the resulting vectors are concatenated by the function **ConcatenateVec** resulting a single vector of size $13^2k'$ which will represent the encrypted message EM sent back out, in addition to the vector FCB containing the positions used to generate the Key. Fig. 2 clearly illustrates the encryption mode.

B. Decryption Mode

The input of our Graph Decryption algorithm is a encryption message EM of size m ($13^2k'$). The function **Decomposition_Vector** decomposes the Vector V of size $13^2k'$ into several k' encrypted block $EMatrixSet_{k'}$ To decrypt one message of size m. The number of blocks is calculated as follows:

$$k' = m \div 13^2 \tag{6}$$

We generate also one key of size $13k'$ from the master key using vector FCB. Sub-keys are then generated to decrypt each block $C_i (i = 1 \dots k')$. Using the formula

$$M_i = C_{i-1} \oplus M'_i \tag{7}$$

Where

$$M'_i = C_i \oplus K_i \tag{8}$$

and

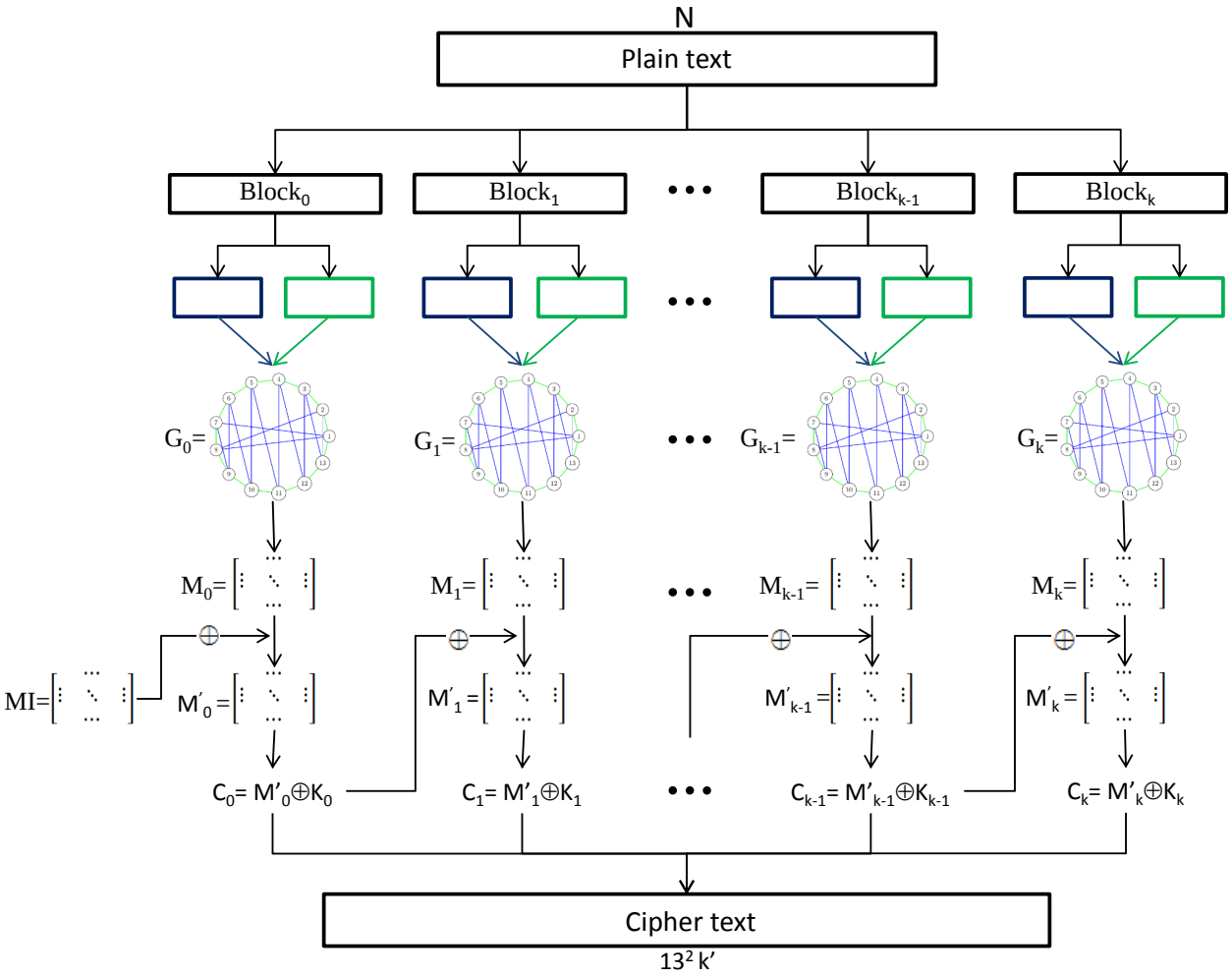

Fig. 2. Encryption Scheme

$$M_0 = IM \oplus M_0' \tag{9}$$

After this stage, each block is decrypted then the disjoint circuits are extracted and the blocks $Block_i$(**Graph_Block**) are reconstructed. Moreover, we convert the ASCII values into characters for each block. Finally, the character blocks are concatenated (**Concatenate_Block**) to form the clear message as shown in Fig. 3.

V. Experimental Results and Analyses

The evaluation of the Encryption technique includes the performance and efficiency of the algorithm, and on the other hand how the scheme can react in terms of robustness against certain attacks such as the Brute-force attack.

A. Statistical Tests

In this part, the DIEHARD test [22] is used to analyze the quality of the random generation of the proposed block cipher. The main purpose of this test is to establish that our algorithm is able to withstand statistical attacks. In other words, a secure block cipher output should be statistically indistinguishable from a random output via the encryption function. For this test to be carried out, a sequence of randomly generated ciphers is first converted into binary to produce a bit-stream larger than 10 MB. Then, this bit-stream is statistically analyzed by subjecting it to the DIEHARD tests. The DIEHARD tests verify the p-value of the randomly generated numbers, where the p-value is in the interval [0.025, 0.975]. The mean values that were obtained are summarized in Table I. Results show that the bit-stream generated using our proposed method has passed all DIEHARD tests. What is more, our encryption system displays satisfactory random and statistically indistinguishable behavior.

B. Brute-Force Attack

Brute-force attacks are a way to find all potential key arrangements using a fast prediction tool. On the assumption that a high-quality machine that takes 10^{-10} seconds to test the validity of each key is used, and that the numbers used in the master key are between 1 and 100, our algorithm has 100^{256} possible keys. A brute-force attack would take about $10^{-10} \times 100^{256}$seconds to obtain the correct key. Thus, a

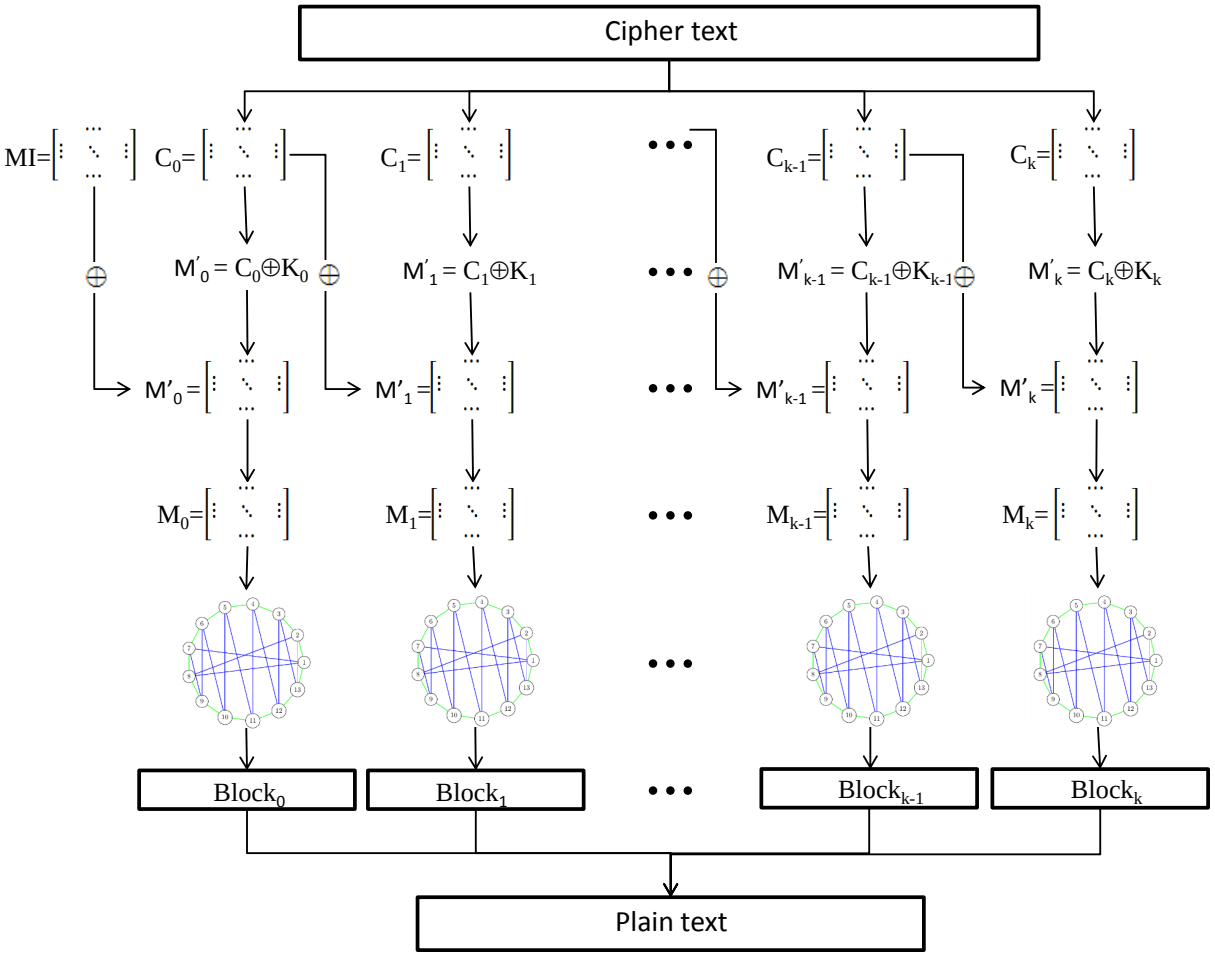

Fig. 3. Decryption Scheme

brute-force attack with an exhaustive key search is impossible within a reasonable timeframe.

To reveal a 25-character message, that is in the case where only one block is used, there are 100 possibilities to determine one of the master key's numbers, which will represent the seed for the BBS generator that's used to generate the vector S_0. That said, the prime numbers used as input parameters for the generator are difficult to determine (because of factorization issues). Therefore, it is almost impossible to find the sub-key if the pq product is sufficiently large.

VI. Conclusion

This paper puts forward a new block-based encryption scheme that utilizes the divide-and-conquer paradigm as well as the fundamental concepts of graph theory to simplify and facilitate processing. Various statistical tests have been carried out to prove that this new algorithm is secure. All of those tests have confirmed that this algorithm resists statistical attacks. Moreover, the BBS-based generator has been used to generate encryption keys for our algorithm, which has further improved key strength. As future work, we aim to design our own pseudo-random generator in order to provide pseudo-keys. We also aim to exploit other graph theory properties for a more robust representation of data.

References

[1] A. J. Menezes, J. Katz, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[2] P. FIPS, "81, des modes of operation," *Issued December*, vol. 2, p. 63, 1980.

[3] W. Meier, "On the security of the idea block cipher," in *Workshop on the Theory and Application of of Cryptographic Techniques*. Springer, 1993, pp. 371–385.

[4] V. Rijmen and J. Daemen, "Advanced encryption standard," *Proceedings of Federal Information Processing Standards Publications, National Institute of Standards and Technology*, pp. 19–22, 2001.

[5] N. P. Smart, "The "naive" rsa algorithm," in *Cryptography Made Simple*. Springer, 2016, pp. 295–311.

[6] ——, "Public key encryption and signature algorithms," in *Cryptography Made Simple*. Springer, 2016, pp. 313–347.

[7] A. J. Menezes, J. Katz, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[8] G. Alexanderson, "About the cover: Euler and königsberg's bridges: A historical view," *Bulletin of the american mathematical society*, vol. 43, no. 4, pp. 567–573, 2006.

input : Clear message of n characters CMC_n, KEK master key, Random square Matrix IM of size 13
output: Encrypted message EM(Vector V of size $169k'$), the vector FCB

begin
  CMA_n=***ASCII_Transfomation***(CMC_n,*n*);
  if *n*%25 == *0* **then**
    k'= $n \div 25$;
  else
    k'= $(n \div 25) + 1$;
  end
  $BlockSet_{k'}$=***Decomposition_Block***(CMA_n,*k'*);
  $K_{k'}$=***Decomposition_Key***(*KEK,k',FCB*);;
  for *element* $Block_i$ *of the set* $BlockSet_{k'}$ **do**
    G_i =**Block_Graph**($Block_i$,13); M_i = **Graph_Matrix**(G_i);;
    if *i==0* **then**
      $M'_0 = IM \oplus M_0$;
    else
      $M'_i = C_{i-1} \oplus M_i$;
    end
    $C_i = M'_i \oplus K_i$;;
    V_i=**Concatenate_Lines**(C_i);
  end
  EM=**Concatenate_Vec**($V_{k'}$);
end

Algorithm 1: Encryption Algorithm

input : Encrypted message EM(Vector V of size $169k'$), KEK master key, the vector FCB
output: Clear message of n characters CMC_n, KEK master key, Random square Matrix IM of size 13

begin
  $k' = m \div 13^2$;
  $EMatrixSet_{k'}$=***Decomposition_Vector***(EM,*k'*);
  $K_{k'}$=***Decomposition_Key***(*KEK,k',FCB*);
  for *element* C_i *of the set* $EMatrixSet_{k'}$ **do**
    $M'_i = C_i \oplus K_i$;
    if *i==0* **then**
      $M_0 = IM \oplus M'_0$;
    else
      $M_i = C_{i-1} \oplus M'_i$;
    end
    G_i=**Matrix_Graph**(M_i);
    $Block_i$= **Graph_Block**(G_i);
  end
  $BlockSet_{k'}$=**Concatenate_Block**($Block_{k'}$,k');
  CMC_n=**Concatenate_Block**($BlockSet_{k'}$,k');
end

Algorithm 2: Decryption Algorithm

TABLE I. DIEHARD TEST

Test name	p-value	Assessment
diehard birthdays	0.26375543	PASSED
diehard operm5	0.37541747	PASSED
diehard rank 32x32	0.95699200	PASSED
diehard rank 6x8	0.09885031	PASSED
diehard bitstream	0.93471890	PASSED
diehard opso	0.73703729	PASSED
diehard oqso	0.17764052	PASSED
diehard dna	0.81870913	PASSED
diehard count 1s str	0.97971001	PASSED
diehard count 1s byt	0.49404053	PASSED
diehard parking lot	0.27155245	PASSED
diehard 2dsphere	0.35355239	PASSED
diehard 3dsphere	0.58482092	PASSED
diehard squeeze	0.91687701	PASSED
diehard sums	0.15611362	PASSED
diehard runs	0.64253136	PASSED
diehard craps	0.63765229	PASSED
marsaglia tsang gcd	0.74963931	PASSED
sts monobit	0.92126172	PASSED
sts runs	0.28893885	PASSED
sts serial	0.50145071	PASSED
rgb bitdist	0.69014502	PASSED
rgb minimum distance	0.57112646	PASSED
rgb permutations	0.59422228	PASSED
rgb lagged sum	0.59927829	PASSED
rgb kstest test	0.10026759	PASSED
dab bytedistrib	0.49551450	PASSED
dab dct	0.08738803	PASSED
dab filltree	0.22157233	PASSED
dab filltree2	0.22430630	PASSED
dab monobit2	0.15458308	PASSED

[9] L. Blum, M. Blum, and M. Shub, "A simple unpredictable pseudo-random number generator," *SIAM Journal on computing*, vol. 15, no. 2, pp. 364–383, 1986.

[10] M. Yamuna, M. Gogia, A. Sikka, and M. J. H. Khan, "Encryption using graph theory and linear algebra," *International Journal of Computer Application. ISSN*, pp. 2250–1797, 2012.

[11] W. M. Al Etaiwi, "Encryption algorithm using graph theory," *Journal of Scientific Research and Reports*, pp. 2519–2527, 2014.

[12] M. Yamuna, k. D. Mukesh, K. Tushar, and R. Tnujoy, "Encryption of multiple messages using hamiltonian circuit and time dependent function," *International Journal of Advanced Scientific Research and Technology*, vol. 3, pp. 2249–6149, 2013.

[13] M. Yamuna, A. Sankar, S. Ravichandran, and V. Harish, "Encryption of a binary string using music notes and graph theory," *International Journal of Engineering and Technology*, vol. 5, no. 3, pp. 2920–2925, 2013.

[14] M. Yamuna, A. Suwathi, and N. Krishnan, "Four digit pin number as a digraph," *International Journal of Computer Application*, no. 4, pp. 100–107, 2014.

[15] D. Sensarma and S. S. Sarma, "Gmdes: A graph based modified data encryption standard algorithm with enhanced security," *Int J Res Eng Technol*, vol. 3, no. 3, pp. 653–60, 2014.

[16] S. Agarwal and A. S. Uniyal, "Prime weighted graph in cryptographic system for secure communication," *International Journal of Pure and Applied Mathematics*, vol. 105, no. 3, pp. 325–338, 2015.

[17] F. AMOUNAS, "An innovative approach for enhancing the security of amazigh text using graph theory based ecc," *International journal of scientific research in science, engineering and technology*, 2016.

[18] P. Amudha, A. C. Sagayaraj, and A. S. Sheela, "An application of graph theory in cryptography," *International Journal of Pure and Applied Mathematics*, vol. 119, no. 13, pp. 375–383, 2018.

[19] S. G. Akl, "The graph is the message: design and analysis of an unconventional cryptographic function," *From Parallel to Emergent Computing, Adamatzky, A. et al., Eds., Taylor & Francis, CRC Press, Boca Raton, Florida*, 2019.

[20] D. Sensarma and S. S. Sarma, "Application of graphs in security," *Inter-*

national Journal of Innovative Technology and Exploring Engineering, 2019.

[21] N. Deo, *Graph theory with applications to engineering and computer science*. Courier Dover Publications, 2017.

[22] G. Marsaglia, "Diehard test suite," *Online: http://www. stat. fsu. edu/pub/diehard*, vol. 8, no. 01, p. 2014, 1998.