Indonesian Journal of Electrical Engineering and Computer Science
Vol. 19, No. 1, July 2020, pp. 248~256
ISSN: 2502-4752, DOI: 10.11591/ijeecs.v19.i1.pp248-256

Fast and robust approach for data security in communication channel using pascal matrix

Oday Kamil Hamid, Riyadh Bassil Abduljabbar, Nazar Jabbar Alhyani
Dijlah University College, Department of Computer Techniques Engineering, Iraq

Article Info

Article history:

Received Jun 21, 2019
Revised Dec 26, 2019
Accepted Jan 27, 2020

Keywords:

Decryption
Encryption
English
Pascal matrix

ABSTRACT

This paper described fast and robust approach of text encryption and decryption based on Pascal matrix. The technique of encryption can be applied on both Arabic and English text. The results shows that the ciphered text unintelligible and rubush for the interuder or hukers. The encryption scheme usually uses a pseudo-random encryption key generated by an algorithm. All this done by using Pascal matrix. Encryption and decryption simulated using MATLAB version 10 and notepad ++to write the input text.

Corresponding Author:

Oday Kamil Hamid,
Department of Computer Techniques Engineering,
Dijlah University College, Iraq, Baghdad.
Email: oday.kamil@duc.edu.iq

1. INTRODUCTION

The exchange data have been advanced in the recent decay, such as e- mail, social mediea and messages between banks and customers…etc. And these informations require protecting from unathurized persons. Therefore, the secure information becomes necessary to protect data through storage or transmition. The technique of transmitting and storing information in particular form that could read and process by only who intended is called cryptography. In same time, the associated methods with scrambling plaintext into cipher text and return back again is called decryption techniques [1]. Figure 1 illustrates this concept of cryptography and decryption [2].

The cryptography meaning if firstly settle and study the technique of massage sending in clandestine methods specifically enciphered shape or disguised shape. Hence, the planned receiver is only having the automation to eliminate the mask of this message and examine it. The cryptography methods has its etymology and krypton from Greek which mean graphing and hidden to write [3]. The plaintext and cipher text is refer to original and disguised message respectively. The encapsulated and sent message is the final form of cryptogram methods and the transform plaintext addicted to code wording process call enciphering or encryption [4]. While, the method of turning the code text addicted to plaintext is call deciphering or decryption technique which is talented by receiver that have familiarity to eliminate camouflage [5]. The cryptography include two types namely symmetric key based on classic algorithms for cryptography using same keys for plaintext encryption and decryption of ciphering wording [6]. These key might be matching or simple transformation among two keys which is in perform stand for the common covert among two or more party which is used to keeping confidential data link [7]. The party's access to the secret key required symmetric key encryption compared with public key of encryption [8] as illustrated in Figure 2 [9].

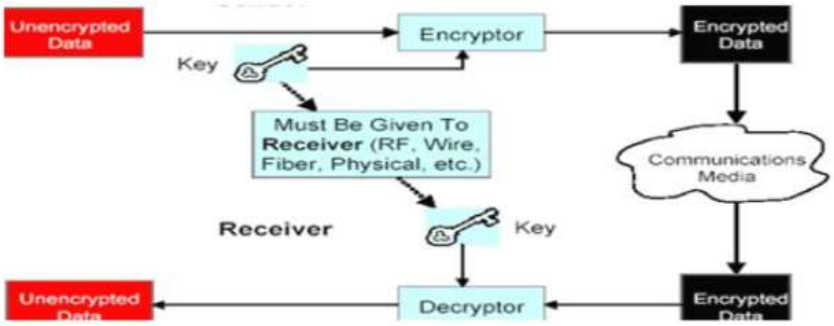

Figure 1. Concept of cryptography

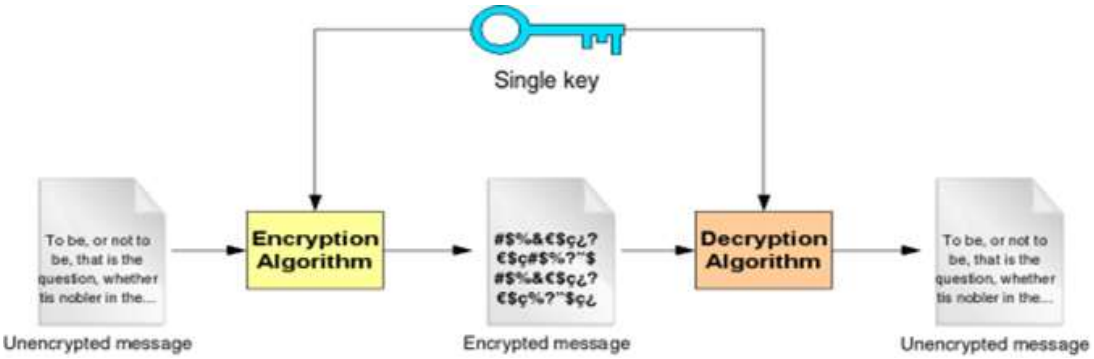

Figure 2. The algorithm of symmetric-key

The cryptographic algorithms required two different keys as public or symmetric cryptography, one is called clandestine or secretive and the others is a community key. Though, the two different pair of keys is liked mathematically. The confidential key is used to decrypt the code text or to produce signature in digital form and the others is community key which is used to encrypt the plaintext or to confirm the digital sign [10]. The asymmetric term stems from different keys to carry out the opposite function every one is the reverse of the other to contrast a conventional symmetric encryption which is realize the same perforation of key [11] as showing in Figure 3 [12]. The data encrypted by the recipient public key could not decrypted without using private matching key in public encryption, while the public key could be used to decrypt information that encrypted by corresponding key [13]. Hence, public key could not used in place of confidential key. In case of lock key is used in public systems, this technique could be used by anyone to drive personal communication to the owner of unlock key. The legal reception which match the private key will make sure when only person able to read the message in the channel. Hence, this will confirm the confidentially of communication between two party as showing in Figure 4 [14].

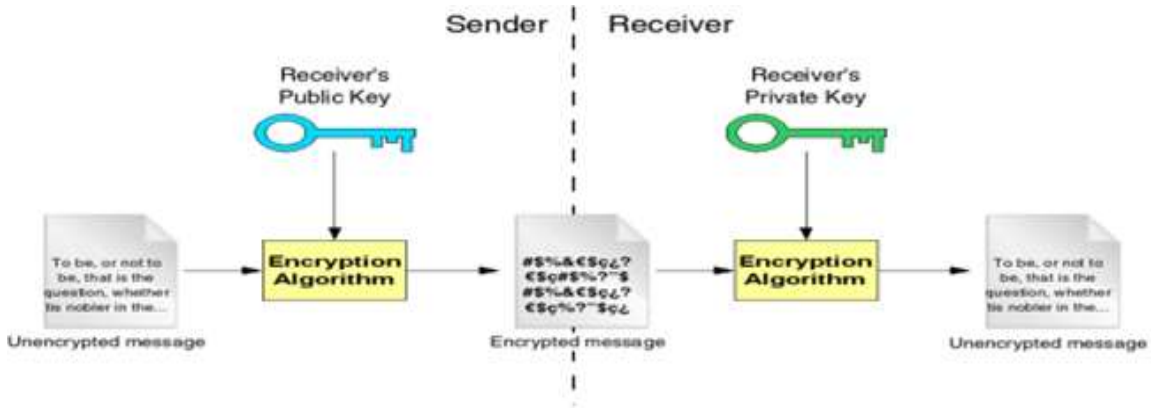

Figure 3. Asymmetric-key algorithm

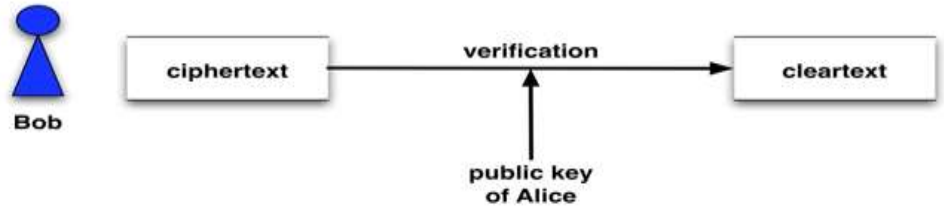

Figure 4. Appear concept of public key

The private keys could be used in data decryption that encrypted by public mating keys in the encryption of public key methods. In same time, the encryption of data by private key could be only decrypted by public matching keys [15]. Though, one could not able to use the private key in place of public key by any way. So, in case of locking key is generated as private keys, the possibility of verifying the documents were locked by other owner in the channel. Due to message encryption by the sender can only opened by public matching key of specific person, then the verification of the sender did hold the private key as illustrated in Figure 5 [16, 17].

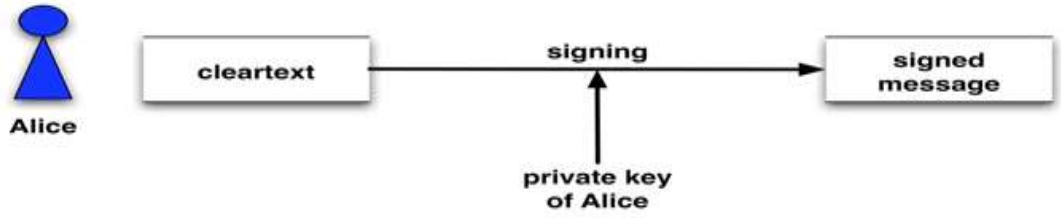

Figure 5. Appear concept of private key

The private and public keys are a couple of keys in cryptography public keys methods. In the case of public locking, the unlock key befall private key and vice versa and this type of key cannot derived by private key [18]. Additionally, if the public keys are the lock keys, this key could be used to launch confidential communication to overcome the protect privacy in the channel. In same time, in case of private keys is the lock keys, the scheme could be used to confirm document that send by the owner of confidential key to avoid the preserve authenticity [19].

2. RELATED WORK

In ciphered text, many techniques have been utilized as described follows; Israa [20] proposed cipher based on modified RSA to increase the complxsity of RSA key by changing three keys. Another proposed [21] for ciphering text and image by three security levels, firstly, compresing and hidden the significant information within compresed data. Then ciphered the compresed data by AES cipher and finally used vedio steganography methods. In order to secure electronic payment via e-banking, Ali [22] suggested secured technique by Hash algorithim. The result shows that the proposed approach is secure against any unatherized persons.

In 2019, Zolidah and Hikma have proposed a cipher based on AES and DES algorithms to encrypt image, text and voice. The proposal based on analyzing various time to DES and AES for several files and each file tested with another five files [23]. In order to improve AES security and reduced time processing, Edjie and Ariel have been proposed AES modification by reducing a round iteration to 6 and added key permutation between them. The experimental result shows that the encryption time improved by 1.27% and decryption time by 1.21% [24]. The paper is ordered as follows; described method in section 3, followed by experimental results in section 4. Finally conclusion and future work are presented in section 5.

3. THE PROPOSAL METHOD

3.1. Pascal's Matrix and Its Inverse

A numbers of system arrange in row resembling a triangle and every row contain a coefficient in the form of $(a + b)^n$ for n = 0, 1, 2, 3, as in [25]. Then, the P Pascal matrix of order n is the real square n×n matrix whose entries are

$$P_{ij} = \binom{i + j + 2}{i - 1} \qquad 1 \le i, j \le n \tag{1}$$

For n = 5

$$P = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 3 & 6 & 10 & 15 \\ 1 & 4 & 10 & 20 & 35 \\ 1 & 5 & 15 & 35 & 70 \end{pmatrix} \tag{2}$$

A Pascal matrix consist of Pascal triangle on its ant diagonals. Pascal matrices are ill-conditioned [26]. However, the inverse of the n×n Pascal matrix is known explicitly. The characteristic polynomial of a Pascal triangle is a reciprocal polynomial .The inverse of pascal's matrix have the same terms of pascal's matrix but with alternative signs. The inverse of a Pascal matrix has integer entries [27].

In this paper the encryption and decryption steps are explained in Figure 6 and 7 respectively. We investigated there is restricted in applied matrix Pascal has dimensions more than 24*24 there were an error ecured in the decryption prossesing. Consequently, an algorithim has been applied a dicision, if the matrix size is greater than 24*24, the size reduced to suitable dimension i.e. less than 24*24 dimensions.

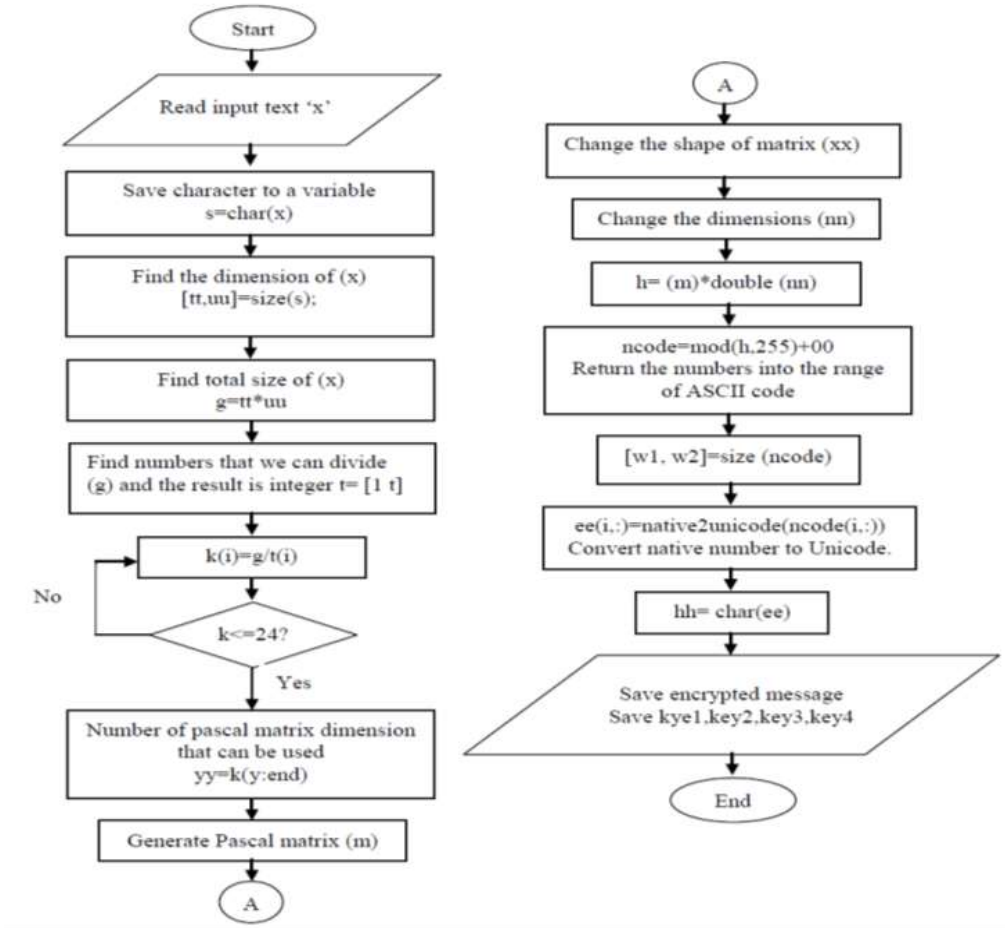

Figure 6. Flow chart for encryption message

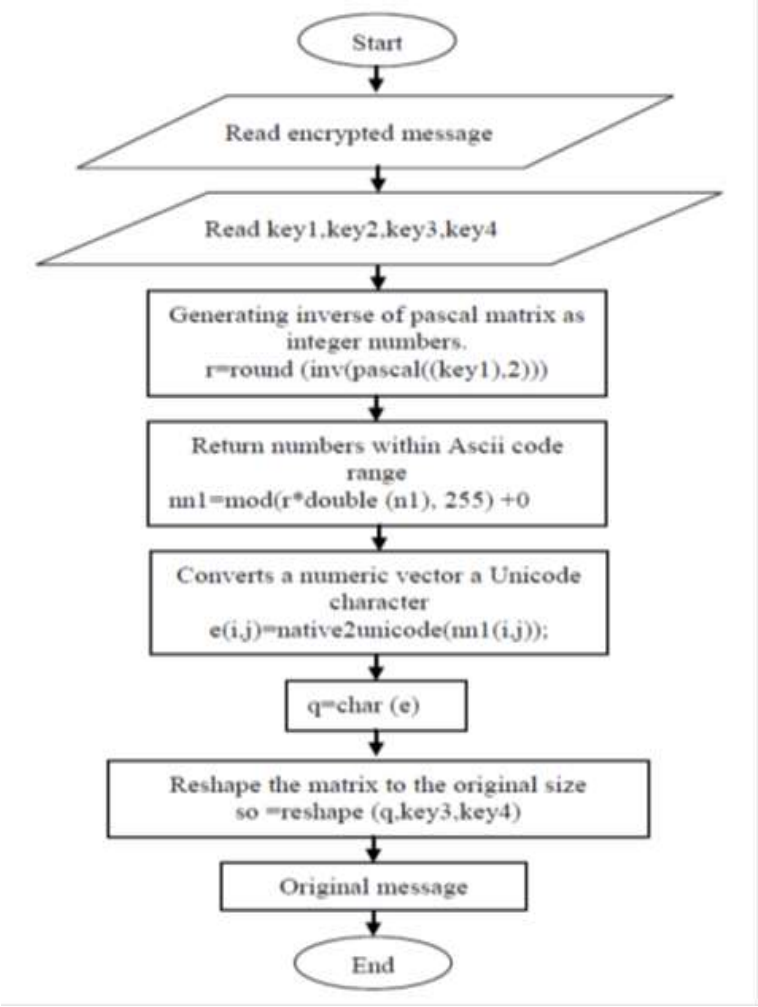

Figure 7. Flow chart for decryption message

4. EXPERIMENTAL RESULTS

4.1. Encryption English Text Procedure

First step is reading the English message that want to be encrypted and decryption this will be done by using Matlab instruction (text read). The message that entered to the program is **"**Data security for plaintext encryption**."**

a. Read the text which saved in the file in the folder of Matlab. Call the file want to encryption and read a white-space or delimiter-separated string and determine the type of file modulation to use it in encryption to calculate spaces and characters and numbers in row and column plus the numbers of characters in the Unicode.
b. Save characters in variable to easy use from Matlab.
c. Find the dimension of original message.
d. Find the size of hole matrix.
e. 5-find the numbers that we can divide hole size of original letter and the result is integer.
f. Finding the position of Pascal matrix rang which it is less than 24.
g. Calculate the number of pascal matrix dimension that can be used.
h. Read size of matrix for Precaution.
i. Returns a transposed and permuted version of for this case, P is a cube root of the identity matrix. Answer of this instruction was shown in Table 1.
j. Change the shape of matrix mean for each matrix has specific shape in the matrix did not under to any rule. The output for this step is shown below:

68	32	110	120
97	102	100	116
116	111	32	32
97	114	101	101
32	32	110	110
115	97	103	99
101	114	108	114
99	97	105	121
117	98	115	112
114	105	104	116
105	99	32	105
116	32	116	111
121	97	101	110

Table 1. Pascal matrix of english encryption

1	1	1	1	1	1	1	1	1	1	1	1	1
-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0
66	55	45	36	28	21	15	10	6	3	1	0	0
-220	-165	-120	-84	-56	-35	-20	-10	-4	-1	0	0	0
495	330	210	126	70	35	15	5	1	0	0	0	0
-792	-462	-252	-126	-56	-21	-6	-1	0	0	0	0	0
924	462	210	84	28	7	1	0	0	0	0	0	0
-792	-330	-120	-36	-8	-1	0	0	0	0	0	0	0
495	165	45	9	1	0	0	0	0	0	0	0	0
-220	-55	-10	-1	0	0	0	0	0	0	0	0	0
66	11	1	0	0	0	0	0	0	0	0	0	0
-12	-1	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0

Generate Pascal matrix depended on largest element inside matrix.

k. Change the dimensions and save it in the new variable.
l. Equal dimension for the last matrix with Pascal matrix in order to make the multiply process. The output for this step is shown below:

1298	1130	1237	1367
-7214	-6683	-7375	-7924
25500	23436	26783	28580
-62442	-55700	-66563	-71543
110644	95102	120461	130718
-145036	-119414	-163186	-178466
141956	111267	166953	183563
-103649	-76781	-128579	-141775
55790	38723	73409	80999
-21552	-13874	-30121	-33201
5671	3345	8392	9228
-913	-486	-1420	-1556
68	32	110	120

m. Return the number into the range of ASCII code. The output for this step is shown below:

23	110	217	92
181	202	20	236
0	231	8	20
33	145	247	112
229	242	101	158
59	181	14	34
176	87	183	218
136	229	196	5
200	218	224	164
123	151	224	204
61	30	232	48
107	24	110	229
68	32	110	120

n. Calculate the size of matrix after (Mode) order.
o. Convert native number to Unicode. The output for this step is shown below:

nظ\ µتى !‘÷p ٍٍe ;µ " °W·ع ˆهؤ بعà¤ {—àج =-è0 knه D nx

p. Reserve the last matrix in new variable in order to deal it with MATLAB. The output for this step is shown below:

« گ h ث X W à (R • پ – ث ں ل W و Q ه

4.2. Decryption of English Text

a. Read the contents of encrypted message in the form of a matrix and save in the new variable. The output for this step is shown below:

23	110	217	92
181	202	20	236
0	231	8	20
33	145	247	112
229	242	101	158
59	181	14	34
176	87	183	218
136	229	196	5
200	218	224	164
123	151	224	204
61	30	232	48
107	24	110	229
68	32	110	120

b. Read the contents of (‘key1’) in the form of a matrix and save in the key1which it represent the number of column of pascal matrix that must be used.
c. Read the contents of (‘key2’) in the form of a matrix and save in the key2which it represent the number of rows of pascal matrix that must be used..
d. Read the number of rows for the original text in the form of a matrix and save in the key3.
e. Read the number of characters in the original text save in the key4.
f. Save the matrix from step 1 in a new variable
g. Generating inverse of pascal matrix as largest element has been stored in the key1 and approximated into integer numbers. The output for this step is shown in Table 2.

Table 2. Inverse pascal’s matrix

0	0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	0	-1	-12
0	0	0	0	0	0	0	0	0	0	1	11	66
0	0	0	0	0	0	0	0	0	-1	-10	-55	-220
0	0	0	0	0	0	0	0	1	9	45	165	495
0	0	0	0	0	0	0	-1	-8	-36	-120	-330	-792
0	0	0	0	0	0	1	7	28	84	210	462	924
0	0	0	0	0	-1	-6	-21	-56	-126	-252	-462	-792
0	0	0	0	1	5	15	35	70	126	210	330	495
0	0	0	-1	-4	-10	-20	-35	-56	-84	-120	-165	-220
0	0	1	3	6	10	15	21	28	36	45	55	66
0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	-12
1	1	1	1	1	1	1	1	1	1	1	1	1

h. Change dimensions matrix from step 6 into new variable.
i. Return numbers for the matrix in step 8 within Ascii code range. The output for this step is shown below:

68	32	110	120
97	102	100	116
116	111	32	32
97	114	101	101
32	32	110	110
115	97	103	99
101	114	108	114
99	97	105	121
117	98	115	112
114	105	104	116
105	99	32	105
116	32	116	111
121	97	101	110

j. Converts a numeric vector for the previos matrix, bytes, from the user default encoding to a Unicode character representation. native2unicode treats bytes as a vector of 8-bit bytes, and each value must be in the range [0,255].

k. Convert the matrix to a character array using the char function in order to deal it with MATLAB program. The output for this step is shown below:

D nx afdt to aree nn sagc erlr caiy ubsp riht ici t to yaen

l. Re shape the generated matrix to the original size of plaintext matrix depending on key3 and key 4

"Data security for plaintext encryption"

5. CONCLUSION AND FUTURE WORK

The encryption and decryption system has been very fast (which mean plaintext contain from 52,640 characters , 8,960 words and 3,000 lines take 28.85 seconds to encrypted it and 1.25 seconds to decrypted) and the usage MATLAB instruction codes make building algorithm simple. The size of input text can up to be very large and the encryption is very strong and none could understand it. The decrypted message form is exactly the same shape of original message even there is white spaces. We investigated there is a limitation in size of Pascal matrix when the dimentions exeeds 24*24. In the future, various extentions can be made, firstly applied the ciphering approach to image and video encryption secondly develop a pascal matrix to make it more suitable for large size matrix.

REFERENCES

[1] M. Frustaci, et al., "Evaluating Critical Security Issues of the IoT World: Present andFuture Challenges," *IEEE Internet Things J.*, vol. 5, pp. 2483-2495, 2018.
[2] Kai M., et al., "How Usable are Rust Cryptography APIs?" *IEEE*, pp. 143-154, 2018.
[3] K. Kurosawa, "*Kurosawa-Desmedt key encapsulation mechanism, revisited*," International Conference on Cryptology in Africa, Springer International Publishing, pp. 51-68, 2014.
[4] A. Kiayias, et al., "*End-to-end verifiable elections in the standard model*," Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer Berlin Heidelberg, pp. 468–498. 2015.
[5] C. M. Chen, et al., "A scalable transitive human-verifiable authentication protocol for mobile devices. Information Forensics and Security," *IEEE Transactions on*, vol. 8, no. 8, pp. 1318-1330, 2013.
[6] A. G. Forte, et al., "*Eyedecrypt—private interactions in plain sight*," in International Conference on Security and Cryptography for Networks, pp. 255-276, 2014.
[7] D. Lateiner, "Signifying Names and Other Ominous Accidental Utterances in Classical Historiography," *Greek, Roman, and Byzantine Studies*, vol. 45, no. 1, pp. 35-57, 2010.
[8] M. T. Goodrich, et al., "*Loud and clear: Human-verifiable authentication based on audio*," in Distributed Computing Systems, 2006 (ICDCS 2006), 26th IEEE International Conference on, pp. 10-10, 2006.
[9] Y. W. Chow, et al., "*A visual one-time password authentication scheme using mobile devices*," in International Conference on Infor-mation and Communications Security, pp. 243-257, 2014.
[10] A. Kanso, et al., "Keyed hash function basedon a chaotic map," *Information Sciences*, vol. 186, no. 1, pp. 249-264, 2012.
[11] M. Y. Valandar, et al., "A new transform domain steganography based on modified logistic chaotic map for color images," *Journal of Information Security and Applications*, vol. 34, pp. 142-151, Jun 2017.
[12] C. L. Chen, et al., "Asecure anonymous e-voting system based on discrete logarithm problem," *Applied Mathematics and Information Sciences*, vol. 8, no. 5, pp. 2571-2578, 2014.

[13] Y. Acar, et al., "*Comparing the Usability of Cryptographic APIs*," in 2017 IEEE Symposium on Security and Privacy (SP), pp. 154-171, May 2017.
[14] W. Neji, et al., "Distributed key generation protocol with a new complaint management strategy," *Security and Communication Networks*, vol. 9, no. 17, pp. 4585-4595, 2016.
[15] A. Beimel, "Secret-Sharing Schemes: A Survey," *Coding and Cryptology*, pp. 11-46, 2011.
[16] B. Lee, et al., "*Providing receipt-freeness in mixnet-based voting protocols*," International Conference on Information Security and Cryptology, Springer Berlin Heidelberg, pp. 245-258, 2003.
[17] Riyadh B., "Fast Approach for Arabic Text Encryption using Genetic Algorithm," *European Journal of Scientific Research*, vol. 144, no. 4, pp. 342-348, 2017.
[18] Riyadh B., "Steganography System using Slantlet," *Journal of Information, Communication, and Intelligence Systems (JICIS)*, vol. 2, no. 1, Feb 2016.
[19] S. E. Haneen A., et al., "*New Text Encryption Method Based on Hidden Encrypted Symmetric Key*," ACIT 2018, Ceske Budejovice, Czech Republic, 2018.
[20] S. A. Israa B, et al., "Modified RSA-based algorithm: a double secure approach," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 6, pp. 2818-2825, Dec 2019.
[21] Dwi A. and Endi S., "Optimization of video steganography with additional compression and encryption," *TELKOMNIKA Telecommunication Computing Electronics and Control*, vol. 17, no. 3, pp. 1417-1424, Jun 2019.
[22] Ali F., et al., "Secured e-payment system based on automated authentication data and iterated salted hash algorithm," *IJPEDS*, vol. 11, no. 1, 2019.
[23] Zolidah K, et al., "Time performance analysis of advanced encryption standard and data encryption standard in data security transaction," *IJEECS*, vol. 16, no. 2, pp. 988-994, Nov 2019.
[24] Edjie M., et al., "File encryption based on reduced-round AES with revised round keys and key schedule," *IJEECS*, vol. 16, no. 2, pp. 897-905, Nov 2019.
[25] N. Wafa, et al., "A secure electronic voting protocol with a simple ballot's encryption function," *Int. J. Security and Networks*, vol. 13, no. 1, 2018.
[26] M. Usman, et al., "A Light weight Encryption Algorithm for Secure Internet of Things," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, 2017.
[27] D. Rachmawati, et al., "*Hybrid Cryptosystem Using Tiny Encryption Algorithm and LUC Algorithm*," IOP Conf. Ser. Mater. Sci. Eng., p. 300, 2018.