ABJAD Arabic-Based Encryption

Ahmad H. Al-Omari
Computer Science Dept., Faculty of Science
Northern Border University, KSA

***Abstract*—The researcher introduced an enhanced classical Arabic-based encryption technique that is essentially designed for Arab nations. The new algorithm uses the shared key technique where the Keyword system Modulus is employed to add randomness and confusion to the table of alphabets being used. The results proved that the technique is resistant to brute force and cryptanalysis attacks. The time needed to break the algorithm is huge and the possibilities of decrypting the cipher text using the language frequency and language characteristics are hard and unfeasible. The technique assumes the existence of a secure channel for the keyword exchange.**

Keywords*—*Arabic-based cryptography; classical encryption; Arabic language encryption; shared key; keyword

I. Introduction

Cryptography is an Arab-born science unlike other sciences like mathematics and physics which were translated from their original language founders, developed and then enriched by western scientists [1]. *David Khan*, who is one of the greatest historians in cryptology, stated that cryptology was born in Arabic world [2]. This fact was confirmed in some Arabic cryptologic treatises in 1980 which were found in Istanbul's *Suleymanye* library [3] in addition to the work of other scholars who wrote about cryptography and cryptanalysis in the Arab world [4], [5], [6].

Data protection mechanisms currently use two main algorithmic approaches, symmetric and asymmetric algorithms. Some examples are the AES and RSA algorithms that have proven their strength and practical use over many years. The development in the field of quantum computing has brought a serious threat to the current state-of-the-art cryptology systems [7]. However, some cryptographic asymmetric systems such as RSA with a four-thousand-bit key are believed not to resist attacks by large quantum computers whereas cryptographic symmetric systems, such as AES-256 bits, can resist attacks by large quantum computers. For instance, to break a single AES encryption, an exhaustive search would take $2^{256} > 10^{75}$ steps requiring billions of years with state-of-the-art ultra-massive computing resources [8] [9]. Therefore, researchers have started to explore new encryption methods that are safe in classical computers as well as quantum computers. Some algorithms said to be post-quantum cryptography that remain secure with the assumption that the attacker has a large quantum computing power [10].

A. Research Problem

Recently, Arab communities encounter a real need for Arabic-Based cryptographic algorithm to be used as a second alternative technique in addition to the available encryption standards in the market. Thus, this research comes to bridge the gap that the Arab communities need. It is worth mentioning that such encryption algorithm will be used solely in Arab language encryption intercommunication.

B. Research Objectives and Limitations

Arabic language is spoken by more than 350 million native speakers in 23 countries of the Arab world and is used by more than 57 countries of the Islamic world. It is also one of the six official languages of the United Nations [11] [12]. This research grant supports the design of Arabic-Based encryption technique that can be used by governments, institutions, public and private sectors or individuals.

The research work aims to achieve the following objectives:

1) developing an Arabic-based encryption technique that is fast, cheap, secure and suitable for Arab communities

2) encouraging Arab researchers to employ modern technology in the service of Arabic language sciences

3) building cryptographic algorithms that use pure Arabic letters

The project is restricted to the following criteria:

1) It assumes the existence of secure Quantum Key Distribution (QKD) protocol like (BB84, SARG04, E91 or any other secure Key Distribution (KD) protocol) [9].

2) It is designed for Arabic alphanumeric data format. which is derived from the Arabic coding character set standard (ISO-8859-6).

3) It does not use the Unicode, ASCII, EBCDIC or any other data format or representation.

4) It is targeted for the Arab language users.

5) The non-Arabic character sets are excluded in this phase of the project.

6) It is limited to text message formats only.

II. Related Work

In his paper, *Ibrahim A. Al-Kadit* proves that Arabs are the origins of cryptography. The researcher discussed the factors behind the Arab advancement in cryptology like translation, linguistic studies, administrative studies, public literacy and the advanced mathematics. The researcher briefly listed some of the famous Arab scientists who have contributed to cryptology as *AL-Khalil*, *Jabir ibn Hayyan (Geber)*, *Thoban al-Masry*, *Al-Kindi*, *Ibn Wahshiyya*, *Mohammad ibn Ahmad ibn Tabataba*, *As'sa ibn Muhadhdhab ibn Mammati, Ibn*

This work is sponsored by the Deanship of Scientific Research at Northern Border University, Arar K.S.A.

Adlan, Ibn Dunainair, Ibn ad-Durihimi, Ali ibn Mohammad ibn Aidamur al-Jaldki and Al-Qalqashandi. The researcher proved that the word encryption have developed from Arab literacy; the word "cipher" means concealment of clear meaning of messages or simply encryption. The Arabic word "*sifr*" stands for the digit "zero" (0). Then it was transformed into European technical terms that mean encryption and which was later converted from "sifr" in Arabic to "cipher" in Latin cipher [13].

Yahya Alqahtani, Prakash Kuppuswamy, Sikandhar Shah have proposed a modified version of the *Vigenère* cipher based on Arabic alphabets. The original *Vigenère* cipher is a method of encrypting alphabetic text by using a series of different *Caesar* ciphers based on the letters of a keyword [14]. The modified version of the *Vigenère* cipher works by adding a keyword repeatedly into the plaintext. The alphabets consist of 28 characters of Arabic letters, 1 blank character and 10 characters for the numbers. So the total number of the alphabets is 39 characters. The addition is carried out using the system modulo 39. That is to say, if the result is greater than 39, we subtract as many multiples of 39 as needed to bring us into the range (0 . . . 38). The above mentioned researchers claimed that they have a better secure algorithm than the original one using the Arabic alphabets and that their work is a milestone in Arabic language secure communication [15].

Haifaa Abdul-Zahra Atee has proposed a new cryptographic algorithm based on Arabic letters. The researcher demonstrated an encryption/ decryption example but she did not provide it with afterwards investigation regardding the strength of the algorithm and did not compare the results with any known classical algorithm [5].

In their work "Hybrid combination of Message Encryption Techniques on Arabic Text", *Mohammed Abdullah Aysan* and *Prakash Kuppuswamy* have adopted the *Caesar* cipher approach to Arabic letters after adding the 28 Arabic alphabet characters in addition to the 10 decimal numerals. Then they proposed generating two keys; the first key is based on a synthetic specific value for each Arabic letter from (0, 1… 38), whereas the second key is the logarithmic value of the generated key (X), ($\log_3(X)$). The researchers have argued that their algorithm is simple, fast and has the advantage of using standard methods. Besides, it consumes less processing time and capacity [16].

On the other hand, *Prakash Kuppuswamy, Yahya Alqahtani* have proposed another symmetric encryption technique that is based on Arabic alphabets. Likewise, the initial key is randomly selected and its inverse is calculated. Then another negative number is selected and its inverse is calculated again before generating the cipher text. The decryption is carried out using the reverse order process. The researchers have argued that their work presents more secure algorithms than being used by similar classical encryptions [4].

None of the related works can be adopted by Arab communities because they are either weak or not well designed. The proposed algorithm in this work might be the outset after adding further enhancements and testing to the algorithm to be strong enough and attack resistance.

III. The Proposed Solution

In this work, we proposed an enhanced classical symmetric encryption algorithm that is based on an old encryption technique invented by al-Kindī who was known as "the Philosopher of the Arab world" [3]. The technique is similar to the Porta Cipher but with a modern renovation [17]. The proposed encryption technique is an enhanced version inspired by some techniques as Playfair, al-Kindy, Caeser and Porta ciphers [18].

Arabic encoding is similar to any other language alphabetic scripts. For instance, the Unicode standard is used for encoding a raw text not as a glyph list. The Unicode Standard specifies an algorithm for the presentation of the text with a bidirectional behavior i.e. Arabic and English [19]. In our project, we do not use the known standards as Unicode, ASCII or EBCDIC data representations but rather we use the Arabic alphanumeric data representation [20].

Arabic letters have many characteristics. For example, it has 28 characters, it has no upper or lower case characters, it views some of the two-character pairs as a single character and it is read and written from right to left. Moreover, some shapes of Arabic Letters change depending on the context; some Arabic letters may have up to four shapes depending on the position of the letter in the word, its predecessor and its successor. Arabic Letters also have an isolated shape, a connected shape, a left-connected shape and a right connected shape.

TABLE I. Forms of Order

0	Weight	ABJAD	Normal
1	0	أ	أ
2	1	ب	ب
3	2	ج	ت
4	3	د	ث
5	4	ه	ج
6	5	و	ح
7	6	ز	خ
8	7	ح	د
9	8	ط	ذ
10	9	ي	ر
11	10	ك	ز
12	11	ل	س
13	12	م	ش
14	13	ن	ص
15	14	س	ض
16	15	ع	ط
17	16	ف	ظ
18	17	ص	ع
19	18	ق	غ
20	19	ر	ف
21	20	ش	ق
22	21	ت	ك
23	22	ث	ل
24	23	خ	م
25	24	ذ	ن
26	25	ض	ه
27	26	ط	و
28	27	غ	ي

Furthermore, Arabic has several diacritics (small vowels) that can be written above or beneath each letter. The use of diacritics is determined by the grammatical state of the word and eventually the meaning of the statement changes accordingly [19] [21]. However, in our research, we will not consider diacritics or language grammar.

In addition to the normal order of Arabic alphabet (as used in dictionaries), Arabic has another order known as “ABJAD” pronounced /ˈæbdʒɑːd/ [22] [23]. The two alphabetical orders are shown Table 1 Forms of order.

The table 1 (forms of order) is read from right to left. The first row (serial) shows the alphabets order, the second row (weight), represents a given numeric equivalent to each letter, the third row (ABJAD) represent the ABJAD order of the Arabic letters and the last row (normal) represents the normal alphabets order.

A. The Solution Description

The Arabic coding of character set (ISO-8859-6) [24] is used to create a modified synthetic table composed of 75 characters that represent the standard alphabets, numbers and special characters. Table 2"Modified Arabic (ISO-8859-6)", represents a matrix *M* of (5x15) rows and columns. The matrix *M(mi,j)* contains 75 characters where *mi,j* represents the ith and jth character of the matrix (M). So ((1<= i <=5) and (1<=j< 15)), as defined in Equation (1).

$$\mathrm{M} = [i-j]_{5X15} = \begin{bmatrix} m1{,}1 & \cdots & m1{,}15 \\ \vdots & \ddots & \vdots \\ m5{,}1 & \cdots & m5{,}15 \end{bmatrix} \qquad (1)$$

The matrix M(mi,j) is the initial matrix and is reconstructed by distributing its content (characters) according to the keyword characters modulus value. The keyword is randomly chosen by the user. Besides, there are no restrictions on the length of the keyword yet it is recommended to be more than 10 characters. The Keyword is used as a shared key between the communicating parties. For the algorithm calculation purposes, a copy of the keyword with non-redundant characters is used. Then the copied Keyword Length (KL) is calculated as defined in Equation 2. The Keyword Position (KP) in Equation 3 determines the insertion position in the table of alphabets. Nevertheless, the insertion includes the Keyword with non-redundant characters followed by the rest of the non-contributing characters in the keyword from Table 2. Using the system modulus 75 adds randomness and confusion to the algorithm and hence will make it hard to brute-force attacks. The keyword insertion process is performed by filling the unique characters of the keyword followed by the rest of Table 2 starting from position *(m1,15),(m1,14)… (m1,1), (m2,15), (m1,14)… (m2,1)…. (m5,15), (m5,14)…. (m5,1)*. Finally, at the end of the table reconstruction process, a new generated matrix (table) that is called the modified *MM(mi,j)* is created for the encryption purposes.

$$KL = \sum(Keyword\ \text{Characters Weights}) \qquad (2)$$

$$KP = KL\ Mod\ 75 \qquad (3)$$

To access the matrix (table), we need two indexes *(r,c)*, the right index *(r)* and the top index *(c)*. The right index *(Row)* has the values *(r1,r2,r3,r4,r5)* and the top index (Column) has the values *(c1,c2,c3,…….c14,c15)*. While the ABJAD alphabets are used to fill in the *(r,c)* pairs, the two indexes *(r,c)* are used to point to the matrix MM elements where *MM(mi,j)* elements are respectively determined by the value of *(r,c)* (i.e. *i=r* and *j=c*).

To determine which character of the ABJAD alphabets is the starting character to fill the *(r,c)* contents, the sum of the (keyword Weights system modulus 28) is used as defined in Equation 4. The resulting value points to the starting character to be inserted in r5. So the next character will be in *r4,r3,r2,r1 ,c15,c14,c13,…….c2,c1*.

$$KP = KL\ Mod\ 28 \qquad (4)$$

Using the keyword system modulus as in Equation (4) assures the randomness in choosing the starting character of the ABJAD alphabets. For each new plaintext character that is going to be encrypted, the ABJAD alphabets will be down-shifted for one character in a circular-round fashion (down-circular-shift). Using the down-circular-shift makes the algorithm more attack-resistant by adding randomness and confusion to the algorithm.

The encryption process is performed in two steps. In the first step, the plain text is divided into distinct characters where each individual character is substituted with the corresponding pairs of characters from the row *(ri)* that is concatenated with column *(cj)* and which are both from *MM(mi,j)* table. The resulting text is a two-character text (S) as defined in Equation (5).

$$\mathrm{S} = MM(mi,j) = \left(\mathrm{ri}\middle|\middle|\mathrm{cj}\right)) \qquad (5)$$

TABLE II. MODIFIED ARABIC (ISO-8859-6)

ض	ص	ش	س	ز	ر	ذ	د	خ	ح	ج	ث	ت	ب	أ
1	0	ي	و	ه	ن	م	ل	ك	ق	ف	غ	ع	ظ	ط
&	%	$	#	"	!		9	8	7	6	5	4	3	2
?	>	=	<	;	:	/	.	-	,	+	*	)	(	'
-	؟	؛	،	~	}	|	{	`	_	^	]	\	[	@

In the second step, the resulting two-character pairs (S) are converted back to one character by substituting the corresponding letter from the original Table 2. The intersection of right index *(r)* and top index *(c)* determines the letter being substituted as defined in Equation 6. Likewise, the whole encryption process is repeated for each plain text character in the same way until the end of the plain text message.

$$(mi,j) = \mathrm{S} = \left(\mathrm{ri}\middle|\middle|\mathrm{cj}\right)) \qquad (6)$$

The decryption process works the same as the aforementioned encryption process but in reverse order.

B. The Algorithm Steps

The whole algorithm is clarified by steps, pseudo code and examples. A detailed explanation is shown with examples in the following section:

The encryption algorithm consists of the following phases:

1) The initialization phase that consists of the following steps:

a) The keyword selection: The keyword selection is the choice of the user and it is recommended to meet the following properties.

i. The Keyword characters should be selected from Table 2.

ii. The *Keyword* length is recommended to be not less than ten-character long and to contain a mixture of characters.

iii. After algorithm calculations, the *Keyword* characters should be unique (i.e. each character appears only once, sans duplicates).

Example: the Keyword “@؟12 بسم الله” becomes “ بسم @؟12الله”.

b) The Keyword calculations: The calculations are performed as in Equations 3 and 4:

i. The *keyword* characters’ weights

Example:

Σ	@	؟	1	2	هـ	ل	أ	sp	م	س	ب
298	60	59	29	30	25	22	0	38	23	11	1

ii. The *keyword* summation modulus 75 is computed as in Equation 3.

Example:

KP= 298 mod 75=73,
where the *keyword* starts

iii. The *keyword* summation modulus 28 is computed as in Equation 4.

Example:

KP= 298 mod 28=23, where the ABJAD
alphabets start from “ص”

c) The table reconstruction: The table reconstruction is built as follows:

i. The *keyword* inside the table starts from the position determined by computing the system modulus 75.

298 mod 75=73, where the *keyword* starts

ii. Filling the tables from the rest of the non-contributing characters in the *keyword* is continued.

d) The indexes reconstruction: In this step, the right (r) and top indexes (j) of the table are reconstructed as follows:

i. The starting letter of the ABJAD alphabets is determined to build the right and top indexes.

298 mod 28=23, where the ABJAD alphabets start

ii. The ABJAD alphabets are written starting from the last r5 then backward until c1.

2) The encryption phase in which the encryption is performed in two rounds:

a) Round-1, one to two characters substitution: Each single character from the plaintext is substituted with two-cipher characters from the first reconstructed table.

b) Round-2, two to one character substitution: The resulted two-cipher characters are substituted with one cipher text character from the second reconstructed table.

In the abovementioned example, the encryption of the plain text “جامعة” is encrypted in two rounds.

i. In the first round, each plaintext character is substituted with two characters and the resulted text is “توثضخذخوضج”.

ii. In the second round, the resulted text is substituted (encrypted) with one ciphertext and the resulted ciphertext is “ستأمج”.

3) The decryption phase in which the decryption is performed in two rounds:

a) Round-1, one to two characters substitution back: Each single character from the cipher text is converted back to two characters from the first reconstructed table.

b) Round-2, two to one character substitution back: The resulted two-cipher characters are converted back to the original plain text characters from the second reconstructed table.

In the decryption process, the ciphertext “ستأمج” is converted back to its original plaintext characters in two rounds:

i. Round-1, in the ciphertext “ستأمج”, each character is converted back to its two-characters equivalent from the second reconstructed table and the result is “توثضخذخوضج”.

ii. Round-2, in the ciphertext “توثضخذخوضج”, each two-character pairs was decrypted back to its original plaintext and the result is “جامعة”.

IV. Discussion and Analysis

The research encryption algorithm is neither classical nor modern; it is better classified as a hybrid approach for it employs mathematics and is inspired by modern encryptions.

The number of the alphabets in Arabic language is more than in English. Thus, the use of 75 characters that are randomly distributed in the modified table MM(mi,j) makes the algorithm better in terms of the attack-resistance than many of the other known modern encryptions.

The algorithm analysis complies with the most common types of attacks like the cryptanalysis and the brute-force attacks. In the brute-force attack, the attacker tries every possible key until an intelligible translation of the ciphertext into plaintext is obtained [18]. The brute-force attack requires that three items should be known by the attacker: the encryption algorithms, the language of the plaintext and the number of the possible keys that could be generated [25].

In our work, to break the key, all the resulted alphabet diagraphs need to be obtained which means that you neeed to choose among the 75 characters minus the keyword length multiplied by the 28 possible characters from the ABJAD alphabets. The mathematical combinational formula (*n choose r*; *C(n,r)*) is the best formula to describe our results. To make it clear, this formula is used when the chosen characters do not need to be repeated and the order does not matter [26]. The formula is also called the Binomial Coefficient as defined in equation (7).

Example: If we choose a keyword of n-character length, the number of possible generated diagraphs (combinations) to break a single letter is calculated according to the following formula:

$$\boldsymbol{Diagraph\ D} = \frac{\boldsymbol{n}!}{\boldsymbol{r}!(\boldsymbol{n}-\boldsymbol{r})!} \quad ..\ (7)$$

Where D is the result, n is the set of characters to choose from, and r is the chosen character.

Example: Suppose the keyword length n=15.

$$\text{Diagraphs} = \frac{75!}{15!(75-15)!} * 28 = 6.38\ \text{X}\ (10)^{16} =$$

63840355228050240 which is more than 63 quadrillion diagraphs.

If we assume a supercomputer that is developed by China's National University of Defense Technology [27] and is with 33.86 Peta-flops (33.86 quadrillion operations/second) that has been used to crack the fifteen-character keyword, then this process will take approximately about $5.9\ \text{X}\ (10)^{37}$ years to break the keyword.

$$\text{Time in Years} = \frac{6.38\ \text{X}\ (10)^{16}}{33.86\ \text{X}\ (10)^{15} * 365 * 24 * 60 * 60} = 5.9\ \text{X}\ (10)^{37}\ \text{Years}$$

In the cryptanalysis attack, using the language characteristics to attack the ciphertext is unfeasible since the encryption algorithm passes through two rounds of encryptions that are previously explained. Hence the attacker will not get benefits of language frequency and language characteristics since the relationship between letters will disappear. For example, the two and three letters that appear together like “ال” or “نيه” will be scrambled and converted to different alphabets in each sub-process. This is also valid for the letters frequency analysis since it is hard for the crypto analysts to get benefits of letters frequency because each letter will not be encrypted to the same cipher-text. That is to say, it will be encrypted to a different cipher-text in each sub-process of the algorithm.

The encryption process guarantees randomness of the table distribution since the same character will be encrypted differently each time. Thus, hackers are not able to get benefit of having two or three combination letters that usually come together since each letter is encrypted separately and independently. By nature, the encryption algorithm disseminates and hides the language characteristics and letters frequency.

We performed some experiment measures to compare our algorithm speed with the well-known algorithms (AES and DES). We used the open source library Crypto++ for C++ programming language on a laptop core i5, 2.5 GHz CPU with operating Windows 7 operating system and we used six different plaintext data size. The collected performance metrics are the encryption and the decryption time.

The encryption speed chart in Table 3 shows that our algorithm speed outperforms the other two-encryption DES and AES algorithms. In the comparison chart (the encryption chart), our algorithm is faster than the DES and the AES in the encryption process especially when the data size gets larger in size.

TABLE III. THE ENCRYPTION TIME

Plaintext Size/KB	DES	AES	Ours
15	3.8	5.07	2.08
30	7.5	17.09	4.2
45	8.5	19.96	5.7
60	8.8	22.91	6.1
75	9.33	29.99	7.4
90	10.7	38.15	8.3

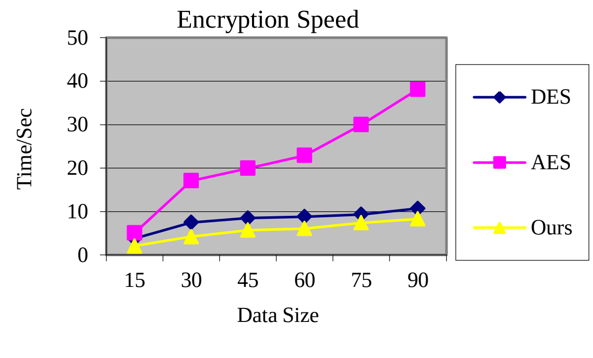

Fig. 1. Encryption Speed.

The Encryption and Decryption speed are shown on Table 3, Table 4, Figure 1 and Figure 2. All the tables and figures show that our algorithm is faster than the DES and AES decryptions especially when the data size grows in size.

TABLE IV. THE DECRYPTION TIME

Plaintext Size/KB	DES	AES	Ours
15	3.6	4.09	2.2
30	4.4	17.04	4.6
45	8.3	19.85	5.9
60	8.9	22.8	6.5
75	9.4	30.3	7.9
90	10.6	38.4	8.6

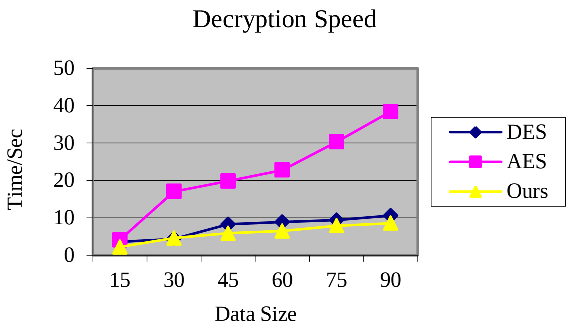

Fig. 2. Decryption Speed.

The proof of concept used in this work aims to confirm that Arabic language can accommodate new technologies especially the encryption which is essentially an Arab-born science. The algorithm uses a shared-key classical encryption technique and gets benefits of mathematics and the spirit of modern encryptions, the fact that assures the flexibility and adaptability of Arabic language and encourages researchers from the Arab world to pay more attention to Arabic-based encryption techniques. The main contribution in this work is designing a new encryption algorithm that is based on the ABJAD-order Arabic alphabets and employing the Modified Arabic (ISO-8859-6) to perform the encryption/decryption processes. The new algorithm is resistant to the brute-force attacks and can relatively perform fast and secure encryption/decryption processes.

V. Conclusion and Future Work

Arabic language has special features that could be positively employed for the benefit of developing cryptographic algorithms specially designed to Arab nations. The research shows that Arabic language can be reactivated to generate more Arabic-based cryptographic techniques that could be used to serve the Arab community.

The results of the research project prove that the presented algorithm is hard to break using brute-force attack; it needs a very long time to obtain the key or to decrypt the message. The Cryptanalysis attack is also very hard to be used since the letter frequency and language characteristics disappear.

In the future, the cryptographic algorithm could be generalized to be used in any other language and will not be limited to Arabic language. The algorithm can also be expanded to include more characters, symbols from other languages, data types and file formats that could be flexibly included. Moreover, other enhancements could be added to the algorithm like rounds of substitutions and permutations in addition to a keyword dynamic change in the encryption process.

Acknowledgment

The authors wish to acknowledge the approval and support of this research study by the grant No SCI-2017-1-7-F-6891 from the Deanship of Scientific Research at Northern Border University, Arar K.S.A.

References

[1] Yahya Meer Alam, M. Hassan at-Tayyan Mohammed Mrayati, al-Kindi's Treatise on Cryptanalysis.: KFCRIS & KACST, 2003.

[2] Monica Borda, Fundamentals in Information Theory and Coding.: Springer, 2011.

[3] Yaḥyá Mīr ʻAlam, M. Hassan At-Tayyan. (et al) Muḥammad Marāyātī, Al-Kindi's treatise on cryptanalysis. Riyadh: KFCRIS & KACST, c2003.

[4] Yahya Alqahtani Prakash Kuppuswamy, "NEW INNOVATION OF ARABIC LANGUAGE ENCRYPTION TECHNIQUE USING NEW SYMMETRIC KEY ALGORITHM," vol. 7, no. 1, 2014.

[5] Haifaa Abdul-Zahra Atee, "DEVELOPMENT OF A NEW WAY TO ENCRYPT THE ARABIC LANGUAGE LETTERS USING THE SYMMETRIC ENCRYPTION SYSTEM," no. 1818653X , 2011.

[6] Ibn Tabataba Treatise Cryptanalysis of Arabic Poetry, "Bushra Mohamed Elamin Elnaim, Hayder Abood S. Wsmi Al-Lam," vol. 5, no. 2, 2017.

[7] Lidong Chen, Dr Özgür Dagdelen (et al) Matthew Campagna, "Quantum Safe Cryptography and Security, An introduction, benefits, enablers and challenges," 979-10-92620-03-0, 2015.

[8] Peter W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM, vol. 41, no. 2, pp. 303-332, 1999.

[9] Nitin, et al. Jain, "Attacks on practical quantum key distribution systems (and how to prevent them)," Contemporary Physics, vol. 57, no. 3, pp. 366-387, Feb. 2016.

[10] Erik Dahmen Daniel J. Bernstein •Johannes Buchmann, "Post-Quantum Cryptography," 2009.

[11] British Council, "ARABIC LANGUAGE AND CULTURE CONFERENCE, PROMOTING THE TEACHING AND LEARNING OF ARABIC IN THE UK," Qatar Foundation, Doha, Qatar, Conference Precedding 2013.

[12] Yusuf Perwej, "Recurrent Neural Network Method in Arabic Words Recognition System," International Journal of Computer Science and Telecommunications, vol. 3, no. 11, pp. 43-48, 2012.

[13] Ibrahim A. Al-Kadit, "ORIGINS OF CRYPTOLOGY: THE ARAB CONTRIBUTIONS," vol. 16, no. 2, 2010.

[14] Aiden A. & Forcinito, Mario A Bruen, Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century.: John Wiley & Sons., 978-1-118-03138-4.

[15] Prakash Kuppuswamy, Sikandhar Shah Yahya Alqahtani, "NEW APPROACH OF ARABIC ENCRYPTION/DECRYPTION TECHNIQUE USING VIGENERE CIPHER ON MOD 39," International Journal of Advanced Research in IT and Engineering, 2013.

[16] Mohammed Abdullah Aysan and Prakash Kuppuswamy, "HYBRID COMBINATION OF MESSAGE ENCRYPTION TECHNIQUES ON ARABIC TEXT: USING NEW SYMMETRIC KEY AND SIMPLE LOGARITHM FUNCTION," International Journal of Scientific Knowledge, vol. 5, no. 5, pp. 37-41, Aug 2014.

[17] Friedrich L. Bauer, Decreypted Secrets Methods and Maxims of Cryptology, 1st ed. Munchen, Germany: Springer, 1997.

[18] William Stallings, Cryptography and Network Security: Principles and Practice, 7th ed.: Pearson, 2017.

[19] Khalil Shihab, "Arabic and Multilingual Scripts Sorting and Analysis ," in 6th WSEAS International Conference on Applied Informatics and Communications, Elounda, Greece, 2006, pp. 157-162.

[20] Irv Englander, The Architecture of Computer Hardware, Systems Software, and Networking: An Information Technology Approach, 5th ed.: Wiley, 2014.

[21] Husni, and Chris Mellish Al-Muhtaseb, "Some Differences Between Arabic and English: A Step Towards an Arabic Upper Model," in 6th International Conference on Multilingual Computing, 1998.

[22] M. V. McDonald, "The Order and Phoetic Value of Arabic Siblants in the "ABJAD"," Semitic Studies, vol. XIX, no. 1, pp. 36-46, March 1974.

[23] Alain George, "Calligraphy, Colour and Light in the Blue Qur'an," Journal of Qur'anic Studies, vol. 11, no. 1, pp. 75-125, 2009.

[24] ISO/IEC JTC 1/SC 2. (1999, Dec.) International Organization for Standardization. [Online]. https://www.iso.org/standard/28250.html

[25] Seymour Bosworth et al., Computer Security Handbook, 6th ed., M. E. Kabay (Editor), Eric Whyne (Editor) Seymour Bosworth (Editor), Ed.: Wiley, 2014.

[26] Waits, Foley, Kennedy Demana, Precalculus: Graphical, Numerical, Algebraic, Teacher's Edition, 7th ed.: PEARSON, 2007.

[27] Shawon S. M. Rahman, Tanvir Ahmed Shaon Abdullah Al- Mamun, "Security Analysis of AES and Enhancing its Security by Modifying S-Box With an Additional Byte ," International Journal of Computer Networks & Communications (IJCNC), vol. 9, no. 2, pp. 69-88, 2017.