AXDC:A New Symmetric Key Algorithm

Khdega A.Yosef Galala
Department of Computer Science
College of Education, Al Jufrah University, Waddan, Libya
kdebh@yahoo.com

Abstract—In many parts of the world, security of information has become critical issue and development of security techniques are still needed to be addressed. Information is vulnerable to attack and hacking is daily and the cost of information crimes is increasing. Therefore, a strongly technique for securing the transmitted information is recommended. For a long time, encryption system is one of the most effective techniques available to achieve network security. Consequently, this paper aims to develop a new encryption algorithm called AXDC and it falls under symmetric-Key of stream cipher. As first step, the proposed algorithm generates the secret key which is then XORed with the corresponding ASCII code of each character. Then the result obtained for each character is spilt into two separate parts and complex operations were performed on each part. The AXDC algorithm work very smoothly for both a small and numerous amount of data.

Keywords - encryption; decryption; symmetric-key; public key;ascii.

I. INTRODUCTION

In the information age, with increasing reliant on the usage of Internet for business, banking, education, shopping, and many others, network security has become a complex matter. There are a numerous technologies and applications carried out to overcome the growing threat. Encryption is one of the primary and a strong technique for providing network security that is required [1].

Encryption is defined as converting the electronic data which is called plaintext to encrypted form known as Ciphertext. The reverse process of encryption is known as decryption and it means converting encrypted form to the electronic data or original data. The process of encryption and decryption requires an encryption key which is referred to a set of numbers or bits used to encrypt plaintext or decrypt Ciphertext [2].

The primary types of encryption systems are symmetric and asymmetric encryption. In the symmetric-key only one key is used for processes of encrypt and decrypt data. The most popular example of this type is DES algorithm [3]. Asymmetric-key used pair keys which are public and private keys. The public key is used to encrypt data and private key is used to decrypt data. RSA algorithm is the famous example of this type [4].

In recent years, there are various forms of strong encryption algorithms which were developed by many researchers, but the initial history of encryption began about 4000 years ago by the Egyptians. After this time and more than 2000 years ago, another basic algorithm was carried out and it is known as Caesar’s shift cipher. Furthermore, in the 1970s the Data Encryption Standard (DES) was introduced by IBM. In 1978 another more common algorithm was invented by Ron Rivest, Leonard Adleman, and Adi Shamir, it is called RSA. Furthermore, in 1985 another public-key algorithm was introduced by ElGamal [5].

This paper aims at developing a new symmetric-key algorithm which will be described in detail in the next section.

II. THE PROPOSED ENCRYPTION ALGORITHM

The proposed algorithm known as AXDC (ASCII XOR Divide Chr) and it is carried out according to two particular stages which are encryption and decryption stage. These stages are show in figure 1 and explained below:

A. Encryption Algorithm

According to the encryption process of the proposed algorithm, we execute the following steps to perform the encrypting of secret message:

Step 1: Enter the secret message and get its length “Msg_len”.

Step 2: Calculate the value of X (X= L+ k) ; where L= the most left digit in the obtained number of the message length multiply with 10; L= L1 *10. For example, If Msg_len = 35 then L1=3 and L=30 while k= the most left digit of number of the key “key” . For example, If key = 57 then k =5. Thus X=35(30+5=35).

Step 3: Read a character from the text and get the corresponding ASCII value of it “Ch”.

Ch1= ASCII (Ch)

Step 4: To get a new value of Ch1, XORed “Ch1” with the secret key “key”.

Ch2=Ch1 $\oplus$ key

Step 5: Divide the value obtained in the previous step by 10 and get “q” the quotient and “R” the remainder.

$q = \text{Int}(Ch2 /10)$

$R= Ch2 \bmod 10$

Step 6: Calculate q1 value

$q1= (q +X) \oplus key$

Step 7: Convert the value obtained in the previous step to character.

$q2= \text{Chr}(q1)$

Step 8: Add the value of secret key to “R” the remainder.

$R1=R + key$

Step 9: Convert the value obtained in the previous step to character.

$R2=\text{Chr}(R1)$

Step 10: Repeat steps 3 to step 9 for each characters.

B. *Decryption Algorithm*

The process of decryption starts on receiving encrypted text and deriving the secret key. The process of the proposed decryption algorithm is explained in details below:

Step1: Enter the Ciphertext and get its length “Msg_len1”.

Step 2: Calculate X value (X= L+ k); (Divide the length of Ciphertext “Msg_len1” by 2).

Step 3: Read pairs of characters from the Ciphertext (q3, R2).

Step 4: Get the corresponding ASCII value of the first character.

$q2= \text{ASCII}(q3)$

Step 5: Calculate q1 value

$q1 = (q2 \oplus key)- X$

Step 6: Multiply the value obtained in the previous step with 10.

$q= q1*10$

Step 7: Get the corresponding ASCII value of the second character “R2”.

$R1= \text{ASCII}(R2)$

Step 8: Subtract “key” from the value obtained in the previous step.

$R=R1-key$

Step 9: Add the value obtained in step 6 to the value obtained in step 8.

$Ch2= q + R$

Step 10: XORed the result obtained in the previous step with key.

$Ch1=Ch2 \oplus key$

Step 11: Convert the value obtained in the previous step to character.

$Ch = \text{Chr}(Ch1)$

Step 12: Repeat steps 3 to step 11 for each characters.

C. *Decryption Key Generation*

The process of generating the secret key according to this work are as follows:

Step 1: Compute the first part of key “keya”

1- Enter three characters and find the ASCII code representation of each character (n1,n2,n3).

2- Compute the average of three values;

$keya = \text{Int}((n1+n2+n3)/3).$

Step 2: Compute the second part of key “keyb ”

1-Read the first character of the massage and get its ASCII code representation “Ch”.

2-Get the reverse number of Ch and divided the result by10;

$keyb = \text{Int}((\text{reverse}(Ch)) /10).$

Step 3: Calculate the value of secret key “key”

$key = (keya - 30) \oplus keyb$

Step 4: Convert the value obtained in the previous step to character.

Example

The above steps will be explained in the following example:

Step 1→ Compute the “keya”

Let n1=“A” and ASCII “A” is 65 in decimal.

Let n2=“m” and ASCII “m” is 109 in decimal.

Let n3=“b” and ASCII “b” is 98 in decimal.

keya = int((65+109+98) /3) = 90.

Step 2→ Compute the “keyb”

Let the first character in the secret message is “A” and ASCII “A” is 65 in decimal.

The reverse 65 is 56.

keyb = Int (56 /10)=5.

Step 3→ Compute the “key”

Key=((90-30) ⊕ 5) = 60 ⊕ 5 =57.

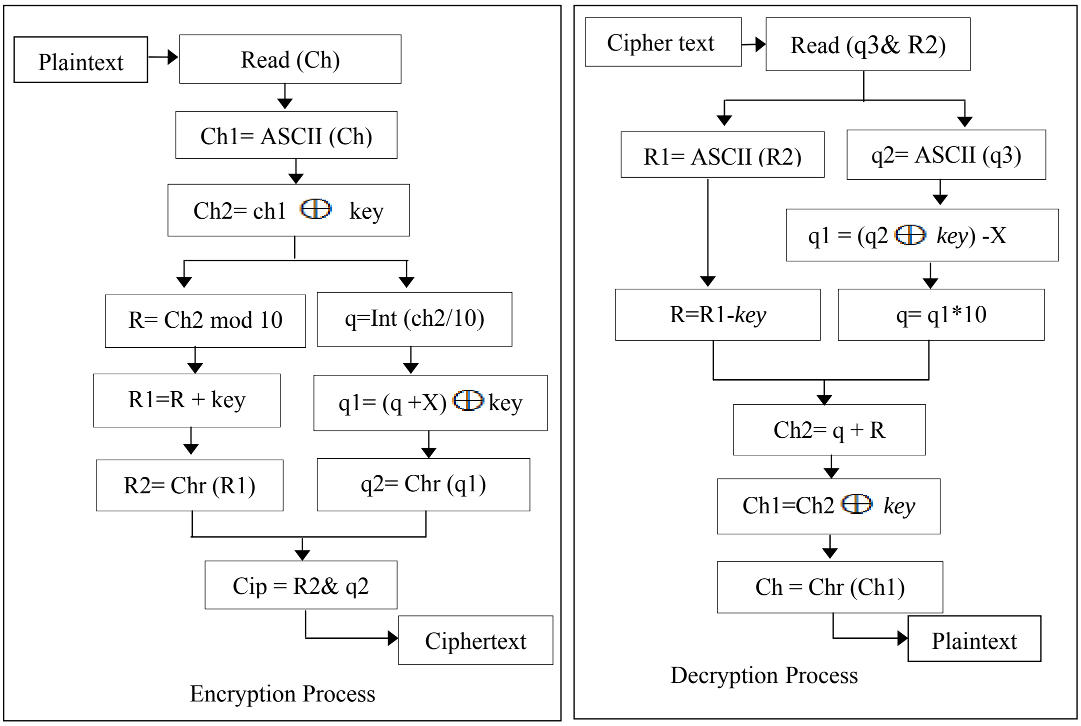

Figure 1. The encryption /decryption process of the proposed algorithm

III. Example

The following example explains how we could encryption-decryption text according to the proposed algorithm.

A. *ENCRYPTION*

Let, the Plaintext is “AXDC Symmetric Encryption Algorithm”, then

Msg_len=35; key=57(according to the above example); X =35(according to the above example).

Step 1 → The result of converting each character to ASCII “Ch1” will be:

658868673283121109109101116114105993269110991141211121161051111103265108103111114105116104109

Step 2 → The result of XORed each value with key “Ch2” will be:

120971251222510664848492777580902512487907564737780868725120859486758077818 4

Step 3→ The result of “q“ the quotient and “R” the remainder will be:

1209712512225106648484927775809025124879075647377808687251208594867580778184

Step 4→ The result of “q1” (q1= (q +x) ⊕ key) and “R1” (R1=R + key) will be:

22572164226222592862206316611861186121591964196218572157286222611864215719621661196019641857186318642862225718622161186319621857196418581861

Step 5 → The result of converting the ASCII values to characters "Ciphertext" will be:

┬ 9┴ @┬ >┬ ; >¶ ?┼ =↨ =↨ =┴ ;‼ @‼ >↨ 9┴ 9 >┬ =↨ @┴ 9‼ >┼ =‼ <‼ @↨ 9↨ ?↨ @ >┬ 9↨ >┴ =↨ ?‼ >↨ 9‼ @ ↨ :↨ =9

B. *DECRYPTION*

Step 1 → The corresponding ASCII value of each character (q2, R1) will be:

22572164226222592862206316611861186121591964196218572157286222611864215719621661196019641857186318642862225718622161186319621857196418581861

Step 2 → The result of "q1" (q1 = ((q2 ⊕ key) –X)*10) and "R"(R=R1-key) will be:

120090712051202205100660480480490270770580090020512048079007056047037078008068072051200805904806705800707801804

Step 3 → The result of "Ch2" (Ch2=q + R) will be:

1209712512225106648484927775809025124879075647377808687251208594867580778184

Step 4 → The result of XORed each value with key "Ch1" will be:

658868673283121109109101116114105993269110991141211121161051111103265108103111114105116104109

Step 5 → Convert the ASCII values into characters "Plaintext" will be:

AXDC Symmetric Encryption Algorithm

IV. Advantages of the Proposed Algorithm

The advantages of the proposed algorithm are:

1. The Ciphertext does not deal with direct characters of languages and this makes it difficult to analyze and hard to break.
2. The secret key is not fixed and its value changes with each entered text, so it will make it difficult to guess. Thus this offers better security.
3. The AXDC algorithm can work very smoothly for both a small and numerous amount of data.
4. The AXDC algorithm works efficiently with both Arabic and English texts.

V. Conclusion

Security is often considered as critical issue when information is presented daily in electronic form. A new symmetric key algorithm was presented in this paper and it will be a new contribution to information security domain. A new algorithm was developed with many security features which make it an effective tool to provide better security for information in electronic form.

References

[1] Andrew S. Tanenbaum and David J. Wetherall, Computer Networks, 5rd ed., Pearson Education,2010.

[2] William Stallings, "Cryptography and network security principles and practices," Prentice Hall, 4rd ed, 2005.

[3] Florim Idrizi, Fisnik Dalipi, Ejup Rustemi, "Analyzing the speed of combined cryptographic algorithms with secret and public key," International Journal of Engineering Research and Development, Vol. 8, no. 2, pp. 45, 2013.

[4] Sunitha K and Prashanth K.S., "Enhancing privacy in cloud service provider using cryptographic algorithm," IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 12, no. 5. pp. 64.

[5] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone, Handbook of applied cryptography, Boca Raton: CRC Press, 1997.