Teaching Basic Cryptography Concepts Through The Creation of a Cryptographic Learning Tool

Celedonio Arroyo-Serrano, Master in Computer Science[1] , Alfredo Cruz, PhD[2], Jeff Duffany, PhD[3]
[1] Polytechnic University of Puerto Rico, Hato Rey, Puerto Rico, alfredcross@gmail.com
[2] Polytechnic University of Puerto Rico, Hato Rey, Puerto Rico, arroyonex@gmail.com
[3] Turabo University, Puerto Rico, jeduffany@suagm.edu

Abstract **·** ***Cryptography is a common technique used to assure data confidentiality and integrity. With the development of a tool designed to provide basic information and capabilities of cryptography, we can reduce the time that it takes for students and IT security professionals to understand and get familiarized with the different types of ciphers that are available, such as mono-alphabetic, poly-alphabetic, and polygraph ciphers. This paper will provide an inside view of the functionalities offered and information related to the ciphers used within the tool, helping to master encryption/decryption concepts faster and more effectively.***

Key Terms · Ciphers, Cryptanalysis, Decryp-tion, Encryption.

Digital Object Identifier
(DOI):http://dx.doi.org/10.18687/LACCEI2016.1.1.170
ISBN: 978-0-9822896-9-3
ISSN: 2414-6390

Teaching Basic Cryptography Concepts Through The Creation of a Cryptographic Learning Tool

Celedonio Arroyo-Serrano, Master in Computer Science[1] , Alfredo Cruz, PhD[2], Jeff Duffany, PhD[3]
[1] Polytechnic University of Puerto Rico, Hato Rey, Puerto Rico, alfredcross@gmail.com
[2] Polytechnic University of Puerto Rico, Hato Rey, Puerto Rico, arroyonex@gmail.com
[3] Turabo University, Puerto Rico, jeduffany@suagm.edu

Abstract *— Cryptography is a common technique used to assure data confidentiality and integrity. With the development of a tool designed to provide basic information and capabilities of cryptography, we can reduce the time that it takes for students and IT security professionals to understand and get familiarized with the different types of ciphers that are available, such as mono-alphabetic, poly-alphabetic, and polygraph ciphers. This paper will provide an inside view of the functionalities offered and information related to the ciphers used within the tool, helping to master encryption/decryption concepts faster and more effectively.*

Key Terms —*Ciphers, Cryptanalysis, Decryp-tion, Encryption.*

INTRODUCTION

In our society information assurance (IA) has taken an important role. Countless transactions that are done over the internet are secured by different algorithms in order to maintain the secrecy and the integrity of the information that is shared. In order to secure this information we rely on *cryptography*. Cryptography concepts are found in some of the Computer Science and related Information Assurance Degree and Certificate courses.

In this paper we propose the use of a visualization technique to understand the basic concepts of cryptography, by implementing a tool that can be distributed in related courses, and can also be used by IA professionals.

Visualization is a process of taking raw data and converting it into a graphical form that is viewable and understandable to humans [1] by creating images, diagrams or animations to communicate a message. Advances in tech-nology have made visualization very informative and easy to practice making the employment of visualization tools in the class room very productive.

The user will be able to differentiate the use of *mono-alphabetic*, *polyalphabetic*, and *polygraph ciphers*. In addition a section on advanced ciphers is added with a basic approach to the *Block cipher* using the Advanced Encryption Standard (AES).

Even though many of the classical ciphers can be solved by hand, the Instructor can engage the student into using the techniques and then verifying with the tool if the encrypted or decrypted message is correct.

Definition of Concepts

- *Cipher:* Algorithm for performing encryption or decryption. Well-defined steps that can be followed as a procedure.
- *Plaintext:* The original intelligible message or data that is fed into the algorithm as input.
- *Ciphertext***:** The scrambled message produced as output.
- *Encryption*: The process of converting from plaintext to ciphertext.
- *Decryption*: The process of restoring the plain-text from the ciphertext
- *Key*: A value used to encrypt or decrypt a message or data.
- *Cryptanalysis*: The branch of cryptology deal-ing with the breaking of a cipher to recover information.

SUPPORTING THEORY

Cryptography is the method of storing and transmitting data in a way that those it is intended for can read and process it [2]. Although the critical goal of cryptography is to hide information from unauthorized individuals, most algorithms can be broken and the information can be revealed if the attacker has enough knowledge, time, desire, and resources. So a more realistic goal of cryptography is to make obtaining the information too work-intensive to be attractive to the attacker [3].

If we start to look back at the history of cryptography we see in many text books that this concept can reach back to 4000 years. The Egyptians used a type of substitution cipher to encipher some of their hieroglyphic writing on monuments. It's also a fact that ancient Hebrews enciphered certain words in the scriptures [2]. Even 2000 years ago Julius Caesar used a simple cipher to send secret messages to his generals. Roger Bacon described several methods to cipher messages [4] in the 1200's. In 1585 the polyalphabetic substitution cipher was

Digital Object Identifier (DOI): http://dx.doi.org/10.18687/LACCEI2016.1.1.170
ISBN: 978-0-9822896-9-3
ISSN: 2414-6390

described by Blaise de Vigenere [5]. Many other ciphers had evolved during that time, producing a clear difference between classical ciphers and modern ciphers.

In 2011 Ma et al. [6] stated that cryptography should be a course offered at colleges and univer-sities. Textbooks and handbooks aid in the teaching of cryptography, but students attracted to this field may post some unique challenges to educators. In particular, Computer Science students find that learning the sophisticated mathematics behind the crypto-systems is a difficult task, while math ma-jors often get lost in the details of the complicated algorithms. Educators need to find a way to help students understand both how and why the algorithms are used [6]. Visualization tools can be an effective way for educators to battle this challenge.

Instructors will be able to teach the basic concepts of *cryptography* by providing this tool in their courses, motivating students to understand the use of classical and advanced ciphers.

Classical and Advanced Ciphers

Classical ciphers have been used throughout history. But now, they are being used to learn the basic concepts of cryptography because they can be solved by hand. With the fast growth of technology many techniques such as cryptanalysis make the encrypted messages insecure and easy to decrypt without even knowing the key or the shift parameter [2, 3].

In the tool we added a basic form of an advanced block cipher using the Advanced Encryption Standard (AES). The student will be able to see the difference between how classical ciphers have evolved in order to be more secure. The AES is very strong, making it very difficult to break with cryptanalysis.

DEMO OF TOOL AND TYPES OF CIPHERS

Upon starting the tool, an interactive screen will permit the user to select any cipher, and the screen will change to the encrypt-decrypt options of each cipher as seen in the Main Menu in Fig. 1.

In Fig. 1 we can observe the menu that is used to select the different options available in the tool. As we can see each cipher is grouped by type. The Monoalphabetic type has the Substitution and Transposition ciphers with different examples: Caesar and Affine for the Substitution ciphers and Railfence for the Transposition cipher. The Polyalphabetic type uses the Vigenere Cipher. The Polygraph type has the Playfair Cipher. AES is included as an example of Advanced Ciphers.

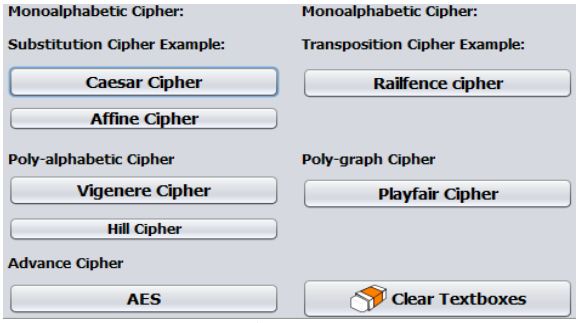

Fig. 1 Main Menu of the Cryptographic Learning Tool

All the ciphers will be explained in the order they appear on the tool and an explanation of the use of the tool will follow the examples provided.

MONO-ALPHABETIC CIPHERS

Mono-alphabetic ciphers use fixed substitution over a plaintext in order to create a ciphertext. The algorithms in these ciphers are based on two sub-types: substitution and transposition ciphers [2].

Substitution Ciphers

Substitution ciphers are a method of encoding by which units of plaintext are replaced with cipher text, according to a regular system.

Caesar Cipher

The Caesar Cipher is a monoalphabetic cipher, composed of the use of the alphabet (in our case the English alphabet). Other types of classical ciphers will use a key and a plaintext in order to encrypt or decrypt a message, but this cipher uses a shift parameter. The shift parameter is responsible for the transformation of the plaintext or the ciphertext. Based on the shift parameter, the alphabets that are aligned together will be rotated to the left or to the right, depending on the number of shifts [2, 5].

In Fig. 2, we can see how the alphabet was transformed using a shift parameter of 3. In this example A was transformed into D, B was trans-formed into E, C was transformed into F, and so on, until the entire alphabet is transformed.

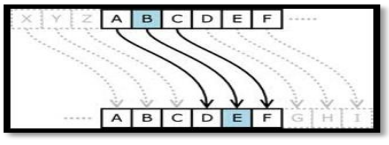

Fig. 2 Transformation of the Alphabet

It's important to mention that once the alphabet reaches the final letter in the alphabet (Z) it will keep on transforming the alphabet starting from A again until the full transformation is completed.

The example of the Caesar Cipher is the first choice available on the menu. In Fig. 3 the user will be able to enter a plaintext and select the shift parameter. As an example, the user enters the plaintext "CryptographyIsEasyToLearn" in the plaintext textbox and can choose up to 26 shifts. In Fig. 3 the shift parameter used is 4. Once the user clicks on the "Encrypt" button the encrypted message in the ciphertext textbox is: "GVCTXSKVETLCMWIEWCXSPIEVR".

Afterwards, the user can copy the ciphertext that was generated into the decryption screen plaintext text box, enters the same shift parameter, resulting in the same plaintext.

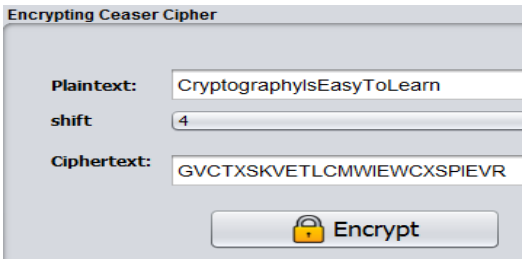

Fig. 3 Encrypting a Message Using the Caeser Cipher

The mathematical description of the Caesar Cipher is one of the easiest to understand. In this case, an array would be created and a series of loops can then shift the alphabet depending on the number of shifts wanted [5]. In equation (1), we can observe that ciphertext will be equal to C, while the plaintext will be equal to P, and the shift parameter will be K. The arithmetic expression is as follows using the size of the English alphabet (26):

$$C = (P + K) \bmod 26 \quad (1)$$

Affine Cipher

The Affine Cipher is a variant of the Caesar Cipher (considered a special Monoalphabetic Cipher) based on how the shift parameter is used to encrypt/decrypt the plaintext [7]. In this cipher a linear equation is used to make the shift of the plaintext different. Modulo *m*, where *m* is an integer the size of the alphabet used, for example, the English alphabet *m = 26*. Each letter in the alphabet is mapped to its numeric equivalent for example, A is 0, B is 1, C is 2, and so on, until reaching Z, which is equal to 25.

In order to select the keys it is important to meet the restrictions that are involved for the *value of a*, and the value of *b:*

- ***a** can't be **0***: It must be any integer .
- ***a:*** Must be relatively prime to *m*. There must not be any common factor with *m*.
- ***b***: Can be any integer in the alphabet.

Equation (2) shows the function notation used in this cipher. This is the equation used to encrypt:

$$f(x) = (ax + b) \bmod m \quad (2)$$

- ***x*** is the plaintext letter.
- ***a*** is the first shift parameter or second key.
- ***b*** is the second shift parameter or second key.
- $f(x)$ is the ciphertext.

Table 1 gives an example of the encryption process solving it by substituting each variable in the equation and Fig. 4 does the same example with the tool.

TABLE 1
Encryption Process of the Affine Cipher

S	18	y = (11*18 + 4) MOD 26 =	20	U
E	4	y = (11*4 + 4) MOD 26 =	22	W
C	2	y = (11*2 + 4) MOD 26 =	0	A

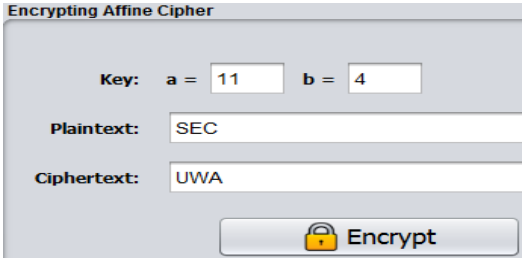

Fig. 4 Encryption Using the Affine Cipher

The value of *a* is *11* which is a prime number, and the value of *b* is *4*. These values were entered by the user on the tool. The plaintext SEC was entered by the user on the plaintext field.

Once the user clicks on the encryption button the equation was solved using equation (2) and the result "UWA" was placed on the ciphertext field. In order to do it by hand, in Table 1, we can see how S which is 18 is used and all the variables are added on equation (2), by solving (11*18+4) we obtain 202 and then we calculate the modulo 26 of 202, giving as a result 20, which is the alphabet equivalent U. This process is repeated with the rest of the letters.

In order to decrypt we use the multiplicative inverse. A multiplicative inverse of an integer *a* modulo *m* is an integer *b*,

in the range *1* to *m*-1, such that $ab \equiv 1 \bmod m$. When *a* and *m* are relatively prime, *b* will exist and as the multiplicative inverse of a and label it a^{-1} [7]. Equation (3) shows the decryption function for this cipher.

$$f^{-1}(x) = a^{-1}(x-b) \bmod m \quad (3)$$

In the example of Table 1 we used the key of *(11, 4)*, where *a = 11*, and b = 4, and *m =26*. First we need to find the multiplicative inverse of *a*. See Table 2, the inverse of *11* is *19*. This table is added in order to provide the user the answer to each inverse because it can be difficult to obtain.

TABLE 2
Multiplicative Inverses Modulo 26

x:	1	3	5	7	9	11	17	25
x^{-1} mod 26:	**1**	**9**	**21**	**15**	**3**	**19**	**23**	**25**

In Table 3 we can see the decryption process using equation (3), for example in order to decrypt the UWA, first we start by solving U. First we have that *a* is 11, so we use Table 2 and search for the inverse of 11 that is 19, we then replace that number on the variable *a* of equation (3). Then we replace *(x-b)* with the numerical equivalent of "U" which is 20, and *b* is the second key which was 4, and we obtain *(20-4)*, which is 16. Now we multiply *19* and *16* and we obtain *304*. Then we use 304 and using modulo 26 we obtain 18 which is the numerical equivalent to "S".

TABLE 3
Decryption Process of the Affine Cipher

U	20	19(20 - 4) MOD 26 =	18	S
W	22	19(22 - 4)MOD 26 =	4	E
A	0	19(0 - 4) MOD 26 =	2	C

Transposition Ciphers

The Transposition Cipher is part of the Mono-alphabetic Cipher but it is a method of encryption by which the positions held by units of plain-text are shifted according to a regular system, so that the ciphertext constitutes a permutation of the plaintext. This means that the plaintext is reordered. It can be described in mathematics as a bijective function where the character's position is used to encrypt and the inverse function to decrypt.

Rail fence Cipher

The *Rail Fence cipher* gets its name from the way in which it is encoded. It is one of the most basic but always used ciphers to show the concepts of transposition ciphers. In the *Rail Fence cipher*, the plaintext is written downwards on successive "rails" of an imaginary fence, then moving up when we get to the bottom. The message is then read off in rows [2, 5].

In Fig. 5 we can see an example using two "rails" and the plaintext "SECURITY". A line shows how the transposition goes from the top row to the bottom row for each letter of the plaintext shifting to the next column each time, vertically.

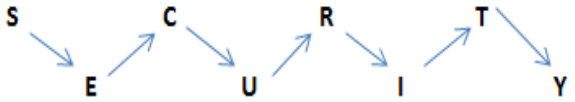

Fig. 5
Rail fence Transposition Example

After all the transposition of the plaintext is done, then we can see the encrypted message. This is done by reading the first row, from left to right and the second row from left to right, giving as a result the encrypted message, "SCRTEUIY".

In Fig. 6, we can see the example discussed above but using the tool interface.

Fig. 6 Encryption Using the Rail fence Cipher

First the user enters the plaintext "SECU-RITY" then selects the number of rails, in this case "2". Once the encrypt button was clicked the ciphertext "SCRTEUIY" was generated and placed in the ciphertext textbox. Transposition ciphers such as the Railfence don't hide letter frequency.

POLYALPHABETIC CIPHERS

In the Polyalphabetic Cipher we can see how substitution is done at different positions of the message. In this cipher a unit from the plaintext is mapped to one of several possibilities (for example to the cipher text and vice versa) making this type of cipher a little bit more secure than the regular mono-alphabetic ciphers [8]. One of the features of this cipher is that multiple alphabets are used. In order to facilitate encryption, all the alphabets are written out in a large table that consists of a 26×26 graph, so that 26 full ciphertext alphabets are available.

Vigenere Cipher

The Vigenere Cipher is considered one of the most popular polyalphabetic ciphers; it was first published in 1585 and was considered unbreakable until 1863. It consists of the alphabet written out 26 times in different rows, as shown in Fig. 7. Each alphabet is cyclically shifted to the left compared to the previous alphabet, corresponding to the 26 possible shifts in the Caesar Cipher [8, 9] (See Mono-alphabetic Ciphers).

It's important to mention that in order to start the encryption process we need to get the plaintext and align each letter on it with the letters of a key as in Table 4. If the key is too short for all the letters, then we keep repeating the key. For example, if the plaintext is "SECURITY" and the key is "POLY" then the alphabet used at each point depends on repeating the key, aligning each character using the same letter on the key until all the characters in the plaintext are covered.

TABLE 4
Plaintext and Key for Vigenere Cipher

Plaintext:	S	E	C	U	R	I	T	Y
Key:	P	O	L	Y	P	O	L	Y

In Fig. 7 we can observe that the top row represents the plaintext letters. The leftmost column of the square represents the key letters. In order to find the ciphertext we select the letter from the plaintext on the top row "S" and then we select the letter from the left column that is part of the key, "P". We then trace those two and select the letter that intersects the column and the row as seen in Fig. 5, this letter is then the ciphertext letter "H".

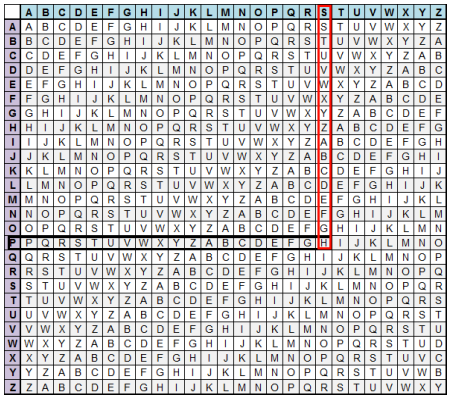

Fig. 7 The Vigenere Square

This same process is done by the tool, but instead the user adds the key (for example "PO-LY") and then enters the plaintext "SECURITY". Once the user clicks on the encryption button, the ciphertext "HSNSGWEN" will be generated in the ciphertext textbox. This can be seen in Fig. 8.

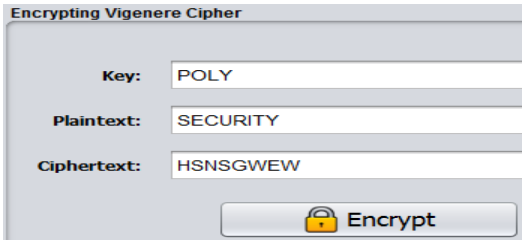

Fig. 8 Encrypting a Message Using the Vigenere Cipher

The decryption is performed by going to the row in the table corresponding to the key which is the left most column, finding the position of the ciphertext letter in this row, and then using the the top row corresponding to the plaintext. By intersecting those two columns the decryption character will be obtained. It is the same operation as in the encryption process but backwards. The decryption screen requires entering the ciphertext and key characters to obtain the plaintext.

The algebraic description of this cipher is as follows: If the letters A–Z are taken to be the numbers 0–25, addition is performed and the result is calculated with modulo 26 to obtain the result as the numerical equivalent of the alphabet letter that is being encrypted. The Vigenere encryption ***E*** using the key ***K*** can be written as in equation (4).

$$C_i = E_k(M_i) = (M_i + K_i) \bmod 26 \qquad (4)$$

For decryption we use *D as* decryption using the *k*, as in equation (5) and we solve it to obtain the plaintext numerical equivalent of the alphabet.

$$M_i = D_k(C_i) = (C_i + K_i) \bmod 26 \qquad (5)$$

POLYGRAPH CIPHER

In a *polygraphic substitution cipher*, plain-text letters are substituted in larger groups instead of substituting single letters. The first advantage is that the frequency distribution is much flatter than that of individual letters making it harder to analyze the letter frequencies, making this type of cipher harder to break [10].

Playfair Cipher

In this cipher, a 5 x 5 grid is filled with the letters of a mixed alphabet. A digraph substitution is then simulated by taking pairs of letters as two corners of a rectangle, and using the other two corners as the ciphertext. This was developed by Charles Weatston. Special rules handle double letters and

pairs falling in the same row or column [8]. In Fig. 9, we can see a 5 x 5 grid. On it, a key was used and the grid was generated. Notice that I and J are together in this graph. This is an important part of the cipher. To generate the key table, first fill in the spaces in the table with the letters of the keyword. In Fig. 9, we used the key "ANNUAL CONFERENCE", (Dropping any duplicate letters) then we filled the remaining spaces with the rest of the letters of the alphabet, in order. To generate the key table, first fill in the spaces in the table with the letters of the keyword.

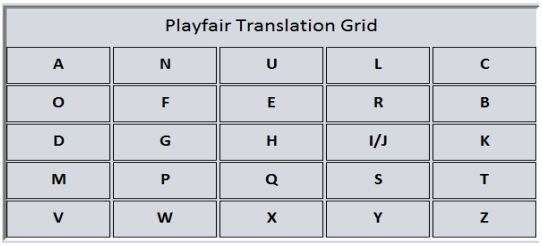

Fig. 9 Playfair Graph

In Fig. 9, we used the key "ANNUAL CONFERENCE", (Dropping any duplicate letters) then we filled the remaining spaces with the rest of the letters of the alphabet, in order. In Fig. 10 we can see an example of the encryption process, using the tool; the user entered the key and the ciphertext.

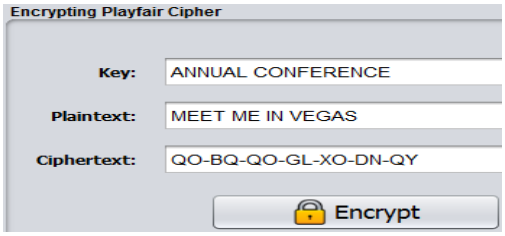

Fig. 10 Encrypting a Message Using the Playfair Cipher

In Fig. 10, while the key was being entered, the graph on Fig. 9 was generated. To encrypt a message, one would break the ciphertext into digraphs (groups of 2 letters) such that, "MEET ME IN VEGAS" becomes "ME-XE-TM-EI-NV-EG-AS", and map them out on the key table. If needed an "X" is added to complete the final digraph, or if two letters are the same, for example in the word "MEET", an "X" needs to be used to separate the double "EE" turning into "ME XE". The two letters of the di-graph will also be used to do the encryption /decryption, and this is done by using a rectangle as the reference.

For example in "ME XE" we are going to start by looking for "ME" on the graph. We look for "M" and then "E" and we create a rectangle with those two, considering each letter as a corner of a rectangle in the key table. In the example in Fig.10, once the user presses the encrypt button the ciphertext is generated resulting in "QO-BQ-QO-GL-XO-DN-QY".

Depending on the type of rectangle we need to use 3 rules to order each pair of letters in the plain-text in order to encrypt:

- If the letters appear on the same row of the table, replace them with the letters to their immediate right (wrapping around to the left side of the row if a letter in the original pair was on the right side of the row).
- If the letters appear on the same column of the table, replace them with the letters below (wrapping around to the top side of the column if a letter in the original pair was on the bottom side of the column).
- If the letters are not on the same row or column, replace them with the letters on the same row but at the other pair of corners of the rectangle defined by the original pair. The or-der is important; the first letter of the encrypt-ted pair is the one that lies on the same row as the first letter of the plaintext pair. To decrypt, invert the last 3 rules.

ADVANCED CIPHERS

A Block cipher is an encryption method that processes the input stream as groups of bytes that are fixed in size, 64, 128 or 256 bits long. The state of a block cipher is reset before processing each block [11].

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data established by the U.S. National Institute of Standards and Technology (NIST) in 2001 [12, 13]. AES is a variant of the Rijndael cipher.

Rijndael is a family of ciphers with different key and block sizes [13]. The AES is byte oriented, that is a bit, 0 or 1. 8 bit together form a byte and is the basic unit of processing data in AES. The key size used specifies the number of repetition of transformation rounds that converts the plaintext into the ciphertext. The size of the key determines the number of cycles. The plaintext, salt and key are represented as arrays of bytes that are formed by dividing these sequences into groups of eight contiguous bits to form arrays of bytes [14]. The best way to describe it is with the 16 bytes of the 128 bit input block arranged in a 4x4 matrix of bytes as in the input bytes matrix of Fig. 11.

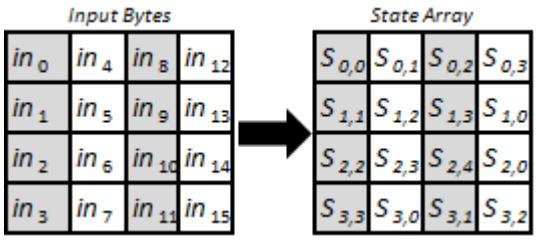

Fig. 11 Input Bytes to State Array

Each round consists on a series of steps covering four different similar stages but done on the state array matrix in Fig. 11. For the decryption we do the same rounds but in the opposite direction. In order to make sure that the key remains secret in this paper we added a *salt* which is any word, and an *Initial Vector*, randomly generated, this is known as key expansion. This is used to add confusion to the key and is performed while filling the input matrix. In Fig. 12 we can see a flowchart of the transformation performed on the data block [12-14].

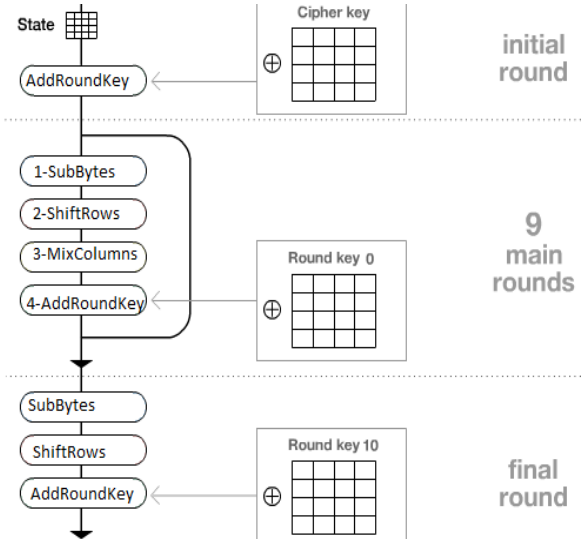

Fig. 12 Encryption Process Flowchart

The four transformations in Fig. 12, are:

1. *SubBytes*(Substitue bytes): in where an *S-box* is used to perform a byte to byte substitution of block.
2. *ShiftRows* (Shift the rows): a simple permuta-tion.
3. *MixColumns* (Mixing the columns): a simple transposition.
4. *AddRoundKey* (Add round key): A simple bit-wise *XOR* of current block with a portion of expanded key.

These steps are used to alter the input matrix. First we start with the *AddRoundKey* stage [14]. In here the key, the *salt*, and the plaintext are used together and converted. The *initial vector* is generated and added. Using the tool we can see an example of the process mentioned in the flowchart, in Fig. 13.

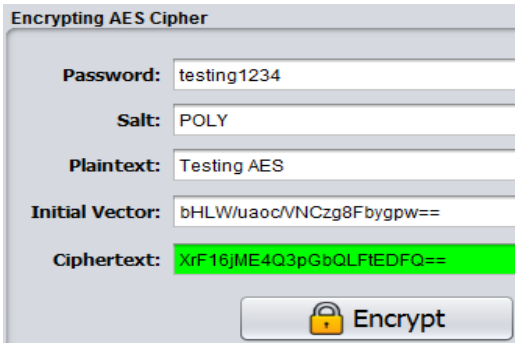

Fig. 13 User Input, Encrypting AES

The user enters the password "testing1234", then he enters the *salt* "POLY" and the plaintext to be encrypted "Testing AES". Once the user clicks on the encryption button the *initial vector* will be randomly generated using the password and the s*alt*, and the ciphertext will be generated and posted, resulting in "XrF16jME4Q3pGbQLFtEDF Q==". The representation of the *initial vector* and the ciphertext, was changed from hexadecimal to base 64. *AddroundKey* is the only stage that makes use of the key and that is the reason why the cipher ends and begins with this stage. The other three stages are in charge of adding confusion, diffusion and nonlinearity, but by themselves would provide no security because they do not use the key. In *SubBytes*, the bytes in the data block are substituted with the values in a matrix called *S-Box*, which is defined by AES as a 16 X 16 matrix of byte values. The *S-Box* contains all possible 256 values. This can be seen in Fig. 14.

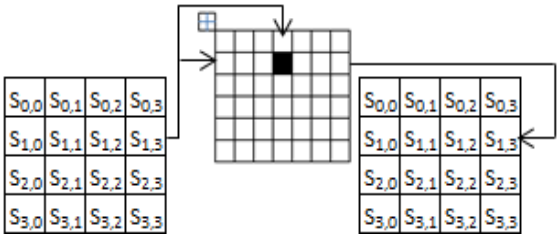

Fig. 14 Substitution of Bytes

In Fig. 14, the upper nibble of byte value of the state is used as a row value of the *S-Box* and the lower nibble of byte value of the state is used as column value of the *S-Box* [13, 14]. These rows and column values serve as indexes into the *S-Box* to select a unique 8 bit output value. In *ShiftRows* the rows of the state array are altered by circularly left shifting them, but the first row is not altered. The 2nd row is shifted by one position, the 3rd by two, and the 4th by three. This can be seen in Fig. 15.

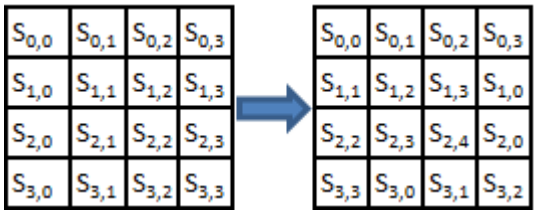

Fig. 15 Shift Row Transformation

In *MixColumns* each column is done discretely and each byte of column is mapped into a new value that is obtained by multiplying the four byte vector by a fixed invertible 4x4 matrix [13, 14]. This can be seen in Fig. 16, in where the product matrix is the sum of products of elements of one row and one column.

$$\begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \begin{pmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{pmatrix} = \begin{pmatrix} s^1_{0,0} & s^1_{0,1} & s^1_{0,2} & s^1_{0,3} \\ s^1_{1,0} & s^1_{1,1} & s^1_{1,2} & s^1_{1,3} \\ s^1_{2,0} & s^1_{2,1} & s^1_{2,2} & s^1_{2,3} \\ s^1_{3,0} & s^1_{3,1} & s^1_{3,2} & s^1_{3,3} \end{pmatrix}$$

Fig. 16 Mixing the Columns Matrix

The *MixColumn* transformation combined with *ShiftRow* transformation ensures that after a few rounds all output bits depend on the input bits.

The last step is the *AddRoundKey*, the entire size state are *XOR'ed* with the 128 bits of round key, making this a very simple transformation, but affecting every bit of the state. It is also the only transformation that uses the key ensuring security in the ciphertext generated [13, 14]. This can be seen in Fig. 17.

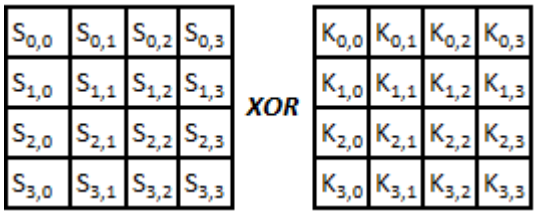

Fig. 17 Add Round Key, XOR

TOOL DESCRIPTION

The tool was created using JAVA, and the IDE used to create the tool is NETBEANS 7.4. The tool includes each of the ciphers that were mentioned in this paper, in addition to the functions that were needed. The tool can be used on any device able to compile a java application; from tablets to computers, using Windows, or any other operating system.

Using the Tool in the Classroom

The tool can be used and/or the code can be modified to add more ciphers. A copy of the tool will be distributed to the professor, class or conference attendees in a USB pen drive, or copied to the computer.

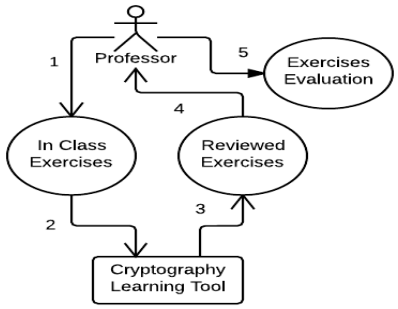

Fig. 18 Real Case Scenario of Tool Interaction

Fig. 18 portrays a real case scenario of a professor using the tool while giving a presentation to a group of students or to attendees in a conference.

1. The professor explains the basic concepts of cryptography, functionalities of the ciphers, and gives a description of each of them, providing exercises. The students try to solve the exercises by hand and then verify their answers with the tool.
2. In this part they might be able to find errors and understand the concepts more in depth.
3. Answers are handed to the professor for in- class discussion.
4. An evaluation is provided to each student based on the results, their effort to understand the concepts, and their participation during the class or conference.

CONCLUSION

The main goal of the tool is to improve the way that the professor can teach the basic concepts of cryptography to students, making it more exciting to them. This impacts the way students learn and challenges them to study the concepts and improve the tool with new and stronger ciphers.

Additional classical and modern ciphers will be added to the tool, starting with the integration of the Hill Cipher. This tool can be further developed to be used in security challenges (such as code-a-thons) by implementing communication protocols such as TCP/IP, broadcasting encrypted messages from IP to IP, and decrypting them within the tool.

ACKNOWLEDGEMENT

This material is based upon work supported by, or in part by, the Nuclear Regulatory Commission (NRC) Grant Fellowship Award under contract/ award # NRC-27-10-511 and is also based upon work supported by, or in part by, the

U.S. Army Research Laboratory and the U.S. Army Research Office under contract/grant number W911NF1110 174.

REFERENCES

[1] Simms, X., & Chi, H. "Enhancing Cryptography Education via Visualization." Proc. of the 49th Annu. Southeast Reg. Conf., Georgia, 2011, pp.344-345.

[2] Harris, S. *CISSP All-in-One Exam Guide.* 5th. ed. McGraw Hill Prof., 2010.

[3] Kuldeep, T., & Richa, A. "Triple Security of File System of Cloud Computing." *Intl. Jrnl. for Res. in Appl. Sci. and Engin. Technol.* 2(1), 2014, pp. 61-72.

[4] Goldstone, L., & Nancy, G., *The Friar and the Cipher*, Double-day, 2005.

[5] Reynard, R., *Secret Code Breaker II*, Smith & Daniel Marketing, 1997.

[6] Ma, J., Tao, J., Keranen, M., Mayo, J., & Kuang, C., "SHAvisual: A Visualization Tool for Secure Hash Algorithm". *Jrnl. of C. Sci. in Coll.*, 27(1), 2011, pp. 81-89.

[7] Barr, T., *Invitation to cryptology*. Upper Saddle River, N.J.: Prentice Hall, 2002.

[8] Dhull, S., Beniwal, S., & Kalra, P., "Polyalphabetic Cipher Techniques Used For Encryption Purpose", *Int. Jrnl of Adv. Res.in CS. and Soft. Engin.* 3(1), 2013, pp. 64-66.

[9] Klima, R., & Sigmon, N.,"Using Graphs to Break Vigenere Ciphers".In Proc. of the 24th Annu.. Int. Conf. on Tech. in Collegiate Mathematics. 2012, pp. 117-126.

[10] Apelbaum, J., *User Authentication Principals Theory and Practice*. Technology Press, 2004.

[11] Block Cipher. (n.d.) *Computer Desktop Encyclopedia.* (1981-2013). Retrieved October 30, 2014 from: http://encyclopedia2.thefreedictionary.com/Symmetric+block+cipher

[12] "Federal Information Processing Standards Publication 197". *United States National Institute of Standards and Technology (NIST).* November 26, 2001.

[13] Daemen, J.; Rijmen, V., "AES Proposal: Rijndael". *National Institute of Standards and Technology*. March 9, 2003, pp. 1.

[14] Kangude, N., Wani, P. and Raut, S, "Advance Encryption Standard". *Intl. Jnlr of C. Sci. Engin. and Tech.* 1(3), 2011, pp. 118-126.