International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

CARV System – Cryptographic Algorithms Representing Visually System

Pyarimol Nair[1], Neethu Krishnan[2]

[1, 2]M.Tech student, Department of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Pathanamthitta, Kerala, India

Abstract: ***In many Engineering and Educational Institutions, the presence of Educational Software systems have a significant role for the improvement of student's attitude and knowledge acquisition. In Engineering and Data Security courses, it covers the topics from cryptographic algorithms, network security and applications and system security. Cryptographic algorithms have become the toughest part among these courses. So visually presenting a topic is easier than theoretically presenting these toughest parts. This paper presents a novel software system for Cryptographic Algorithms Representing Visually (CARV), which visually represent the complex cryptographic algorithms. This system allows users to execute and understand each and every step in complex cryptographic algorithms (RSA, Diffie- Hellman, DES and AES) in easier way.***

Keywords: AES, algorithm visualization, DES, RSA, Diffie-Hellman, data security

1. Introduction

Network security is a complicated subject, so usually it is handled by well trained and experienced experts. However, as more and more people become alert, an increasing number of people come to understand the basics of security in the networked world. In this modern e-world, confidentiality and authentication has become the two major services and it becomes indispensable in the internet world.

It has become more important to understand the basics of network security, cryptographic algorithms, system security etc. Therefore, many schools and colleges provide classes and courses to understand these areas. Many research programs and specialization programs have conducted based on these areas. Cryptographic algorithms are the toughest topic among them, since confidentiality and authentication services are achieved by using various combinations of different cryptographic algorithms.

Since these cryptographic algorithms are very complex and tough to understand, it is very difficult to convince the internal working of each algorithm. Therefore we decided to introduce a new software system that help to make a keen understand about the main cryptographic algorithms like RSA, Diffie-Helman, DES,AES etc. In this system, each step of these each algorithms are explained in detail and are presented visually. So that it is very useful for the students to understand these complex algorithms.

2. Motivation and Related Works

AESvisual [1] is a visualization tool used for visually representing the AES algorithm. This tool allows users to visualize all the major steps of AES encryption and decryption. But in AESvisual tool, it does allow users to enter his own input and does not have a better organized and clear view of certain steps and it does not have a web- based version.

ECVisual [2] is a visualization tool represents the elliptic curve based ciphers. This tool allow users to visualize elliptic curve over real and finite fields of prime order, and performs arithmetic operation and thereby do encryption and decryption and convert plaintext to a point on an elliptic curve. But it does not have better visualization, operation scheme and design.

Grasp [3] is a visualization tool for learning about security protocols. It usually represents Diffie-Hellman protocol. And this representation limits to only conceptual level.

In digital Lego system [4] cryptographic operations are represented as pieces of a puzzle and security protocols are presented as structures. These structures are built from these pieces of puzzles. But it does not cover some areas of cryptography.

Previously described tools have many disadvantages and some of the steps of cryptographic algorithms are not visually represented. So with the CARV tool, we aimed to fulfil all the needs of students who study cryptography and to make them understand each and every step of different complex cryptographic algorithms in detail and in visual.

3. Implementation

CARV system visualizes mainly four cryptographic algorithms. They are RSA, Diffie-Helman, DES and AES. This system is implemented in interactive level. So students are able to execute the algorithms forward and backward. They can configure the algorithm parameters before the execution begins. Each and every steps of algorithm execution are explained in detail and in visualized manner. CARV system is designed in such a way that it has very minimum chance for the occurrence of errors. The CARV system is implemented in JAVA programming language. NetBeans IDE is used as the development environment.

3.1 RSA Implementation

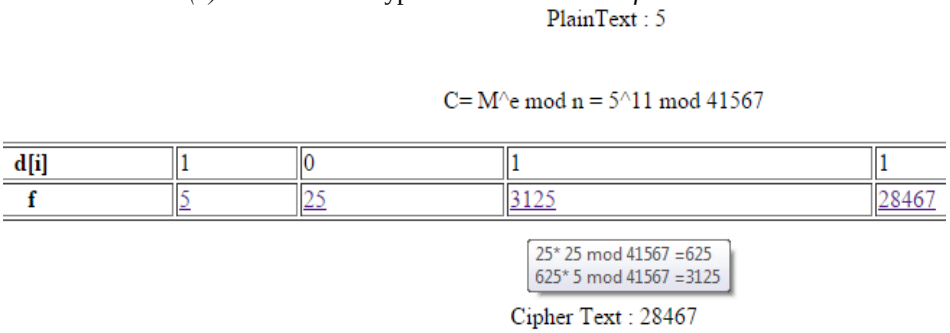
RSA is an asymmetric public key cryptography algorithm. Asymmetric cryptographic algorithm means, it uses same

keys for the process of encryption and decryption. RSA algorithm can be explained as below.

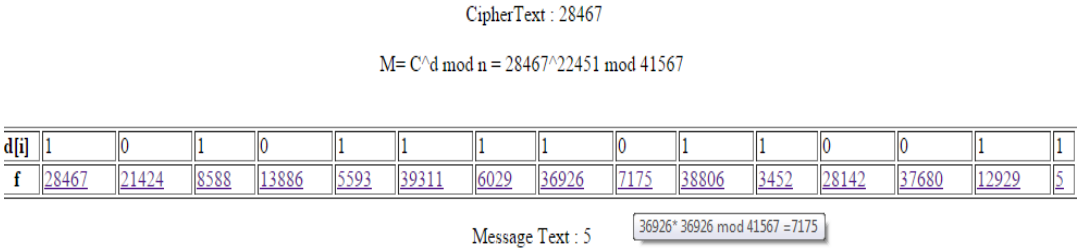
RSA algorithm consists of mainly three processes. They are: Key Generation, Encryption and Decryption. Key Generation process is the process of finding public key pairs (e and n) and private key pairs (d and n). So start with the selection of two prime numbers (p and q). To find n, multiply p with n. Next step is to find e. To calculate e, it should satisfy following two conditions: GCD(Φ(n), e) = 1and 1<e<Φ(n). Euler's totient function (Φ(n)) is calculated using the formula, $\Phi(n) = (p - 1) * (q - 1)$. After that d is calculated using the formula, $d = e^{-1} \bmod \Phi(n)$. For the Encryption process, we use the formula, Cipher, $C = M^e \bmod n$. For the decryption process, we use the formula, Decrypted Message, $M = C^d \bmod n$.

3.1.1Visual Representation of RSA

The tool described in [6] helps to illustrate RSA algorithm. The parameters p, q, e and d can be selected by students. Fast exponential algorithm is used for the encryption and decryption calculations. By positioning the mouse pointer on the top of the intermediate result, the student gets the tip that shows the way that the result was calculated. In Figure1 (a) and Figure 1 (b), *d(i)* shows the bit representation of value of e and d. And f shows the intermediate result.



(a) Visual Representation of Encryption in RSA using Fast Exponentiation Algorithm.



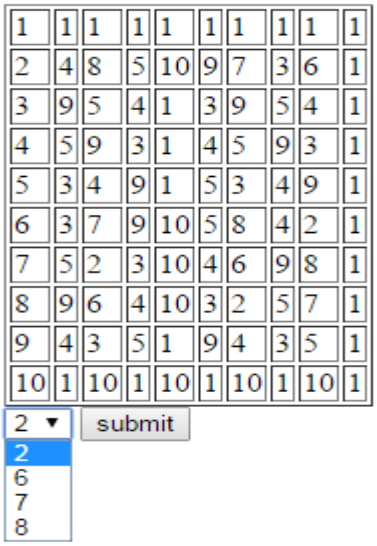
(b) Visual Representation of Decryption in RSA

Figure 1: Visual Representation of RSA

3.2 Diffie – Hellman Implementation

Diffie–Hellman algorithm is also an asymmetric public key cryptographic algorithm. It is a key agreement protocol. It is mainly used for the secret exchange of keys between two users. Diffiie-Hellman algorithm can be explained as below. For visualization, Diffie-Hellman consists of following processes. First of all, the two users should agree on a prime number p and its primitive root a, which is less than p. Then each user should select their private number $X1$ and $X2$ and calculate their public number $Y1$ and $Y2$ using the formula, $Y1 = a^{X1} \bmod p$ and $Y2 = a^{X2} \bmod p$. Then this public values are exchanged between the two users. The common secret key value is calculated by two users using the following formula $K = Y2^{X1} \bmod p$ and $K = Y1^{X2} \bmod p$.

3.2.1Visual Representation of Diffie-Hellman



(a) All Possible Primitive Roots, a, of Prime Number 11. The Dropdown Box Shows Roots Less Than Prime Number.

User A's Key

K = Y2^X1 mod p = 7^10 mod 11

d[i]	1	0	1	0
f	7	5	10	1

K = 1

5* 5 mod 11 =3
3* 7 mod 11 =10

User B's Key

K = Y1^X2 mod p = 1^9 mod 11

d[i]	1	0	0	1
f	1	1	1	1

K = 1

(b) Secret Key 1 is Shared Between Two Users

Figure 2: Visual Representation of Diffie-Hellman

3.3 DES Implementation

DES is a symmetric block algorithm. Several iterations of plain text lead to the cipher text. Place table titles above the tables. The size of input message or plain text is 64 bits. Key size is 56 bit. Round key size is 48 bits. And number of iterations is 16. DES algorithm can be explained as follows.

First, 64 bits plain text is given as input. An initial permutation is done based on predefined permutation table. Permutation is the process of reordering of bit position for each of the input. After permutation, the 64 bits are divided into 32 bits each. 56 bit key is permuted based on permutation table and a left round shift is done and again permuted once again. Then these inputs are entered into 16 round. Each round consists of XOR operation with subkey, substitution and permutation. At the end a 32 bit swap and an inverse permutation is done to obtain the cipher text. Same step is executed for decryption also.

3.3.1Visual Representation of DES

Cipher Test | Initial Permutuation | Expansion | SubKey | S-box | Permutation | Final Permutation

Input message:
00000001 00100011 01000101 01100111 10001001 10101011 11001101 11101111

After initial permutation
11001100 00000000 11001100 11111111 11110000 10101010 11110000 10101010

R: After E expansion
01111010 00010101 01010101 01111010 00010101 01010101

R: After XOR with the subkey
01100001 00010111 10111010 10000110 01100101 00100111

R: After S boxes
01011100 10000010 10110101 10010111

R: After P permutation
00100011 01001010 10101001 10111011

Right half at round #1
11101111 01001010 01100101 01000100

(a) Steps in First Round.

R: After XOR with the subkey
01011111 11000101 11010100 01110111 11111111 01010001

R: After S boxes
10110010 11101000 10001101 00111100

R: After P permutation
01011011 10000001 00100111 01101110

Right half at round #15
01000011 01000010 00110010 00110100

R: After E expansion
00100000 01101010 00000100 00011010 01000001 10101000

R: After XOR with the subkey
11101011 01010111 10001111 00010100 01010110 01011101

R: After S boxes
10100111 10000011 00100100 00101001

R: After P permutation
11001000 11000000 01001111 10011000

Right half at round #16
00001010 01001100 11011001 10010101

After 16 rounds
00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

After final permutation
10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

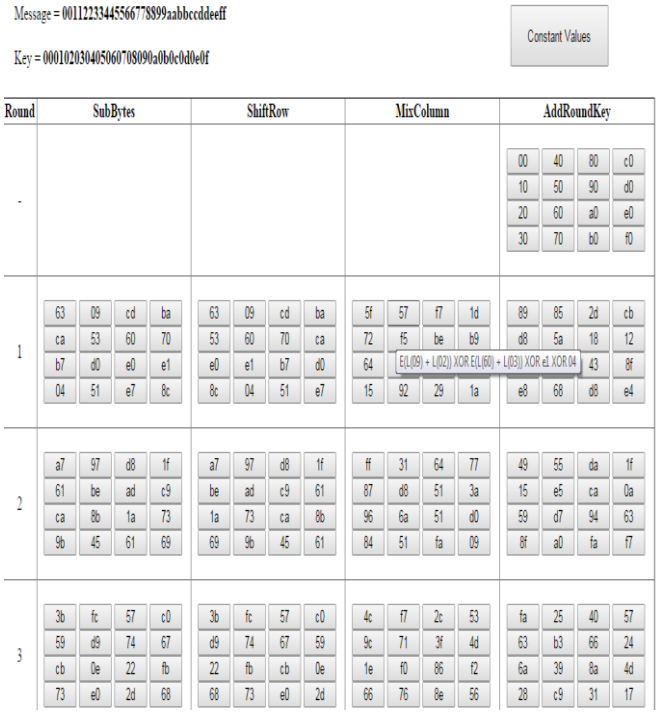
(b) 16th Round of DES Algorithm with Detailed Description about Each Step.

Figure 3: Visual Representation of DES

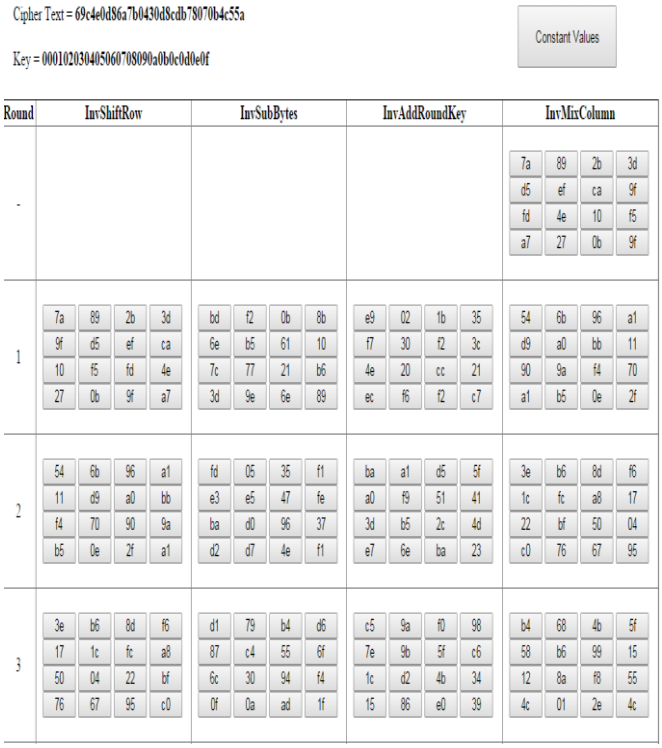
3.4 AES Implementation

AES is also a symmetric block algorithm. Here the plain text size is 128 bits. The key size can be 128 /192/256 bits. The round key size is 128 bits. It consists of 10/12/14 rounds based on key size. AES algorithm can be explained as follows. The plain text and key is given as input. Initially, Add Round Key operation is performed. That means an initial XOR with the original key is done. After that based on substitution matrix, Byte Substitution operation is performed. Then Shift Row operation and Mix Column operation is done. Mix Column operation is done based on E-Table and L-Table. At last Add Round Key operation is performed. 10 rounds will complete to get the cipher text. In decryption, after initial Add Round Key operation, Inverse Shift Row, Inverse substitute Byte, Add Round Key and Inverse Mix Column operations are performed.

3.4.1Visual Representation of AES



(a) AES Encryption Steps



(b) AES Decryption Steps

Figure 4: Visual Representation of AES

4. Conclusion

In this paper, we introduce a novel CARV system for visually representing complex cryptographic algorithms. This system really helps students to understand each and every step of cryptographic algorithms like RSA, Diffie-Hellman, DES and AES. The student itself can configure the parameters before the execution begins. To the best of our knowledge, AES algorithm is implemented in detail at first, and it is done in this system. The students can control the execution backward and forward, so that they can thoroughly understand these complex cryptographic algorithms. Hence CARV system is much beneficial for the students.

References

[1] Jun Ma, Jun Tao, Melissa Keranen, Jean Mayo, Ching-Kuang, "AESvisual : A Visualization Tool for the AES Cipher",Michigan Technological University, Houghton, USA, 2014

[2] J. Tao, J. Ma, M. Keranen, J. Mayo, and C. K. Shene, "ECvisual: A visualization tool for elliptic curve based ciphers," in Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ., Feb. 2012, pp. 571–576.

[3] D. Schweitzer, L. Baird, M. Collins, W. Brown, and M. Sherman, "GRASP: A visualization tool for teaching security protocols," in Proc. 10th Colloquium Inf. Syst. Security Educ., Jun. 2006, pp. 75–81.

[4] W. Wang, A. Lu, L. Yu, and Z. Li, "A digital lego set and exercises for teaching security protocols," in Proc. Colloquium Inf. Syst. Security Educ., 2008, pp. 26–33.

[5] W.Stallings, "Cryptography and Network Security" 4th ed. Englewood Cliffs, NJ, USA: Prentice Hall,2005.

[6] D. Schweitzer and L. Baird, "The design and use of interactive visualization applets for teaching ciphers," in Proc. IEEE Inf. Assurance Workshop, Jun. 2006, pp. 69–75.

Author Profile

Pyarimol Nair received the B.Tech degrees in Computer Science and Engineering from M.G University, Kerala at Mount Zion College of Engineering in 2012. And now she is pursuing her M.Tech degree in Computer Science and Engineering under the same university in Mount Zion College of Engineering.

Neethu Krishnan received the B.Tech degrees in Computer Science and Engineering from M.G University, Kerala at Mount Zion College of Engineering in 2012. And now she is pursuing her M.Tech degree in Computer Science and Engineering under the same university in Mount Zion College of Engineering.