COALA-System for Visual Representation of Cryptography Algorithms

Remya V
PG scholar, CSE
Younus College of Engineering and Technology,
Vadakkevila, Kollam, Kerala, India, pin: 691010

Rageena. M
Associate Professor, IT
Younus College of Engineering and Technology,
Vadakkevila, Kollam, Kerala, India, pin: 691010

***Abstract* - In the modern era people live their entire life online and the security for e-services are of utmost importance. Most of the engineering programs especially computer science has two or more security related subjects, but lack of active learning and practical experience. Cryptographic algorithms which solve security problems relay on specific mathematical areas such as modular arithmetic, probability and number theory. Unfortunately learners feel difficulty to follow the concepts due to the underlying sophisticated mathematics; it necessitates use of Interactive pedagogical tools that makes the standard content easier to learn and understand. This work, COALA describes an interactive visualization tool. which helps the learner's community to understand the mathematical concepts behind public cryptographic algorithms. This paper presents a novel software system for CryptOgraphic Algorithm visuAl representation (COALA) which was developed to understand the four basic cryptographic algorithms DES, AES, RSA and Diffie Helman key exchange protocol at its base level. The system allows users to follow the execution of these complex algorithms on real world examples in a step by step detailed view with the possibility of forward and backward navigation. The system presents all the intermediate results and details of every operation in the algorithm.**

Index terms- Data Security, AES, Algorithmic visualization, DES, RSA, Diffie- Hellman.

I. INTRODUCTION

Security has been a major issue since the time communication evolved. Everyone needs to have a method to communicate to a known person securely without a third person being able to understand the message. Over the centuries many people have proposed various algorithms or ways to encrypt and decrypt the messages. Some of these algorithms have been weak and easily broken while others have proven to be more rigid. However, most of these algorithms have been broken and there is a constant need for more rigid and powerful algorithms as the world is opening up to the Internet.

Some algorithms are simple and easy to understand whereas the more complex algorithms need some kind of visualization to understand. Hence, there is a clear necessity for a visualization tool, which provides step-by-step in-detail analysis of the encryption and decryption process of basic algorithms. The tool COALA gives user a feel of how a message would be encrypted in the backend.

II. MOTIVATION AND RELATED WORK

Algorithm visualization tools have been used in the education process for a long time and a lot of different tools have been created. But there is a relatively low number of tools that are used for security especially for cryptographic algorithms. Some of the visualization tools include :

Visualization applet[10] is a visualization tool in which people input text to be encrypted and the key. The applet shows only some rounds as a diagram with textual description of operations.

Grasp[11] is a visualization tool which provides protocol specification language that allows an arbitrary number of actors and message passing with appropriate commands needed for security protocols. Users can edit protocol commands during visualization allowing them to observe "what if" scenarios and they can control the protocol execution by moving a step forward or backward resetting or finishing the execution.

ECVisual[12] is a tool for visualization of elliptic curves based ciphers. This tool allow users to visualize elliptic curve over the real field and over a finite field of prime order perform arithmetic operations, do encryption and decryption and convert plain text to a point on an elliptic curve.

Digital Lego[13] In this atomic security units such as cryptographic operations have been represented as pieces of puzzle and security protocols are presented as structures built from these pieces.

DESVisual[14] is a visualization tool for the DES algorithm. It visually presents the execution of initial permutation and the first round of the DES algorithm, using 8 or16 bits input and 6 or 10 bits key, as a diagram.

Previously mentioned tools were developed to meet very specific requirements with a narrow scope of algorithms covered and parts of algorithms that are visualized. These tools only partially fulfill the need to improve the understanding of algorithms. Therefore the COALA tool is aimed to close the gap between the available tools and the needs of Data security.

III KEY COALA DESIGN ISSUES

COALA system is designed to be at the responding level of engagement [4], which means that students are answering questions concerning the visualization presented by the system. In order to have high interactivity [7], the COALA system was designed to enable students to control the execution of algorithms forward and backward, it allows them to configure the algorithm parameters before starting an execution, and it makes it possible for them to follow the result of every operation in any time. Abstractions used to present concepts are quite intuitive and thus easy to understand. The user interface in the COALA system is designed to show only relevant information at a time and it includes only the minimum set of necessary options. Windows, tabs etc are used to separate operations in algorithms and dropdown lists, textboxes etc are used for selection of inputs. Buttons are used to prompt underlying basic calculations in every operation. Algorithms presented in the COALA system are the ones that were found difficult for students to cope up with and at the same time the ones that represent concepts on which they are based on.

The COALA system was implemented in the Java programming language, using NetBeans IDE as a development environment. The Swing API was used for GUI development.

IV. DATA ENCRYPTION STANDARD

A. Short Description

The DES algorithm consists of initial permutation, 16 identical iterations and inverse of initial permutation. the expansion permutation which permutes and extends the right 32 bits to 48 bits in order to match the size of a round key. This expanded value is XORed with the round key for that iteration and the result is reduced to 32 bits using the substitution. After the substitution, the value is permuted once again, this time without the changes in the size, and XORed with the left 32 bits in order to make the right 32 bits for the next iteration. The left 32 bits for the next iteration are the right 32 bits from the current iteration. The iteration key creation includes an initial permutation of the original key and a left circular shift and a permutation in each iteration to create a 48-bit iteration key.

B. COALA for DES

The operations in the DES algorithm are executed on a bit level and hence in order to fully represent details of operations COALA decided to represent data and iteration keys as a matrix on the bit level.

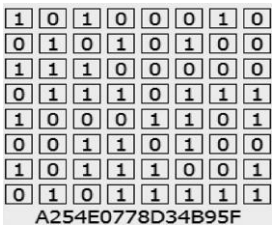

Fig 4a

One iteration of the algorithm execution is presented at one time in the COALA system. Since the DES algorithm iteration consists of several operations, each operation has its own tab in the window. The following window shows the first iteration of DES encryption. It represents the PC1 permutation. The initial key and the result of the PC1 permutation are shown and for each bit of the result the position in the initial key can be viewed (Fig 4b).

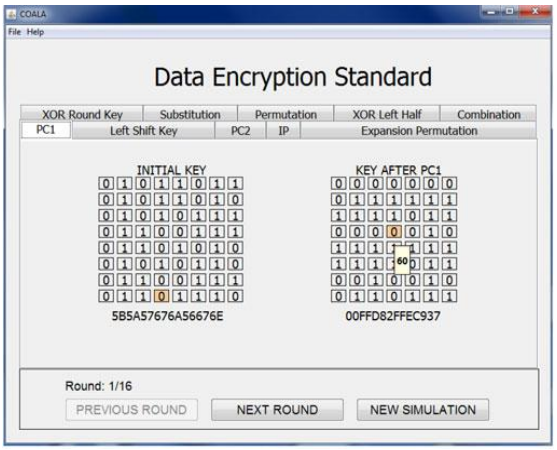

Fig 4b.Permutation step

Fig. 4c shows the substitution for the first iteration of the DES algorithm for encryption. The result of the XOR with the round key is shown divided in the eight groups of six bits and the result of the substitution is presented as the eight groups of four bits. Between these two values there are eight buttons representing eight S-box operations and by pressing any of them a student can obtain details about substitution for the pressed S-box (Fig. 4d).

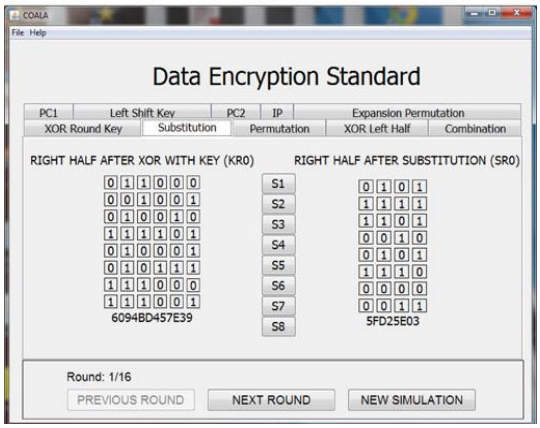

Fig. 4c. Substitution step

Fig 4d. S box-Example

V. AES

A. Short Description

AES is a iterated symmetric block cipher that operates with data several iterations before producing cipher text. On 2002 the National Institute of science and technology has selected a block cipher called RIJNDAEL (name by the inventors Vincent Rijimen and Joan Daemen) as the symmetric key encryption to encrypt sensitive but unclassified American federal information.

The AES algorithm consists of the initial XOR with the original key, nine identical iterations and the incomplete final iteration. Each iteration in the AES algorithm starts with the substitute bytes operation, which substitutes each byte of the state with a byte determined by a specially designed substitution matrix. The iteration continues with the shift rows operation, which performs a byte circular left shift on each row of the state matrix. The mix columns operation is next for all iterations except the last one, which does not have this operation. The mix columns operation makes that each byte in a column is calculated based on all four bytes in that column. Each iteration finishes with a simple XOR of the state matrix with the iteration key. Iteration keys are created by using the expansion function on the original key. The detailed explanations of the algorithms can be found in [15].

B. COALA Implementation

An important design choice was the way the data is represented in COALA. Since the operations in the AES algorithm are executed on byte level, it was decided that states and iteration keys are represented divided to bytes, as a 4x4 byte matrix, (Fig 5a) which is consistent with their original definition in the algorithm.

38	A9	1F	04
4E	26	0B	E3
78	21	4D	95
6A	02	47	AE

Fig.5a Byte matrix

Execution of the algorithm is organized in the way that one iteration is presented at one time in the COALA system. Tabs in the window are used to show each operation of the current iteration. The COALA system provides additional help while executing an algorithm. In each operation parts that are connected are highlighted with the same color when a mouse crosses some of these parts. Binary representation of XOR operation can be viewed as a tool tip where it is possible. Fig. 5b presents an add round key operation at the beginning of the first iteration of the AES algorithm for encryption, where both a tooltip and color are used as an aid. The initial state and the initial key are used in the XOR function and the result of this operation is shown.

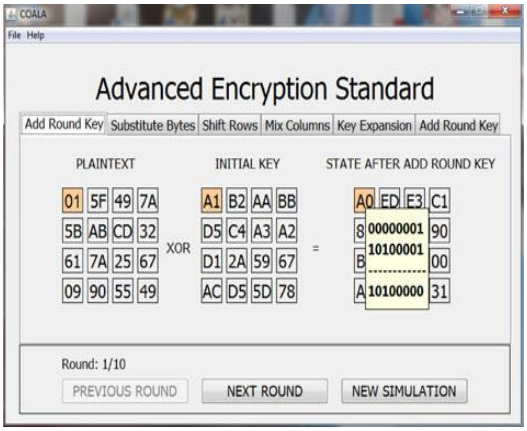

Fig 5b. Add round key step

Fig.5c shows the substitute bytes operation of the AES algorithm for encryption. The input state is shown on the left, and the resulting state after this operation is shown on the right. In the middle there is a matrix of buttons 4 X 4, in which each button can be pressed to show the S-box and how each byte of the input state is substituted in this operation.

Fig 5c Substitution byte step

The COALA system enables users to view details of some complex operations like mix column operation. Fig.5d depicts the mix columns operation of the first iteration of the AES algorithm for encryption where color is used as an aid. Here the multiplication of a state and the multiplication constant matrix is shown, along with the resulting state after the operation. Fig. 5e shows how the first byte of the state is obtained in the mix columns operation. This window is shown when a user clicks on the first button of the resulting state in the mix columns operation.

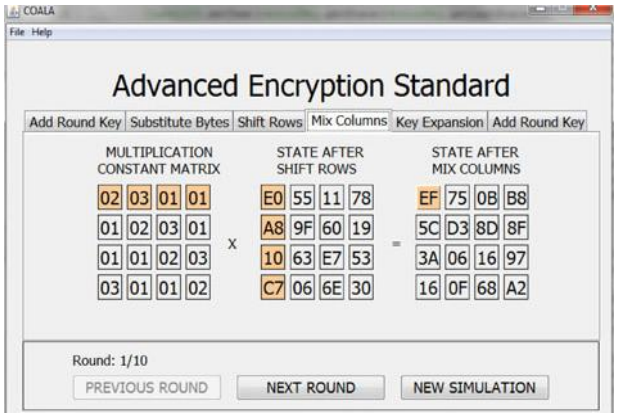

Fig 5d. Mix Column step

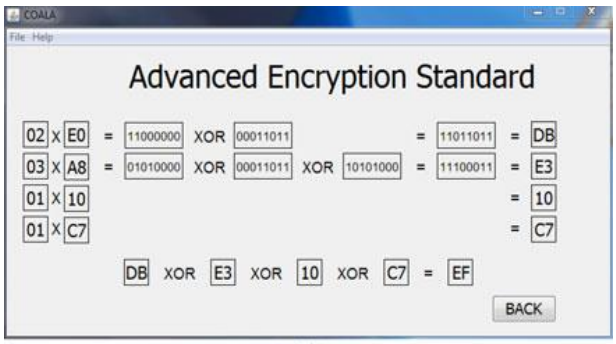

Fig 5e.Visual representation of Mix column step

The key expansion operation in AES, in which every word of the expanded key is generated from the previous words of the key, is also represented using buttons. Fig. 5f gives a portion of the extracted key that is needed for the first iteration of the AES algorithm for encryption. The initial key is shown, and so is the key for the first iteration.

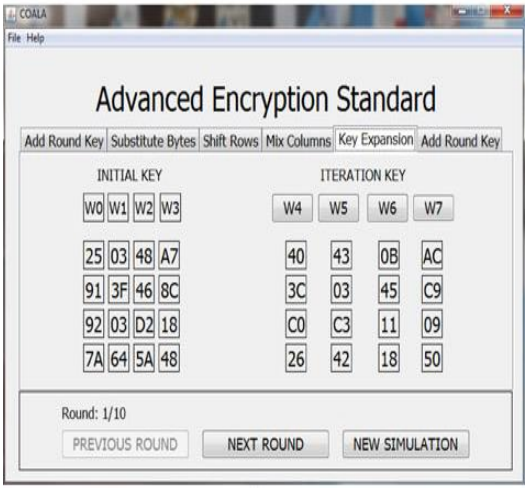

Fig 5f. Part of expanded key

Fig. 5g shows how the fourth word (W4) of the portion of the expanded key is obtained. This window is shown when a button W4 is pressed in the key expansion operation window.

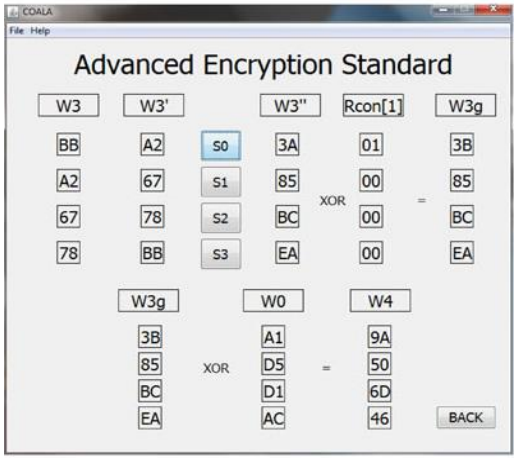

Fig 5g. Key generation

VI. COALA for RSA

A.Short Description

The RSA algorithm involves three steps: key generation, encryption and decryption. RSA involves a public key and a private key. The public key can be known by everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted in a reasonable amount of time using the private key.

The keys for the RSA algorithm are generated the following way: Choose two distinct prime numbers p and q. Compute n = p*q. n is used as the modulus for both the public and private keys. Its length, usually expressed in bits, is the key length. Compute φ(n) = (p − 1)(q − 1), where φ is Euler's totient function. Choose an integer e such that 1

< e < φ(n) and gcd (e, φ(n)) = 1; i.e., e and φ(n) are co-prime. Determine'd' as d ≡ e−1 (mod φ(n)). The public key consists of the modulus n and the public (or encryption) exponent e. The private key consists of the modulus n and the private (or decryption) exponent d, which must be kept secret. Encryption of a message M, which must be less than n, is done using public key and formula M^e mod n. Decryption of a message C, which must be less than n is done using private key and formula C^d mod n.

B.COALA Implementation

The key operation in RSA is an algorithm for fast exponentiation which is used for the calculation of RSA encryption and decryption. Fig.6a shows the visual representation of the algorithm for fast exponentiation in COALA. By positioning the mouse pointer on any intermediate result, a student can obtain a tooltip that shows how the result was calculated (Fig. 6a shows a tooltip for column 4). In Figure d(i) are bits from the binary representation of the exponent (659) and f represents intermediate results for each step of the algorithm for fast exponentiation.

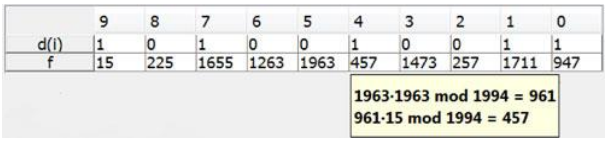

Fig. 6a. Visual representation of Fast Exponentiation

In COALA the key generation process for the RSA algorithm starts with a user choosing two private prime numbers p and q from two dropdown lists (p≠q). Then the public n is calculated by multiplying p and q. Also F(n) is calculated as (p -1)(q- 1). Next, a student chooses a public e from a dropdown list. The offered values for e fulfill required conditions (gcd(φ(n), e)=1 and 1 < e < φ(n)). Finally, the value for the private d is selected by a student from the dropdown list, which is populated based on the selected value for e, so that it fulfills the required condition (d ≡ e−1 (mod φ(n)). At the end, the public key (e,n) and the private key (d, n) are formed and the algorithm is ready to be used for encryption and decryption. The visualizing window is shown in Fig 6b.

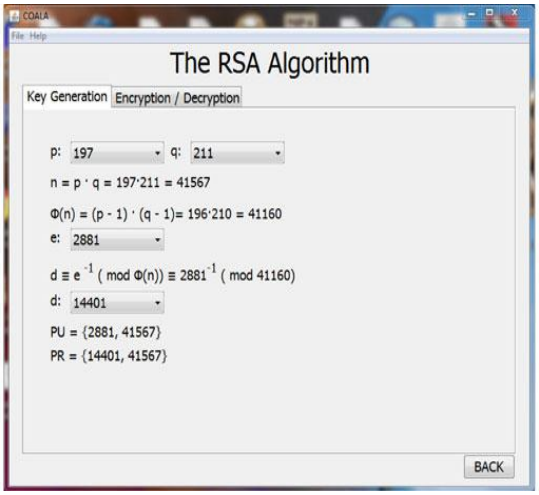

Fig 6b.Key Generation

The usage of the RSA algorithm for the encryption/ decryption (Fig. 6c) starts with a student entering a plaintext value for which the cipher text is calculated. The algorithm used for the calculation is the fast exponentiation algorithm. The table has as many columns as is the length of the binary representation of the exponent. The way that the value is calculated is shown as a tooltip when mouse is positioned on a column (Fig.6a). As a demonstration for each ciphertext value the calculation of the plaintext value is also represented.

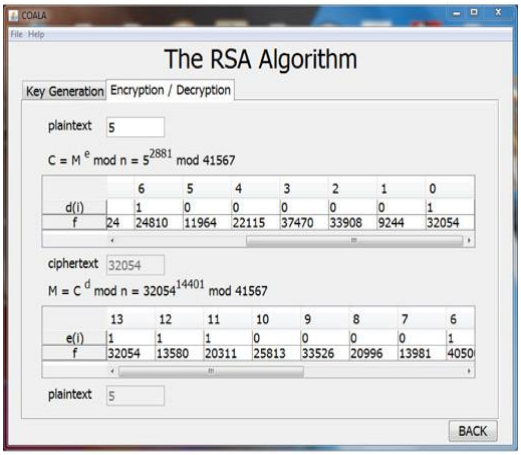

Fig 6c. Execution

VII.DIFFIE HELMAN KEY EXCHANGE PROTOCOL

A.Short Description

Diffie-Hellman is an asymmetric public key cryptography algorithm which is used for key exchange between two users. The main steps in Diffie Helman algorithm are: Select a prime number 'q' and one of its primitive root 'a'.Then users selects private value X_a , X_b and computes a^{Xa} mod q, a^{Xb} mod q respectively. The

selected values q & a will form the global public elements a^{Xa} mod q & a^{Xb} mod q will be the public keys for users A & B respectively. The Common shared secret key calculated for the session is $a^{Xb.\ Xa}$ mod q.

B.COALA for Diffie helman key exchange

In COALA visual representation was used to demonstrate details like representation of all possible values for a primitive root of a prime number. Fig. 7a is a table used to determine which values between 1 and 996 could be a primitive root for a prime number 997. There is one row in the table for each potential primitive root and only those who create a sequence without repetition are primitive roots for 997.

	a^988	a^989	a^990	a^991	a^992	a^993	a^994	a^995	a^996
986	792	261	120	674	562	797	206	725	1
987	185	144	554	442	565	332	668	299	1
988	993	36	673	922	675	904	837	443	1
989	442	452	372	15	877	960	296	623	1
990	156	902	665	330	681	218	468	712	1
991	100	397	609	334	987	60	637	166	1
992	501	486	561	186	67	662	678	598	1
993	491	30	877	480	74	701	187	249	1
994	675	966	93	718	837	480	554	332	1
995	74	849	296	405	187	623	748	498	1
996	1	996	1	996	1	996	1	996	1

Fig.7a Primitive root calculation

The global public elements selection process for the Diffie-Hellman algorithm in COALA starts with a user choosing a prime number q from a dropdown list. After that, the table which makes it possible to calculate primitive roots for q is shown.

In each row of the table sequence of different values is highlighted as shown in fig 7a. Only rows that are completely highlighted represent primitive roots for q and these values are offered in a dropdown list for 'a'. Then user chooses a value for 'a' which completes the global public elements selection process. The selection window for global elements q and a are shown in Fig.7b.

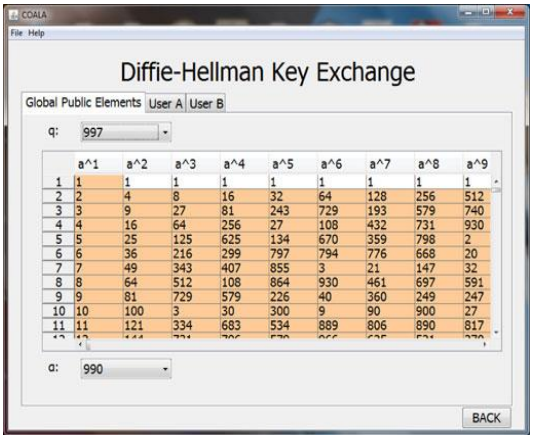

Fig 7b. Selection of global elements

COALA system shows the shared common secret key generation for two communicating parties in window containing tab for each user. The usage of the algorithm for user A (Fig.7c) for the Diffie-Hellman algorithm starts with a student choosing a private number X_a from a dropdown list, which offers only values that fulfills the required condition ($1 \leq X_a \leq q-1$). After that, the public value Y_a is calculated based on the selected value of X_a. At the end, the secret key for a user A, K_a is calculated based on the private value X_a and the public value Y_a. User B (Fig.7d) similarly calculates secret key, K_b. COALA clearly shows us that the secret key value calculated for a user A, K_a and the secret key value calculated for a user B, K_b are the same.

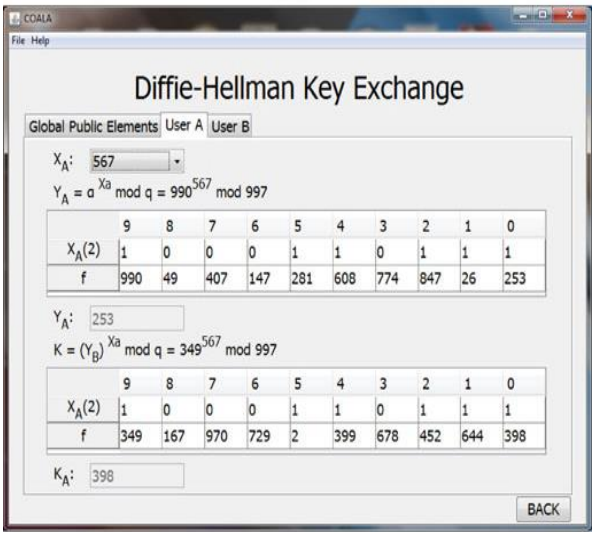

Fig 7c.Common secret key generation

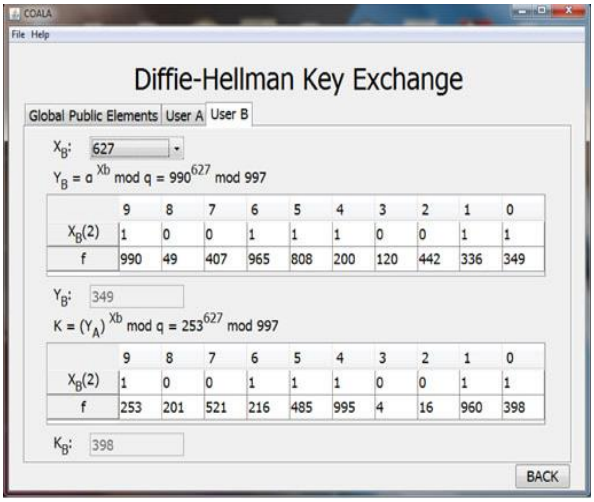

Fig 7d. Common secret key generation

VIII.ADVANTAGES OF COALA

COALA helps to understand the complicated mathematical concepts behind the cryptographic algorithms more easily and quickly. The system present all details and inner working of different components of the algorithm. It also helps us to learn the concepts and practice the computation steps using the user friendly interface .

IX. CONCLUSION

The paper presents a visualization tool which helps to understand the complicated mathematical concepts behind the cryptographic algorithms DES, AES, RSA , Diffie Helman more easily and quickly. The interaction between users and a visualization system is more important than the visualization contents which is provided efficiently by COALA system. COALA can be used in educational system where its active learning techniques increases students' knowledge accqusition and holds their attention longer than lecturing alone. With the tool, instructors are able to present all details and inner working of different components of the algorithm.

X.FUTURE ENHANCEMENT

COALA gives a step by step insight into working of four basic cryptographic algorithms : DES, AES, Diffie helman, RSA. The work runs as a stand-alone application and could later be converted into a web application so that the user need not download the application and instead run it from the browser.

REFERENCES

[1] Z. Stanisavljevic, V. Pavlovic, B. Nikolic, and J. Djordjevic, "SDLDS—System for digital logic design and simulation," IEEE Trans. Educ., vol. 56, no. 2, pp. 235–245, May 2013.

[2] G. Roßling,€ M. Schuer,€ and B. Freisleben, "The ANIMAL algorithm animation tool," ACM SIGCSE Bull., vol. 32, no. 3, pp. 37–40, Jul. 2000.

[3] C. A. Shaffer, M. L. Cooper, A. J. D. Alon, M. Akbar, M. Stewart, S. Ponce, and S. H. Edwards, "Algorithm visualization: The state of the field," ACM Trans. Comput. Educ., vol. 10, no. 3, article 9, pp. 1–22, Aug. 2010.

[4] J. Urquiza-Fuentes and J. A. Vēlazquez-Iturbide, "A survey of successful evaluations of program visualization and algorithm animation systems," ACM Trans. Comput. Educ., vol. 9, no. 2, article 9, pp. 1–24, Jun. 2009.

[5] C. D. Hundhausen, S. A. Douglas, and J. T. Stasko, "A meta-study of algorithm visualization effectiveness," J. Vis. Languages Comput., vol. 13, no. 3, pp. 259–290, Jun. 2002.

[6] T. Naps, G. Roessling, V. Almstrum, W. Dann, R. Fleischer, C. Hundhausen, A. Korhonen, L. Malmi, M. McNally, S. Rodger, and J. Vel_azquez-Iturbide, "Exploring the role of visualization and engagement in computer science education," SIGCSE Bull., vol. 35, no. 2, pp. 131–152, 2003.

[7] D. Schweitzer and W. Brown, "Interactive visualization for the active learning classroom," ACM SIGCSE Bull., vol. 39, no. 1, pp. 208–212, Mar. 2007.

[8] D. Schweitzer, D. Gibson, and M. Collins, "Active learning in the security classroom," in Proc. IEEE 42nd Hawaii Int. Conf. Syst. Sci., Jan. 2009, pp. 1–8.

[9] X. Yuan, R. Archer, J. Xu, and H. Yu, "A visualization tool for wireless network attacks," J. Educ., Inf. Cybern., vol. 1, no. 3, pp. 53–58, 2009.

[10] 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.

[11] 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.

[12] 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.

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

[14] J. Tao, J. Ma, J. Mayo, C. K. Shene, and M. Keranen, "DESvisual: A visualization tool for the DES cipher," J. Comput. Sci. Colleges, vol. 27, no. 1, pp. 81–89, 2011.

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