ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Performance Analysis of Hybrid Cryptographic Algorithm- A^{3}D Algorithm

-a New Age Cryptographic Algorithm

Anjan K Koundinya [1], Abhijith C [2], Arunraj[3], Deekshith N[4], Srinath N K[5], Jibi Abraham[6]

Assistant Professor, Dept. of Computer Science and Engineering, R. V. College of Engineering, Bangalore, India[1]

B.E Student, Dept. of Computer Science and Engineering, R. V. College of Engineering, Bangalore, India[2,3,4]

Professor and Dean Student Affairs, Dept. of Computer Science and Engineering, R. V. College of Engineering, Bangalore, India[5]

Professor, Dept. of CEIT, College of Engineering Pune(COEP), Pune, India [6]

ABSTRACT: The information exchanged over the internet may not be secured over public domain. Protecting the information transmitted over the network is as important as data security issue becomes proliferated. Cryptographic algorithms that exist may not provide security to variable application needs due to inherent demerits. With the advancement of network technology, internet attacks are also versatile and the traditional encryption algorithms may not suffice securing the information over network. The Hybrid Cryptographic algorithm is secured algorithm and computational advances to reduce time in encryption. One such algorithm is the A3D algorithm which has the benefit over many public key algorithms. The paper is intended to discuss the performance of the algorithm and its suitability with generic application of network security.

KEYWORDS: Multilevel random number generator; Elliptic curve cryptography(ECC); Jacobian symbol; Goldwasser- Micali.

I. INTRODUCTION

Every commercial application on network requires different levels of security, certain application requires high security and confidentiality of information over the network[1]. To enhance the confidentiality various measures have been taken up to improve cryptographic algorithm. These measures may not have remarkable effect on attacks that compromises security.Cryptographic algorithm need to be designed with a balance on computational power and security. Most of the algorithms either symmetric or asymmetric caters to any one of these needs. Hence to implement an algorithm thattakes case of these two factors is tedious or difficult, however certain changes in the existing algorithms like inclusion of point curve of Elliptic curve cryptography in GoldWasser Micali Algorithm would answer the needs.

A. **Elliptic curve cryptography(ECC)**

Elliptic Curve Cryptography is a promising asymmetric cryptographic algorithm with an approach to public-key cryptography based on the algebraic structure of elliptic curves over finite fields [2][3][4]. The primary benefit ofECC is that it requires a smaller key size compared to other cryptographic algorithms [5]. This reduces storage and transmission requirements, which leads to faster processing. This is very useful for implementing encryption on small devices with limited resources in terms of power, CPU and memory [6]. It is also very helpful in handling many encrypted sessions for large web servers. The strength of an asymmetric encryption algorithm such as ECC is found in the complexity of computing the inverse of the function used to generate the key. Creating the key is straight forward, but finding the inputs that were used to create the key is computationally infeasible. In ECC, the computationally intense problem is called “Elliptic Curve Discrete Logarithm Problem”, and involves the difficulty in computing the

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

discrete logarithm (exponent) from the result. And there are many hybrid cryptographic algorithms which uses ECC cryptosystem as a base making it an ideal choice.

B. **GoldWasser Micali Cryptosystem**

Goldwasser-Micali cryptosystem is an asymmetric key encryption[7] algorithm, which is based on probabilistic public-key encryption scheme. It is highly secured algorithm as the ciphertext generated will be several times larger than the initial plaintext [8]. Since the algorithm uses probabilistic encryption technique, a given plaintext may produce different ciphertexts each time it is encrypted. This has significant advantages, as it prevents from recognizing intercepted messages by comparing them to a set of known ciphertexts.

A probabilistic model of data encryption and the first implementation of this model is presented along with the required proof [10]. The security of this implementation is proved with the calculation of quadratic residue modulo composite numbers whose factorization is unknown. Survey of public key cryptosystems based on trapdoor systems is done in detail. The public key cryptosystems and their security are outlined. The quadratic residuosity problem is used for the mathematical explanation. An introduction to probabilistic encryption \cite{prob2} along with some key definitions is given. The quadratic residuosity is major concept involved in this encryption technique. The Goldwasser-Micali Encryption Scheme is provided along with proofs required to understand the scheme. Probabilistic encryption is the use of randomness in an encryption algorithm, i.e. the ciphertext generated will be different for each and every encryption of the same plaintext. Goldwasser- Micali is one such asymmetric probabilistic algorithm. Performance analysis of the algorithm is done based on the metrics like encryption time, decryption time and size of cipher text with varying plain text sizes and the results are compared with that of RSA. The mathematical model of the A^3DAlgorithm is discussed in [22].

II. RELATED WORK

Internet provides essential communication between billions of people and is being increasingly used as a tool for commerce. Security becomes a tremendously important issue to deal with. There are many aspects to security and many applications, ranging from secure commerce and payments to private communications and protecting passwords. One essential aspect for secure communications is cryptography \[19].

Cryptography is the science of coding and decoding messages so as to keep these messages secure. Cryptography not only protects data from theft or alteration, but can also be used for user authentication. In general, there are two types of cryptographic schemes which are categorized based on the number of keys that are employed for encryption and decryption typically used to accomplish these goals:

- **Secret key (or symmetric) cryptography**: Uses a single key for both encryption and decryption.
- **Public-key (or asymmetric) cryptography:** Uses separate keys for encryption and decryption.

In secret key cryptography, a single key is used for both encryption and decryption [1]. The sender uses a key to encrypt the plaintext and sends the ciphertext to the receiver. The receiver applies the same key to decrypt the message and recover the plaintext. Because a single key is used for both functions, secret key cryptography is also called as symmetric encryption. With this form of cryptography, it is obvious that the key must be known to both the sender and the receiver; that, in fact, is the secret. The biggest difficulty with this approach, of course, is the secret distribution of the key.

If it is a two-key cryptosystem in which two parties could engage in a secure communication over a communication channel without having to share a secret key [19]. In public key cryptography, one of the keys is designated the public key and may be advertised as widely as the owner wants. The other key is designated the private key and is never revealed to another party. Public-key cryptography algorithms that are in use today for key exchange or digital signatures include RSA, Diffie-Hellman and Digital signature Algorithm.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

III. SCOPE OF THE RESEARCH WORK

The components of hybrid cryptographic algorithm require keys to establish connection between the sender and receiver. Generation of these keys (private and public) will be based on multilevel random numbers. This involves two efficient pseudo random number generating algorithms namely Park Miller algorithm and Mersenne Twister algorithm. Park Miller algorithm generates 32 bit sequence of pseudo random numbers[14]. This algorithm is comparatively less complex in computation, power efficient and flexible [15]. Mersenne Twister (MT) algorithm is one of the most widely used pseudo random number generator for high performance computing applications such as financial computing [16]. Initially, a random seed is passed to Park Miller algorithm. The generated output is fed to Mersenne Twister as seed. Hashing function SHA-2 [17] [18] is used to eliminate any possibilities of finding initial seed used for multilevel random number system. The part of hash string is considered for key generation process.

Development of hybrid cryptographic algorithm involves combining the features of two efficient well known cryptographic algorithms namely Elliptic Curve Cryptography and Goldwasser-Micali algorithm. Elliptic curve cryptography offers equal security for a smaller key size compared to RSA [8], thereby reducing the processing overhead. Whereas, Goldwasser-Micali algorithm uses bitwise probabilistic encryption based on the quadratic residuosity concept which is more secured than RSA [11]. The designed hybrid cryptographic algorithm acts as a common interface to these algorithms.

Finally, the integrated cryptosystem is used in the process of encryption and decryption with the help of generated secret keys. Bitwise encryption and decryption operation is done based on the concept of quadratic residue. Operations at decryption stage will be reverse process of encryption operations.

The implementation of proposed hybrid cryptographic algorithm is restricted to ASCII character set. The designed algorithm made to work independently and it will not be integrated with any existing security suits as integration requires more work. The designed algorithm is limited to work in sequential processing environment.

IV. SYSTEM DESIGN

In any system, each component that makes complete system, contributes to the system with its own function. It is important to identify those components in order understand those in a better way. The initial design process of identifying these components, establishing a connection between those and understanding about their organization within the system is known as system organization. In this section overall system organization is described along with brief description of components that makes the system. Figure 1 shows the system organization which involves certain components that makes entire system. Each block represents a module or an entity involved with the system. Each module has its own function that is to be performed and is itself comprised of smaller tasks.

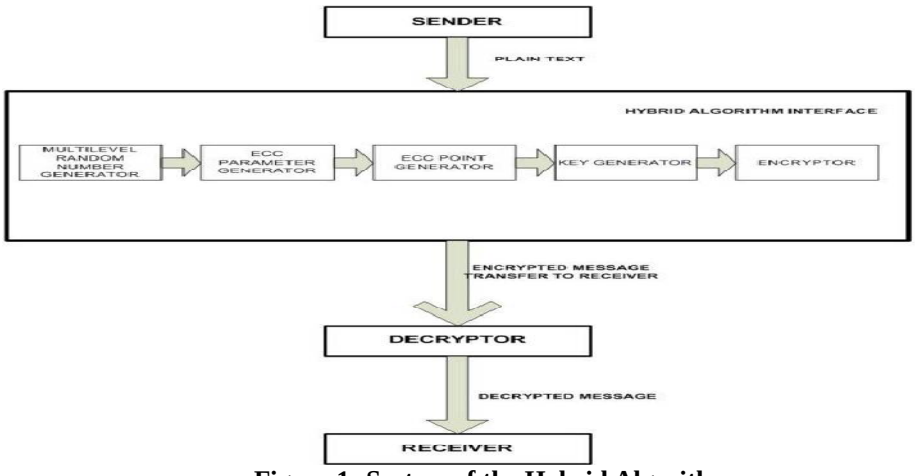

Figure 1: System of the Hybrid Algorithm

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Sender is a person who wants to send a message securely to receiver over a communication channel using the hybrid cryptosystem. The message which sender wants to send is called cipher text. Sender needs to create a file which contains the message he wants to send. Then using user interface he needs to browse the file to encrypt it. The encrypted message will be transferred to the receiver over a communication channel. Before delivering the messageto receiver, the cipher text is converted back to original message by decryptor. The random number generator [9][10][11][13][14] is important component of hybrid algorithm interface. This module computes pseudo random numbers and provides it to key generation process and ECC parameter generator.To finds suitable values for the elliptic curve parameters [6][20] - ***p, a and b***. Parameter p is obtained using random number generator. Parameters a and b should be generated such that, they should satisfy condition the condition:

$$4A^3 + 27B^2 \bmod P = 0$$

To computes the point P(X,Y) such that, it should lie on the ECC curve $Y^2 = X^3 + AX^2 + B$. Since ECC is implemented using prime field F(P), the condition:

$$Y^2 \bmod P = X^3 + AX + B \bmod P$$

should be satisfied by the generated point [20][21]. This component generates key required by encryptor and decryptor [21]. Using ECC parameters base point G is generated using point generator. Scalar multiplication is performed as k(G) which generates ECC public key. The points coordinates are converted as nearest primes as (p,q) which is used as Goldwaser Micali Algorithms private key. From the private key, public key (n,y) is obtained as explained in mathematical design in this paper. Using keys generated by key generator, encryptor will encrypt the message using bit-wise encryption technique [22]. Similarly, using bit-wise decryption technique the original message is retrieved by receiver.

V. PERFORMANCE ANALYSIS AND RESULTS

This section contains the results of the various performance analysis of developed hybrid cryptographic algorithm. The results are represented in tabular and graphical form. The evaluation metrics have been listed and the results are based on those metrics. The following parameters [3][4] have been considered for performance analysis:

- Plaintext size to ciphertext size ratio
- Encryption time
- Decryption time
- Total execution time
- Key generation time

Plaintext size to ciphertext size ratio-In this analysis, size of ciphertext is measured for particular sizes of plaintext. After five iterations, the average values are tabulated in Table 1. Both plaintext size and ciphertext size are in kB. The table is represented in graphical form in Figure 2, which shows linear increment in size of ciphertext with increase of plaintext size.

Table 1- Comparison of plaintext size and ciphertext size

Plaintext size(kB)	Ciphertext size(kB)
1	97
2	156
3	219.6
4	320
5	417.8
6	570.8
7	662.2
8	763.6

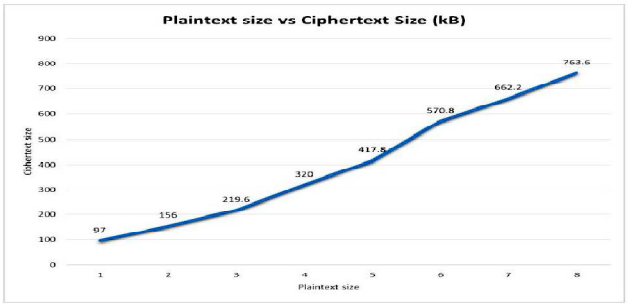

Figure 2: Plaintext size vs Ciphertext size

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Ratio of plaintext size to Ciphertext size-This analysis shows the change in size of ciphertext with respect to ciphertext size by computing the ratio of these two and same is given in table 2. The results of previous analysis are considered for this computation. The figure 3 is the graph of calculated ratio. By referring the graph, it can be concluded that the computed ratio is almost constant.

Table 2- Ratio of plaintext size to ciphertext size

Plaintext size(kB)	Ciphertext size(kB)	Ratio
1	97	0.0103
2	156	0.0128
3	219.6	0.0128
4	320	0.0125
5	417.8	0.0120
6	570.8	0.0105
7	662.2	0.0106
8	763.6	0.0105

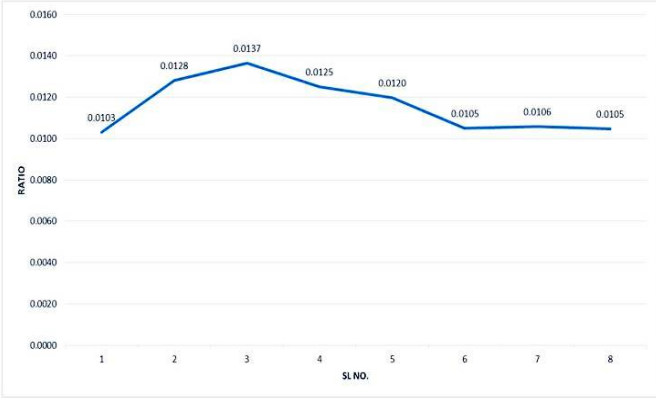

Figure 3: Ratio of plaintext size to Ciphertext size

Performance of random number generator-Random nu numbers which are unpredictable. In this analysis, performance of random number generator i.e. randomness of numbers generated is analysed. For this analysis, some numbers are generated using the random number generator and a graph is plotted. Graph in Figure 4 shows that, the numbers generated are very random.

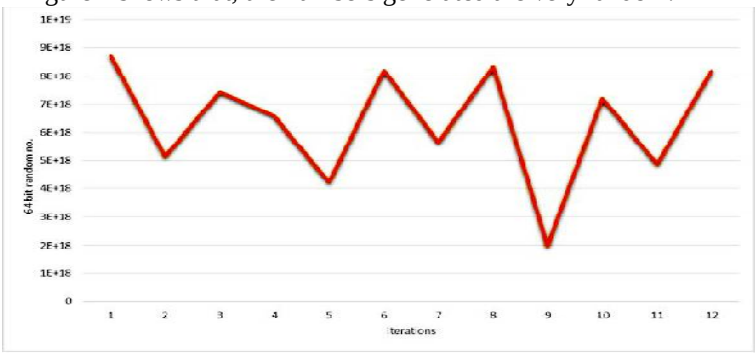

Figure 4: Performance of random number generator

Key generation time -Key generation time is the time required by key generator module to generate keys, which are then used in the process of encryption and decryption. In this analysis, key generation time with varying plaintext size is measured. During the analysis, it is observed that for first iteration, the key generator takes more time than that of for further iterations. This is due to initial computations that take place only in first iterations. So, faster execution is observed during subsequent iterations. Average key generation time of different iterations is given in table 3. Figure 5 shows that the key generation time becomes constant from second iteration.

Table 3: Key generation time

Iteration	Key generation time(sec)
1	0.3361
2	0.01976
3	0.01732
4	0.01413
5	0.01541

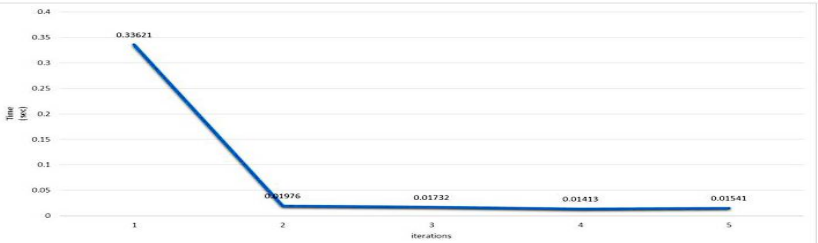

Figure 5: Key generation time

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Encryption time-Encryption time is the time taken by encryptor module to convert plaintext into ciphertext using keys generated by key generator. In this performance analysis, the encryption time is measured by varying plaintext size. In table 4, encryption time during first iteration and average encryption time during subsequent iterations with different plaintext sizes are tabulated. The same results are used to plot a graph which is shown in Figure 6.

Table 4: Encryption time

Text size(kB)	Enc. time (first iteration)(sec)	Enc. time(further iterations)(sec)
1	1.905	0.769
2	2.871	1.038
3	3.050	1.393
4	4.008	2.241
5	5.632	3.182
6	6.359	5.209
7	9.476	6.131
8	10.481	7.818

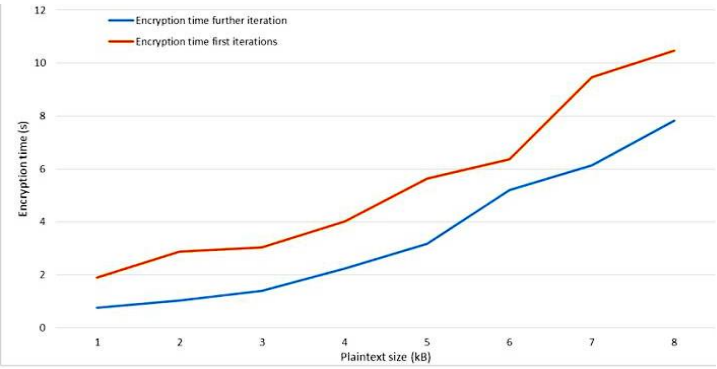

Figure 6: Encryption time

Decryption time-Decryption time is the time taken by decryptor module to get back plaintext from ciphertext. Decryption time with varying plaintext size is measured during first iteration and subsequent iterations separately and the same is tabulated in table 5. It is observed that, the decryption time during first iteration is more compared to that during subsequent iterations. The same values are plotted in graph and are shown in figure 7. The graph shows linear increment in decryption time with the increase in plaintext size.

Table 5: Decryption time

Text size(kB)	Dec. time (first iteration)(sec)	Dec. time(further iterations)(sec)
1	0.369	0.140
2	0.424	0.225
3	0.509	0.332
4	0.685	0.437
5	0.829	0.548
6	1.043	0.729
7	1.138	0.831
8	1.290	0.960

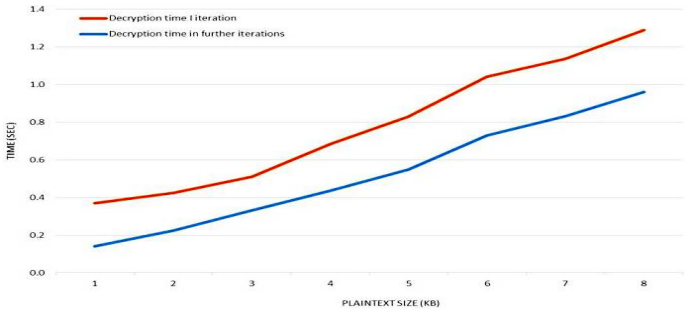

Figure 7: Decryption time

Total execution time-Total execution time is the time taken by designed cryptosystem to complete its execution. This includes time taken by random number generator, key generator, encryptor and decryptor to complete their respective processes. It is observed that, total execution time gets reduced after first iteration. Table 6 shows the execution time for first iteration and average execution time during subsequent iterations. Figure 8 is the graph obtained by plotting same results.

Table 6: Total Execution time

Text size(kB)	Exe. time (first iteration)(sec)	Exe. time (further iterations)(sec)
1	2.618	0.786
2	3.670	1.056
3	3.898	1.409
4	5.048	2.258
5	6.772	3.199
6	7.730	5.225
7	10.939	6.146
8	12.086	7.834

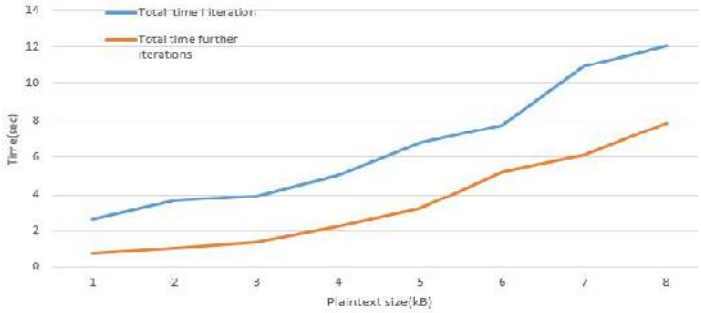

Figure 8: Total Execution time

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

Performance analysis has been conducted based on certain standard evaluation metrics. It is observed that, the system execution time is more during first iteration and it becomes constant in subsequent iterations. Also, the generated ciphertext is large when compared with the plain text size. The ratio of plaintext size to ciphertext size remains constant in all the iterations.

VI. CONCLUSION AND FUTURE WORK

Cryptographic algorithms are integral part of the secured communication over the internet. The existing well known algorithms may not be feasible against security attacks in future. Hybrid cryptosystem developed in this project provides a shield against brute force attacks and the cryptosystem mainly concentrates on increased level of security. Key features of ECC algorithm and Goldwasser-Micali algorithm have been implemented in this project successfully. An effective random number generator have been developed to strengthen the cryptosystem. The developed cryptographic algorithm uses 64-bit key for encryption and decryption. The encrypted message is semantically secure because of the bit-wise probabilistic encryption technique used and hence process of deciphering is impossible for the intermediaries. Performance analysis shows that the ratio of plaintext size to ciphertext size is constant. Also, the encryption and decryption time are less in comparison with other existing standard algorithms under similar specifications.

ACKNOWLEDGMENT

Prof. Anjan K Koundinya would like to thank Late. Dr. V.KAnanthashyana, former Head, Dept. of CSE, MSRIT, India for igniting the passion for research. His vision and aspiration has inspired me to pursue and reach completion of my Ph.D research work.

REFERENCES

1. Ayushi, "A Symmetric Key Cryptographic Algorithm", International Journal of Computer Applications, Vol. 1(1), 2010, pp. 01-05.
2. Sravana Kumar D, Suneeth, Chandrasekhar A, "Encryption of data using Elliptic Curve over finite fields", International Journal of Distributed and Parallel Systems (IJDPS), Vol.3(1), 2012 , pp. 301-308.
3. Pavithra S, Ramadevi E, Study And Performance Analysis of Cryptography Algorithms", International Journal of Advanced Research in Computer Engineering & Technology, Vol. 1(5), July 2012, pp. 82-86.
4. Diaa Salama Abd Elminaam, Hatem Mohamed Abdual Kader, Mohiy Mohamed Hadhoud, \Evaluating The Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol. 10(3), May 2010, pp. 213-219.
5. Moncef Amara, Amar Siad, "Elliptic Curve cryptography and its applications", Proceeding of the 7th International Workshop on Systems, Signal Processing and their Applications (WOSSPA), France, 2011, pp. 247-250.
6. Prabu M, Shanmugalakshmi R, "A Comparative and Overview Analysis of Elliptic Curve Cryptography Over Finite Fields", Proceeding of the International Conference on Information and Multimedia Technology, 2009, pp. 495-499.
7. Nafeesa Begum J, Kumar K, Sumathy V, "Multilevel Access Control in Defense Messaging System Using Elliptic Curve Cryptography", Proceeding of the second International conference on Computing, Communication and Networking Technologies,2010.
8. Song Ju, "A Lightweight Key Establishment in Wireless Sensor Network Based on Elliptic Curve Cryptography", IEEE, 2012, pp.138-141.
9. Shafi Goldwasser, Silvio Micali, "Probabilistic Encryption", Journal of computer and system sciences, 1984, pp. 270-299.
10. Orhio Mark Creado, Xianping Wu, Yiling Wang, Phu Dung Le, "Probabilistic Encryption: A Comparative Analysis against RSA and ECC", Proceeding of the Fourth International Conference on Computer Sciences and Convergence Information Technology,2009, pp. 1123-1129.
11. Sammy Kwok, Edmund Lam, "FPGA based High speed True Random Number Generator for Cryptographic Applications",IEEE, 2006.
12. Bang-Ju Wang, Hong-Jiang Cao, Yu-Hua Wang, Huan-Guo Zhang, "Random Number Generator of BP Neural Network Based on SHA-2 (512)", Proceeding of the Sixth International Conference on Machine Learning and Cybernetics, Hong Kong, 2008, pp. 2708-2712.
13. Rohith S, Bharatesh N, FPGA Implementation of Park-Miller Algorithm to Generate Sequence of 32-Bit Pseudo Random Key for Encryption and Decryption of Text Message, International Journal of Engineering Research and Applications, Vol. 3(5), Sep-Oct 2013, pp. 1246-1252.
14. Jingjing Lan, Wang Ling Goh, Zhi Hui Kong, Kiat Seng Yeo, A Random Number Generator for Low Power Cryptographic Application, Proceeding of the International SOC design Conference (ISOCC), 2010, pp. 328-331.
15. Yuan Li, Jiang Jiang, Hanqiang Cheng, Minxuan Zhang, Shaojun Wei, An Efficient Hardware Random Number Generator Based on the MT Method, Proceeding of the 12th International Conference on Computer and Information Technology, 2012, pp. 1011-1015.
16. Wanzhong Sun, Hongpeng Guo, Huilei He, Zibin Dai, Design and Optimized Implementation of the SHA-2 (256, 384, 512) Hash Algorithms, 2007, pp. 858-861.
17. Sklavos, Koufopavlou O, On The Hardware Implementation of the SHA-2 (256, 384, 512) Hash Functions, 2003, pp. 153-156.
18. Joseph Migga Rizza, Guide to Computer Network Security, Springer International Edition, ISBN: 978-03872-0473-4, 2009.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 5, May 2016

19. Apostolos Fournaris P, Odysseas Koufopavlou, "Creating an Elliptic Curve Arithmetic unit for use in Elliptic Curve cryptography", IEEE, 2012, pp. 1457-1464.
20. Kimmo Jarvinen U, Jorma Skytta O, "High-Speed Elliptic Curve Cryptography Accelerator for Koblitz Curves", Proceeding of the 16th International Symposium on Field-Programmable Custom Computing Machines, Finland, 2008, pp. 109-118.
21. Julien Bringer, Herve Chabanne, Malika Izabachene, David Pointcheval, Qiang Tang, Sebastien Zimmer, "An Application of the Goldwasser-Micali Cryptosystem to Biometric Authentication", Proceeding of the 12th Australasian Conference on Information Security and Privacy, Queensland, Australia, 2007, pp. 1-10.
22. Anjan K, Abhijith C, Arunraj, Deekshith N, Design and Mathematical Model of Hybrid Cryptographic Algorithm- A3D Algorithm, IJARCCE, Vol 3, Issue 6, Jun 2014

BIOGRAPHY

Anjan K has received his B.E degree from Visveswariah Technological University, Belgavi, India in 2007 And his master degree from Department of Computer Science and Engineering, M.S. Ramaiah Institute of Technology, Bangalore, India. He has been awarded Best Performer PG 2010 for his academic excellence. He is pursuing Ph.D in Computer Science and Engineering from VTU, Belgavi. He is currently working as Assistant Professor in Dept. of Computer Science and Engineering, R V College of Engineering, Bengaluru, India.

Srinath N K has his M.E degree in Systems Engineering and Operations Research from Roorkee University, in 1986 and PhD degree from AvinashLingum University, India in 2009. His areas of research interests include Operations Research, Parallel and Distributed Computing, DBMS, Microprocessor. His is working as Professor and Dean Student Affairs, Dept of Computer Science and Engineering, R V College of Engineering.

Jibi Abraham has received her M.S degree in Software Systems from BITS, Rajasthan, India in 1999 and PhD degree from VisveswariahTechnologicalUniversity, Belgavi, India in 2008 in the area of Network Security. Her areas of research interests include Network routing algorithms, Cryptography, Network Security of Wireless Sensor Networks and Algorithms Design.She is working as Professor and Head in Dept. of CEIT, College of Engineering Pune.

Abhijith C has received his B.E degree in Computer Science and Engineering from R V College of Engineering, Bangalore, India in 2014. His areas of research interests include Computer Networks, Cryptography, Network Security. He is currently working as a Graduate Software Engineer at Aurigo Software Technologies Pvt. Ltd.

Arunraj has received his B.E degree in Computer Science and Engineering from R V College of Engineering, Bangalore, India in 2014. His areas of research interests include Computer Networks, Cryptography, Network Security. He is currently working as a Graduate Software Engineer at Sapient.

Deekshith N has received his B.E degree in Computer Science and Engineering from R V College of Engineering, Bangalore, India in 2014. His areas of research interests include Computer Networks, Cryptography, Network Security. He is currently working as a Graduate Software Engineer at Tesco Hindustan Service Centre.