International Journal of Security and Its Applications
Vol. 11, No. 3 (2017), pp.17-28
http://dx.doi.org/10.14257/ijsia.2017.11.3.02

Data Encryption using Advanced Encryption Standard with Key Generation by Elliptic Curve Diffie-Hellman

Samiksha Sharma* and Vinay Chopra

Department of Computer Science and Engineering, DAV Institute of Engineering and Technology, Jalandhar, Punjab 144008, India

Department of Computer Science and Engineering, DAV Institute of Engineering and Technology, Jalandhar, Punjab 144008, India
samikshadogra1992@gmail.com, vinaychopra@yahoo.co.in

Abstract

In today's era ubiquitous computing is worldwide adopted. Internet is the main root for providing a ubiquitous network for communication between different people across the world, such communications can be through a wireless channel or wired channel that helps in getting messages or confidential information exchanged between different groups of people. Data security is of utmost importance because of wireless communications on insecure network. So the concept of cryptography is emerged which is nothing but known as an art of writing a secret code and it provide diverse set of services for protecting data over network such as authentication, confidentiality, non-repudiation and integrity. Cryptography offers wide range of algorithms which can help to guard communications over an insecure network such as symmetric encryption techniques which uses one key for encryption and decryption. For a symmetric cipher security can be compromised as it uses a single key, with this an advantage comes out while using an asymmetric security technique that makes use of a pair of keys to secure communications over unsafe channels. In this paper the positive characteristics of both the techniques discussed above are taken and a hybrid approach is used to guard messages on timid wireless medium. AES which is known as symmetric algorithm is combined with ECDH algorithm that is asymmetric by nature and is an amalgam of ECC and Diffie-Hellman – anonymous key agreement protocol. Different text files are taken as input to the model with varying sizes. Encryption and decryption is performed using Advance encryption standard (AES) whereas ECDH will help in securing the communication for a session set up between client and server by generating key for AES. Also Diffie-Hellman will provide security by establishing a shared secret between client and server after successful key agreement. At last analysis of proposed model is done on the basis of different metrics like storage, encryption time, decryption time, correlation and avalanche effect. Proposed approach has been proven effective in reducing the gaps discovered in the present literature.

Keywords*: Advance Encryption Standard; Elliptic Curve Diffie-Hellman; Encryption; Decryption; Correlation; Avalanche Effect*

1. Introduction

In present era the requirement of internet for wireless communication is rising day by day and thus there is a need of security to guard such communication by users on insecure wireless channel. Data sent over the communication channels is susceptible to attacks because of sensitive information it contain. To defend the data

* Corresponding Author

ISSN: 1738-9976 IJSIA

from external threat the concept of Cryptography is emerged. Cryptography is defined as "An art of writing a secret code" [1], Methodology of writing such code is cipher and text is converted into cipher text which is commonly called Encryption whereas the reverse practice of converting a cipher text into normal text is known as Decryption. Cryptography can be categorized as classical and modern, classical cryptography techniques were used to foil eavesdropping and message interception problems whereas the modern cryptography techniques are more secure and used for high speed communications. Modern cryptography techniques are more secure than the classical ones and are widely used such as DES, 3DES, AES, ECC, ECDH, RSA *etc*. Figure 1 represents the terminology of encryption and decryption process.

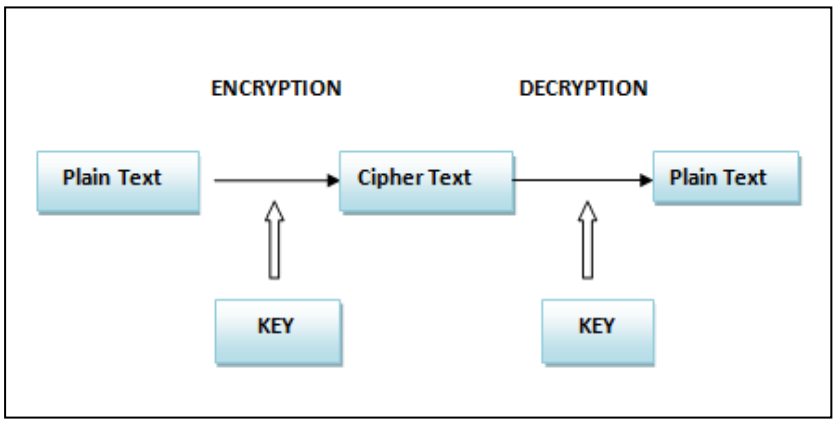

Figure 1. Encryption/Decryption Terminology

Plain Text: Original text which user uses for the communication purpose is termed as plain Text. For example bob sends "how are you" to alice here the plain text is "how are you".
Cipher Text: Plain text is transformed into a message(cipher text) which cannot be interpret by third party out of communication.
Example : "bye" is converted into "#@a%".
Encryption: Encryption is a procedure of transforming the plain text into the cipher text which is in non-readable form.
Decryption: Decryption is a procedure of transforming the encrypted text back into the plain text.
Cryptography must ensure four basic data protection requirements which are authentication, privacy, integrity and non – repudiation. From [2] we can define these requirements as:
Authentication -Where we have to verify user's identity involved in communication.
Privacy - To ensure no third person can intercept the message.
Integrity - Ensures that original message and received are identical *i.e.* no alteration of data.
Non-repudiation- Here we need to verify the sender's identity.

2. Literature Survey

This section will have the sight of all research work done in field of Advance Encryption Standard. In the past years lot of research has been done in the area of Advance encryption cryptography Technique, various cryptography algorithms have been evaluated on the basis of different metrics. Many cryptography models were developed using the rewarding features of AES cryptography which are proved effective in enormous number of fields. The survey presents the research work where AES is compared with different other algorithms and has been proven

effective in terms of computation speed and security. At last it covers the previous discovered hybrid AES models where AES is implemented with other cryptography algorithms such as RSA, DES, ECC and 3DES. Major Highlights are mentioned below.

Table 1. Literature Review covering the Major Findings in the Field of Advanced Encryption Standard

S.No.	Proposed Work
[3]	Presents comparative analysis between various symmetric techniques and at the end it is concluded that AES requires medium memory size as compared to other symmetric techniques and the strength of the algorithm in perspective of security is excellent.
[4]	Accomplishes comparison between different encryption techniques such as AES, DES and RSA on the behalf of different metrics like memory requirement and computation time. AES require less time for encryption than RSA and DSA.
[5]	Implemented three cryptography techniques DES, AES and RSA and compared their performance on the basis of simulation time. The results on different datasets proved that AES requires less time for encryption and decryption.
[6]	Proposed a model for timing evaluation based on the random generation of numbers method to calculate consumption of time of well known three cryptography algorithms RSA, AES and DES.
[7]	Stated for cloud security AES is considered as best cryptography technique. Also it provides shield adjacent to different attacks such as differential attack, recovery attack, key attack and square attack.
[8]	Demonstrated that on accumulating additional rounds (Nr) 16 to AES more computational time is required to break the security of algorithm. Also it compared AES with other algorithms like DES, TDES and proved AES faster.
[9]	Presents a comparison study is done for AES and DES which concludes that for less memory requirement AES is better. For the same file size it requires 10.2 MB and DES requires 43.3 MB, also the simulation time of DES is greater than AES.
[10]	Proposed a hybrid approach combining AES and RSA and gave the mathematical theory behind AES and RSA. It also covers problems related to computation, different possible attacks with the preventive measures.
[11]	Introduced a hybrid approach to secure Bluetooth transmissions where AES keys are encrypted by RSA and this approach takes the advantages of both AES and RSA thus highly secure.
[12]	Presents an approach where ECC plays an important role in encryption and decryption of data whereas for setting up a secure session Diffie-Hellman key exchange is there, that provide security using shared secret.
[13]	Investigated an approach where a secure replica of SMS model is made using AES and ECC for E-commerce payment. Various mobile E-commerce models have been studied and the ideal one among them has been chosen after application of AES-ECC.
[14]	Proposed a system with a hybrid approach using AES-ECC which has improved the creation of digital signature and superior authentication. Also in terms of flexibility and versatility the design has been proven good.

[15]	Presents amalgam of elliptic curve cryptography and Diffie-Hellman an anonymous key agreement protocol to provide forward secrecy in web applications using HTTPS.

AES has been widely adopted as it has been proved secure and most efficient in terms of storage requirements and computation speed. Also the encryption time of AES is less than other algorithms so far discussed. AES is thus faster and simple in terms of architectural design. Great results have been discovered by integrating AES with other cryptography algorithms and are widely used in enormous number of fields. Irrespective of all the work which has been done in this domain there are still some of the research gaps found in the present literature [8], In previous work, evaluation of different techniques is done on the basis of encryption time against different file sizes. When file size will increase encryption execution time will increase which is considered in the previous work but most importantly storage space for encrypted file size will also increase that will definitely affect the efficiency of the algorithm. So this approach may not be useful while dealing with cloud data security. The key generation method used in previous work is Polybius square. Ciphers generated by this technique are not secure because of monoalphabetic substitution *i.e*. same digits are used to identify the character each time, thus it can be exposed to frequency analysis. In previous work, only one parameter is considered which is encryption execution time whereas for validating a cryptography algorithm we need to have knowledge of strength, performance and weakness of the algorithms that can be identified by several metrics that include encryption time, decryption time, storage, avalanche effect, correlation, architecture, flexibility and many more that give deeper insight of the performance of algorithm.

3. A Brief Preface to AES

AES is announced as a federal information Processing standard by NIST (National institutes of standards and technology) in 2001. AES is recurrently used encryption technique due to its high security, efficiency and simplicity. It uses the same key for both encryption and decryption process and known as symmetric block cipher. It uses three block ciphers AES-192, AES-128, AES-256. There are different rounds of processing according to the block size such as 10 rounds for 128- bit key, 12 rounds for 192-bit key and 14 rounds for 256-bit key. Different steps for encrypting data with AES are given below:

Key Expansion- Rinjndael's Key Schedule is used to calculate the round key using the cipher key.

Initial Round- Add round key: Bitwise XOR operation is used to combine each byte of the state with the derived round key.

Different Rounds of Processing

- Sub Bytes: every Byte is replaced with another using the lookup table, a non linear kind of substitution.
- Shift rows: This is called transposition step where each row will by cyclically shifted to number of times required.
- Mix columns: four Bytes of each column are combined in a state matrix.

Final Round

- Sub Bytes
- Shift Rows
- Add Round Key

So, the final round will not have mixing of columns. During Decryption the processing rounds will be same but the only difference is Inverse of every processing round will be executed. If in encryption we have sub bytes then in decryption it will be Inverse sub

bytes. Similarly for shift rows and mix columns in encryption there will be Inverse shift rows and inverse mix columns for decryption.

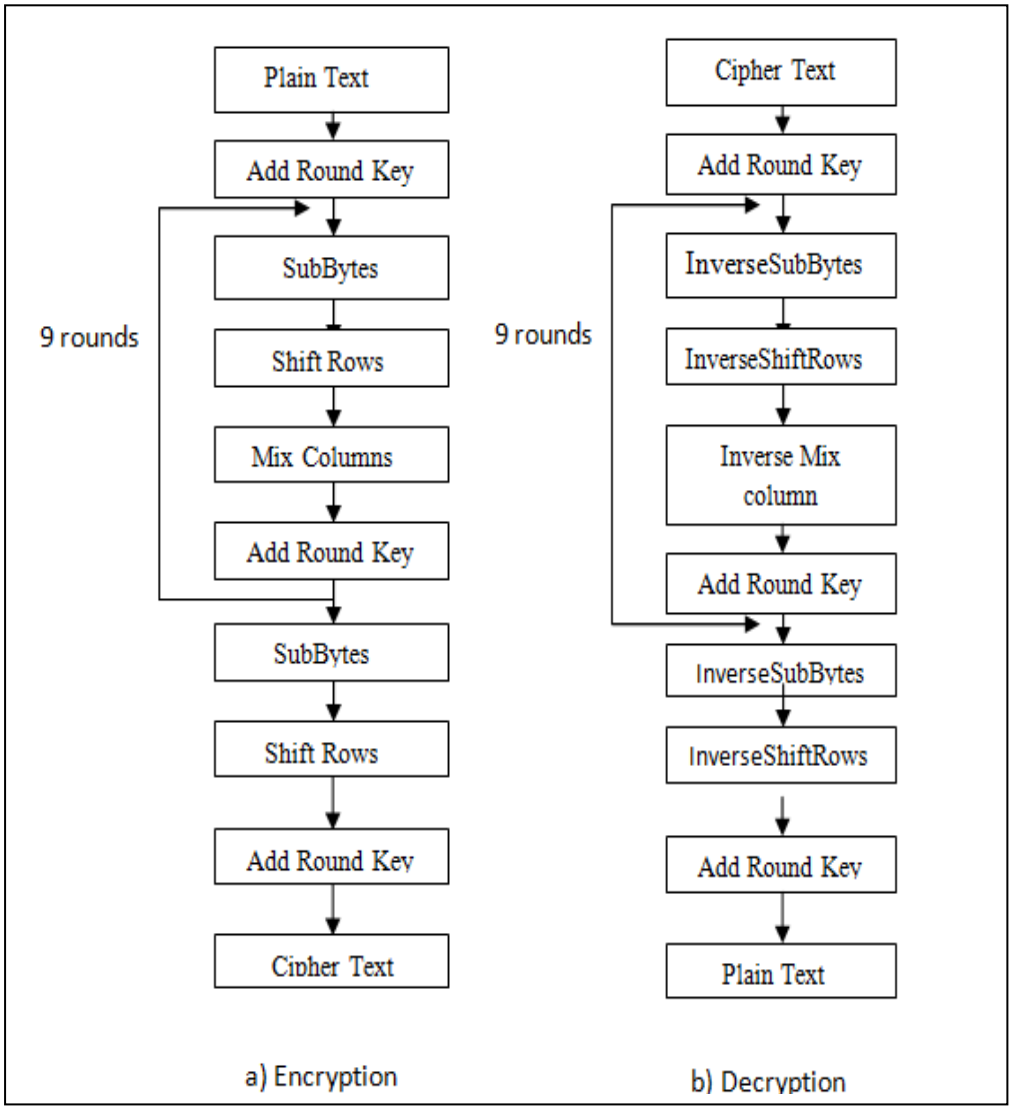

Figure 2. AES-128 bit Block Encryption/Decryption

4. Proposed Methodology

In Proposed model AES and ECDH are used for text file encryption. Input text file is transformed into encrypted form using AES algorithm with key generated by ECC and Diffie-Hellman will help in generating a shared secret which is then combined with ECC key and uploaded to the server. After successful key agreement client would be able to decrypt. The above methodology is implemented in JAVA 8 using Eclipse an open source platform that allows a developer to create a customized development environment (IDE). The above experiment is conducted on different text files with different sizes (KB), evaluation is done on the basis of different parameters like correlation, Avalanche effect, storage, encryption time and decryption time. Given below Figure 3 illustrate the complete process of AES-ECDH Encryption and Decryption.

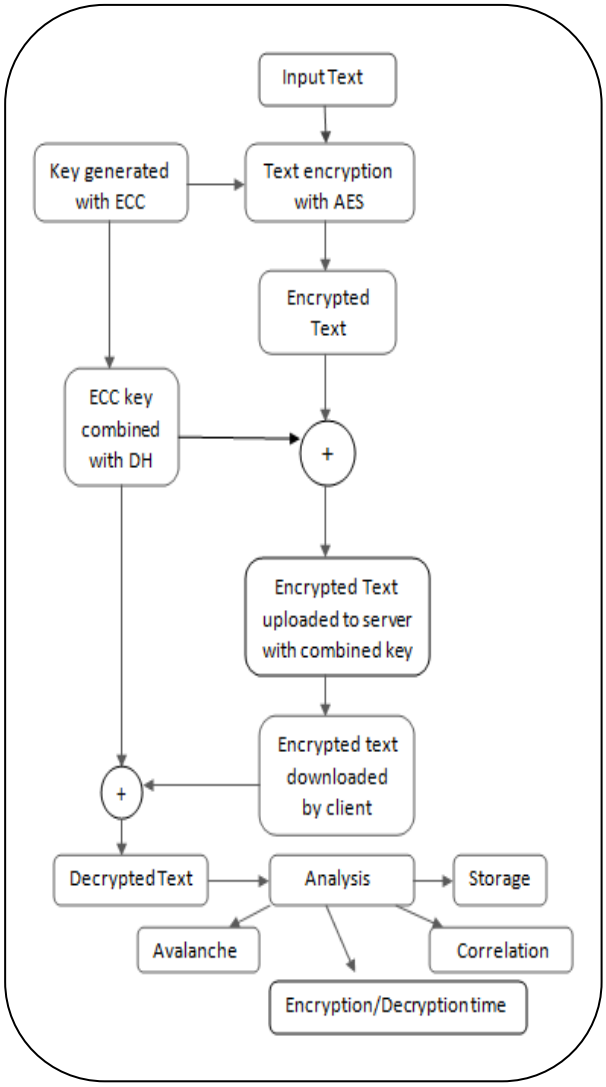

Figure 3. AES-ECDH Encryption/Decryption

Step 1: Different text files of different sizes are taken as input, Size of files in Kb. Set of files taken as input are license.txt, new.txt, new1.txt, readme.txt and example.txt. Client will request a file from the server and after the client's request file will be selected by server for Encryption.

Step 2: After taking text file as input, Elliptic curve will generate different private and public key pairs. Encryption is performed by Advance encryption standard and AES will encrypt the text file using one of the key pairs generated by Elliptic curve defined over a field. From the different key pairs one key will be kept secret with the server say **'d'** and one key will be secret to the client say **'e'**.

Step 3: Elliptic Curve Diffie – Hellman will establish the shared secret between client and server by making a successful key agreement between both of the communicating parties. Let's see how the agreement will be made in order to generate the shared secret among client and server.

- Initially they have to agree upon domain parameter ***(p, a, b, n, G, h)***.
- In the above specified domain parameters **'p'** is known as field on which elliptic curve is defined, **'a'** and **'b'** are values defined over the curve, **'G'** is called generator point which is fixed for a curve and known to both communicating parties, **'n'** a prime

order of generator point **'G'** and **'h'** is known as co-factor that tells about number of points on the curve, for h=1 points will be uniformly distributed over the curve.

- Each party will have a key pair generated by elliptic curve, one private key owned by both server and client. Say server has **'d'** and client has **'e'** as their corresponding private keys then they both will generate public key. **'G'** is known as generator point and will be fixed for both of them, known to both.
- Server will compute its public key say ***Q(a) = dG*** similarly client will computes its public key say ***Q(b) = eG***.
- Both will calculate shared secret now for that they will exchange their public keys with each other which is ***Q(a)*** is passed to client and ***Q(b)*** is passed to server.
- Server will compute ***dQ(b)*** and client will compute ***eQ(a).***
- Let shared secret be denoted by **'S', S = *dQ(b)= e Q(a) = deG.***

So, the shared secret will be in between server and client for a particular session and if a third party wants to break shared secret, they have to solve the discrete logarithm problem. When AES will be encrypting the input text file on the other side Diffie-Hellman will be performing the key agreement between client and server. If key agreement will be successful only then client will be able to decrypt the encrypted text file.

Step 4: AES will encrypt the message using the key generated by Elliptic Curve Cryptography. After encryption is done, the encrypted text is uploaded on the server with the combined key *i.e*. another key as shared secret obtained from Diffie-Hellman. Client will download that encrypted file from the server and decrypt that file using the combined key formed by ECC and DH but only after the successful key agreement that will establish a shared secret among client and server.

Step 5: After decryption is successfully done client will have the original file. At last after encryption and decryption is successfully done, Analysis of AES-ECDH is performed on the basis of certain metrics like encryption time, decryption time, storage for encrypted files, avalanche effect and correlation. Encryption time gives the time taken to convert the original text file into the cipher file; Decryption time denotes the time taken to convert the cipher file back to the original file. Coming toward storage that is essential to measure that signify the size of encrypted or cipher file formed after encryption. Most importantly Avalanche effect that will tell about variation in the cipher file by making a bit change in the original input file and correlation will signify the dependence between the cipher file and the original file *i.e*. whether the relationship among the files is linear and increasing or linear but decreasing, also if correlation is less then both cipher file and original file are dissimilar and it is difficult for intruder to know original text from cipher text.

5. Results and Analysis

The above methodology is implemented in JAVA 8 using Eclipse an open source platform that allows a developer to create a customized development environment (IDE). The above experiment is conducted on different text files with different sizes (KB). Files with different sizes are taken as input to the system and evaluation is done on the basis of different parameters like correlation, Avalanche effect, storage, Encryption time and Decryption time.

5.1. Parameter Evaluation using AES-ECDH

Different text files are taken as input and encrypted using AES-ECDH hybrid approach. Performance of the used hybrid approach is evaluated on the basis of different parameters given in table II below. Three metrics were taken to analyze the proposed methodology storage, encryption time and decryption time. Different text

files are taken as input with varying file size, after encryption we can see file size will increase and encryption and decryption time of hybrid approach is better.

Table 2. Evaluating Storage, Encryption Time and Decryption Time

File Name	Original file Size (kb)	Encrypted file size (kb)	Encryption Time (ms)	Decryption Time (ms)
License.txt	7650	13811	17	7
new.txt	152135	275527	37	27
new1.txt	581469	1057809	57	54
readme.txt	2878	5217	10	3
example.txt	4106	7447	11	4

In Table III, evaluation of AES-ECDH is done on the basis of avalanche effect and correlation taking again different files as input. Let's have a brief look on what is Avalanche effect and Correlation coefficient.

Avalanche Effect: Avalanche effect is the property of an algorithm which tells us how much change in bits will be there in the encrypted text if we change one bit of plain text. In Table 3 below every input text file holds good percentage of avalanche effect. if an algorithm holds good percentage of avalanche effect then it's too difficult for the intruder to break the security. Also enormous number of cipher texts can be generated by making just one bit variation.

$$\text{Avalanche} = \frac{\text{Number of Flipped bits in the Cipher Text}}{\text{Number of bits in the Cipher Text}}$$

Correlation: Correlation tells about the dependence among two variables or one can say similarity or relationship between the two variables so for system to be more secure correlation must be 0 or nearest to 0 which indicates no similarity between two variables. Here in the proposed work encrypted text and plain text files are checked whether they exhibit some linear relationship or not. Correlation can be calculated using formula given below where x, y are two variables, r be the calculated relationship among them and s is the standard deviation.

$$r = \frac{\sum_1^n (xi - \mu x)(yi - \mu y)}{(n-1)Sx\,Sy}$$

Table 3 represents the calculated avalanche effect and correlation among different files. Correlation is almost nearest to 0 which implies that there is almost negligible relationship between plain text and encrypted text, so it is difficult to obtain plain text by encrypted text.

Table 3. Evaluating Avalanche Effect and Correlation

File Name	Original File Size	Avalanche	Correlation
License.txt	7650	76.32	0.3
new.txt	152135	76.99	0.33
new1.txt	581469	79.5	0.26
readme.txt	2878	77.23	0.29
example.txt	4106	74.16	0.39

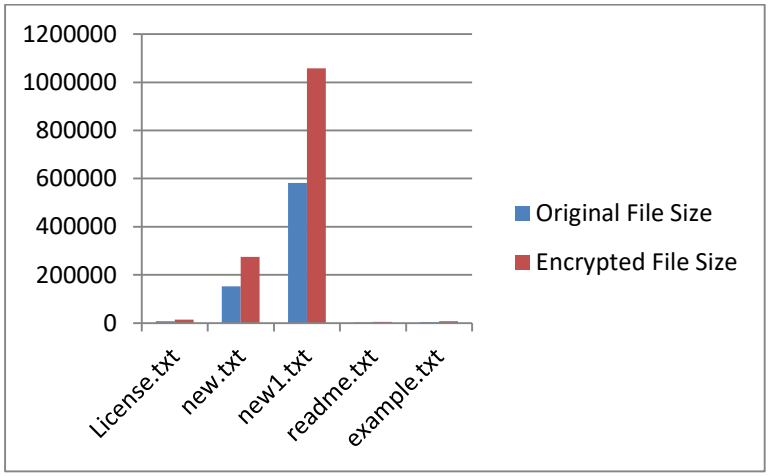

Figure 4. Encrypted File size of Different Files

Figure 4, Represents the graphical outputs of storage which is encrypted file size against the original file sizes taken as input. Storage is large than original file size for every set of input file. Figure 5 gives graphical outputs of encryption and decryption time. Computation speed of algorithm is much high as it requires much less time for encryption and decryption.

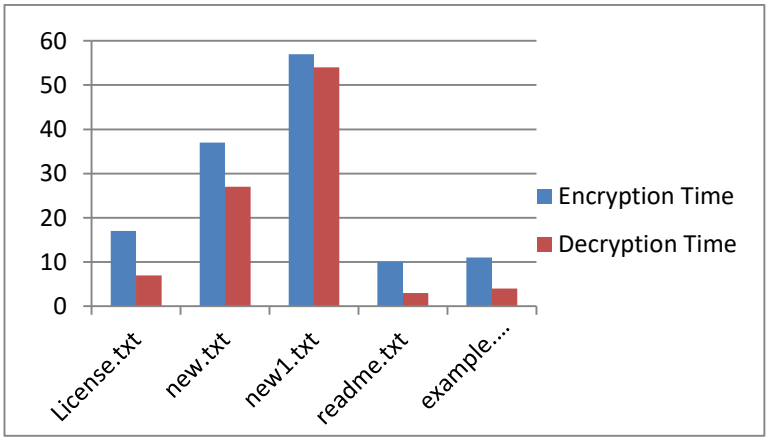

Figure 5. Encryption/Decryption Time of Different files

Figure 6 and Figure 7 gives avalanche effect and correlation against files taken as input to the hybrid system. Avalanche is quite high that enhance the security of the system as number of ciphers can be formed by making a single bit variation in the text without compromising the security. Avalanche for License.txt is 76.32, new.txt is 76.99, new1.txt is 79.5, readme.txt gives 77.23 percentage and example.txt exhibits 74.16. So avalanche is much good in percentage for all inputs files that ensure long term security.

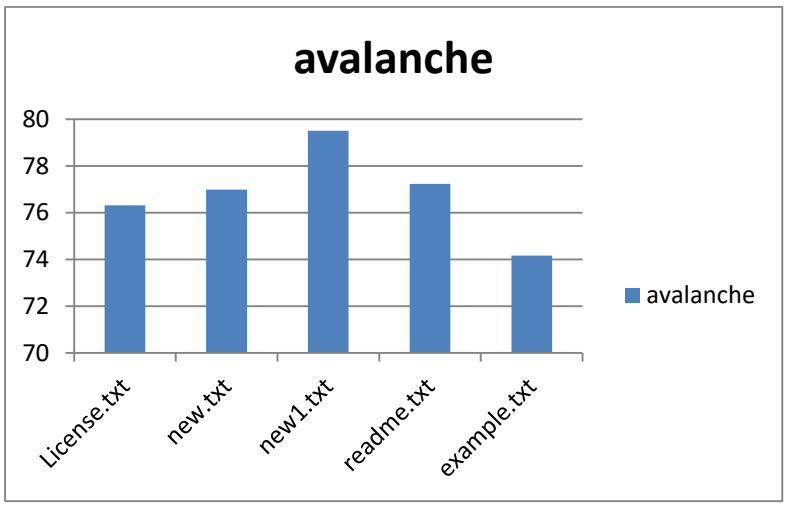

Figure 6. Avalanche Effect in Different Files

Figure 7 exhibits the correlation that can be seen from Table 3, almost nearest to zero which implies that there's no similarity between the original text and the cipher text or there is much less dependency between the original input file and cipher file. Correlation for License.txt is 0.3, new.txt is 0.33, new1.txt is 0.26, readme.txt gives 0.29 and example.txt exhibits 0.39. Correlation is almost closer to '0' which signify no similarity between original and output cipher file.

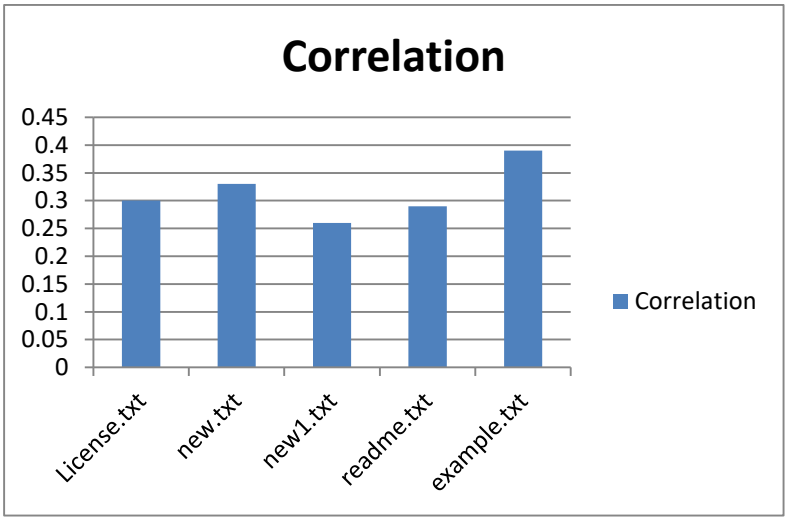

Figure 7. Correlation in Different Files

6. Conclusion

Data security is to safeguard the information which is getting exchanged between two parties communicating over an insecure network. Cryptography provides wide range of algorithms to protect such communications so that information can be transmitted securely over the wireless medium and provide authentication, data integrity, privacy and non-repudiation. This paper propose a hybrid model combining the characteristics of AES algorithm which is a symmetric technique and ECDH which is widely known as asymmetric technique to guard the communication from external threat. Proposed model is implemented on client server system, where client will communicate with the server and manage all information securely with the help of AES-ECDH hybrid model. Different text files of different sizes are taken as input; the key for encryption is generated with the help of elliptic curve cryptography while encryption and decryption is performed with the help of Advanced Encryption Standard (AES). After encryption all encrypted files are uploaded to the server and Diffie-Hellman will establish a shared secret between the client and server. When key agreement will be successful for that particular session client will be able to decrypt the message securely. At last the analysis is performed on the basis of different parameters such as encryption time, decryption time, storage, correlation and avalanche. Key generation is improved with Diffie-Hellman and the proposed approach is proved much effective in terms of avalanche and correlation that clearly demonstrate with the output results, the hybrid approach is much secure and the security of combined AES-ECDH is difficult to break. Obtained results show that the impact of this hybrid approach is significant and better than other algorithms. Future research may focus towards further improvement in the key management using some other cryptography model to get better security with less resource utilization in terms of storage, encryption and decryption time.

References

[1] D. E. Denning, "Cryptography and Data Security", Addison-Wesley Publishing Company, America, (**1982**).
[2] G. C. Kessler, "An overview of cryptography", [Online], Available: http://www.garykessler.net/library/crypto.html#purpose, (**1998**).
[3] P. Patil, P. Narayankar, N. D. G. and M. S. M, "A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish", Procedia Computer Science, (**2016)**; Nagpur, India.
[4] P. Prajapati, N. Patel, R. Macwan, N. Kachhiya and P. Shah, "Comparative Analysis of DES, AES, RSA Encryption Algorithms", International Journal of Engineering and Management Research, vol. 4, no. 1, **(2014)**, pp. 132-134.
[5] P. Mahajan and A. Sachdeva, "A study of Encryption algorithms AES, DES and RSA for security", Global Journal of Computer Science and Technology, vol. 13, no. 15, **(2013)**.
[6] Y. Wang and M. Hu, "Timing evaluation of the known cryptographic algorithms", International Conference on Computational Intelligence and Security, (**2009**).
[7] V. R. Pancholi and B. P. Patel, "Enhancement of Cloud Computing Security with Secure Data Storage using AES", International Journal for Innovative Research in Science and Technology, vol. 2, no. 09, **(2016)**, pp. 18-21.
[8] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security", Optik-International Journal for Light and Electron Optics, vol. 127, no. 04, **(2016)**, pp. 2341-2345.
[9] A. K. Mandal, C. Parakash and A. Tiwari, "Performance evaluation of cryptographic algorithms: DES and AES", Electrical, Electronics and Computer Science (SCEECS), (**2012**).
[10] A. A. Hasib and A. A. M. Haque, "A comparative study of the performance and security issues of AES and RSA cryptography", Third International Conference on Convergence and Hybrid Information Technology, **(2008)**.
[11] K. Rege, N. Goenka, P. Bhutada and S. Mane, "Bluetooth Communication using Hybrid Encryption Algorithm based on AES and RSA", International Journal of Computer Applications, vol. 71, no. 22, **(2013)**.

[12] R. R. Ahirwal and M. Ahke, "Elliptic curve diffie-hellman key exchange algorithm for securing hypertext information on wide area network", International Journal of Computer Science and Information Technologies, vol. 4, no. 2, **(2013)**, pp. 363-368.
[13] A. Pourali, M. V. Malakooti and M. H. Yektaie, "A Secure SMS Model in E-Commerce Payment using Combined AES and ECC Encryption Algorithms", The International Conference on Computing Technology and Information Management (ICCTIM), (**2014)**.
[14] B. Ji, L. Wang and Q. Yang, "New Version of AES-ECC Encryption System Based on FPGA in WSNs", Journal of Software Engineering, vol. 9, no. 1, **(2015)**, pp. 87-95.
[15] N. Jha and B. Patel, "Forward Secrecy For Google HTTPS using Elliptic Curve Diffie-Hellman Key", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), vol. 1, no. 9, **(2012)**.

Authors

Samiksha Sharma, she received her B. Tech degree in Computer Science and Engineering from Beant College of Engineering and Technology, Gurdaspur, Punjab, India in 2014 and is currently pursuing M.Tech degree in Computer Science and Engineering from DAV Institute of Engineering and Technology, Jalandhar, Punjab, India. Her research area includes Cryptography and Network Security.

Vinay Chopra, he received his M.E. degree from Thapar University, Patiala, Punjab, India in 2004. He received his Doctorate Degree from Punjabi University, Patiala, Punjab, India in 2008. He is presently working as an Assistant Professor in the Department of Computer Science & Engineering holding 13 years of expertise in his research domain in DAV Institute of Engineering and Technology, Jalandhar, Punjab, India. He is appointed as life member of Punjab Academy of Sciences and his research interests include software engineering, Computer Graphics and Automata.

