IJCSE International Journal of Computer Sciences and Engineering Open Access

Research Paper Vol.7, Special Issue.1, Jan 2019 E-ISSN: 2347-2693

Securing e-Learning Transactions using Digital Signature

Anup Pasari[1*], Kh Amirul Islam[2], Sunil Karforma[3], Sripati Mukhopadhyay[4]

[1,2,3,4]Dept. of Computer Science, The University of Burdwan, Burdwan, West Bengal, India

**Corresponding Author: anup.pasari@gmail.com, Tel.:+91 8170995888*

Available online at: www.ijcseonline.org

***Abstract*—**E-Learning is one of the most effective applications of Information and Communication Technology (ICT). In e-Learning environment participants such as Teacher, Student and Administrator perform their transaction electronically through Internet which is inherently insecure. During transaction process intruders may manipulate the message. Digital signature may be used to detect any change caused by intruders. To ensure information authenticity, the digital signature is the most suitable replacement of hand written signature. In this paper, we have applied ElGamal Digital Signature Algorithm (ElGamal DSA) combined with Secure Hash Algorithm (SHA-256) to check the authenticity of the sender and we have also used International Data Encryption Algorithm (IDEA) to impose security of digital document during transaction between student and administrator of an e-Learning system. The proposed model accommodates security of ElGamal Digital Signature Algorithm in combination with SHA-256 as hashing technique and IDEA encryption technique. The performance of the proposed model is analyzed with the help of supporting tables and charts.

***Keywords*—**Electronic Learning, Public Key Cryptography, ElGamal Digital Signature Algorithm, Secure Hash Algorithm, International Data Encryption Algorithm, Data Encryption Standard, Advanced Encryption Standard, Message-Digest Algorithm.

I. INTRODUCTION

E-Learning transaction involve uploading students documents like pass certificate, identity proof, photograph, signature and downloading documents like admit, mark sheet and certificate. Now a days the intruder can easily manipulate, copy and delete the transaction due to exploiting the loopholes of Internet which is the main backbone of E-Learning as Internet is an open access electronic media to all users [17]. During submission of students' document electronically using Internet the hacker can perform different types of attacks like synflooding, man in the middle attack, Denial-of-service and Eavesdropping etc [1]. As a result the information of a fake student or wrong information about a valid student may be submitted at the server end. To prevent against such types of attacks, digital signature may be used as an authentication tool and IDEA may be used to prevent unauthorised access [1][2][4][6]. The digital signature is a mathematical technique used to validate authenticity and integrity of a message and electronic document [5][6]. The digital equivalent of handwritten signature or a digital signature offers inherent security to solve the problems of digital communications based on public key cryptography. In other words a digital code that can be attached with the electronically transmitted message which is uniquely identify the sender and ensure that the information has not been alerted during transmission [4]. ElGamal digital signature algorithm is utilising the benefits of discrete logarithmic problems to provide the security of the one-way function of exponentiation in modular rings [19]. During ElGamal digital signature creation process we have also used strong one way hashing technique likes SHA-256 to create the message digest derived from the certificate which would be submitted by the student. The SHA-256 is designed by United States National Security Agency (NSA) [23]. Underlying structures of SHA-2 scheme referred one way compression functions to form a specialised block cipher and allows a message with a maximum length less than 264-bits and produce an output of 256-bits message digest containing 64 hexadecimal digits. The function takes an input message and portions into 512 bit blocks (16 x 32 bits) for processing. Each block processed through 64 iterations to produce an output block [21].IDEA is one of the symmetric key encryption algorithms that is widely used for different security purposes [13][17]. IDEA operates with 64-bit block of data with the help of 128-bit key and produce 64-bit cipher block in 8 steps of identical transformation and one output transformation [16][18]. The fundamental design of the algorithm provides the security by using three different algebraic operations are bitwise Exclusive OR, modular addition and modular multiplication [4][6][7].IDEA provides high level security based on keeping the algorithm not a secret, but importance upon the secret key transformation [9]. In this paper we have proposed the combination of public key cryptography based ElGamal digital signature algorithm and SHA-256 and IDEA encryption technique to impose the security of digital document transaction [7][23]. As the objective is to provide the strong signature that requires minimum space on the certificate [20]. Our proposed system accommodates security in all types of online transaction including certificate, marksheet, valid id, assignment submission and communications.

The section 2 contains an architecture framework that is needed to implement any e-Learning environment. Proposed scheme is explained in section 3. Finally we have written conclusion including future scope in section 4.

II. ARCHITECTURE FRAMEWORK

Architecture of e-Learning system is shown in figure 1. There are two entities – client, web server and database server. When a user wants to send some digital document from the client side system, the most way to communicate with the server is via a web browser which is connected to the internet. The standard protocol HTTP and FTP are used to make the document transaction between client and server. The system requires more security than an ordinary web browser and the firewall are able to provide [15]. Extra security functionality included in cryptographic techniques.

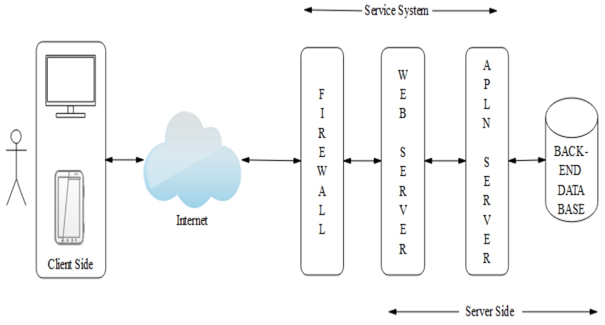

Figure 1: E-Learning Architecture using Firewall

2.1. Client Side: At the client end a user can submit a particular document i.e. mark sheet, certificate, assignment and others. Before making the transaction first the file encrypted using IDEA and a secure hash SHA-2 performed to create the message digest from the encrypted file. Then the signature is generated from the hash file using ElGamal digital signature algorithm. The encrypted file and the signature called digitally signed message is sent to the server for the completion of the transaction. The flowchart for client side application program is given in figure 2.

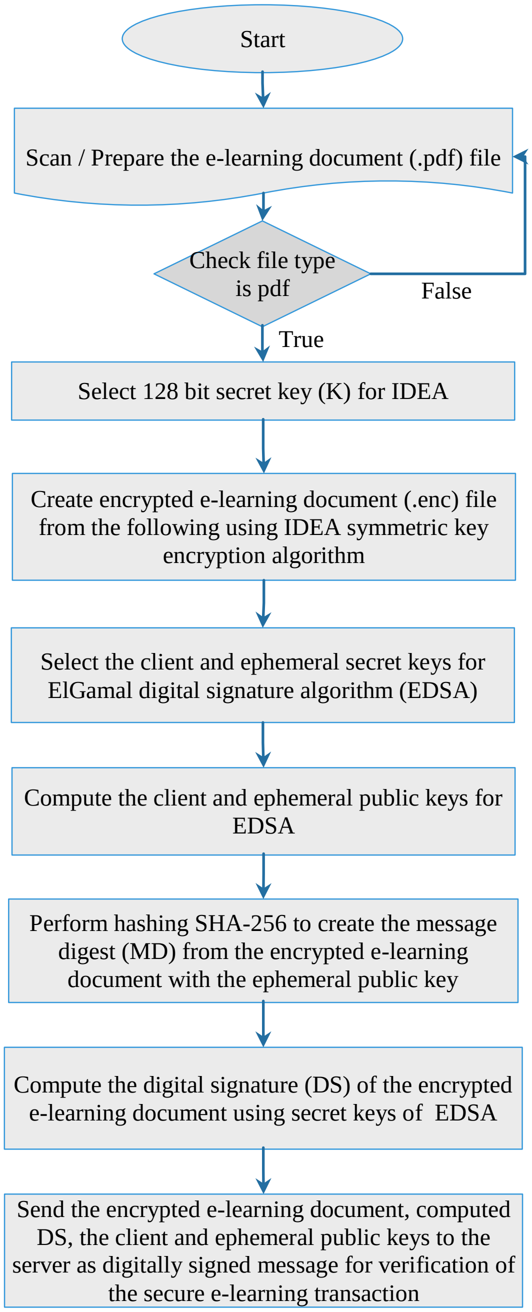
Start
Scan / Prepare the e-learning document (.pdf) file
Check file type is pdf
False
True
Select 128 bit secret key (K) for IDEA
Create encrypted e-learning document (.enc) file from the following using IDEA symmetric key encryption algorithm
Select the client and ephemeral secret keys for ElGamal digital signature algorithm (EDSA)
Compute the client and ephemeral public keys for EDSA
Perform hashing SHA-256 to create the message digest (MD) from the encrypted e-learning document with the ephemeral public key
Compute the digital signature (DS) of the encrypted e-learning document using secret keys of EDSA
Send the encrypted e-learning document, computed DS, the client and ephemeral public keys to the server as digitally signed message for verification of the secure e-learning transaction

Figure 2: Flowchart for Client Side Application

2.2. Server Side: The server side receives the client message and separate the DS from the message and recalculate the digital signature from the encrypted file using the same algorithm as client side. If the calculated signature and received signature both are same then the server will accept the file otherwise report to the client that error has occurred during the transaction process. After accepting the encrypted file the server will apply IDEA decryption to get the original document and the document is stored in the server back end database. The flowchart for server side application program is given in figure 3.

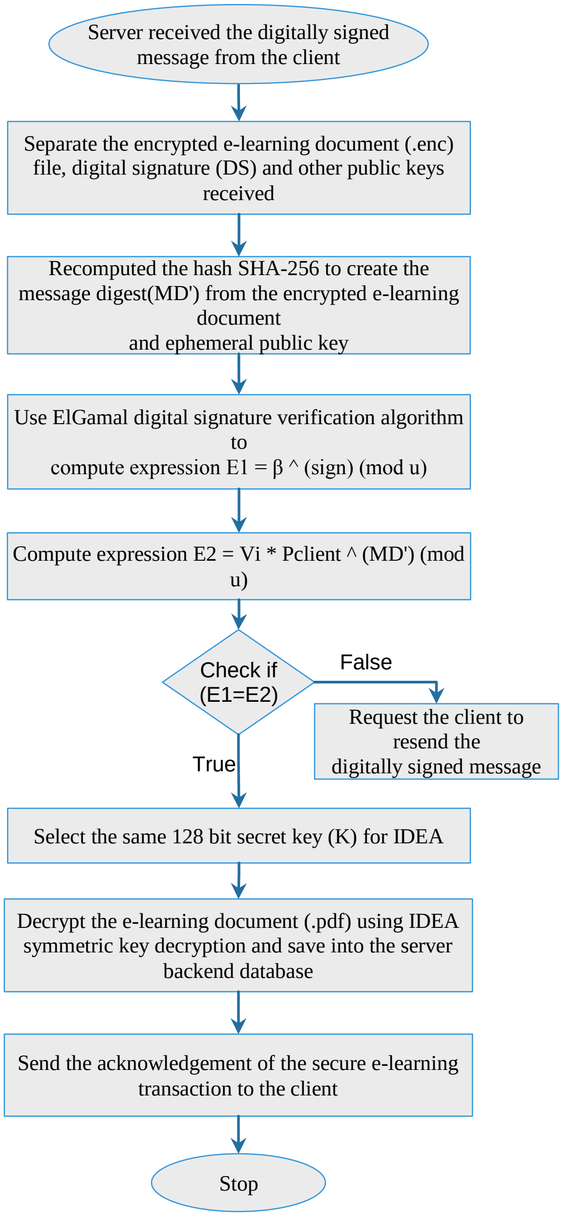
Server received the digitally signed message from the client
Separate the encrypted e-learning document (.enc) file, digital signature (DS) and other public keys received
Recomputed the hash SHA-256 to create the message digest(MD') from the encrypted e-learning document and ephemeral public key
Use ElGamal digital signature verification algorithm to compute expression E1 = β ^ (sign) (mod u)
Compute expression E2 = Vi * Pclient ^ (MD') (mod u)
Check if (E1=E2)
False
Request the client to resend the digitally signed message
True
Select the same 128 bit secret key (K) for IDEA
Decrypt the e-learning document (.pdf) using IDEA symmetric key decryption and save into the server backend database
Send the acknowledgement of the secure e-learning transaction to the client
Stop

Figure 3: Flowchart for Server Side Application

2.3. Database Sever: The database server contains the information about the user and their transaction. Database server is connected with the web server through application server. The web server gets the information from the client and execute for checking its authenticity. The authenticated document saved to database and intimation sent to the client via web browser.

III. Proposed Scheme

Our proposed scheme is based on International Data Encryption Algorithm [4][6][16], Secure Hash Algorithm-2 [3][11][21] and ElGamal Digital Signature [10][12][19] scheme.

3.1. Client Side Application: The digital document is uploaded by the client in a specific file format (.pdf) is considered as plaintext and encrypted using symmetric encryption technique IDEA before sending. The encryption is performed using a block cipher with 64-bit plaintext and 128-bit key [14].

3.1.1. Encryption:

The 64-bit plaintext block is partitioned into four 16-bit sub-blocks (X1, X2, X3, and X4), as all the algebraic operations used in the process operate on 16-bit numbers. The key generation of IDEA uses the same process to produce a total of 52 (= 8 level x 6 key in each level + 4 key for output transformation) different 16-bit sub-blocks (K1 to K52) from the 128 bit key. Initially the key consist of 128 bit, from which 8 sub-keys (K1 to K8) are generated from which K1 to K6 are used in first round and reaming K7 and K8 are used as first two sub-keys in second round. For reaming sub-keys the 128 bit key are shifted left circularly by 25 bits and next 8 sub-keys (K9 to K16) are generated. Rest of the sub-keys are generated using the same techniques. Table 1 displays the sub keys used in different round.

Table 1: Encryption sub keys of different round.

Round No	Sub keys of Decryption
1	K1, K2, K3, K4, K5, K6
2	K7, K8, K9, K10, K11, K12
3	K13, K14, K15, K16, K17, K18
4	K19, K20, K21, K22, K23, K24
5	K25, K26, K27, K28, K29, K30
6	K31, K32, K33, K34, K35, K36
7	K37, K38, K39, K40, K41, K42
8	K43, K44, K45, K46, K47, K48
Output Transformation	K49, K50, K51, K52

IDEA Encryption Algorithm:

Add* represents addition modulo 2^{16} and Multiply* represents multiplication modulo $2^{16} + 1$

For each round R_i 4 data block (X1, X2, X3, and X4) and 6 key block (K1 to K6)

Step 1: Multiply* X1(i) and K1.
Step 2: Add* X2(i) and K2.
Step 3: Add* X3(i) and K3.
Step 4: Multiply* X4(i) and K4.
Step 5: EX-OR the result of Step1 and Step 3.

Step 6: EX-OR the result of Step 2 and Step 4.
Step 7: Multiply* the result of Step 5 and K5.
Step 8: Add* the result of Step 6 and Step 7.
Step 9: Multiply* the result of Step 8 and K6.
Step 10: Add* the result of Step 7 and Step 9.
Step 11: EX-OR the result of Step 1 and Step 9 as X1(i+1) of next step.
Step 12: EX-OR the result of Step 3 and Step 9 as X2(i+1) of next step.
Step 13: EX-OR the result of Step 2 and Step 10 as X3(i+1) of next step.
Step 14: EX-OR the result of Step 4 and Step 10 as X4(i+1) of next step.

At the last round of the encryption process (after 8th round) one time output transformation takes place

Step 1: Multiply* X1(9) and K49.
Step 2: Add* X2(9) and K50.
Step 3: Add* X3(9) and K51.
Step 4: Multiply* X4(9) and K52.

3.1.2. Signature Computation Algorithm:

At the end of encryption encrypted file in a specific format (.enc) is generated is considered as message for the next signature generation using ElGamal DSA.

Step 1: Select a secret key for the clientS_{client} .

Step 2: Calculate the public key for the client $\mathrm{P}_{\text{client}} = \beta^{S_{\text{client}}} (modu)$.

Where, β is a random number serve as the generator and u is a large prime number serve as modulus

Step 3: Select an ephemeral secret key R_i .

Step 4: Calculate the ephemeral public key $\mathrm{V}_{\mathrm{i}} = \beta^{R_{\mathrm{i}}} (modu)$.

Step 5: Calculate the hash value as message digest MD of the message combined with ephemeral public key V_{i} using hash function. $MD_i = H(msg_i || \mathrm{V}_{\mathrm{i}})$.

Step 6: Compute the signature$\mathrm{sign}_{\mathrm{i}} = R_i + MD_i * S_{\text{client}} (modu - 1)$.

Send the message (encrypted digital document), signature and the public keys $\mathrm{P}_{\text{client}}$ and V_{i} .

H (M): (Secure Hash Function using SHA-256)

The encrypted document (M) is padded to divide in N no blocks M_1, M_2, M_3, ..., M_N. Each block M_i consist of 16, 32-bit words $M_{i,0}$, $M_{i,1}$, $M_{i,2}$, $M_{i,3}$, ... $M_{i,15}$ a total of 512-bit block.

Basic operations:

AND, XOR and OR, denoted by ∧, ⊕ and ∨, respectively.

Bitwise complement, denoted by '.

Integer addition modulo 2^{32}, denoted by P + Q.

Each of them operates on 32-bit words.

For the last operation, binary words are translated as integers written in base 2.

RotR(P, n) denotes the circular right shift of n bits of the binary word P.

ShR(P, n) denotes the right shift of n bits of the binary word P.

P || Q denotes the concatenation of the binary words P and Q.

Functions:

$Ch(A, B, C) = (A \wedge B) \oplus (A' \wedge C)$---------(1)
$Maj(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$,
$\Sigma_0(A) = RotR(A, 2) \oplus RotR(A, 13) \oplus RotR(A, 22)$,
$\Sigma_1(A) = RotR(A, 6) \oplus RotR(A, 11) \oplus RotR(A, 25)$,
$\sigma_0(A) = RotR(A, 7) \oplus RotR(A, 18) \oplus ShR(A, 3)$,
$\sigma_1(A) = RotR(A, 17) \oplus RotR(A, 19) \oplus ShR(A, 10)$,

Constants: The 64 binary words K_t (t=0 to 63) are set to the initial values by the first 32-bit fractional parts of the cube roots of the first 64 prime numbers.

The 8 binary variables $H_{0,i}$(i=0 to 7) are set to the initial values by the first 32-bit fractional parts of the square roots of the first 8 prime numbers.

Each Block of e-learning document (M) is $M_i = M_{i,0} \| M_{i,1} \| M_{i,2} \| M_{i,3} \| .... \| M_{i,15}$ $1 \leq i \leq N$

Secure Hash Algorithm using SHA-256:

```
For i = 1 to N
    Step 1: Prepare the message schedule W
    for t = 0 to 15
    $W_t = M_{i,t}$
    for t = 16 to 63
    $W_t = \sigma_1^{256}(W_{t-2}) + W_{t-7} + \sigma_1^{256}(W_{t-15}) + W_{t-16}$
    Step 2: Initialize the working variables
        $a = H_{i-1,0}$; $b = H_{i-1,1}$; $c = H_{i-1,2}$; $d = H_{i-1,3}$;
        $e = H_{i-1,4}$; $f = H_{i-1,5}$; $g = H_{i-1,6}$; $h = H_{i-1,7}$;
    Step 3: for t = 0 to 63
        $T_1 = h + \sum_1 e + Ch(e, f, g) + W_t + K_t$
        $T_2 = h + \sum_0 a + Maj(a, b, c)$
        $h = g$; $g = f$; $f = e$; $e = d + T_1$;
        $d = c$; $c = b$; $b = a$; $a = T_1 + T_2$
    Step 4: Compute the new values of $H_i$
        $H_i,0 = a + H_{i-1,0}$; $H_i,1 = b + H_{i-1,1}$;
        $H_i,2 = c + H_{i-1,2}$; $H_i,3 = d + H_{i-1,3}$;
        $H_i,4 = e + H_{i-1,4}$; $H_i,5 = f + H_{i-1,4}$;
        $H_i,6 = g + H_{i-1,6}$; $H_i,7 = h + H_{i-1,7}$;

return ($H_{N,0} \| H_{N,1} \| H_{N,2} \| H_{N,3} \| H_{N,4} \| H_{N,5} \| H_{N,6} \| H_{N,7}$) as message digest.
```

3.2. Server Side Application:

After receiving the encrypted e-learning document and the digital signature along with ephemeral and client public keys, server continues the transaction verification process through the following steps with the help of ElGamal digital signature algorithm.

3.2.1. Signature Verification Algorithm:

Step 1: Calculate the message digest MD$'$ of the encrypted digital document combined with ephemeral public key V_i using hash function. $MD'_i = H(msg_i||V_i)$.

Step 2: Compute $E_1 = \beta^{sign_i}(modu)$.

Step 3: Compute $E_2 = V_i * P_{client}^{MD'_i}(modu)$.

Step 4: Compare E1 and E2 if matched accept the transaction otherwise decline to client.

Step 5: Decrypt the encrypted document using IDEA decryption.

Step 6: Store the digital document in the server database.

3.2.2. Decryption:

The decryption process is identical to the encryption process, but the order of the round keys is inverted, and the sub-keys for the odd rounds are inversed. For instance, the values of sub-keys KD1 to KD4 are replaced by the inverse of K49 to K52 for the respective group operation, KD5 and KD6 of each group should be replaced by K47 and K48 for decryption. Table 2 displays the decryption sub keys of different round and the Table 3 displays the relation between sub keys of encryption and the sub keys of decryption where $K49^{-1}$ is the multiplicative inverse of K49 and -K50 is additive inverse of K50 in Annexure A.

Table 2: Decryption sub keys of different round.

Round No	Sub keys of Decryption
1	KD1, KD2, KD3, KD4, KD5, KD6
2	KD7, KD8, KD9, KD10, KD11, KD12
3	KD13, KD14, KD15, KD16, KD17, KD18
4	KD19, KD20, KD21, KD22, KD23, KD24
5	KD25, KD26, KD27, KD28, KD29, KD30
6	KD31, KD32, KD33, KD34, KD35, KD36
7	KD37, KD38, KD39, KD40, KD41, KD42
8	KD43, KD44, KD45, KD46, KD47, KD48
Output Transformation	KD49, KD50, KD51, KD52

3.3. Performance Analysis:

In addition, to improve the accuracy of our scheme for timing measurements, program was executed 100 times for each input file and we report the average of the times thereby obtained. The comparison of Message-Digest Algorithm (MD5) with Secure Hashing Algorithm (SHA) are made on the basis of time and the security in bits [14]. The symmetric key encryption algorithm Data Encryption Standard (DES), IDEA, and Advanced Encryption Standard (AES) are also compared on the basis of time and security in bits [14].

An experimental result of time taken by each steps of the E-learning transaction for generating encrypted digital document, signature computation using message digest, signature verification and decrypting the original digital document is shown in Table 4. In the following experiment we have used different document of different file size.

Table 4: Time elapsed in different steps of the transaction for different document size

File Name (.pdf)	File Size (in KB)	Client Side Application Time in Milliseconds		Server Side Application Time in Milliseconds	
		Document Encryption	Signature Computation	Signature Verification	Document Decryption
A	246	22.8	4.4	4.4	21.8
B	558	39.8	9.4	9.5	42.4
C	1178	77.5	19.0	19.1	80.1
D	2142	136.9	35.4	35.0	138.8
E	4265	267.5	69.5	69.0	270.2

Figure 4 shows the graphical representation of average execution time of different steps of the transaction in milliseconds.

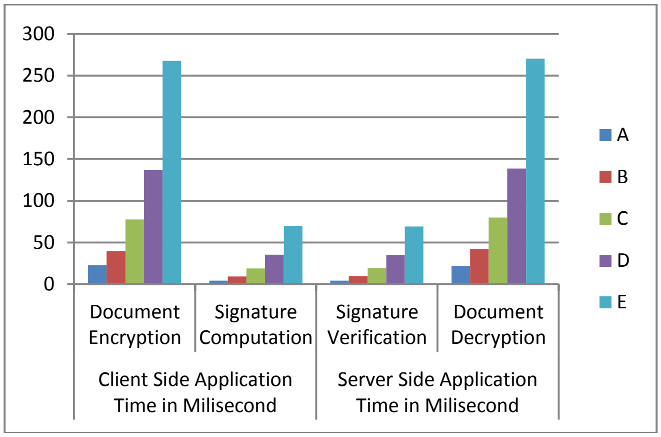

Figure 4: Comparison of processing times in different steps for same file types different sizes

We compare the execution time of DES, IDEA and AES algorithms for encryption and decryption. An experimental result of time is taken by the symmetric encryption algorithms for generating encrypted digital document in client side application and the decryption of the original digital document by the server side application. Figure 5 shows the average encryption as well as decryption time of different cryptographic algorithm in milliseconds.

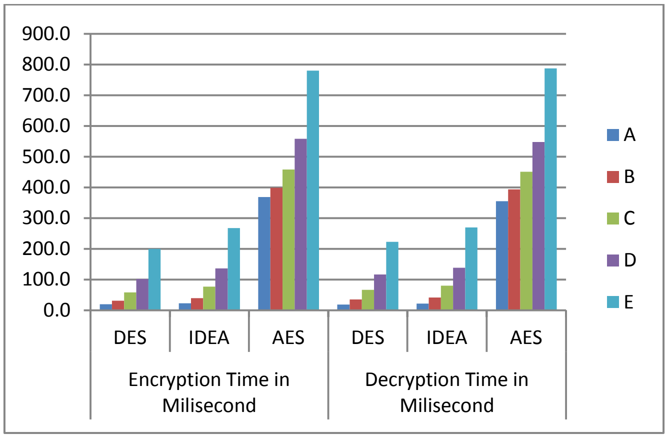

Figure 5:Comparison of encryption and decryption times for various common symmetric encryption algorithms

We have also compared the execution time of MD5, SHA-1 and SHA-256 hashing algorithms. An experimental result of time taken by the algorithms for generating message digests for ElGamal Digital Signature. Figure 6 shows the average signature generation as well as signature verification time of the algorithms in milliseconds.

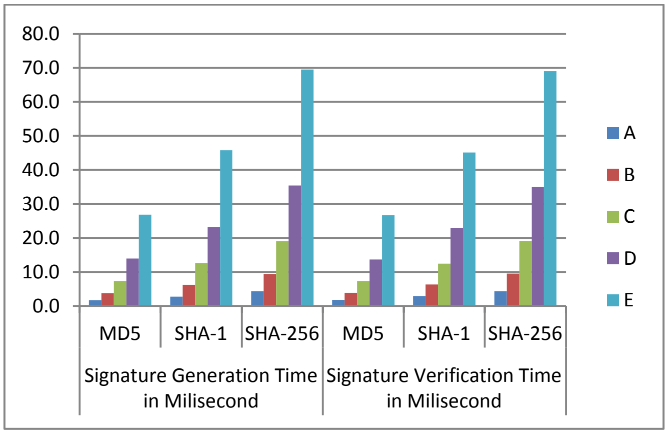

Figure 6: Timing comparison of signature generation and verification using different hashing algorithms

3.4. Result Analysis:

From the Figure 5 concludes execution time for IDEA and DES are almost same but the security in bits with respect to execution time (Security in bits / execution time) is very high for IDEA is shown in Figure 7.

Similarly for the Figure 6 concludes execution time for SHA-256 is greater than others but in terms of avalanche effect with the time SHA-256 is better than others. Avalanche effect of different common hashing algorithm using one byte information and time are shown in Figure 8.

To check the efficiency and strength of these algorithms, authors have developed these algorithms and compare it with other. Java implementation has used to implement these algorithms using Blue J Software. Intel Pentium Dual Core i7 2.50 Ghz, 8 GB of RAM and Window-7 Professional SP1,have used in which performance data is collected.

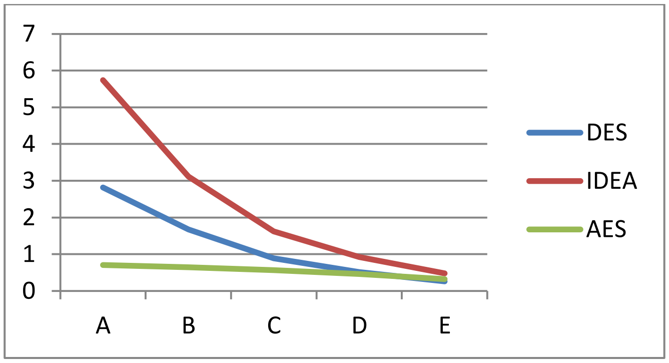

Figure 7: Comparison of security in bits with average encryption and decryption times for various common symmetric encryption algorithms.

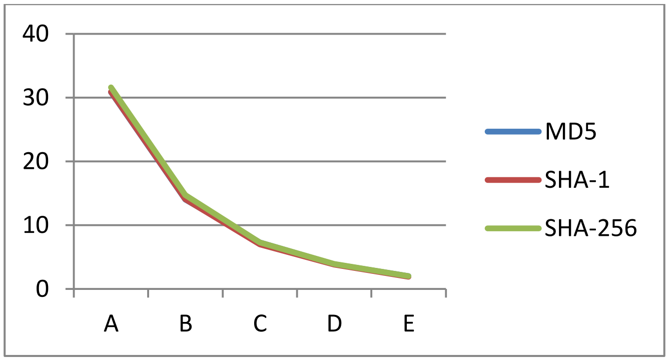

Figure 8: Comparison of avalanche effect with average hashing times for various common hashing algorithms.

IV. CONCLUSION AND FUTURE SCOPE

In order to control the manipulation of the digital documents like admit, certificate, mark sheet, assignment by the intruders, a scheme is proposed by applying IDEA, SHA-256 and ElGamal Digital Signature Algorithm. These algorithms provide strong cryptographic security and authenticity of digital documents used in e-learning at the cost of a few computational overhead which are shown with supporting data in the performance analysis section. The proposed model may be used successfully in m-learning environment as well as similar kind of e-services such as e-governance, m- governance, etc. In future the security of proposed model may be improved further by implementing Elliptic Curve version of ElGamal Digital Signature Algorithm (ECEDSA) using SHA-3 as hashing techniques needed for digital signature computation and verification.

REFERENCES

[1] Fadia Ankit, “Network Security”, Macmillan Publishers India Ltd., Second Edition, 2006.

[2] Forouzan Behrouz A, “Data Communications and Networking”, Tata McGraw-Hill Publishing Company Ltd, Fourth Edition, 2007.

[3] Hook David, “Beginning Cryptography with Java” Wiley India (P.) Ltd., First Edition, 2005.

[4] Kahate Atul, “Cryptography and Network Security”, Tata McGraw-Hill Publishing Company Ltd, Second Edition, 2003.

[5] Menezes, Oorschot, Vanstone, “Handbook of Applied Cryptography”, CRC press, Fifth Printing, 2001

[6] Schneier Bruce, “Applied Cryptography”, Wiley India (P.) Ltd., Second Edition, 2010.

[7] Stallings William, "Cryptography and Network Security: Principles and Practice", Pearson Education, Sixth Edition, 2014.

[8] Tanenbaum, Wetherall, “Computer Networks” Pearson Education, Fifth Edition, 2011.

[9] A. Aravind, K. Kumar V.G, S. Rai, Nisha, "Implementation of Two Light Weight Cryptographic Algorithms", IOSR Journal of Electronics and Communication Engineering (IOSR-JECE), Vol. 12, No. 4, Ver. II, pp. 43-47, Jul.-August 2017.

[10] A. Mousa , "Security and Performance of ElGamal Encryption Parameters" Journal of Applied Sciences, Vol. 5, pp. 883-886, 2005.

[11] A. Swaminathan, Y. Mao, Min Wu, "Robust and Secure Image Hashing" IEEE Transactions on Information Forensics and Security, Vol. 1, No. 2, pp. 215-230, June 2006.

[12] C. Chang, W. Liao, "A remote authentication scheme based upon ElGamal's signature scheme" Elsevier Science Ltd Computers & Security, Vol. 13, pp 137-144, 1994.

[13] K. D. Sharma, H. K Verma, A. Kumar, "Study and Performance Analysis of IDEA with Variable Rounds", International Journal of Advanced Research in Computer Science and Software Engineering, Vol. 2, No. 5, pp. 102-105, May 2012.

[14] K. Raghuvanshi, P. Khurana and P. Bindal, "Study and Comparative Analysis of Different Hash Algorithm", Journal of Engineering Computers & Applied Sciences (JECAS), Vol. 3, No. 9, Online pp. 1-3, September 2014.

[15] N. Barik, S. Karforma, "Risks and Remedies in E-Learning System", International Journal of Network Security & Its Applications (IJNSA), Vol.4, No.1, pp.51-59, January 2012.

[16] O. Almasri, H.Mat Jani2, "Introducing an Encryption Algorithm based on IDEA", International Journal of Science and Research (IJSR), Vol. 2 No. 9, pp. 334-339 September 2013.

[17] P. Ghosh, S. Karforma, "Application of International Data Encryption Algorithm in E-Learning Security: An UML (Unified Modelling Language) based approach", In the Proceedings of the 2010 International Conference on Computing and Systems (ICCS 2010), West Bengal, India, pp. 96-102, November 2010.

[18] S. Omer, A. Babiker, "Comparison of Various Encryption Algorithms and Techniques for improving secured data Communication” IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 17, No 1, Ver. III, pp 62-69, Jan – Feb. 2015.

[19] W. Lee a, C. Wu a, W. Tsaur, "A novel deniable authentication protocol using generalized ElGamal signature scheme" Information Sciences Elsevier Inc, Vol. 177 pp, 1376–1381, September 2006.

[20] Commonlounge, *www.commonlounge.com.* [Online], https://www.commonlounge.com/discussion/35a1c2baa00b447f9275e8f71b02ef29.

[21] ResearchGate, www.researchgate.net, [Online], https://www.researchgate.net/file.PostFileLoader.html?id=534b393ad3df3e04508b45ad&assetKey=AS%3A273514844622849%401442222429260.

[22] Wikipedia, *en.wikipedia.org.* [Online],https://en.wikipedia.org/wiki/International_Data_Encryption_Algorithm.

[23] Wikipedia, *en.wikipedia.org.* [Online], https://en.wikipedia.org/wiki/SHA-2.

Authors Profile

Mr. Anup Pasari pursed Bachelor of Computer Application from University of Burdwan, Burdwan in 2012 and Master of Computer Application from University of Burdwan in year 2015. He is currently pursuing Ph.D. in Department of Computer Sciences, University of Burdwan, Burdwan since 2018. He is a member of IEEE & IEEE computer society since 2018. He has published 01 research papers in conference including IEEE and it’s also available online. His main research work focuses on Cryptography Algorithms, Network Security, Cloud Security and Privacy, Computational Intelligence based education.

Mr. Kh Amirul Islam pursed Bachelor of Computer Application from Dumkal Institute of Engineering & Technology in 2012 and Master of Computer Application from University of Burdwan in year 2015. He is currently pursuing Ph.D. in Department of Computer Sciences, University of Burdwan, Burdwan since 2016. He has published some research papers in reputed journal and in conference including IEEE and it’s also available online. His main research work focuses on Cryptography Algorithms, Network Security, Cloud Security and Privacy, Computational Intelligence based education.

*Prof. Sunil Karforma*has completed B.E. (Computer Science and Engineering) and M.E. (Computer Science and Engineering) from Jadavpur University. He has completed Ph.D. in the field of Cryptography. He is presently holding the post of Professor as well as Head of the Department in the Department of Computer Science, The University of Burdwan. Network Security and E-commerce is his field of interest in research area. He has published research papers in reputed National and International journals and proceedings.

Prof. Sripati Mukhopadhyay, M. Tech, Phd. is a former professor and head of Department of Computer Science, University of Burdwan. Hehas served North Bengal University , Vidyasagar University , Visva-Bharti etc. He has 33 years of teaching and research experience. His research interests include Computational Intelligence, Software Engineering and Data Base Systems.

Annexure-A

Table 3: Relation between encryption key and decryption key.

Round No	Sub keys of Decryption	Relation with Encryption keys
1	KD1, KD2, KD3, KD4, KD5, KD6	$K49^{-1}$, -K50, -K51, $K52^{-1}$, K47, K48
2	KD7, KD8, KD9, KD10, KD11, KD12	$K43^{-1}$, -K44, -K45, $K46^{-1}$, K41, K42
3	KD13, KD14, KD15, KD16, KD17, KD18	$K37^{-1}$, -K38, -K39, $K40^{-1}$, K35, K36
4	KD19, KD20, KD21, KD22, KD23, KD24	$K31^{-1}$, -K32, -K33, $K34^{-1}$, K29, K30
5	KD25, KD26, KD27, KD28, KD29, KD30	$K25^{-1}$, -K26, -K27, $K28^{-1}$, K23, K24
6	KD31, KD32, KD33, KD34, KD35, KD36	$K19^{-1}$, -K20, -K21, $K22^{-1}$, K17, K18
7	KD37, KD38, KD39, KD40, KD41, KD42	$K13^{-1}$, -K14, -K15, $K16^{-1}$, K11, K12
8	KD43, KD44, KD45, KD46, KD47, KD48	$K7^{-1}$, -K8, -K9, $K10^{-1}$, K5, K6
Output Transformation	KD49, KD50, KD51, KD52	$K1^{-1}$, -K2, -K3, $K4^{-1}$