Enhancing Security Of El Gamal Encryption Scheme Using Rsa And Chaos Algorithm For E-Commerce Application

Jerome P. Magsino, Edwin R. Arboleda, Reynaldo R. Corpuz

Abstract: The credit card number can be secured by hiding the original data to a ciphertext. Different methods of enciphering can be used but some of those are prone to any brute force attack especially those that have been used by many. This paper proposes to hybrid the El Gamal encryption scheme with RSA and chaos algorithm. The authenticity of the system and its speed has been tested to prove the efficiency of the new system.

Index Terms: ciphertext, cryptography, decryption, encryption, hybrid encryption, ElGamal; RSA, Chaos, E-Commerce

1. INTRODUCTION

Electronic Commerce or E-Commerce is the buying and selling of products and services using Information and Communications Technology (ICT). It includes order accepting, order evaluating, supplying of order, billing and money transfer [1].E-Commerce is an outcome of globalization and technology outbreak of the 21st century. The growing of technology contributes to the fast-changing lifestyle of the people[2]. Shopping over the internet became an alternative to shopping at a physical store[3]. Shopping on the internet has a number of advantageous reasons why one should buy on e-commerce websites. This includes convenience, timeliness, and broad range of varieties, lesser effort and the purchase of products from overseas. These reasons make the general public to use the new trend: E-Commerce, in buying and selling of products as well as services [4]. As all good things have their drawbacks, E-Commerce also has its disadvantages. Security issues may pose a threat to consumers who purchase on E-Commerce websites[5]. The authenticity of the website is also a risk on the consumer's part. The products displayed on the website might not be the actual merchandise to be given when sold. These uncertainties must be considered whenever shopping online[6].

In an online transaction, cashless payment is the most common scenario [5]. The credit card replaces the use of cash. Although credit cards are designed to rely on physical signatures for authentication, this mechanism is rendered useless in e-commerce. The security features of a credit card become useless whenever it is used in an e-commerce application. Any person who has the knowledge to a credit card number and its expiration date can buy anything over the Internet. Credit card fraud is considered a cybercrime but no specific section in RA 10175 or the Cybercrime Prevention Act of 2012 of the Philippines focuses on this matter. There is a need for a combination of legislation and technical solutions to secure the privacy of consumers globally[7]. Although the El Gamal encryption scheme has been a well-known aid in data security issues, it has its own deficiencies [8][9][10].This paper intends to solve the technical problem of credit card fraud. This paper proposes to solve or to help solve the problem regarding the security in E-Commerce by introducing an encryption scheme into the process of E-Commerce. The encryption scheme to be used in fortifying the security of E-Commerce is the El Gamal encryption scheme to be highbred with RSA and Chaos algorithm.

2 RELATED LITERATURE

The parent cryptosystems must be review to analyze their algorithm. El Gamal cryptosystem, RSA cryptosystem, and Chaos algorithm are taken into consideration.

2.1. El Gamal Cryptosystem

El Gamal proposed a public-key cryptosystem in 1985. The El Gamal algorithm is mostly used for both encryption/decryption and digital signatures. The security of the El Gamal scheme uses on the difficulty of calculating the discrete logarithms over GF(p) where the variable p is a large prime[11]. To describe the El Gamal system, choose a prime number p and two random numbers, g, and x, with a relation g < p and x < p, where x is a private key. The random number g is a primitive root modulo p. The public key is defined by y, g, and p. Then calculate

y = gx (mod p). To encrypt the message m, 0 < m < p−1, first pick a random number k such that gcd(k, p−1) = 1. The encrypted message can be taken as the pair (r, s) as follows:

$$r = g^k \pmod p \tag{1}$$

$$s = (y^k m \pmod p)) (m \pmod{p-1})) \tag{2}$$

- *Jerome P. Magsino is from the Department of Computer and Electronics Engineering, College of Engineering and Information Technology, Cavite State University, Indang,Cavite. E-mail: jeromemagsinoece@gmail.com*
- *Edwin R. Arboleda is from the Department of Computer and Electronics Engineering, College of Engineering and Information Technology, Cavite State University. Indang, Cavite. E-mail: edwin.r.arboleda@cvsu.edu.ph*
- *Reynaldo R. Corpuz is an affiliate of Isabela State University-Cauayan Campus. Email: reynaldo.r.corpuz@isu.edu.ph*

To decrypt the value of m, divide s by rx such that s/rx = m (mod p − 1). To put a signature on a given message m, first choose a random number k in a condition that gcd (k, p − 1) = 1, and compute m = xr + ks (mod p − 1) in the method of extended Euclidean algorithm to solve (Rhee, 2003).

2.2 RSA Cryptosystem

Rivest, Shamir, and Adleman have introduced the RSA cryptosystem which was first publicly known as a public-key cryptosystem [12].

The RSA algorithm can be described as follows:

(i) Key generation.

a. Choose two large prime numbers p1 and p2.

b. Compute n = p.q.

c. Randomly chooses a number e smaller than n, such that e and < P(ri) are relatively prime, where <P(n) = (p1-1)(p2-1). Compute d such that ed=l(mod e<P(ri)). Then (n,e) are announced as the public key and d is kept secret.

(ii) Encryption.

The sender computes $C=M^e$ mod n. And C is the encrypted message.

2.3. Chaos Algorithm

Chaos Algorithm proposes the use of Random number generators (RNG) in determining the keys on a cryptosystem[10]. Random number generators in key generation create a phenomenon called avalanche effect. Avalanche effect is so-called avalanche because a slight change in the keys will change the whole entity of the ciphertext. Chaos algorithm is used in systems where encryption/decryption speed is necessary[13].

2.4. RSA and El Gamal

The RSA cryptosystem depends on the IFP or Integer factorization problem[14] and the El Gamal cryptosystem depends on the DLP or Discrete logarithm problem[13][15]. The addition of these two cryptosystems gives additional difficulties in the processing of data being encrypted.

3. METHODOLOGY

Suppose Alice is a customer on an e-commerce website. She wants to purchase an item on the website. To make a valid transaction, Alice needs to send her Credit card number to Bob who is a vendor on that e-commerce website. Inputting such data like the Credit card number to the internet is a risky act. Eve, the hacker, can acquire Alice's Credit card number easily if the data inputted by Alice is not secure. In order to solve this problem, we will introduce an encryption scheme that will secure the Credit card number inputted by Alice, so that it will be protected from Eve's unauthorized access but will be accessible only to the desired receiver which is Bob. Suppose 2 3 8 9 7 6 is the Credit card number of Alice. The E-commerce website must contain the following processes in order to protect its customer's data.

3.1 Cryptographic Algorithm

For Key Generation

1. Choose a prime number p, such that p > 10.
2. Generate the next prime number q, such that q > p.
3. Generate φ(n) ; φ(n) = (p-1)(q-1).
4. Choose a random number e, such that gcd(e, φ(n)) = 1
5. Generate the decryption key d by using the Euclidean algorithm
6. Choose a private key x; such that x < p
7. Choose 6 random keys.
8. Generate y_n; $y_n = g_n^x$(mod p)
9. Generate a number k, such that gcd(k , p-1) = 1.

For Encryption

1. Input the data to be encrypted.
2. Generate r; $r_n = g_n^k$ (mod p).
3. Generate b; b = (m)(d).
4. Generate for s; $s_n = (y_n^k$ (mod p)) (b_n).

For Decryption

1. Input the keys p, x, e, (r_1, s_1), (r_2, s_2), (r_3, s_3), (r_4, s_4), (r_5, s_5), (r_6, s_6).
2. Generate the decryption key d by using the Euclidean algorithm
3. Generate b_n; $b_n = \frac{s_n}{r_n{}^x \ (mod\ p)}$
4. Generated m_n; $m_n = b_n/d$.

3.2 Cryptographic Block Diagram

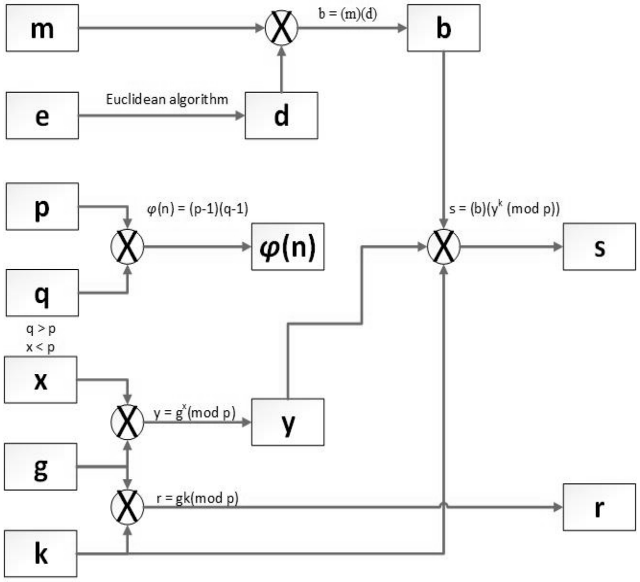


Figure 1. Proposed Cryptosystem for Encryption

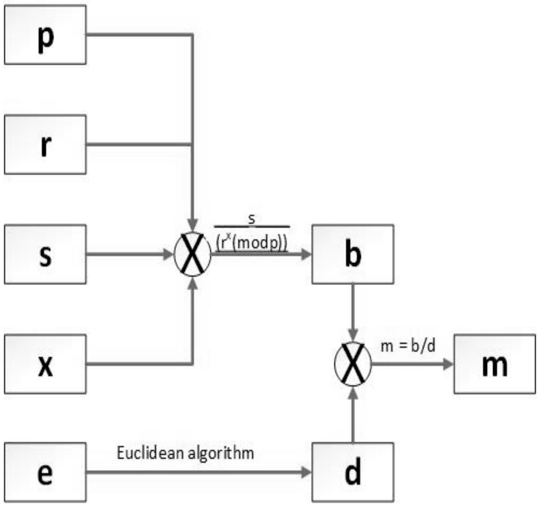


***Figure 2**. Proposed Cryptosystem for Decryption*

3.3 Algorithm Implementation

3.3.A Key Generation

Step1: Choose a prime number p, such that p > 10.
p = 11
Step2: Generate the next prime number q, such that q > p.
q = 13
Step3: Generate φ(n) ; φ(n) = (p-1)(q-1).
φ(n) = (11-1)(13-1)
φ(n) = (10)(12)
φ(n) = 120
Step4: Choose a random number e, such that gcd(e, φ(n)) = 1
e = 7, gcd (7, 120) = 1, they are coprime
e = 7
Step5: Generate the decryption key d by using Euclidean algorithm
e = 7
d = 103
Step6: Choose a private key x; such that x < p
x = 6
Step7: Choose 6 random keys.
Denoted as g_1, g_2, g_3, g_4, g_5, and g_6
$g_1 = 3$; $g_4 = 4$;
$g_2 = 5$; $g_5 = 6$;
$g_3 = 7$; $g_6 = 8$;
Step8: Generate y_n; $y_n = g_n^x (\text{mod } p)$

$y1=g_1^x(\text{mod}11)$ $=3^6(\text{mod}11)$ $= 3$	$y2=g_2^x(\text{mod}11)$ $=5^6(\text{mod}11)$ $= 5$	$y3=g_3^x(\text{mod}11)$ $=7^6(\text{mod}11)$ $= 4$
$y_4=g_4^x(\text{mod } 1)$ $=4^6(\text{mod}11)$ $= 4$	$y_5=g_5^x(\text{mod}11)$ $=6^6(\text{mod}11)$ $= 5$	$y_6=g_6^x(\text{mod}11)$ $=8^6(\text{mod } 11)$ $= 3$

Step9: Generate a number a, such that gcd(k , p-1) = 1.
p = 11
p-1 = 10
k = 3; gcd(3 , 10) = 1
k = 3
Thus public and private keys have been generated.
Keys p, k, e, d, y_1, y_2, y_3, y_4, y_5, and y_6 will be used for encryption
Keys p, x, and e will be sent to Bob for decryption.

3.3.B Encryption

Step1: Input the data to be encrypted.

$m_1 = 2$;
$m_2 = 3$;
$m_3 = 8$;
$m_4 = 9$;
$m_5 = 7$;
$m_6 = 6$;

Step2: Generate r; $r_n = g_n^k$ (mod p).

$r_1=g_1^k(\text{modp})$ $=3^3(\text{mod}11)$ $= 5$	$r_2=g_2^k(\text{modp})$ $=5^3(\text{mod}11)$ $= 4$	$r_3=g_3^k(\text{modp})$ $=7^3(\text{mod}11)$ $= 2$
$r_4=g_4^k(\text{modp})$ $=4^3(\text{mod}11)$ $= 9$	$r_5=g_5^k(\text{modp})$ $=6^3(\text{mod}11)$ $= 7$	$r_6=g_6^k(\text{modp})$ $=8^3(\text{mod}11)$ $= 6$

Step3: Generate b; b = (m)(d).

$b_1= (m_1)(d)$ $= (2)(103)$ $= (206)$	$b_2 = (m_2)(d)$ $= (3)(103)$ $= (309)$	$b_3= (m_3)(d)$ $= (8)(103)$ $= (824)$
$b_4= (m_4)(d)$ $= (9)(103)$ $= (927)$	$b_5 = (m_5)(d)$ $= (7)(103)$ $= (721)$	$b_6= (m_6)(d)$ $= (6)(103)$ $= (618)$

Step4: Generate for s; $s_n = (y_n^k \text{ (mod p)}) (b_n)$.

$s_1=(y_1^k(\text{modp})) (b_1)$ $=(3^3(\text{mod}11))(206)$ $= (5) (206)$ $= 1030$	$s_2=(y_2^k(\text{modp})) (b_2)$ $=(5^3(\text{mod}11))(309)$ $= (4) (309)$ $= 1236$
$s_3=(y_3^k(\text{modp})) (b_3)$ $=(4^3(\text{mod}11))(824)$ $= (9) (824)$ $= 7416$	$s_4=(y_4^k(\text{modp})) (b_4)$ $=(4^3(\text{mod}11))(927)$ $= (9) (927)$ $= 8343$
$s_5=(y_5^k(\text{modp})) (b_5)$ $=(5^3(\text{mod}11))(721)$ $= (4) (721)$ $= 2884$	$s_6=(y_6^k(\text{modp})) (b_6)$ $=(3^3(\text{mod}11))(618)$ $= (5) (618)$ $= 3090$

The cipher text shall be generated $σ_n = r_n, s_n$
σ1 = 5, 1030
σ2 = 4, 1236
σ3 = 2, 7416
σ4 = 9, 8343
σ5 = 7, 2884
σ6 = 6, 3090
These cipher text will be sent to Bob through the internet.

3.3.C. Decryption

Now Bob has received the ciphertext sent by Alice. We will try to decode the encrypted data from the Ciphertext using the decryption keys sent earlier.

Step1: Input the keys p, x, e, (r_1, s_1), (r_2, s_2), (r_3, s_3), (r_4, s_4), $(r_5,$

s_5), (r_6, s_6).

Step2: Generate the decryption key d by using the Euclidean algorithm

e = 7
d = 103

Step3: Generate b_n;

$$b_n = \frac{s_n}{r_n{}^x\,(mod\ p)}$$

$b_1 = \frac{S_1}{r_1{}^x\,(mod\ p)} = \frac{1030}{5^6\,(mod\ 11)} = \frac{1030}{5} = 206$	$b_2 = \frac{S_2}{r_2{}^x\,(mod\ p)} = \frac{1236}{4^6\,(mod\ 11)} = \frac{1236}{4} = 309$	$b_3 = \frac{S_3}{r_3{}^x\,(mod\ p)} = \frac{7416}{2^6\,(mod\ 11)} = \frac{7416}{9} = 824$
$b_4 = \frac{S_4}{r^x\,(mod\ p)} = \frac{8343}{9^6\,(mod\ 11)} = \frac{8343}{9} = 927$	$b_5 = \frac{S_5}{r_5{}^x\,(mod\ p)} = \frac{2884}{7^6\,(mod\ 11)} = \frac{2884}{4} = 721$	$b_6 = \frac{S_6}{r_6{}^x\,(mod\ p)} = \frac{3090}{6^6\,(mod\ 11)} = \frac{3090}{5} = 618$

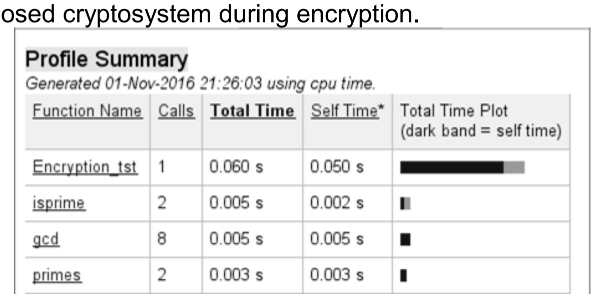
Step4: Generated m_n; $m_n = b_n/d$.

m_1 = b1/d = 206/103 = 2	m_1 = b2/d = 309/103 = 3	m_1 = b3/d = 824/103 = 8
m_1 = b4/d = 927/103 = 9	m_1 = b5/d = 721/103 = 7	m_1 = b6/d = 618/103 = 6

Thus, the Credit Card number has been decrypted to its original form 2 3 8 9 7 6.

4. RESULTS AND DISCUSSION

To verify the authenticity of the proposed encryption scheme, the cryptographic algorithm is simulated using MATLAB. The MATLAB encryption function accepts data inputs and generates the ciphertext which is to be sent to a client. The decryption function accepts a generated ciphertext and transforms it into the original message. The speed of the proposed cryptographic algorithm has also been tested. The speed of the new algorithm has been compared to the speed of the original RSA and El Gamal Cryptosystem. The computer used in the study has a 64-bit Windows 10 Pro operating system. The processor is an Intel Core i5-5200U CPU. And has 4096MB of RAM. Figure 3 shows the runtime of the proposed cryptosystem during encryption.

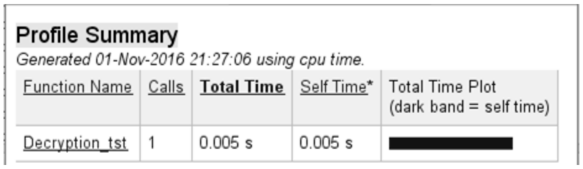


Profile Summary
Generated 01-Nov-2016 21:26:03 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Encryption_tst	1	0.060 s	0.050 s	▇▇▇▇▇▇▇▇░
isprime	2	0.005 s	0.002 s	▮
gcd	8	0.005 s	0.005 s	▮
primes	2	0.003 s	0.003 s	▮

Figure 3. *Runtime during encryption of the proposed system.*

Figure 4 shows the runtime of the proposed cryptosystem during decryption.

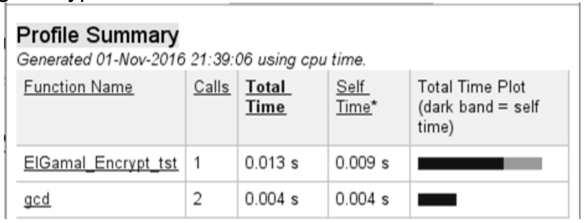


Profile Summary
Generated 01-Nov-2016 21:27:06 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
Decryption_tst	1	0.005 s	0.005 s	▇▇▇▇▇▇▇▇▇

Figure 4. *Runtime during encryption of the proposed system.*

Figure 5 shows the runtime of the El Gamal cryptosystem during encryption.

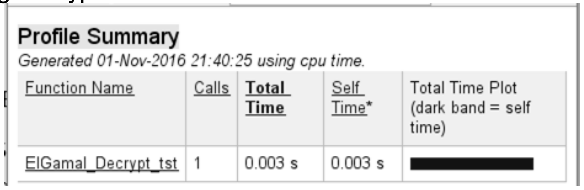


Profile Summary
Generated 01-Nov-2016 21:39:06 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
ElGamal_Encrypt_tst	1	0.013 s	0.009 s	▇▇▇▇▇▇░░░
gcd	2	0.004 s	0.004 s	▇▇▇

Figure 5. *The runtime of the El Gamal cryptosystem during encryption.*

Figure 6 shows the runtime of the El Gamal cryptosystem during decryption.

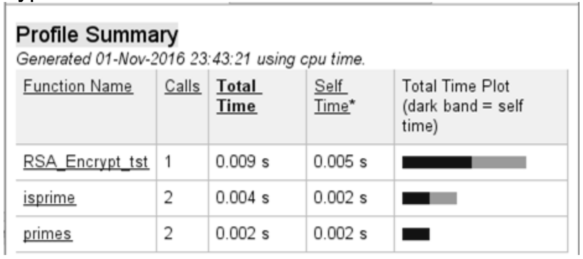


Profile Summary
Generated 01-Nov-2016 21:40:25 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
ElGamal_Decrypt_tst	1	0.003 s	0.003 s	▇▇▇▇▇▇▇▇▇

Figure 6. *The runtime of the El Gamal cryptosystem during decryption*

Figure 7 shows the runtime of the RSA cryptosystem during encryption.

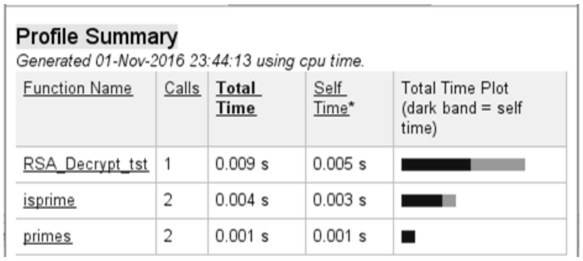


Profile Summary
Generated 01-Nov-2016 23:43:21 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
RSA_Encrypt_tst	1	0.009 s	0.005 s	▇▇▇▇▇░░░░
isprime	2	0.004 s	0.002 s	▇▇░░
primes	2	0.002 s	0.002 s	▇▇

Figure7. *The runtime of the RSA cryptosystem during encryption.*

Figure 8 shows the runtime of the RSA cryptosystem during decryption.



Profile Summary
Generated 01-Nov-2016 23:44:13 using cpu time.

Function Name	Calls	Total Time	Self Time*	Total Time Plot (dark band = self time)
RSA_Decrypt_tst	1	0.009 s	0.005 s	▇▇▇▇▇░░░░
isprime	2	0.004 s	0.003 s	▇▇▇░
primes	2	0.001 s	0.001 s	▮

Figure 8 *Runtime of the RSA cryptosystem during decryption*

Self-time is the time spent in a function excluding the time spent in its child functions. Child functions are the functions used in the test of the cryptosystem, is prime, gcd and primes are some examples. The total time is the overall time that the cryptosystem spent in executing.

Table 1 *Comparison between the proposed cryptosystem, El Gamal, and RSA.*

Cryptosystem	Encryption	Decryption
Proposed	0.060s	0.005s
El Gamal	0.013s	0.003s
RSA	0.009s	0.009s

5. CONCLUSION

The tests have proved that the algorithm of the new cryptosystem is more secure than its parent cryptosystems (RSA and El Gamal). But the speed of the new system is slightly slower than its parents. The RSA algorithm can be slower depending on the chosen encryption key. Overall the new cryptosystem is found efficient to use in credit card number encryption.

6. REFERENCES

[1] Y. A. Nanehkaran, “An Introduction To Electronic Commerce,” Int. J. Sci. Technol. Res., vol. 2, no. 4, pp. 2–5, 2013.

[2] G. Wang, T. N. Wong, and C. Yu, “Computational method for agent-based E-commerce negotiations with adaptive negotiation behaviors,” in International Conference on Computational Science, ICCS 2011 Computational, 2011, vol. 4, pp. 1834–1843.

[3] R. Jia, R. Li, M. Yu, and S. Wang, “E-commerce Purchase Prediction Approach By User Behavior Data,” in 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), 2017.

[4] S. G. E. Garrett and P. J. Skevington, “An introduction to eCommerce,” BT Technol. J., vol. 17, no. 3, pp. 11–16, 1999.

[5] H. Jebur, H. Gheysari, and P. Roghanian, “E-Commerce Reality and Controversial Issue,” Int. J. Fundam. Psychology Soc. Sci., vol. 2, no. 4, pp. 74–79, 2012.

[6] T. M. Nisar and G. Prabhakar, “Journal of Retailing and Consumer Services What factors determine e-satisfaction and consumer spending in e-commerce retailing ?,” J. Retail. Consum. Serv., vol. 39, no. May, pp. 135–144, 2017.

[7] X. He and C. Li, “The Research and Application of Customer Segmentation on E-commerce Websites,” in 2016 6th International Conference on Digital Home, 2016.

[8] W. Stallings, Cryptography and Network Security (4th Edition).

[9] M. Y. Rhee, Internet Security: Cryptographic Principles, Algorithms, and Protocols. Chichester, West Sussex, England ; Hoboken, NJ: J. Wiley, 2003.

[10] M. Enriquez, D. W. Garcia, and E. Arboleda, “Enhanced Hybrid Algorithm of Secure and Fast Chaos-based, AES, RSA, and ElGamal Cryptosystems,” Indian J. Sci. Technol., vol. 10, no. July 2017.

[11] T. Elgamal, “A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms,” IEEE Trans. Inf. THEORY, vol. 31, no. 4, pp. 469–472, 1985.

[12] R. L. Rivest, A. Shamir, and L. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” Commun. ACM, vol. 21, no. 2, pp. 120–126, 1978.

[13] E. R. Arboleda, “Secure and Fast Chaotic El Gamal Cryptosystem,” Int. J. Eng. Adv. Technol., vol. 8, no. 5, pp. 1693–1699, 2019.

[14] J. M. B. Espalmado and E. R. Arboleda, “DARE Algorithm : A New Security Protocol by Integration of Different Cryptographic Techniques,” Int. J. Electrical Comput. Eng., vol. 7, no. 2, pp. 1032–1041, 2017.

[15] E. R. Arboleda, J. L. Balaba, and J. C. L. Espineli, “Chaotic Rivest-Shamir-Adlerman Algorithm with Data Encryption Standard Scheduling,” Bull. Electr. Eng. Informatics, vol. 6, no. 3, pp. 219–227, 2017.