Implementation of Improved DES Algorithm in Securing Smart Card Data

Ariel M. Sison
School of Computer Studies
Emilio Aguinaldo College
Manila, Philippines
ariel@eac.edu.ph

Bartolome T. Tanguilig III
College of Information Technology Education
Technological Institute of the Philippines
Quezon City, Philippines
bttanguilig_3@yahoo.com

Bobby D. Gerardo
Institute of Information and Communications Technology
West Visayas State University
Lapaz, Iloilo City, Philippines
bgerardo@wvsu.edu.ph

Yung-Cheol Byun
Faculty of Telecommunications and Computer Engineering, Cheju National University
Jeju City, Korea
ycb@jejunu.ac.kr

***Abstract*— Although smart cards have already provided secure portable storage device, security is still a major concern to electronic data systems against accidental or unlawful destruction or alteration during transmission or while in storage. One way of ensuring security is through encryption so that only the intended parties are able to read and access the confidential information. The software simulation result proved that the inclusion of the Odd-Even substitution to DES has provided additional confusion technique to DES and was essential in providing adequate security whilst having the stability and speed of handling encryption and decryption processes.**

Keywords-DES; AES; Triple DES; Blowfish; Confusion; Brute Force Attack; Linear Cryptanalysis; Differential Cryptanalysis; Smart Card; Card Skimming.

I. INTRODUCTION

Smart cards have been used in security-sensitive applications from identification and access control to payment systems. Smart cards store confidential information and can deliver secure and accurate identity verification much more difficult to counterfeit than ordinary magnetic stripe cards [2]. These cards can be utilized to provide secure and strong authentication and have been designed to provide greater performance, portability, efficiency, and interoperability. Smart cards enable business establishments to automatically identify, track, and capture information electronically [3].

However, confidential information is vulnerable to potential intruders who may intercept and extract or alter the contents of information during transmission or while in storage [4], [14], [37]. Anyone can interpose a computer in all communication paths and thus can alter or copy parts of messages, replay messages, or emit false material [7]. Secure communication channel is difficult to achieve or there is minimal reliance of network-wide services [8]. As data crosses over an unsecured channel, it is already susceptible to eavesdropping, illegal retrieval, and intended modification [5]. A problem confronting security in an open network includes how to identify the individual making the transaction, whether the transaction has been altered during transmission, or how to safeguard the transaction from being redirected or read to some other destination [6]. Criminals use this opportunity to steal identities and commit fraud [2]. Confidential information can be the subject of manipulation and misuse.

Security measures are needed to safeguard data. According to Zibideh and Matalgah [37], encryption is a vital process to assure security. Encryption is necessary before any data is passed between physical vulnerable networks and decrypted back to plaintext when it is read back from the storage. Encryption is any form of coding, ciphering, or secret writing [17], and a practical means to achieve information secrecy [19].

According to Grabbe [8], Data Encryption Standard (DES) is one of the most widely used symmetric encryption algorithm and its design idea is still used in numerous block ciphers. A symmetric encryption uses series of numbers and letters and some shifting of characters (bits) to alter the message. Over the last three decades, DES has played major role in securing data [9] since it was adopted as Federal Information Processing Standard (FIPS) in November 1977 [10]. It has been endorsed by the United States of America as the standard of encryptions [23], [24].

However, all encryption techniques are subject to attacks. Just like any other encryption techniques (e.g., IDEA, RC5, RC6), DES is no exception. Intruders have exploited its weaknesses to bypass secure encryption to steal sensitive information since it has been publicly known as a standard of encryptions. One major concern surrounding DES security is the key length (56 bits). Intruders have devised attacks that can work against it. Attacks known to have successfully broken DES security are Brute Force (exhaustion attack), Differential Cryptanalysis [18], [19], [20], [36], and Linear Cryptanalysis [21], [22], [36].

 ISBN: 978-1-61208-246-2

While smart card is a secure portable storage device used for several applications, there is a need to look into the security aspects of the device as it has introduced an array of security and privacy issues. Information inside the card could potentially be exploited by an undetected modification or unauthorized disclosure caused by poor design or implementation. Data can be stolen without leaving the users wallet or bag through an illegal activity called *card skimming*. Card skimming involves the intruder hiding a device inside a bag close to a victim's card proximity to steal the data without the victim's knowledge [33].

Since smart cards are becoming prevalent for payment mechanisms (e.g., pay TV access control, transport, supermarket, banks, and cashless vending machines), sending personal information (e.g., health cards, government ID cards), and for security access (e.g., authentication and controlled access to resources); therefore, appropriate measures to protect and secure data during transmission or while in storage must be implemented.

II. REVIEW OF RELATED LITERATURE

A. Evolutions of DES

DES has gone through many enhancements and served as basis for later techniques in the field of encryption. One of its successors is the Triple DES (3DES or TDEA). According to Dhanraj et al. [25], 3DES uses 48 rounds to encrypt the data. Using this technique gives the data three levels of security making it highly resistant to differential cryptanalysis and boosting the security. However, since 3DES involves going through DES three times, its performance also takes three times as long to encrypt and decrypt [26]. The 3DES works by forward and inverse encryptions. DES encrypts with K_1, K_2 and K_3. To decrypt data it starts with K_3, then with K_2 and with K_1 [29].

Ammar et al. [27] proposed an extended DES called Random Data Encryption Algorithm (RDEA). New features added to the DES include pseudo randomized cipher key for encryption and protocols for sending cipher key embedded in the ciphertext. Random generator sequence length and its memory capacity have hampered the RDEA's overall efficiency this along with its weakness to linear attacks to the S-Box and its key scheduling.

Fan Jing et al. [25] proposed the idea of TKE (Two Key), which is versatile in the sense that it can perform faster and works easily with hardware. Aside from these advantages, it also has high-level security like the DES. However, TKE requires two keys and its data block has different length. This gives the process a heavier load slowing the encryption.

Blowfish encryption encrypts a 64-bit block of data using a key with lengths ranging from 32 to 448 bits. The encryption itself is a sixteen round encryption revolving around utilizing S-Boxes and complex key schedules [30]. The strength of the Blowfish lies in the fact that in its full round form cryptanalysis techniques have no effect on it. However, anything less than four rounds are susceptible to cryptanalysis and the algorithm is not immune to brute force attacks [31].

DES has also been used in conjunction with other encryption techniques. Hamami et al. [28] proposed fusion of DES and Blowfish encryption. The proposed fusion aimed to strengthen the key generation of the DES. It encrypts a 64-bit data block using two keys by initial permutation followed by sixteen rounds of crossover iterations using two keys and going through a final inversed permutation. Its weaknesses are the same with regular Blowfish although it offers more resistance to Brute Force attacks but the two keys added to the encryption slowed the process.

AES (Advanced Encryption Standard) is the successor to the DES as a standard for encryptions [32]. AES encrypts data blocks of 128 bits using keys of 128, 192, 256 bits. AES works through phases. First, round keys derived from the main key through key schedules are expanded. In the next rounds or iterations, the plain text is subjected to left shifts as well as column mixing using the derived round keys and S-Boxes. This phase repeats for a final round with the exemption of the column mixing. Although AES has tighter security compared to DES or 3DES, it also has the longest encryption time and load due to all the variables needed to process encryption. AES key schedule has also come under scrutiny since its key schedule is too simple and can be exploited by cryptographers through newer cryptanalysis techniques [34].

B. Known Attacks on DES

Among known attacks on DES, the common techniques used were Brute Force and cryptanalysis techniques. Brute Force is the most basic and effective form of attack on any encryption system to date [35]. It attacks the encryption head on by trying every possible key in a turn. Although Brute Force is proven to work successfully, the machine used and time consumed by the method proved non feasible [23]. Differential Cryptanalysis works by presuming the attacker has a piece of the original plaintext using this knowledge. The attacker diminishes the security of the encryption until he deciphers the key. To break the full 16 rounds, differential cryptanalysis requires 2^{47} chosen plaintexts. Linear Cryptanalysis works like differential cryptanalysis although it only needs 2^{43} known plaintexts due to its linear nature. The number of rounds used by the DES defines these types of attacks. The shorter the rounds give higher probabilities of success for such techniques. Analysts gain knowledge on the security margins needed by DES through these attacks.

Both cryptanalysis attacks require the attacker to gain a part of the plaintext, which gives the method tricky prerequisites.

 ISBN: 978-1-61208-246-2

III. DESIGN ARCHITECTURE OF THE MODIFIED DES

A. Key Encryption Process

In Figure 1, the encryption process starts by converting a 64-bit key into a binary value. The result is reduced to 56-bit and went through the Odd-Even substitution. The Odd-Even substitution process substitutes 1 for every even position and 0 for every odd position in the 56-bit block. Afterwards, the 56-bit block is divided into two 28-bit halves (C_0 and D_0). Each half contains 28-bits and performs the left shift. After shifting is applied, the two halves are combined and reduced to 48 bits. The pipes (||) demonstrate the combination of the two halves. The 48-bit produced is now the first key. The 56 bits (C_1 and D_1) are used to generate the remaining keys and undergo the same steps in generating the first key. This will be performed for 16 rounds. The graphical illustration of key generation process presented in Figure 1 is simulated in Table I. After 16 rounds of iterative operation, the 16 keys produced are shown in Table II.

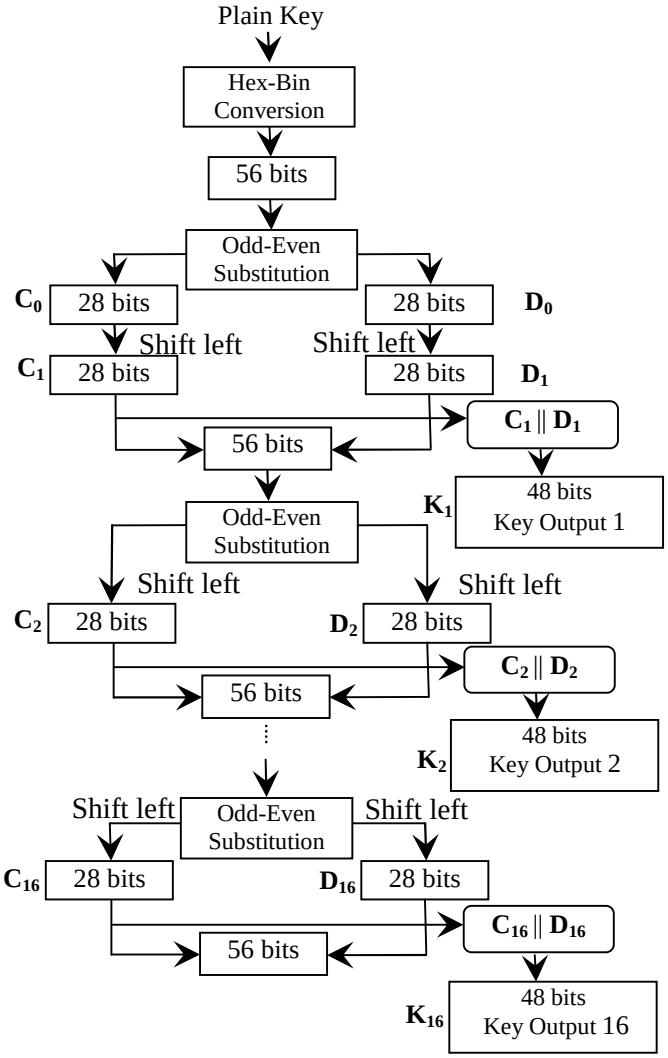

Figure 1. Modified DES algorithm key generation process.

The *Odd-Even* substitution process provided additional confusion to DES. Confusion is one of the two basic techniques of cryptography [35] that is achieved through the XOR operations making the relationship between the ciphertext and the key complex as possible. The enhancement was simple that it does not slow down the whole process of encryption.

TABLE I. ILLUSTRATION OF THE MODIFIED KEY GENERATION PROCESS

Step	Process	Result	
1	Convert the key *p@SSWoRD123 45TiP* from hexadecimal to binary value	01110000010000000101001101010011010111011011101101001001000100000110001001100010001100110011001000011010101010001101001010100000	
2	Reduce the result to 56 bits using the permuted choice 1.	00000000111111110010000101010111110010110000001000001101	
3	Apply Odd-Even substitution to the result.	01010101010101010101010101010101010101010101010101	
4	Divide the result into two halves.	C_0 01010101010101010101010101	D_0 01010101010101010101010101
5	Shift left both C_0 and D_0.	10101010101010101010101010	10101010101010101010101010
6	Assign C_0 to C_1 and D_0 to D_1.	C_1 10101010101010101010101010	D_1 10101010101010101010101010
7	Combine C_1 and D_1 to produce the 56-bits then apply permuted choice 2 to produce the 48-bit key output.	01101110101011000001101010111100111001100100000010 (6eac1abce642)	
8	Combine C_1 and D_1 in Step 6 to generate the next keys (C_2 and D_2, C_3 and D_3 ... C_{16} and D_{16}) and repeat Step 3 to Step 7. Perform this for 16 rounds.		

TABLE II. KEY GENERATION RESULT

Key	Value
1	6eac1abce642
2	6eac1abce642
3	9153e54319bd
4	9153e54319bd
5	9153e54319bd
6	9153e54319bd
7	9153e54319bd
8	9153e54319bd
9	6eac1abce642
10	9153e54319bd
11	9153e54319bd
12	9153e54319bd
13	9153e54319bd
14	9153e54319bd
15	9153e54319bd
16	6eac1abce642

 ISBN: 978-1-61208-246-2

B. *Plaintext Encryption Process*

In the plaintext encryption, the input is of any length. This is a significant contribution to DES since any size or length of plaintext could already be encrypted. The plaintext is then subsequently divided into 64-bit block plaintext. This is shown in Figure 2. For example, if the given plaintext is *I will meet you at 7:00am today at the park*, this will be grouped into 8 characters per block where each character is 8 bits. The result is shown in Table III. Take note that at Block 6, the length of the remaining characters is *3*, which is less than 8 characters. The system automatically padded 5 spaces to make it 8 characters. Each block is encrypted using the 16 keys generated in Table II.

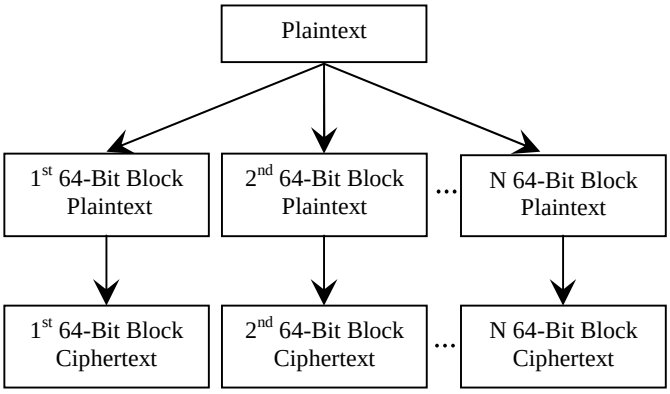

Figure 2. Division of the plaintext into 64-bit block.

TABLE III. 64-BIT BLOCK PLAINTEXT

Block	Plaintext
1	I will m
2	eet you
3	at 7:00a
4	m today
5	at the p
6	ark

In Figure 3, the plaintext is converted to 64 bits and is divided into two 32-bit halves (L_0 and R_0). The value of R_0 is first assigned to L_1 ($L_1 = R_0$) before undergoing E-Bit selection process to produce the 48 bits needed. The result will be XORed (⊕) to the first key in Table 2. The XOR result will then be grouped to 8 blocks. Each block consists of 6 bits. Afterwards, the S-Box $_{(1..8)}$ substitution will be applied. Each S-Box $_{(1..8)}$ has 4 rows and 16 columns with a corresponding value. For example, if the first block is 101000, get the first and the last bits. So $10_2 = 2$ denotes row 2. The remaining middle 4 bits $0100_2 = 4$ denotes column 4. S-Box$_1$ [2][4] will return the value of *13* or *D* in hexadecimal value. Repeat this for the other remaining 7 blocks. Permutation function is then applied to the result to produce 32 bits. Next is to XOR L_0 with the permutation value. The result is then assigned to R_1. $L_{n+1} = R_n$ are swapped to proceed to the next round. This means that $L_2 = R_1$, $L_3 = R_2$, so on and so forth. Iteratively perform this for 16 rounds. Table IV simulates the graphical illustration of Figure 3.

Table IV discussed the encryption process of the plaintext in the first block of Table III.

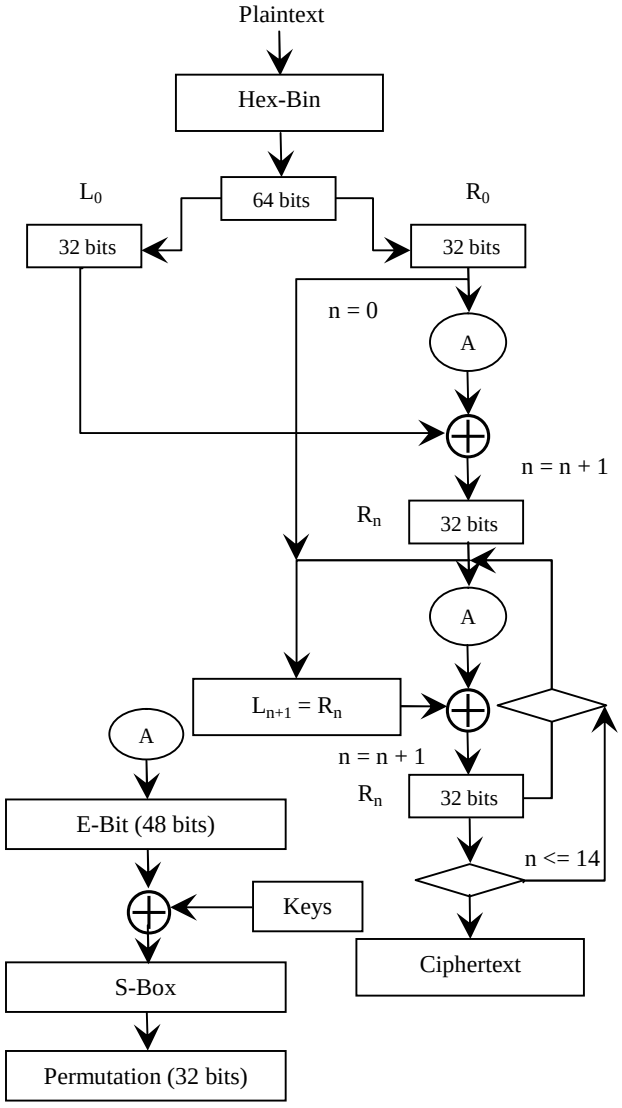

Figure 3. Plaintext encryption process.

TABLE IV. ILLUSTRATION OF THE ENCRYPTION PROCESS OF THE PLAINTEXT PER 64-BIT BLOCK

Step	Process	Result	
1	Convert the text *I will m* from hexadecimal to binary value.	01001001001000000111011101101001011011000110111000001000001101101	
2	Apply initial permutation to the result then convert to hexadecimal value.	bd04b48d00feb904	
3	Divide the result into two halves to form the Left and Right values.	L_0	R_0
		bd04b48d	00feb904

4	Apply E-bit selection table to the result of R_0.	0017fd5f2808
5	Using the first key in Table II, perform XOR with R_0.	6ebbe7e3ce4a
6	Apply S-Boxes substitution to the result.	5f766bef
7	Apply permutation function to the result.	5c7ffcb7
8	Perform XOR with L_0 and the result.	e17b483a
9	Assign the value of R_0 to L_1.	L_1
		00feb904
10	Assign the result of Step 8 to R_1.	R_1
		e17b483a
11	Repeat Steps 3 to 10 having L_1 and R_1 as input to the next round. Perform this for 16 rounds using the remaining keys in Table II. This will produce 16 round output block as shown in Table VI for decryption.	
12	Concatenate the value of R_{16} *(1ba732b1)* and L_{16} *(69bbb462)* from Table VI. Apply inverse on initial permutation (IP^{-1}) to the result. This is now the encrypted value of *I will m*.	Encryption Value: f17618e06dbf8239
13	Read the next plaintext in Table III then perform Steps 1-12 to have the next encrypted value.	

After performing the steps in Table IV, the complete encrypted values of Table III are shown in Table V.

TABLE V. ENCRYPTION RESULT OF THE PLAINTEXT PER 64-BIT BLOCK

Plaintext (64-bit block)	Ciphertext
I will m	f17618e06dbf8239
eet you	a0032d56e3fda715
at 7:00a	2460491262022a41
m today	228db1b8c8102503
at the p	7846ab84750c1e3b
ark	f2d0e39d5784b196

C. Decryption Simulation Process

Although encryption and decryption use the same algorithm, the key processing is performed in reverse order during the decryption process and the input is the ciphertext. Development of the decryption process is necessary to make sure that the modified DES algorithm can decrypt the ciphertext back to its original form.

After finishing the encryption process illustrated in Table IV, the 16 round output block is also generated for each ciphertext block and is shown in Table VI. The decryption starts by applying E-Bit selection table to L_1 *(00feb904)* and K_1 *(6eac1abce642)* getting the result of *6ebbe7e3ce4a*. S-Boxes substitution is then applied to the XOR result to have *5f766bef* followed by permutation function to obtain *5c7ffcb7*. XOR *5c7ffcb7* and R_1 *(e17b483a)* to get the new value for R_1 = *bd04b48d*. This process is shown in Figure 4.

Finally, by concatenating R_1 || L_1 yields to *bd04b48d00feb904*. Thus, $X = IP^{-1} = (L_1 \,||\, R_1) =$ *492077696c6c206d*. Subsequently convert this hexadecimal to ASCII value. Hence, the plaintext *I will m* is recovered.

TABLE VI. ENCRYPTION BLOCK OF THE FIRST CIPHERTEXT

Index	L	R
1	00feb904	e17b483a
2	e17b483a	f6960904
3	f6960904	cd843f3f
4	cd843f3f	75853adf
5	75853adf	b62ff04d
6	b62ff04d	727534d3
7	727534d3	48c03366
8	48c03366	4b5a3d7d
9	4b5a3d7d	8542aa8b
10	8542aa8b	512adb84
11	512adb84	3bbf9555
12	3bbf9555	0d816157
13	0d816157	2bde0e6e
14	2bde0e6e	bd378359
15	bd378359	69bbb462
16	69bbb462	1ba732b1

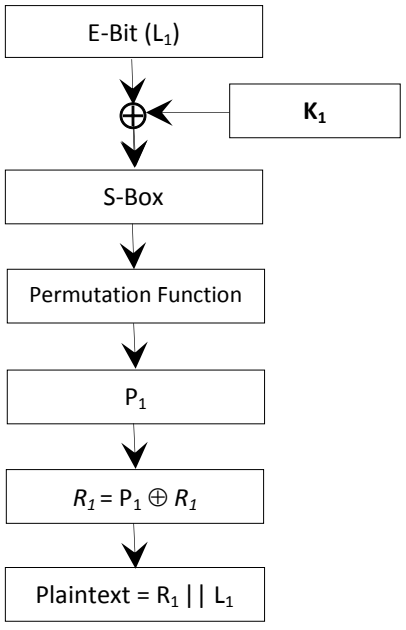

Figure 4. Ciphertext decryption process.

D. Implementation

The smart card device used in the study was ACR122U type. The unit is a contactless reader/writer with an effective proximity of 5 mm. The software was written in Java and

developed using the Eclipse Integrated Development Environment. The average running time of the five attempts with the modified DES was 365.2 milliseconds while 355.8 milliseconds with the typical DES. There is a relatively slight difference in the utilization of the CPU's memory using the typical and the modified DES. This is presented in Table VII. The Average CPU usage is the average of the five attempts in executing both algorithms.

Although there is no performance comparison on how fast the data was written in the card, the encrypted data was successfully written in and read from the card without any problem.

TABLE VII. CPU UTILIZATION COMPARISON

CPU Processor (Intel Core i3 2.67 GHz, 2GB RAM)	Minimum CPU Usage	Maximum CPU Usage	Average CPU Usage
Typical DES	10%	18%	14%
Modified DES	18%	25%	21.2%

IV. CONCLUSION AND FUTURE WORK

Smart card is like an *electronic wallet* replacing all of the things we carry around in our wallets, including credit cards, licenses, cash, and even family photographs and is like carrying digital credentials [4]. There is no doubt that smart cards will be the next generation of the highest level of security card technology that will soon replace magnetic stripes, bar code, and some proximity technologies. It will soon play significant role in personal identity verification.

According to Gong-bin et al. [36], improvement and perfection to DES are still very important. Encrypting information ensures that only the intended parties are able to read and access the confidential information inside the smart card. The inclusion of the Odd-Even substitution to DES ensures that even the data is intercepted by other networks or is redirected to other destinations; its integrity and confidentiality will not be compromised. More so, the Odd-Even substitution has provided additional confusion to the complexity of security capable of resisting cryptanalysis whilst having the stability and speed of handling encryption and decryption processes. Aside that this enhancement can be easily implemented to smart card, it can encrypt or decrypt any length of plaintext and does not require intensive processing, memory, and time compare to AES or 3DES.

Future plans for the study include separate keys for each 64-bit block. Hopefully, additional keys will provide the algorithm more resistance to all known attacks of DES.

REFERENCES

[1] C. Chang and K. Hwang, "Some forgery attacks on a remote user authentication scheme using Smart Cards", Informatica, 2003, Vol. 14, No. 3, pp. 289-294.

[2] W. Wang, Y. Yuan, and N. Archer, "A contextual framework for combating identity theft", IEEE Security & Privacy, Vol. 4, Issue 2, pp. 30-38, 2006.

[3] A. Reid, "Is society smart enough to deal with smart cards?", Computer Law & Security Report, Vol. 23, Issue 1, pp. 53-61, 2007.

[4] B. Lewis, "Making Smart Cards work in the enterprise", SANS Institute InfoSec Reading Room, SANS Institute 2002, http://www.sans.org/reading_room/whitepapers/authentication/making-smart-cards-work-enterprise_138 [retrieved: April, 2011].

[5] P. Rakers, L. Connell, T. Collins, and D. Russel, "Secure contactless smartcard ASIC with DPA protection", IEEE Journal of Solid-State Circuits, Vol. 36, Issue 3, pp. 559-565, 2001.

[6] J. McAndrews, "E-money and payment system risks", Contemporary Economic Policy, Vol. 17, Issue 3, pp. 348-357, July 1999.

[7] R. Needham and M. Schroeder, "Using encryption for authentication in large networks of computers", Communications of the ACM, Vol. 21, No. 12, pp. 993-999, December 1978.

[8] O. Grabbe, "The DES algorithm illustrated", Laissez Faire City Times, Vol. 2, No. 28. (http://www.doc88.com/p-281792313650.html) [retrieved: April, 2011].

[9] K. Rabah, "Theory and implementation of data encryption standard: a review", Information Technology Journal, Vol. 4, pp. 307-325, Asian Network for Scientific Information, 2005.

[10] Data Encryption Standard, Federal Information Processing Standards Publication (FIPS Pub) 46, National Bureau of Standards, Washington, DC, 1977.

[11] M. Hwang and L. Li, "A new remote user authentication scheme using smart cards", IEEE. Transactions Consumer Electron, Vol. 46, No. 1, pp. 28-30, 2000.

[12] A. Awasthi and S. Lal, "An enhanced remote user authentication A. scheme using smart cards", IEEE Transactions on Consumer Electronics, Vol. 50, No. 2, pp. 583-586, May 2004.

[13] X. Zhao and F. Zhang, "A new type of id-based encryption system and its application to pay-tv systems", International Journal of Network Security, Vol. 13, No. 3, pp. 161-166, November 2011.

[14] T. Diament, H. Lee, A. Keromytis, and M. Yung," The efficient dual receiver cryptosystem and its application", International Journal of Network Security, Vol. 13, pp. 135-151, November 2010.

[15] M. Peyravian and N. Zunic, "Methods for protecting password transmission", Computers & Security, Vol. 19, No. 5, pp. 466-469, 2000.

[16] R. Needham and M. Schroeder, "Using encryption for authentication in large networks of computers", Communications of the ACM, Vol.21, Issue 12, pp. 993-999, December 1978.

[17] L. Gilman, "Encryption of data", Encyclopedia of Espionage, Intelligence, and Security. (http://www.faqs.org/espionage/Ec-Ep/Encrption -of-Data.html) [retrieved: April, 2011].

[18] E. Biham and A. Shamir, "Differential Cryptanalysis of DES-like Cryptosystems", Journal of Cryptology, Vol. 4, pp. 3-72, IACR, 1991.

[19] D. Coppersmith, "The data encryption standard (DES) and its strength against attacks", IBM Journal of Research and Development, Vol. 38, Issue. 3, pp. 243-250, May 1994.

[20] N. Courtois, "The best differential characteristics and subtleties of the Biham-Shamir attacks on DES", IACR Cryptology ePrint Archive, Vol. 2005, pp. 202, 2005, (http://eprint.iacr.org/2005/202).

[21] P. Junod, "Linear cryptanalysis of DES", http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.12.9652 [retrieved: June, 2012].

[22] C. Harpes, G. Kramer, and J. Massey, "A generalization of linear cryptanalysis and the applicability of Matsui's piling up lemma", Theory and Application of Cryptographic Techniques - EUROCRYPT, Vol. 921, pp. 24-38, 1995.

[23] M. Smid and D. Branstad, "The Data Encryption Standard Past and Future", Proceedings of the IEEE, Vol .76, Issue 5, pp. 550-559, May 1988, in press.

[24] United States Department of Commerce/National Institute of Standards and Technology, FIPS PUB 46-3 Reaffirmed October 1999.

 ISBN: 978-1-61208-246-2

[25] American National Standard for Financial Institution Message Authentication (wholesale), ANSI x9.9-1986 (revised), American Bankers Association, X9 Secretariat, American Bankers Association, 1986.

[26] Dharaj, C. Nandini, and M. Tajuddin, “An enhanced approach for secret key algorithm based on data encryption standard”, International Journal of Research and Reviews in Computer Science (IJRRCS) Vol. 2, No. 4, pp. 943-947, ISSN: 2079-2557, Science Academy Publisher, United Kingdom, August 2011.

[27] America National Standard for Personal Identification Number (PIN) Management and Security, ANSI x9.8-1982, American Bankers Association, Washington, D.C., 1982.

[28] A. Al-Hamami, M. Al-Hamami, and S. Hashem, “A proposed modified data encryption standard algorithm by using fusing data technique”, World of Computer Science and Information Technology Journal, Vol. 1, No. 3, pp. 88-91, 2011.

[29] W. Barker, “Recommendation for the triple data encryption algorithm (TDEA) block cipher”, U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, Gaithersburg, MD, United States, 2004.

[30] B. Schneier, ”Description of a new variable-length key, 64-bit block cipher (blowfish)”, Fast Software Encryption, Cambridge Security Workshop, pp. 191-204, Springer-Verlag London, UK, 1994.

[31] S. Vaudenay, “On the weak keys of blowfish”, Fast Software Encryption, Cambridge, United Kingdom, Lecture Notes in Computer Science No. 1039, pp. 27-32, Springer-Verlag , 1996.

[32] H. Alanazi, B. Zaidan, A. Zaidan, H. Jalab, M. Shabbir, and Y. Al-Nabhani, "New comparative study between DES, 3DES and AES within nine factors“, Journal of Computing, Vol. 2, Issue 3, pp. 152-157, March 2010.

[33] Consumer Reports Magazine: June 2011, ”Newer cards can be hijacked, too”, http://www.consumerreports.org/cro/magazine-archive/2011/june/june-2011-toc.htm [retrieved: June, 2012].

[34] A. Bogdanov, D. Khovratovich, and C. Rechberger, ”Biclique cryptanalysis of the full AES”, ASIACRYPT'11 Proceedings of the 17th international conference on The Theory and Application of Cryptology and Information Security, pp. 344-371, Springer-Verlag Berlin, Heidelberg, 2011, in press.

[35] C. Shannon, "Communication theory of secrecy systems", Bell System Technical Journal, Vol. 28, pp. 656-715, 1949.

[36] Q. Gong-bin, J. Qing-feng, and Q. Shui-sheng “A new image encryption scheme based on DES algorithm and Chua’s circuit”, IEEE International Workshop on Imaging Systems and Techniques, pp. 168-172, Shenzhen, China, May 2009, in press.

[37] W. Zibideh and M. Matalgah, “Modified-DES encryption algorithm with improved BER performance in wireless communication”, 2011 IEEE Radio and Wireless Symposium (RWS), pp. 219-224, Phoenix, AZ, January 2011, in press.

 ISBN: 978-1-61208-246-2