A Comparison of Data Encryption Algorithms with the Proposed Algorithm: Wireless Security

Shadi R. Masadeh, Shadi Aljawarneh, Nedal Turab
Faculty of Information Technology
Isra University
P.O. Box 22, Al-Isra University Postoffice Amman,
Jordan 11622
{Masadeh, shadi.jawarneh, Nedalturab}@ipu.edu.jo

Aymen M. Abuerrub
MIS Dept, Management and Finance Science Faculty
Al-Ahliyya Amman University
Amman, Jordan
aaburub@ammanu.edu.jo

Abstract - **Encryption algorithms play a main role in wireless network security systems. However, those algorithms consume a significant amount of computing resources such as CPU time, and packet size. In an attempt to remedy the wireless network security issue, a novel work has been deployed to secure the transmitted data over wireless network, called a secure WiFi (sWiFi) algorithm. The sWiFi algorithm is based on developed HMAC cryptography algorithm. This paper also provides evaluation of five encryption algorithms: AES (Rijndael), DES, 3DES, Blowfish, and the proposed algorithm (sWiFi). We examine a method for analyzing trade-off between efficiency and security. A comparison has been conducted for those encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different platforms and different encryption/decryption speed. The experimental evaluation shows that the sWiFi algorithm could provide an extra level of wireless security with relatively higher performance compared with other existing algorithms for e-content delivery applications over different zones of a wireless network.**

Keywords:-Date Encryption Standard (DES), Advanced Encryption Standard (AES),Blowfish, Wi-F- (Wirleess Fideleity).

I. INTRODUCTION

Wireless networks have important role for users and organizations. The wide spread for using wireless networks makes the need for protection of transferred data. Cryptography algorithms are the substantial core of the information security. Furthermore, those algorithms consume a significant amount of computing resources such as CPU time.

Multiple cryptography algorithms are being used nowadays for securing communication channels using public-key exchange includes DES, AES, Triple DES and Blowfish. A public-key exchange depends on a key which is generated through time and mathematical process to encrypt the data that is sent over the unsecured channels of the Internet [1, 2]. It should be noted that main strength of symmetric key encryption depends on the size of key used.

However, wireless networks fall into several categories, depending on the size of the physical area that they are capable of covering. Thus, security and wireless communication will remain an interesting subject for years to come [1, 2].

The common encryption algorithms include DES, 3DES, Blowfish, and AES. DES uses one 64-bits key. 3DES uses three 64-bits keys while AES uses various (128,192,256) bit keys. In Blowfish, various (32-448) bit keys are used [3, 4, 5, 6]. Short definitions of the most common encryption algorithms are mentioned as follows:

DES: (Data Encryption Standard) was the first encryption standard to be used by National Institute of Standards and Technology. DES is 64 bits key size with 64 bits block size. However, several attacks and methods recorded the drawbacks of DES, which made it an insecure block cipher [6].

3DES is an improvement of DES; it is 64 bit block size with 192 bits key size. The encryption method is similar to the one in the original DES but applied 3 times to increase the encryption level and the average safe time. But a number of studies indicated that 3DES is slower than other block cipher methods in terms of performance [6].

Blowfish is a block cipher (64). It takes a variable-length key, starting from 32 bits to 448 bits. Blowfish is an unpatented, license-free, and is available free for all clients. Blowfish has nearly variants of 14 rounds [5].

AES (called Rijndael) is also a block cipher. It has variable key length of 128, 192, or 256 bits. It encrypts data blocks of 128 bits in 10, 12 and 14 round relying on the key size. AES seems fast and flexible-- it can be implemented on different platforms especially in small devices [10]. Also, AES has been carefully tested for many security applications [7-9, 11].

In this paper, a novel algorithm has been developed to secure the data over a wireless network called a secure Wireless Fidelity (sWiFi) algorithm. The experimental testing has provided evaluation of four encryption algorithms (i.e. AES, DES, 3DES, and Blowfish) compared with the developed sWiFi algorithm. The sWiFi adopts asymmetric encryption technique. Asymmetric encryption algorithms are almost 1000 times slower than Symmetric algorithms, because they involve more computational processing power [3].

Our review findings have illustrated that the existing cryptography algorithms relied on a data splitting model that

designed by Fiestel form IBM [12-14]. The Feistel structure has the advantage that encryption and decryption operations are very similar, even though identical in some cases, involving only a reversal of the key schedule. Therefore the size of the code or circuitry required to deploy such a cipher is approximately halved. For the above reasons, the proposed algorithm has taken advantage of the Feistel cipher in the sWIFI design.

In brief, the sWIFI is split into two halves; the second half is divided into four parts for key purpose. The second step is to use the bit shifting and logical OR-ing the data in the four parts in a certain pattern to generate the key and encrypt the generated key which is used to encrypt the first half of the plain text and disrupt the probabilistic phenomena of the letters in the language, and then shuffle the encrypted data within the blocks so there would be no resemblance to the original data.

The rest of this paper is organized as follows. Related work is described in Section 2. Section 3 describes the design of the proposed algorithm (sWiFi). Experimental Designs are discussed in Section 4. Finally conclusions and future work are drawn section 5.

II. RELATED WORK

In this section, we have surveyed a number of studies that make comparison in terms of performance between the common encryption algorithms including AES, DES, 3DES, RC2, Blowfish, and RC6.

Hirani and others [15] concluded performance comparisons between the common encryption algorithms. The results of comparisons stated that AES is faster and more efficient than other encryption algorithms. When the transmission of data is taken into account there is insignificant difference in performance of different symmetric key schemes.

A study in [16] is performed for different common secret key algorithms including DES, 3DES, AES, and Blowfish. Their performance was compared by encrypting input files of varying contents and sizes. The results illustrated that Blowfish had a good performance compared to other encryption algorithms. It also showed that AES had a better performance than 3DES and DES. In addition, it showed that DES has almost triple throughput of 3DES.

Ruangchaijatupon et al. [4] proved that the energy consumption of different common symmetric key encryptions on hand-held devices. They found that after only 600 encryptions of a 5 MB file using 3DES, the remaining battery power is 45% and subsequent encryptions are not possible as the battery dies rapidly.

A Crypto++ Library [17] is a free C++ class library of cryptographic schemes. It evaluates the most commonly used encryption algorithms. It is also shown that Blowfish and AES have the best performance compared with other encryption algorithms.

In [16, 18], they present a performance evaluation of selected symmetric encryption algorithms. The selected algorithms are AES, DES, and 3DES, RC6, Blowfish and RC2. Several points can be concluded from the simulation results. First, in the case of changing packet size, it was concluded that Blowfish has better performance than other common encryption algorithms used, followed by RC6. Secondly, they found that 3DES still has low performance compared to algorithm DES. Thirdly, they found RC2 has disadvantage over all other algorithms in terms of time consumption. Fourthly, they found AES has better performance than RC2, DES, and 3DES. However, they conducted the experiments only one platform: Windows OS.

Salama et al. [19] conducted a comparison between encryption algorithms (AES, DES, and 3DES, RC2, Blowfish, and RC6) at different settings for each algorithm such as different sizes of data blocks, different data types, CPU time, and different key size. Simulation results are given to demonstrate the effectiveness of each algorithm. The algorithms were tested on two different hardware platforms. The results indicated that the Blowfish had more efficient compared to other algorithms. Also it showed that AES had a better performance than 3DES and DES.

Elkilani et al. [18] tested the encryption algorithms such as RC4, AES, and XOR to find out the overall performance of real time video streaming. The results showed that there was time overhead of AES which was less than the overhead using RC4 and XOR algorithm. Therefore, AES is a feasible solution to secure real time video transmissions.

III. DESIGN OF THE SECURE WIRELESS FIDELITY ALGORITHM

We have designed a novel algorithm in attempt to solve the problem of insecure zones over a wireless networks. This algorithm is called a secure Wireless Fidelity (sWiFi).

A. The design process of sWiFi algorithm

We have adopted an Automata Theorem (AT) to design of sWiFi architecture. The AT states if there exists a finite set of characters representing a language L, closed over a character set ($\sum$). It should be noted that the character set ($\sum$) in the sWiFi design denotes to the ASCII code set, so that $\sum$= {W| all characters of the ASCII character set*}.

W represents the set of all words W. In other words, W includes all the words that could be assembled by all legal characters in the language's alphabet in any order with any number of repetitions.

Since L is an ASCII character set, then there exists a function that applies on L elements, and consequently the generated output would still be a subset of L. Assume that S is a function to scramble L, and then the Encryption Key Generation Formula is:

Equation 1: Encryption Key Generation

$$(L) = \sum_{m=0}^{n-1} W \mid P[m] >> K \mid P[n-m] << K\}$$

Where ∑ is the character set of all the ASCII codes, n is the number of bytes in a given string or block, m is a loop iterater to access individual alphabets in the given subset, S is the Scrambled information (Cipher Text) and P is the Plaintext information, K is the number of bits to shift by, I is the size of the plain text in bits. (i.e. K= I / 8).

The S function applies the following:

1. **A logical right shift** to the upper side of a byte, and
2. **Logically left shifted** lower part of another byte over the field of legal alphabets of the language L.
3. **Logically OR** Lower and upper side of byte.

Note that S (L) and P (L) are subsets of L. By induction of testing our work, the function P (L) is the reverse of S (L). Through applying the function S (L) over the scrambled text will produce the original plaintext of the original message. This algorithm will be applied to the encryption key which will be used in encrypting the message.

Since the encryption key was used in ciphering the first part of the plain text, then there is a need to reverse the encryption cycle of the encryption key. So that the decrypted key is 1's complemented and then applied to the decryption process of the first half.

Equation 2: HMAC Generation

HMAC= SHA-1 ("secret key" +SHA-1 ("64 bit Ciphered Text " + "secret key")).

In this paper, a Message Authentication Code (MAC) has been used in our secure environment to ensure the data integrity. HMAC is used to compute the checksum of 64 bit Ciphered Text. We compute a MAC using one-way hash functions (SHA-1) and a changeable secret key SK to create a special fingerprint for 64 bit Ciphered Text. The function in Equation 2shows how to compute the MAC value using SHA-1 hash function.

IV. EXPERIMENTAL DESIGN

Vista and Linux results were obtained from the same personal computer running both as a multi-boot system on an AMD Turion 64x2 machine.

The following tasks that will be performed are shown as follows:

- ✓ A comparison is conducted between the results of the selected different encryption and decryption schemes in terms of the encryption time, and packet size.
- ✓ A study is performed on the effect of changing packet size on CPU work load for each selected cryptography algorithm.

A. Empirical results:

An Empirical result is used to show the efficiency of the Encryption algorithms vs. sWIFI algorithm. Table 3 and Figure 5 show the real time in seconds for per algorithm with using different sample size of files (900MB, 510MB,145MB) with different Encryption algorithms such as AES-128, DES, Triple DES , Blow Fish and sWIFI algorithm , these results have been applied by different operating system such as **Windows XP , Windows Vista and Linux .**

In Figures 5, 6 and 7, we show the performance of cryptography algorithms in terms of sharing the CPU load for encryption process. With a different data block size

	WINDOWS VISTA	WINDOWS XP	LINUX
Encryption	900MB	900MB	900MB
AES_128	31	86.2	86.2
DES	42.2	109	109
3DES	93.2	174.3	174.3
Blowfish	34.8	98.3	98.3
sWIFI	30.1	64.2	63.1

Table4: Comparison between different OS with 900 MB

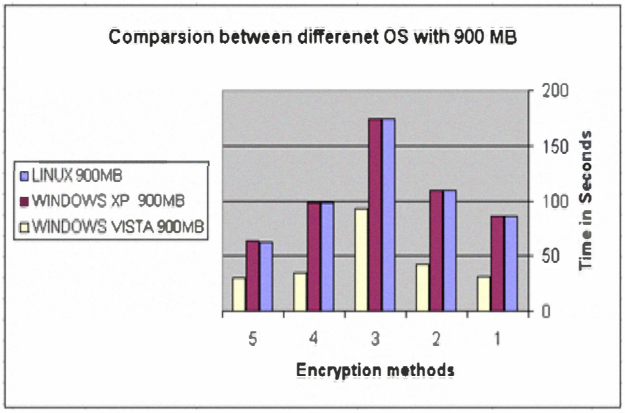

Figure 5: Time consumption for encrypt deferent text data: Comparison between different OS with 900 MB

	WINDOWS VISTA	WINDOWS XP	LINUX
Encryption	510MB	510MB	510MB
AES_128	10.8	53.8	53.8
DES	18.3	64.7	64.7
3DES	48	97.3	97.3
Blowfish	13.6	58.2	58.2
sWIFI	8.4	44.6	35.3

Table5: Comparison between different OS with 510 MB

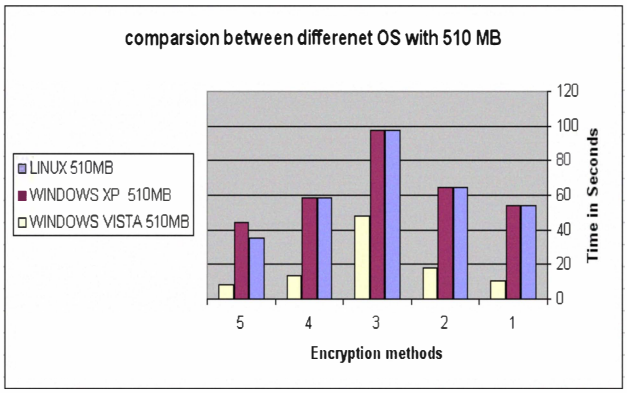

Figure 6: Time consumption for encrypt deferent text data: Comparison between different OS with 510 MB

	WINDOWS VISTA	WINDOWS XP	LINUX
Encryption	145MB	145MB	145MB
AES_128	2.9	16.1	16.1
DES	5.5	19.5	19.5
3DES	14.5	27.4	27.4
Blowfish	4	18.7	18.7
sWIFI	2.5	17.7	10.6

Table6: Comparison between different OS with 145 MB.

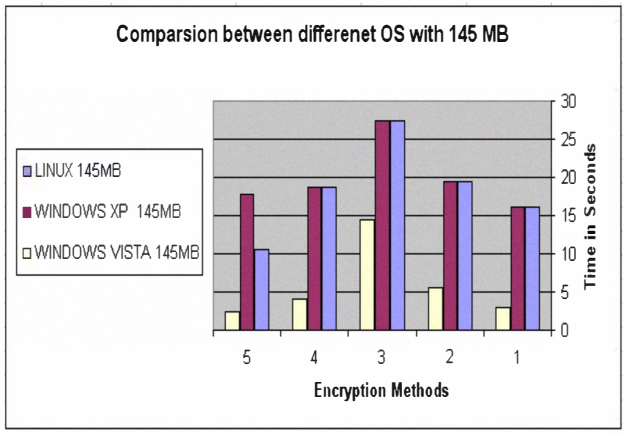

Figure 7: Time consumption for encrypt deferent text data: Comparison between different OS with 145 MB

The variations in the readings for the encryption algorithms is due to many factors out of which are the number of processes are running in the algorithm and process management scheme that is adopted by the operating system in use.

In Windows XP environment the operating system is using a task switching and priority scheduling scheme in which the running task is given the highest priority that is why in Windows XP, the timing is better off than that in other OS.

The second factor is that the Windows XP operating system is designed to be run be a single user where in other OS is originally designed to be used by multi-users at the same time, which is why the processes might run much faster than that running on other OS.

The third factor is that the tasks are co-operative multitasking in Windows XP and that the working set tuner adjusts the sets according to memory needs using the clock algorithm, which allows the process to run uninterrupted until it makes a special request that tells the kernel that it may switch to another task.

The fourth factor is that Windows uses a hybrid kernel where the kernel itself contains tasks such as the window manager and Inter-Process-Communications manager, where in other OS the kernel is monolithic and the services are external to the kernel.

Despite the fact that the Windows XP platform is faster in Hardware, the results are still indicating that the sWIFI algorithm is on a par with the most advanced encryption algorithms in performance.

V. CONCLUSION AND FUTURE WORK:

A comparison has been conducted for those encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different data types, data transmission through wireless network and finally encryption/decryption speed. Simulation results are given to demonstrate the effectiveness of each algorithm.

The developed sWIFI algorithm is built around a 64 bit encryption / decryption and might be further expanded to cover 128 to reach up to 512 bits and more, and will still be efficient in both Speed and size which could be implemented into a hardware solution. In part of future work, we will conduct comparative study between sWiFi and other encryption algorithms on different platforms and different battery power consumption.

REFERENCES

1. S. Masadeh W.Salameh. End to end keyless self encrypting/decrypting streaming cipher. In: Information Technology & National Security Conference. 2007.

2. A. Nadeem MYJ. A performance comparison of data encryption algorithms. In: First International Conference on Information and Communication Technologies. 2005:84- 89.

3. Hardjono, Security in Wireless LANS and MANS, Artech House Publishers, 2005.

4. N. Ruangchaijatupon and P. Krishnamurthy, \Encryption and power consumption in wireless LANs-N," The Third IEEE Workshop on Wireless LANs, pp. 148-152, Newton, Massachusetts, Sep. 27-28,2001.

5. B. Schneier, The Blowfish Encryption Algorithm, Oct. 25, 2008. (http://www.schneier.com/blow¯sh.html)

6. W. Stallings, Cryptography and Network Security, 4th Ed, pp. 58-309, Prentice Hall, 2005.

7. J. Daemen and V. Rijmen. AES Proposal: Rijndael, National Institute of Standards and Technology, July 2001.

8. J. Daemen and V. Rijmen, The Design of Rijndael, Springer-Verlag, 2002.

9. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M.Stay, D. Wagner, and D. Whiting, \Improved crypt-analysis of Rijndael," Seventh Fast Software Encryption Workshop, pp. 19, Springer-Verlag 2000.

10. K. Naik and D. S. L. Wei, \Software implementation strategies for power-conscious systems," Mobile Networks and Applications, vol. 6, pp. 291-305, 2001.

11. J. Daemen and V. Rijmen, \Rijndael: The advanced encryption standard," Dr. Dobb's Journal, pp. 137-139, Mar. 2001.

12. B. Schneier, J. Kelsey, " Unbalanced feistel networks and block cipher design. In: Proceedings of the Third International Workshop on Fast 12 Software Encryption. London, UK: Springer-Verlag. ISBN 3-540-60865-6; 1996:121-144.

13. B. Schneier. Cryptography, security, and the future 1999;:59doi: http://doi.acm.org/10.1145/318536.318553

14. N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, S. Lucks, T .Kohno. "Helix fast encryption and authentication in a single cryptographic primitive. In: Proc. Fast Software Encryption 2003, volume 2887 of LNCS. Springer-Verlag; 2003:330-346.

15. S. Hirani, Energy Consumption of Encryption Schemes in Wireless Devices Thesis, University of Pittsburgh, Apr. 9, 2003, Retrieved October 1,2008 http://portal.acm.org/citation.cfm?id=383768.

16. A. Nadeem and M. Y. Javed, \A performance comparison of data encryption algorithms," Information and Communication Technologies, ICICT 2005, pp.84-89, 2005.

17. Results of Comparing Tens of Encryption Algorithms Using Different Settings-Crypto++ Benchmark, Retrieved Oct. 1, 2008. (http://www.eskimo.com/weidai/benchmarks.html)

18. W.S.Elkilani, "H.m.Abdul-Kader, "Performance of Encryption Techniques for Real Time Video Streaming, IBIMA Conference, Jan 2009, PP 1846-1850

19. D. Salama, A. Elminaam and etal, "Evaluating The Performance of Symmetric Encryption Algorithms", International Journal of Network Security, Vol.10, No.3, PP.216–222, May 2010.