Increase the Security of Web Service without Effect on its Performance by Using Blowfish Cryptography Algorithm

Shaymaa Mohammed Jawad Kadhim, Manjusha Joshi, Shashank Joshi

Abstract: Day by day network and internet applications are becoming very popular. Sensitive information requires ensured information security and safety measures. Security is the most challenging aspect in the internet and network applications.

Encryption algorithm offers the necessary protection against the data intruders' attacks by converting information from its normal form into an unreadable form.

In the first part of the work described in this paper, we are going to use the Blowfish algorithm and apply it on a web service (employing management system) and in second part we will provide a fair comparison between two web services one without applying on it the algorithm and the other using Blowfish.

The comparison is made on the basis of these three parameters :

Response time, MTBF(mean time between failure) and MTTR(mean time to repair).

Index Terms— blowfish, cryptography, performance, Web Service.

I. INTRODUCTION

Internet and networks are admiring day by day in our life. widespread for using internet , wired and wireless networks , and web services makes the need for protection of information.

Cryptographic algorithms play a major role for data user security. As the complexity of algorithm is high the risk of breaking the original plaintext from that of cipher text is less. Greater complexity means greater security. Encryption is the process of encoding plain text into cipher text (secure data). Decryption is the reverse of the encryption process by which cipher text is converted to plain text, as shown in figure (1) .

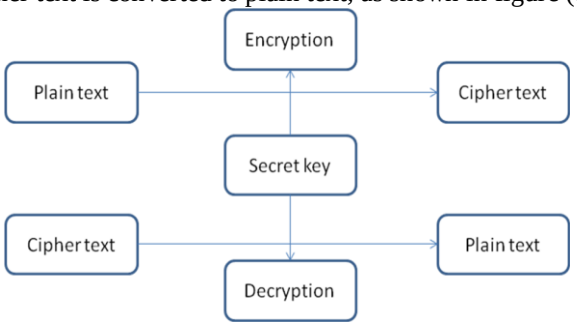

Figure (1) shows the encryption and decryption process by using the same key (symmetric key cryptographic algorithm).

There are many Encryption algorithms which are used for user data security.

They are divided into two cryptographic mechanisms, depending on what keys are used, which are symmetric and asymmetric encryptions. In symmetric encryption only one key is used for both encrypt and decrypt the data. Strength of the symmetric encryption depends upon the size of the key. For the same algorithm, encryption using the longer key is tough to break than one using smaller key. While in Asymmetric encryption two keys are used, one is used to encrypt and other is used to decrypt the data [1]. In the following part we will describe briefly the algorithm that we are going to use it in this paper:

Blowfish is a symmetric encryption algorithm, meaning that it uses the same secret key to both encrypt and decrypt user data. It takes a variable-length key, from 32 bits to 448 bits, making it ideal for securing data. Blowfish is also a block cipher, meaning that it divides a data up into fixed length blocks during encryption and decryption. The block length for blowfish is 64bits [2].

The Blowfish algorithm is Feistel network, iterating a simple encryption function 16 times. The block sixe is 64 bits, and the key can be any length up to 448 bits. Blowfish was designed in 1993 by Bruce schneir as a fast, free alternative to existing encryption algorithm, it is unpatented and license-free, and is available free for all uses.

Blowfish is suitable for applications where the key does not change often, like a communications link or an automatic file encryption. it is significantly faster than most encryption algorithms when implemented on 32-bit microprocessors with large data caches [3].

The algorithm consists of two parts: a key expansion part and a data encryption part. Key expansion converts a key at the most 448 bits into several sub key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistael network. Each round consists of a key dependent permutation, and a key- and data- dependent substitution. In figur (2) the blowfish algorithm.

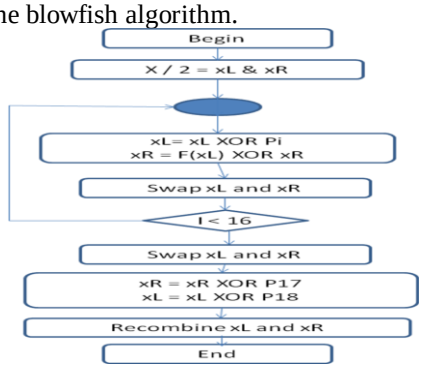

Figure (2) : shows the blowfish algorithm

Manuscript Received November, 2013.

Shaymaa Mohammed Jawad Kadhim, follow up and planning departrment, Ministry of transport, Baghdad, Iraq,

Mrs.manjusha Joshi, research scholar, computer Dept BVU Pune, India.

Dr.Shashank Joshi, Professor , computer engineering Dept BVDUCOE, pune, India.

II. PREVIOUS WORKS

1) The paper [4] provides a fair comparison between three most common symmetric key cryptography algorithms.
2) The paper [5] presents the implementation of Blowfish algorithm.
3) The paper [6] describes a new method to enhance the security of Blowfish algorithm.
4) The paper [7] describes Embedded systems programming.
5) The paper [8] presents of includes a mechanism that implements the Blowfish algorithm with a 64 bit length key with as improved security assurance.

III. PROPOSED ARCHITECTURE

For our work we first use the blowfish algorithm to provide security in a web service that we have already coded (an employing management system) .

In a second step we analyze the performance of the web service by measuring the three parameters response time, MTBF (mean time between failure) and MTTR (mean time to repair).

1) Applying blowfish algorithm on the web service.

Blowfish algorithm uses a large number of subkeys. These keys must be recomputed before any data encryption or decryption. The P-array consists of eighteen 32-bit sub keys: P1,P2,P3……P18.

There are four 32-bit S-boxes with 256 entries each:

S1,0,S1,1……S1,255.
S2,0,S2,1……S2,255.
S3,0,S3,1……S3,255.
S4,0,S4,1……S4,255.

ENCRYPTION

Blowfish has 16 rounds as shown if figure (3) . The input is a 64bit data element ,x. divide x into two 32-bit halves : xL,xR.

Then for I = 1 to 16
xL=xL XOR Pi
xR=F(xL) XOR xR
swap xL and xR

after the sixteenth round , swap xL and xR again to undo the last swap. Then , xR=xR XOR P17 and xL=xL XOR P18.Finally , recombine xL and xR to get the cipher text.

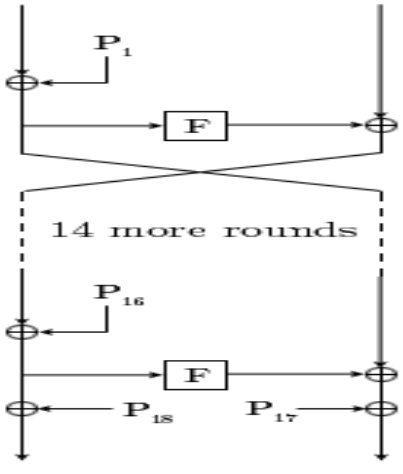

Figure (3) : shows the 16 rounds of blowfish algorithm.

Decryption is exactly the same as encryption, except that P1,P2,…….P18 are used in the reverse order.

Generating the subkeys :

The subkeys are calculated using the Blowfish algorithm:

1) Initialize first the P-array and then the four S-boxes, in order, with a fixed string.
2) XOR P1 with the first 32 bits of the key , XOR P2 with the second 32bits of the key, and so on for all bits of the key (possibly up to P14). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits.
3) Encrypt the all-zero string with the Blowfish algorithm with the modified subkeys described in steps (1) and (2).
4) Replace P1 and P2 with the output of step (3).
5) Encrypt the output of step (3) using the Blowfish algorithm with the modified subkeys.
6) Replace P3 and P4 with the output of step (5).
7) Continue the process, replacing all entries of the P array, and then all four S-boxes in order, with the output of the continuously changing Blowfish algorithm. In total, 521 iterations are required to generate all required subkeys. Applications can store the subkeys rather than execute this derivation process multiple times [9].

This blowfish algorithm is applied in our web service (employing management system). All data are stored in a data base that the web service is connected to. Data are stored in cipher text so no one can understand it, without breaking the algorithm itself.

2) result and analysis

The score of our work is to provide a performance analysis between a web service without any security and the same web service applying security on it by using blowfish algorithm. We used a private web service, which is an employing management system in different operating systems.

On the first operating system, which is windows 7 with 32 bit architecture, we used our web service without applying any security algorithms on it. This means the data will be store in it's database in plain text , so anyone can easily enter it and read it's information and even modify it .

On the second OS, Window7 64bit, we used the same web service (Employing management system), but this time we applied the cryptography algorithm Blowfish to provide security, so all data will stored in cipher text.

The web service is coded in the programming language C#.

The parameters that were used to analyze the performance are

1) Response time.
2) MTBF (mean time between failure)
3) MTTR (mean time to repair)

A. The first set of experiments were conducted by measuring the response time from both web services, the web service with the cryptography algorithm and the web service without the algorithm. The result is shown in figure (4) below : The response time that of the encrypted web service (green bar) is higher than in the non-encrypted web service (blue bar). The results were measured in millisecond though. the difference is small and can almost be neglegted compaired to the gain of

providing a good security to the our web service.

Figure (4) shows response time in encrypted and non-encrypted systems

B. The second set of experiments was conducted measuring the MTBF (mean time between failure) from both web services, the web service using the cryptography algorithm and the web service without the algorithm. The result is displayed in figure (5) below. For the MTBF we notice a similar result as for the response time: The value for the encrypted web service is slightly higher but the difference is only a few milliseconds.

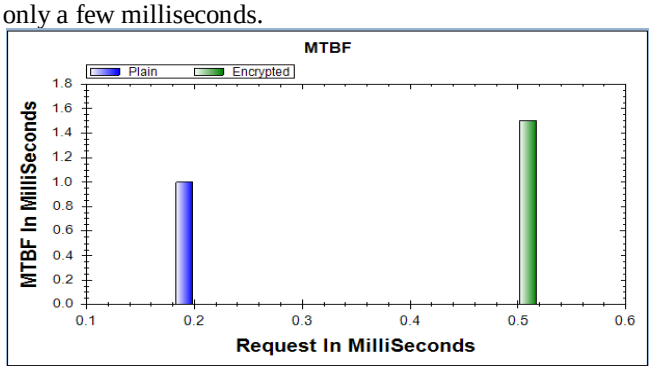

Figure (5) shows mean time between failure in encrypted and non encrypted systems

C. The third set of experiments was conducted by measuring the MTTR (mean time to repair) from both web services, the web service with cryptography algorithm and the web service without the algorithm. The result is shown in figure (6) below. The time required by the encrypted web service is almost the time that required by non-encrypted web .

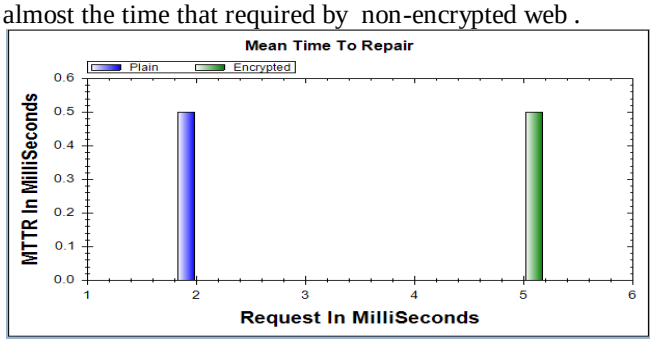

Figure (6) shows mean time to repair in encrypted and non encrypted systems

IV. CONCLUSION

We have created our own web service which is an employing management system. The web service was set up on two windows operating systems, Windows7 with 32 bit architecture and windows 7 64bit. On Windows7 32 bit we used the web service without any security whereas on Windows7 64 bit we used the same web service applying the blowfish cryptographic algorithm as security feature.

In the second part of our work we compared the performance of the web service with and without security. The parameters measured in our work were Response time, MTBF (mean time between failure) and MTTR (mean time to repair). The results displayed above as figures and charts show that a web service that provides good security (Blowfish) can almost have the same performance as services without any security.

FUTURE WORK

Customer satisfaction is the main aim of every system. We should provide better, faster and more secure systems to the customer.

By applying the Blowfish algorithm we could provide a much better security standard on our web service without a crucial effect on it's performance. All aims could be reached at minimal expenses.

Our future work will focus on necessary hardware improvements, thus improving the performance with security at the same time.

REFERENCE

1. Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encrption Standard. "Dr. Dobb's Journal, March 2001,PP.137-139.
2. Bill Gatliff, courtesy of "Embedded systems programming" jul 15 2003 (11:00 AM).
3. http://www.design-reuse.com/articles/5922/encrypting-data-with-the-blowfish-algorithm.html.
4. jawahar Thakur, Nagesh Kumar "DES,AES and Blowfish: Symmetric key cryptography algorithms simulation based performance analysis" .
5. Kevin Allison , Keith Feldman , Ethan Mick " Blowfish".
6. Gurjeevan Singh*, Ashwani Kumar** , K.S.Sandha*** "A Study of New Trends in Blowfish Alogrithm".
7. Bill Gatliff "Embedded Systems Programming".
8. Ch Panchamukesh , Prof.T.Venkat Narayana Rao , A.Vijay Kumar "An Implementation of Blowfish Encryption Algorithm using KERBEROS Authentication Mechansim".

AUTHOR PROFILE

First Author Shaymaa mohammed jawad kadhim received her B.E. degree in computer engineering from Baghdad university ,college of engineering in 2000, doing her the M.E. Computer Engineering at Bharati Vidyapeeth Deemed University Pune. she is currently working in transport ministry in Iraq / Baghdad her research interests include web technologies and security.

Second Author Mrs. Manjusha Joshi received his B.E. degree in Computer from Walchand College of Engineering, Sangli in 1991, the M.E. Degree in Computer Engineering from Bharati Vidyapeeth Deemed University Pune and currently pursuing her Ph. D. in Computer Engineering Department Bharati Vidyapeeth Deemed **University College of Engineering, Pune. Her** research interests include software engineering, SPI, distributed systems.

Third Author Shashank Joshi received his B.E. degree in Electronics and Telecommunication from Govt. College of Engineering, Pune in 1988, the M.E. and Ph. D. Degree in Computer Engineering from Bharati Vidyapeeth Deemed University Pune. He is currently working as the Professor in Computer Engineering Department Bharati Vidyapeeth Deemed University College of Engineering, Pune. His research interests include software engineering. Presently he is engaged in SDLC and secure software development methodologies. He is innovative teacher devoted to Education and Learning for the last 23 yrs.