A SECURE SESSION MANAGEMENT SYSTEM USING MD5 ALGORITHM

M. Nusret SARISAKAL[1] **Dogal ACAR** [2] **Selcuk SEVGEN** [3]

[1,2,3] *Istanbul University, Faculty of Engineering, Department of Computer Engineering,*
34850, Avcilar, Istanbul, Turkey

[1]e-mail: nsarisakal@istanbul.edu.tr [2]e-mail: dogal@istanbul.edu.tr [3]e-mail: sevgens@istanbul.edu.tr

Keywords: *Security, Cryptography, JAVA, MD5 Algorithm*

ABSTRACT

This paper focuses on appearing of data security problem because of becoming widespread of free circulation of electronic data in the internet environment and also explains a session management system that uses MD5 cryptography algorithm, which is most common data security algorithm, for providing the security of an insurance web site. And it explains the MD5 algorithm and its usage.

1. INTRODUCTION

Cryptography covers a wide area since 4000 years ago the Egypt up to world wars and contemporary internet applications. Cryptography has been used specially in military services and diplomacy. In this area cryptography has been used as a tool for protecting national strategies and informations.

In 1960s by becoming widespread of computer and communication systems, in this sectors the need of security in digital environment has appeared. In the beginning of 1970s in IBM, by the studyings of Feistelin, in 1977 the ***DES*** (Data Encryption Standard) came to scene. This algorithm became the most popular cryptography system in the world. Still most of the e-buisness systems are using this algorithm [1].

The most remarkable devolopment in cryptography science was realised by the publications of Diffie and Hellman in 1976 with the name ***New Directions in Cryptography.*** In this study they suggest the ***Public Key*** concept which has made a revolution in cryptography science. And for key changing they devoleped more efficent security methodologoies by using the advantage of the attribute of the seperated logarithm problem which is: can not be controlled easily.

In 1976 Rivest Sahmir and Adleman realised the first application of public-key cryptosystem and signing system. Today this application is mentioning as ***RSA.*** The data must be compressed with a secure method before it was encrypted by the public-key cryptosystems like RSA with private-key.

MD5 algorithm takes an input with any length and it generates an output with 128 bit length which is named as **finger-print** or **message digest**.

MD5 algorithm was designed for digital signing applications. MD5 algorithm was designed to run fast in 32 bit computers for long files. MD5 algorithm does not need wide tables, which are used in algorithm, and it can be coded shortly.

MD5 algorithm is the improved version of MD4 algorithm. Although MD5 algorithm is slower than MD4, it has a simple design in comparasion with MD4. MD4 was designed as a very fast algorithm. As it was began to use before the critical evaluations had been made, the security risk of algorithm was increased.

MD5 algorithm is providing higher security, as it is running slowly in comparasion with MD4. While the MD5 algorithm was designed, suggestions of many participants were taken and many arrangements and extra optimizations were made [2,3].

DEFINITIONS

In algorithm:

- ✓ ***Word*** is a 32 bit and ***byte*** is a 8 bit quantity
- ✓ The + symbol representes the additions of ***word***s (example modulo- 2^32 addition)
- ✓ **X << s** represents the circular left shift of X by **s** positions ($0 \leq s \leq 31$)
- ✓ Not (X) represents bitwise complement of X
- ✓ X ᵛ Y represents bitwise **or** of X and Y
- ✓ X ⊕ Y represents bitwise **xor** of X and Y
- ✓ XY represents bitwise **and** of X and Y

2. MD5 ALGORITHM

Suppose that we have a expression with the length b bit. We will find message digest of this expression. b is a positive integer. b can be zero. And it must not be a multiple of 8. It can be at any length. Representation of the bits of message is as follow:

$\mathbf{m_0\ m_1\ ...\ m_{(b-1)}}$

As we apply the five steps below we will get the message digest.

2.1 ADDING THE PADDING BITS

The length of message bit is padded until it gives 448 for modulo 512. By this operation we provide a message bit that is 64 bit less then the multiple of 512. Even if the padding operation gives 448 for modulo 512, the padding operation is made. The padding operation is made as follows: first a 1 bit is added then 0 is added until the message bit gives 448 for modulo 512. As a result minimum 1 bit or maximum 512 bit is being added.

2.2 ADDING THE LENGTH

The 64 bit of message bit before the padding was made is added to expression we got at the step 2.1. If the value of b is greater than 2^{64} then we use low-order 64 bits. (These bits are added as two 32 bit words which the first one is low-order). At the end of this step our message becomes the multiple of 512.

Let's $m_{[0...b-1]}$ represent the words of message and b be the multiple of 16.

2.3 MD BUFFER

For computing the message digest a buffer with four words is used. All the A,B,C and D are 32 bit registers. These registers are initialized to the following values in hexadecimal, low-order bytes first:

word A: 01 23 45 67
word B: 89 ab cd ef
word C: fe dc ba 98
word D: 76 54 32 10

2.4 PROCESSING THE MESSAGE AS BLOCKS WITH 16 WORDS

We will take the three 32 bit words as input and produce a 32 bit word with using four function as follows:

$$F(X,Y,Z) = XY \vee \text{not}(X)\, Z$$
$$G(X,Y,Z) = XZ \vee Y\, \text{not}(Z)$$
$$H(X,Y,Z) = X \oplus Y \oplus Z$$
$$I(X,Y,Z) = Y \oplus (X \vee \text{not}(Z))$$

F acts as a conditional function for every bit case such as if X is true and Y not then Z is result. If the bits of X, Y, and Z are independent and unbiased, the each bit of F (X,Y,Z) will be independent and unbiased.The inputs of H function are **xor** and **parity** function.

The functions G, H, and I are similar to the function F, in that they act in **bitwise parallel** to produce their output from the bits of X, Y, and Z, in such a manner that if the corresponding bits of X, Y and Z are independent and unbiased, then each bit of G (X,Y,Z), H (X,Y,Z) and I (X,Y,Z) will be independent and unbiased. The function H is the bit-wise **xor** or **parity** function of its inputs.

In this step the 64 bit T [1...64] table, which is constructed from sine function, is used. (Table 1) T [i] represents the i'th member of table the value of this variable is the integer part of 4294967296 times of abs(sin (i)). (i is radian) [2,3]. All functions uses the T table as fallows:

Y_q is the current 512 bit block being processed.
S : step

$$S_F\,(ABCD,Y_q,T[1..16]) \rightarrow S_G\,(ABCD,Y_q,T[17..32]) \rightarrow S_H\,(ABCD,Y_q,T[33..48]) \rightarrow S_I\,(ABCD,Y_q,T[49..64])$$

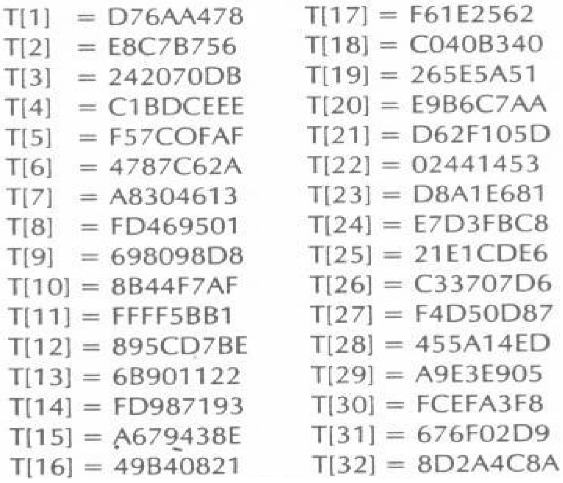
Each step takes A B C D one by one and transmits the results to other function at last S_I gives results and these are added with A B C D. These steps obtain the ***128 bit message digest***. These steps work as fallow:

$$a \leftarrow b + CLS_s\,(a + g(b,c,d) + X[k] + T[I])$$

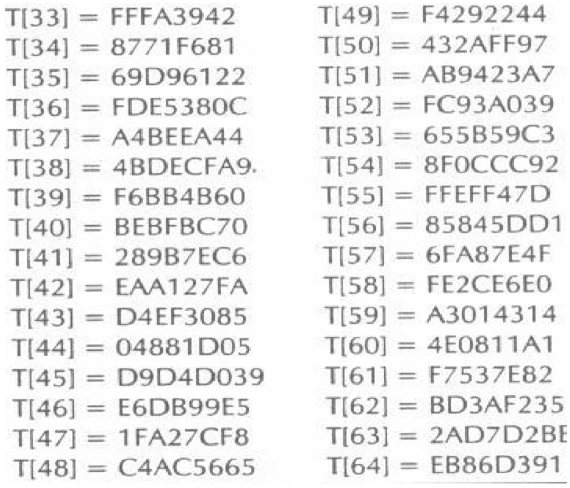
where

a,b,c,d : four words of the buffer
g : one of the pr,mitive functions
CLS$_s$: circular left shift of the 32-bit argument by s bits
X[k] : M[qx16+k]=the k'th 32 bit word in the q'th 512 bit blockof the message
T[I] : the i'th 32 bit word in t table.
+ : addition modula 2^{32}

Table 1. Constructed from the sine function.



T[1] = D76AA478	T[17] = F61E2562
T[2] = E8C7B756	T[18] = C040B340
T[3] = 242070DB	T[19] = 265E5A51
T[4] = C1BDCEEE	T[20] = E9B6C7AA
T[5] = F57COFAF	T[21] = D62F105D
T[6] = 4787C62A	T[22] = 02441453
T[7] = A8304613	T[23] = D8A1E681
T[8] = FD469501	T[24] = E7D3FBC8
T[9] = 698098D8	T[25] = 21E1CDE6
T[10] = 8B44F7AF	T[26] = C33707D6
T[11] = FFFF5BB1	T[27] = F4D50D87
T[12] = 895CD7BE	T[28] = 455A14ED
T[13] = 6B901122	T[29] = A9E3E905
T[14] = FD987193	T[30] = FCEFA3F8
T[15] = A679438E	T[31] = 676F02D9
T[16] = 49B40821	T[32] = 8D2A4C8A



T[33] =	FFFA3942	T[49] =	F4292244
T[34] =	8771F681	T[50] =	432AFF97
T[35] =	69D96122	T[51] =	AB9423A7
T[36] =	FDE5380C	T[52] =	FC93A039
T[37] =	A4BEEA44	T[53] =	655B59C3
T[38] =	4BDECFA9.	T[54] =	8F0CCC92
T[39] =	F6BB4B60	T[55] =	FFEFF47D
T[40] =	BEBFBC70	T[56] =	85845DD1
T[41] =	289B7EC6	T[57] =	6FA87E4F
T[42] =	EAA127FA	T[58] =	FE2CE6E0
T[43] =	D4EF3085	T[59] =	A3014314
T[44] =	04881D05	T[60] =	4E0811A1
T[45] =	D9D4D039	T[61] =	F7537E82
T[46] =	E6DB99E5	T[62] =	BD3AF235
T[47] =	1FA27CF8	T[63] =	2AD7D2BE
T[48] =	C4AC5665	T[64] =	EB86D391

2.5 OUTPUT

The message digest produced as output is A, B, C, D. That is, we begin with the low-order byte of A, and end with the high-order byte of D [2,3].

3. JAVA

Java applets are programs that can be showed on browsers which has java support. A java applet is like an image but it differs with being dynamic and interactive. To connect a java applet to a web page first a java applet is written and complied and then this file is referred in this web page. When web page is loaded on web browser the java interpreter runs the applet and browser shows the result on screen. The first web browser that supports java is HotJava. Today most of web browsers support Java [4].

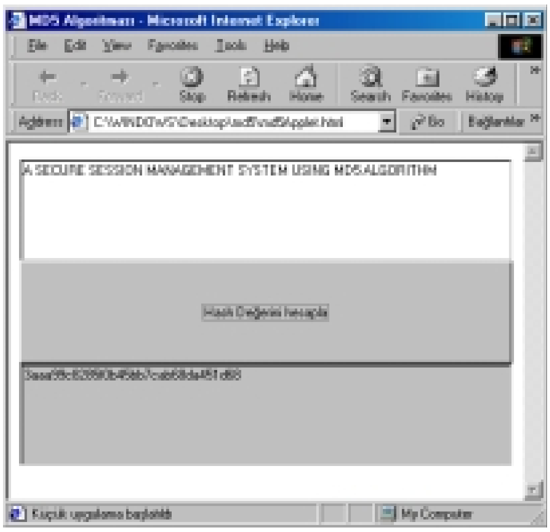
Java is independent from platform. In spite of it was not developed as an internet programming language it became an internet language at a short time. Some of the optimistic specialists think that java will make possible to surf on internet without using PCs. And some of them think that PCs will be used but the dependence to Microsoft will hide. The great advantage of Java is to be independent of platform in other words to run on all of the digital tools. These tools are computers, machines, cars, electricity meters, thief alarms etc. [5].

We coded the MD5 algorithm as java applet. In this applet hash function of any input is computed with MD5 algorithm.

Example:
Input:
A SECURE SESSION MANAGEMENT SYSTEM USING MD5 ALGORITHM

Hash Function as Output:
3aaa99c8285f0b45bb7cab68da451d68

In figure 1, the running of java applet on web browser and computing of hash function is seen.



A SECURE SESSION MANAGEMENT SYSTEM USING MD5ALGORITHM

3aaa99c8285f0b45bb7cab68da451d68

Figure 1.

4. A SESSION MANAGEMENT SYSTEM USING MD5 ALGORITHM

We developed a session management system using MD5 algorithm with PHP script. In php script a MD5 function is running. The sites on the web must know the users' information for giving private service to user. The interaction between user and web site is occuring on a process which is mentioned as **session**. When a session starts, program gives a session id to user. If user wants anything from site, program (on browser) sends his/her session id, in this way program recognizes the user and gives private service to user or continues giving service where user has been staying. In other words if server assigns an id to a user, any request that comes with this id is accepted as it comes from this user.

Anyone, that can guess this id, can make operations in the name of this user. To avoid this objectionable situation session ids must not able to be estimated easily and between session ids there must not be relations that can be a understood easily.

By generating session ids with MD5 algorithm we will avoid hackers who will estimate the ids by using the relations between them.

In figure 2, the screen, when user comes to main page of site, is seen. The page index.php3 is taking an empty parameter with the name sesID. This shows that the visitor has not loged on to site yet. When visitor logs on to system a sesID is produced using visitors username and password. If visitor wants to do any operation, acceptability of sesID parameter will be checked. If parameter is not acceptable user will direct to new logon page.

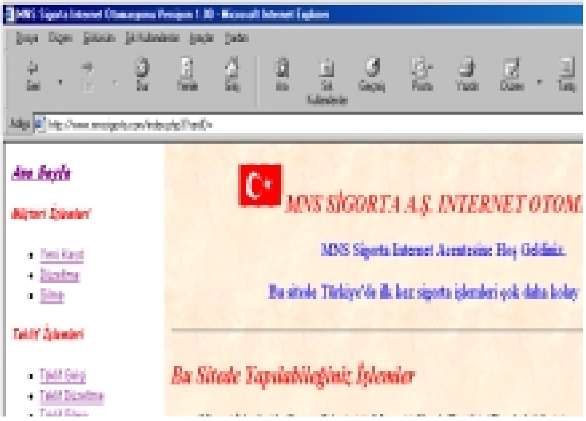

Ana Sayfa
Müşteri İşlemleri
Yeni Kayıt
Düzeltme
Silme
Teklif İşlemleri
Teklif Girişi
Teklif Düzeltme
MNS SİGORTA A.Ş. INTERNET OTOM
MNS Sigorta İnternet Acentesine Hoş Geldiniz.
Bu sitede Türkiye'de ilk kez sigorta işlemleri çok daha kolay
Bu Sitede Yapılabileceğiniz İşlemler

Figure 2.

In figure 3 the name, surname, firm, etc. and username and password of member is taken. Username and password will be used to define the identity of user. If the username and password is acceptable according to user identity in system data, user will logon to system and will take a session id with the name sesID. SesID will be produced with MD5 algorithm for providing security. User will make operations on web site with using this sesID and web site will direct and recognize user according to his/her sesID that was assigned at the begining.

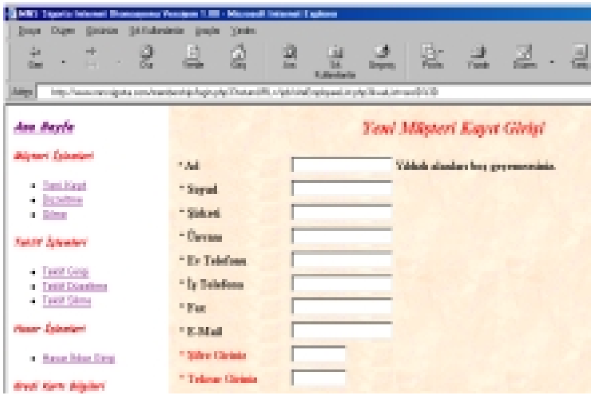


Figure 3

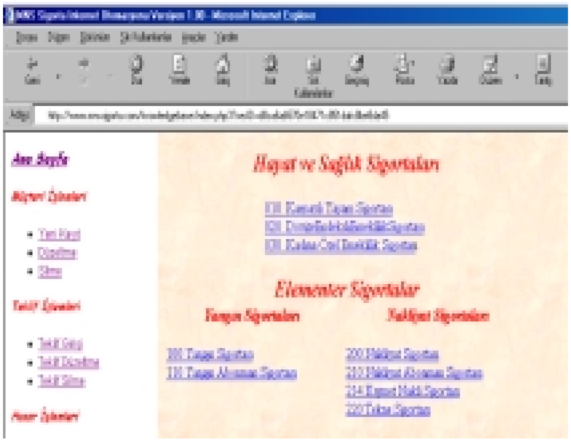


Figure 4

In figure 4, the sesID variable that was assigned to user for this session is seen. In this way we can recognize the user that is making operations on site and avoid the estimating of id by anybody.

5. CONCLUSION AND FURTHER PLANS

In this application MD5 algorithm is applied as a function in PHP script. In the future we will use the MD5 algorithm in a Java applet, as we mentioned before, in different applications.

The java applets on internet runs on web browser so using of system resources on server decrease. As the burden on server decreases the performance of web site increases.

In this application we used MD5 algorithm for providing the security of an insurance web site. In the future we will use different and more robust algorithms and try to make the performance test of these algorithms.

REFERENCES

1. Stinson D. R., Cryptography Theory and Practice, CRC Press, 1995, Florida

2. Ronald L. Rivest, The MD5 Message-Digest Algorithm RFC1321, April 1992.

3. William S., Network And Internetwork Security Principles And Practice, Prentice-Hall, Inc. 1995, New Jersey.

4. http://www.sun.com/java

5. Chorafas D. N, Java – A Contarariann View 157-179, Visual Programming Technology, McGraw-Hill, New York, ISBN 0-07-011685-7, 1997.