CONFERENCE PROCEEDINGS

CRYPTOLOGY 2012

Cryptology 2012

Proceedings of the 3rd International Conference on Cryptology and Computer Security

4th June - 6th June 2012
Langkawi, Kedah, Malaysia

Editors :
Hailiza Kamarulhaili
Yahya Abu Hasan
Azman Samsudin
Muhammad Rezal Kamel Ariffin
Mohamad Afendee Mohamed
Mohd Rushdan Md Said
Goi Bok Min
Heng Swee Huay
Rabiah Ahmad
Nor Azman Abu
Moesfa Soeheila Mohamad

Jointly Organized By:

USM UNIVERSITI SAINS MALAYSIA

MSCR

UPM UNIVERSITI PUTRA MALAYSIA

CyberSecurity MALAYSIA

Proceedings of the 3rd International Conference on Cryptology and Computer Security

4th June – 6th June 2012
Langkawi, Kedah, Malaysia

Jointly Organized By:

UNIVERSITI SAINS MALAYSIA

MSCR

UPM UNIVERSITI PUTRA MALAYSIA

CyberSecurity MALAYSIA

UNIVERSITI SAINS MALAYSIA

School of Mathematical Sciences
11800 Universiti Sains Malaysia,
Pulau Pinang, MALAYSIA
Website : http://math.usm.my
E-mail : dean_mat@usm.my

ISBN 978-967-394-084-4
9 789673 940844

TABLE OF CONTENTS

PROTECTION OF TEXTS USING SHA1 AND BASE64

[1]Mohammad A. Ahmad, [1]Imad Alshaikhli and [2]Hanady Mohammad Ahmad
[1]*Department of Computer Science, International Islamic University of Malaysia, 53100 Jalan Gombak Kuala Lumpur, Malaysia*
[2]*Department of Computer, Basic Education College, Public Authority of Applied Education and Training, 34053 Alshamiya, Kuwait,*
malahmads@yahoo.com, imadyaseen39@yahoo.com, hanadym.1359@windowslive.com

Abstract:

Protection of information is a prerequisite demand in the world of computers today. Protection of information can be accomplished in different methods. The main objective of the use of the protection of information is to protect data and information in order to achieve privacy. This paper discusses two methods of protection of information, an encryption method called Base64, which is a set of encoding schemes that convert the same binary data to the form of a series of ASCII code. Also, The SHA1 hash function is used to hash the encrypted file performed by Base64. As an example of an ASCII code, Arabic letters are used to represent the texts. So using the two protection methods together will increase the security level for protecting the data.

*Keywords: **Encryption, Hash, Base64, SHA1***

Introduction

Protection of information is a prerequisite demand in the world of computers today. Protection of information can be accomplished in different methods. The main objective of the use of the protection of information is to protect data and information in order to achieve privacy. The encryption process combines mathematics and computer science. Cryptography consists of a set of algorithms and techniques to convert the data into another form so that the contents are unreadable and unexplainable to anyone who does not have the authority to read or write on these data. The main objective of the use of encryption algorithms is to protect data and information in order to achieve privacy. The protection mechanism choices are applied based on the data sensitivity. For example, the data bank "ex, clients accounts" needs to be protected by latest security and protection mechanisms. In fact, with the available tools for intrusion in the internet today, computer intruders can hack to secure systems easily. Consequently, combining more than one protection mechanisms is so crucial to achieve the highest level security against intruders. (Imad F. Alshaikhli, 2011)

There are several functions for the protection processes to protect the information and files from intrusion. It is possible to employ encryption in various fields. In this paper, an encryption method is presented to protect the texts. One way to protect the texts from changes is by encryption. This paper will explain the method of encryption using Base64. The first step of the encryption method using Base64 is to convert text to unreadable text and create the ASCII for each character and convert it to a binary number. Then we convert the binary number to a decimal number and find the character that corresponds to the decimal number, and in so doing, the text will be rendered incomprehensible by the encryption process. Also, Secure Hash Algorithm "SHA1" is used as protection mechanism associated with Base64 encryption method. SHA1 is an algorithm that is used to verify data integrity through the creation of a 160-bit from data input (which may be a message of any length); the product is claimed to be as unique to that specific data as a fingerprint is to the specific individual. (Rivest, 1992)

SHA1and Base64 are used together to increase the security level of the data that needs protection. The details are explained in this paper.

Proposed System

Computer security is a major challenge for all computer users, and use of encryption protects data and information from modification. Many businessmen, professionals, and home users employ encryption to protect their data and to maintain strict confidentiality. The system proposed in this paper is to encrypt the texts through the use of the Visual Basic program, as well as the use of encryption method of Base64 and hash function SHA1.

The particular choices for the 64 characters required for the base varies between implementations. The general rule is to choose a set of 64 characters that is both part of a subset common to most encodings, and also printable. This combination leaves the data unlikely to be modified in transit through information systems, such as email, that were traditionally not 8-bit clean.[1] For example, MIME's base64 implementation uses A–Z, a–z, and 0–9 for the first 62 values. Other variations, usually derived from Base64, share this property but differ in the symbols chosen for the last two values; an example is UTF-7.

SHA1 Hash

Definition of SHA1Hash

SHA1 (Secure Hash Algorithm 1) is message-digest algorithm, which takes an input message of any length < 2^64 bits and produces a 160-bit output as the message digest. Based on the SHA1 RFC document, the SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. The original specification of the algorithm was published in 1993 as the Secure Hash Standard, FIPS PUB 180, by US government standards agency NIST (National Institute of Standards and Technology). This version is now often referred to as "SHA0". SHA-0 was withdrawn by the NSA shortly after publication and was superseded by the revised version, published in 1995 in FIPS PUB 180-1 and commonly referred to as "SHA1". (D. Eastlake Septemper 2001)

The technique of Hash SHA1

The SHA-1 is called secure because it is computationally infeasible to find a message which corresponds to a given message digest, or to find two different messages which produce the same message digest. Any change to a message in transit will, with very high probability, result in a different message digest, and the signature will fail to verify. (D. Eastlake Septemper 2001)

Definitions of Bit Strings and Integers

The following terminology related to bit strings and integers will be used:
a. A hex digit is an element of the set {0, 1, ... , 9, A, ... , F}. A hex digit is the representation of a 4-bit string. Examples: 7= 0111, A = 1010.

b. A word equals a 32-bit string which may be represented as a sequence of 8 hex digits. To convert a word to 8 hex digits each 4-bit string is converted to its hex equivalent as described in (a) above. Example:1010 0001 0000 0011 1111 1110 0010 0011 = A103FE23.

c. An integer between 0 and 2^32 - 1 inclusive may be represented as a word. The least significant four bits of the integer arerepresented by the right-most hex digit of the word representation.
Example: the integer 291 = 2^8+2^5+2^1+2^0 =256+32+2+1 is represented by the hex word, 00000123. If z is an integer, 0 <= z < 2^64, then z = (2^32)x + y where 0 <= x < 2^32 and 0 <= y < 2^32. Since x and y can be represented as words X and Y, respectively, z can be represented as the pair of words (X,Y). d. block = 512-bit string. A block (e.g., B) may be represented as a sequence of 16 words. (D. Eastlake Septemper 2001)

Operations on WordThe following logical operators will be applied to words:

a. Bitwise logical word operations
X AND Y = bitwise logical "and" of X and Y.
X OR Y = bitwise logical "inclusive-or" of X and Y.
X XOR Y = bitwise logical "exclusive-or" of X and Y.
NOT X = bitwise logical "complement" of X.

Example:

```
      01101100101110011101001001111
XOR   01100101110000010110100110110111
      -------------------------------
  =   00001001011110001011101111001100
```

b. The operation X + Y is defined as follows: words X and Y represent integers x and y, where 0 <= x < 2^32 and 0 <= y < 2^32. For positive integer's n and m, let n mod m be the remainder upon dividing n by m. Compute z = (x + y) mod 2^32. Then 0 <= z < 2^32. Convert z to a word, Z, and define Z = X +Y.

c. The circular left shift operation S^n(X), where X is a word and n is an integer with 0 <= n < 32, is defined by S^n(X) = (X << n) OR (X >> 32-n).
In the above, X << n is obtained as follows: discard the left-most n bits of X and then pad the result with n zeroes on the right (the result will still be 32 bits). X >> n is obtained by discarding the right-most n bits of X and then padding the result with n zeroes on the left. Thus S^n(X) is equivalent to a circular shift of X by n positions to the left. (D. Eastlake September 2001)

Message Padding

SHA-1 is used to compute a message digest for a message or data file that is provided as input. The message or data file should be considered to be a bit string. The length of the message is the number of bits in the message (the empty message has length 0). If the number of bits in a message is a multiple of 8, for compactness. we can represent the message in hex. The purpose of message padding is to make the total length of a padded message a multiple of 512. SHA-1 sequentially processes blocks of 512 bits when computing the message digest. The following specifies how this padding shall be performed. As a summary, a "1" followed by m "0"s followed by a 64-bit integer are appended to the end of the message to produce a padded message of length 512 * n. The 64-bit integer is the length of the original message. The padded message is then processed by the SHA-1 as n 512-bit blocks. (D. Eastlake Septemper 2001)

Suppose a message has length l < 2^64. Before it is input to the SHA-1, the message is padded on the right as follows:

a. "1" is appended. Example: if the original message is "01010000", this is padded to "010100001".

b. "0"s are appended. The number of "0"s will depend on the original length of the message. The last 64 bits of the last 512-bit block are reserved for the length l of the original message.

Example: Suppose the original message is the bit string

01100001 01100010 01100011 01100100 01100101.

After step (a) this gives

01100001 01100010 01100011 01100100 01100101 1.

Since l = 40, the number of bits in the above is 41 and 407 "0"s are appended, making the total now 448. This gives (in hex)

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000.

c. Obtain the 2-word representation of l, the number of bits in the original message. If l < 2^32 then the first word is all zeroes. Append these two words to the padded message.

Example: Suppose the original message is as in (b). Then l = 40 (note that l is computed before any padding). The two-word representation of 40 is hex 00000000 00000028. Hence the final padded message is hex

61626364 65800000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000000

00000000 00000000 00000000 00000028.

The padded message will contain 16 * n words for some n > 0. The padded message is regarded as a sequence of n blocks M(1) , M(2), first characters (or bits) of the message. (D. Eastlake Septemper 2001)

The technique of Base64

The Base64 method is used to protect the text and files from changes and that is discussed in this paper (Baccala, 1997). The Base64 method involves finding all the ASCII characters, converting them to binary numbers, and then dividing the binary number for the text to 6 bits and converting them to their corresponding values in Base64.

Base64 Mechanism

To encrypt this line using Base64:
- الحمدلله رب العالمين

1. First find the ASCII code for each character.

Letter	ASCII
ا	199
ل	225
ح	205

2. Second, convert the ASCII number of the characters to a binary number.

Letter	ASCII	Binary
ا	199	11000111
ل	225	11100001
ح	205	11001101

3. Third, divide the Binary number to parts and identify a number of bits so that the total is less than or equal to 64 bits. In this example, the Binary number divided to 6-bit.

Letter	ASCII	Binary	Divided binary
ا	199	11000111	11000111
ل	225	11100001	11100001
ح	205	11001101	11001101

4. Fourth, convert parts of the binary number, which has been divided into a decimal number.

Letter	Divided binary	Index
ا	110001	49
ل	111110	62
ح	000111	7
م	001101	13

5. Next, find the character (Char) that corresponds to the number (Value) in the Index Table below.

Index Table

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Letter	Divided binary	Index	Base64-encoded
ا	110001	49	x
ل	111110	62	+
ح	000111	7	H
م	001101	13	N

6. Then perform the encryption for the letters.

Letter	Base64-encoded
ا	x
ل	+
ح	H
م	N

x+HN ← الح

The Steps of Encrypting the Text

Letter	Ascii	Binary	Divided binary	Index	Base64-encoded
ا	199	11000111	110001	49	x
ل	225	11100001	111110	62	+
ح	205	11001101	000111	7	H
م	227	11100011	001101	13	N
د	207	11001111	111000	56	4
ل	225	11100001	111100	60	8
ل	225	11100001	111111	63	/
ه	229	11100101	100001	33	h
" "	32	00100000	111000	56	4
ر	209	11010001	011110	30	e
ب	200	11001000	010100	20	U
" "	32	00100000	100000	32	g

ا	199	11000111	110100	52	0
ل	225	11100001	011100	28	C
ع	218	11011010	100000	32	G
ا	199	11000111	100000	32	G
ل	225	11100001	110001	49	X
م	227	11100011	111110	62	+
ي	237	11101101	000111	7	H
ن	228	11100100	011010	26	A
			110001	49	X
			111110	62	+
			000111	7	H
			100011	35	J
			111011	59	7
			011110	30	E
			010000	16	Q

Conclusion and Future Work

This paper presented two methods of protection, Base64 encryption and Secure Hash Algorithm 1 function to protect the text from being changed. The most important points raised by the paper include:

1. Use of a Base64 encryption method to protect of the texts from modification. This relies on finding the ASCII for each character, converting them to binary numbers, then dividing them into a number of bits and converting them to their corresponding values in Base64.
2. Use of the Visual Basic program for the application program.
3. Use of SHA1 hash function for more security so that each file has its own hash number. When any change occurs in the files, it will change the original hash number and the user will know the file is compromised.

In the future, the protection mechanisms algorithms will be developed. The developed algorithms will be applied and used to protect the electronic Holy Quran from being tampered, changed or modified. More precisely, this paper is the first stage of other series of papers that will lead to a complete project of protecting the different formats of the electronic Holy Quran.

References:

Binark, I., Eren, H., & İhsanoğlu, E. (1986). World bibliography of translations of the meanings of the Holy Qur'an: printed translations, 1515-1980 (Vol. 1): Research Centre for Islamic History, Art, and Culture.

Blaze, M., & Keromytis, A. D. (2000). DSA and RSA key and signature encoding for the KeyNote trust management system.

D. Eastlake, P. J. (Septemper 2001). "Secure Hash Function 1." Network Working Group 10 pages. definition SHA1. (2011, march 12). retrieved from http://searchsecurity.techtarget.com/definition/SHA1 Den Boer, B., & Bosselaers, A. (1994). Collisions for the compression function of MD5.

Imad F. Alshaikhli, M. A. A. (2011). Security Threats of Finger Print Biometric in Network System Environment. [Journal]. Advanced Computer Science and Technology Research, 1(1), 15.

Josefsson, S. (2006). The base16, base32, and base64 data encodings.

Klima, V. (2006). Tunnels in hash functions: SHA1 collisions within a minute.

Morin, R. C. (2001). How to base64.

Quran, H., & Ahmad-UK, F. (1996). Al Islam. The Review of Religions.

Rivest, R. (1992). The SHA1 message-digest algorithm.

Touch, J. D. (1995). Performance analysis of SHA1. ACM SIGCOMM Computer Communication Review, 25(4), 77-86.

Tuszynski, J. (2008). caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc. R package version, 1.

Wang, X., & Yu, H. (2005). How to break SHA1 and other hash functions. Advances in Cryptology–EUROCRYPT 2005, 561-561.

http://quran.alahmad.net