IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661, ISBN: 2278-8727Volume 6, Issue 6 (Nov. - Dec. 2012), PP 49-51
www.iosrjournals.org

Count based Secured Hash Algorithm.

[1]Mayank Swarnkar, [2]Dr. Shekhar Verma
[1,2]Indian Institute of Information Technology-Allahabad

Abstract: *This paper introduces a new approach to the Secured hash algorithm. The proposed algorithms takes the variable length input bit sequence and generate the fix length output as according to the algorithm. There are many constant values, used functions, pre-processing and hashing in SHA[1][2][3][4]. Here introducing new technique in which shifting of bits in pre-processing are not fixed but depends on the counts of bits in input sequence.*

Keywords: *Block Cipher, Hash Function, Linked List, Secured Hash Algorithm.*

I. Introduction:

Security is an important aspect in the data privacy[6]. One of the weapon against security attack is Secured Hash Algorithm [3][4]. This Algorithm is widely used in Digital signatures[7], authentications[1], password protections, protocol security[HAR] etc. Secure hash algorithm takes variable size input and results in fixed size output blocks.[1][4]

A series of secured hash algorithm has been published by NIST i.e. National Institute of Standard and Technology [2][3][4][5][6]. Initially Secured Hash Algorithm version 0 (SHA-0) has been given as a security aspect. Next comes the next version of Secured Hash Algorithm i.e. SHA-1 which works on 256 bits. Then comes SHA-2 which includes SHA 256, SHA 384, SHA 512[1][5].The output so obtained is known as message digest or simply digest.

These Algorithms are iterative algorithms hence they are enable the determination of a message’s integrity. It means any change to the message will, with a very high probability, result in a different message digest. This property is useful in the generation and verification of digital signatures[7] and message authentication codes[HAR], and in the generation of random numbers (bits).

Secured Hash Algorithm completes in two steps which are pre-processing and hashing[5]. Pre-processing includes message padding, message parsing and setting initial values for algorithm. These algorithms enable the determination of a message’s integrity: any change to the message will, with a very high probability, results in a different message digest. This property is useful in the generation and verification of digital signatures and message authentication codes, and in the generation of random numbers (bits).

II. Count Based Secured Hash Algorithm:

This section describes the new function used in secured hash algorithm. In previous work when SHA functions were used , fixed rotation is applied to the input message whereas here the numbers of bits are counted and on the basis of count result the rotation is performed.

III. Parameter Using :

a, b, c, …, h Working variables that are the w-bit words used in the computation of the hash values, $H^{(i)}$.

$_{H}(i)$ -The i^{th} hash value. $H^{(0)}$ is the initial hash value; $H^{(N)}$ is the final hash value and is used to determine the message digest.

H_j The j^{th} word of the i^{th} hash value, where H^{i} is the left-most word of hash value i.

Kt- n Constant value to be used for iteration t of the hash computation.

k- Number of zeroes appended to a message during the padding step.

L -Length of the message

M, in bits -m Number of bits in a message block,

$M^{(i)}$. M Message to be hashed.

$_{M}(i)$ -Message block i, with a size of m bits.

M(i,j) -The j^{th} word of the i^{th} message block, where M^{i}_{0} is the left-most word of message block i.

n- Number of bits to be rotated or shifted when a word is operated upon.

N- Number of blocks in the padded message.
T- Temporary w-bit word used in the hash computation.
w- Number of bits in a word.

Wt - The tth w-bit word of the message schedule.

IV. Counting Algorithm:

```
Input - $M^{(i)}$ as Linked List of length i; // Dynamic data Structure for different length input.
Initialize Count = 0
While(list->next == NULL) // traversing till list ends
If( list->data == 1) // counts the number of 1's in the message.
then
Count = Count++ ; // increment count till message ends.
End loop
$\beta = ROTR^{count}(x)$; // count variable
```

Above Algorithm is designed to improve the security in the Hash algorithms used. Since SHA algorithm is known to all i.e. adversary knows the algorithm. But if the algorithm is known but still output depends on the input message and also depends on the number of 1's in the message then security improves to the greater extent.

In this algorithm the input bits are taken as the data field in the linked list. When all the data is inserted in the linked list, It becomes complete. Since input message is of variable length hence dynamic data structure Linked list is used. A count variable is used which acts as a counter and goes on incrementing when 1's are encountered during scanning of the linked list.

A new count variable β is introduced which will be the resultant when input message sequence x is rotate to the right by the number of 1's in the Message sequence x. Hence the output always depends on the number of 1's counted. Security improves as it does not depends only on the prior known Algorithm but also depends on the input message bits.the digest also depends on the number of 1's encountered. Hence it is improved version of security through SHA.

V. Modified Sha-256 Function

SHA 256 uses six logical functions, where each function works on 32-bits, which are represented as x, y, and z. The result of each function is a new 32-bit word.

$Ch(x, y, z) = (x \wedge y) \oplus (x \wedge z)$

$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$

$\sum_0^{256}(x) = ROTR^{2}(\square) \oplus ROTR^{13}(\square) \oplus ROTR^{22}(\square)$

$\sum_1^{256}(x) = ROTR^{6}(\square) \oplus ROTR^{11}(\square) \oplus ROTR^{25}(\square)$

$\sigma_0^{256}(x) = ROTR^{7}(\square) \oplus ROTR^{18}(\square) \oplus SHR^{3}(\square)$

$\sigma_1^{256}(x) = ROTR^{17}(\square) \oplus ROTR^{19}(\square) \oplus SHR^{10}(\square)$

VI. Modified SHA-384 And SHA-512 Function

SHA-384 and SHA-512 each use six logical functions, where each function operates on 64-bits, which are represented as x, y, and z. The result of each function is a new 64-bit word.

$Ch(x, y, z) = (x \wedge y) \oplus (x \wedge z)$

$Maj(x, y, z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$

$\sum_0^{512}(x) = ROTR^{28}(\square) \oplus ROTR^{34}(\square) \oplus ROTR^{39}(\square)$

$\sum_1^{512}(x) = ROTR^{14}(\square) \oplus ROTR^{18}(\square) \oplus ROTR^{41}(\square)$

$\sigma_0^{512}(x) = ROTR^{1}(\square) \oplus ROTR^{8}(\square) \oplus SHR^{7}(\square)$

$\sigma_1^{512}(x) = ROTR^{19}(\square) \oplus ROTR^{61}(\square) \oplus SHR^{6}(\square)$

Rest all remains the same as the SHA Algorithms [2][3][4][5][6]. Same steps of preprocessing and hashing are implemented on the Count based security hash algorithm. Same constants are used in all SHA 256, SHA 384 and SHA 512[2][3][4][5][6].

VII. Conclusion:

Count based Secure Hash Algorithm enhances the security in the general SHA[5]. In this paper it is proposed that security increases if digest depends on the content of the input message. Count based algorithm works with higher time complexity as compared to the general Secured hash algorithm but provides better degree of security in the world of cryptography. SHA normally used in digital signatures and in authentication purpose. Count based secured hash algorithm can be used with triggered data with usable counts of 1's in inputs to improve the security. Pre-processing and hashing steps remains the same as in the SHA with only differences in the algorithm as mentioned earlier. Pre-processing step changes in this paper as new parameter □ has been introduced which rotates the bits to the right or shift the bits to the right according to the number of 1's in the Input message. This parameter helps to increase the security in the message with different values of □ in each message digest.

References:

[1] Chu-Hsing Lin, Chen-Yu Lee, Yi-Shiung Yeh, Hung-Sheng Chien and Shih-Pei Chien, "Generalized Secure Hash Algorithm: SHA-X"

[2] National Institute of Standards and Technology, "Secure hash standard, "Federal Information Processing Standards Publications *FIPS PUB* 180,May. 1993.

[3] National Institute of Standards and Technology, "Secure hash standard," Federal Information Processing Standards Publications *FIPS PUB* 180-1, 1995.[HAC] A. Menezes, P. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*, CRC Press, Inc., October 1997.

[4] National Institute of Standards and Technology, "Secure hash standard, "Federal Information Processing Standards Publications *FIPS PUB* 180-2", 2001.

[5] National Institute of Standards and Technology, "Secure hash standard," Federal Information Processing Standards Publications *FIPS PUB* 180-2", 2002.

[6] National Institute of Standards and Technology, " Data encryption standard (DES)," Federal Information Processing Standards Publications *FIPS PUB* 46-3", 1999.

[7] Hassan. M. Elkamchouchi, Abdel-Aty M. Einarah Esam A. A. Hagras," A New Secure Hash Dynamic Structure Algorithm (SHDSA) for Public Key Digital Signature Scheme", May 2006