International Journal of Engineering Sciences & Emerging Technologies, August 2013.
ISSN: 2231 – 6604 Volume 6, Issue 1, pp: 29-36 ©IJESET

DESIGN AND IMPLEMENTATION A NEW SECURITY HASH ALGORITHM BASED ON MD5 AND SHA-256

R. Roshdy[1], M. Fouad[2], M. Aboul-Dahab [3]

[1]Department of Electrical & Computer Engineering, Higher Technological Institute, 10th of Ramadan City, Egypt.
eng_roda@yahoo.com

[2]Department of Electronics & Communications Engineering, Zagazig University, Zagazig, Egypt.
fouadzu@gmail.com

[3]Department of Electronics & Communications Engineering, Arab Academy for Science, Technology and Maritime Transport, Cairo, Egypt.
mdahab@aast.edu

ABSTRACT

A cryptographic hash function has an important role in cryptography to achieve certain security goals such as authenticity, digital signatures, digital time stamping, and entity authentication. They are also strongly related to other important cryptographic tools such as block ciphers and pseudorandom functions .Due to the previous merits we present a proposal for a new secure hash algorithm based on the combination of some functions of SHA-256 (Secure Hash Algorithm 256) -with its message expansion modification- and MD5 (Message Digest 5) based on double-Davis-Mayer scheme to overcome the weakness existing in these functions. The proposal hash algorithm has been designed to satisfy the different levels of enhanced security and to resist the advanced hash attacks by increasing the complexity degree of the proposed hash algorithm. The security analysis of the proposed hash algorithm is compared to SHA256 and gives more security and highly acceptable results as shown in our security results and discussions.

KEYWORDS: *Hash function, SHA, Message Digest, compressed function, collision resistance, cryptography security.*

I. INTRODUCTION

A cryptographic hash function H is an algorithm which takes a message of variable length as input and produces a fixed length string as output referred as hash code or simply hash of the input message [1]. The hash value is appended to the message at the source at a time when the message is assumed to be correct. The receiver authenticates that message by re-computing the hash value [2]. A strong cryptographic hash function H is usually expected to satisfy a number of requirements, namely collision resistance, preimage resistance, second preimage resistance [3,12].

Message Digest (MD) describes a mathematical function that can take place on a variable length string. MD5 is a hash function designed by Ron Rivest as a strengthened version of MD4 [4]. It can compress any length of data into an information digest of 128 bits while this segment message digest often claims to be a digital fingerprint of the data [5]. In 1993, B. den Boer and A. Bosselaers found a kind of pseudo-collision for MD5 which consists of the same message with two different sets of initial values. This attack discloses the weak avalanche in the most significant bit for all the chaining variables in MD5 [4]. In 2002, the National Institute of Standards and Technology (NIST) produced a revised version of the standard, FIPS 180-2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512. These new versions have the same underlying structure and use the same types of modular arithmetic and logical binary operations as SHA-1. In 2005, NIST announced the intention to phase out approval of SHA-1 and

move to a reliance on the other SHA versions by 2010. Shortly thereafter, a research team described an attack in which two separate messages could be found that deliver the same SHA-1 hash using 2^{69} operations, far fewer than the 2^{80} operations previously thought needed to find a collision with an SHA-1 hash [2]. SHA-256 is also not secure enough as it has an attack for 46 (out of 64) steps of the compression function with practical complexity [6] and preimage attacks on 41 steps SHA-256 [7]. In 2007 it was a combination between MD5 and SHA-1 with hash code length 160 bits [8], in 2012 it was a combination between MD5 and SHA-1 with hash code length 256 bits [9] .

In this paper we combine the compression function of MD5 with SHA-256 to have a good diffusion so that the output in each round will be spread out and not to be equal with the same output in the next coming stages. This is done with XOR-ing each stage output with next input. At the same time Double-Davies-Meyer scheme will ensure the diffusion and will resist against hackers to reach the minimum distance.

The paper is organized as follows: The next section describes SHA-256 & MD5. Section 3 describes our proposed hash algorithm; equations, block diagram and comparison between hash code of SHA-256, MD5 and the proposed algorithm are given. Results and discussion of the Security analysis are presented in section 4. We end by conclusion and future directions about what can be proposed in the future in section 5.

II. DESCRIPTION OF SHA-256 AND MD5

2.1. SHA-256

The message, M, is padded by Appending the bit "1" to the end of the message, followed by k zero bits, so that its length is congruent to 448mod512. The padded message is parsed into N 512-bit message blocks, $M^{(1)}$, $M^{(2)}$,.., $M^{(N)}$, by appending 64-bit block . The initial hash value, $H^{(0)}$=IV is set, consist of eight 32-bit words, in a hexadecimal form .

SHA-256 uses a message schedule of sixty-four 32-bit words .The words of the message schedule are labelled W_0, W_1,…, W_{63}.

The following steps describe the algorithm:

- Prepare the message schedule,{W_t}

$$W_t = \begin{cases} M_t^{(t)} & 0 \le t \le 15 \\ \sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16} & 16 \le t \le 63 \end{cases}$$

Where

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

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

- Initialize the eight working variables, a, b, c, d, e, f, g, and h, with the (i-1)st hash value:

For t=0 to 63
{

$$T_1 = h + \sum_1^{\{256\}}(e) + Ch(e,f,g) + K_t^{\{256\}} + W_t$$

$$T_2 = \sum_1^{\{256\}}(a) + Maj(a,b,c)$$

h = g, g = f, e = d+T1, d = c, c = b, b = a,
a = T1+T2
}
Where
Ch(x, y, z) = (x ∧ y) ⊕ (x ∧ z)
Maj(x, y, z) = (x ∧ y) ⊕ (x ∧ z) ⊕ (y ∧ z)
$K_t^{\{256\}}$ is a sequence of sixty-four constant 32-bit words

$$\sum_{0}^{\{256\}}(x) = ROTR^{2}(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$$

$$\sum_{0}^{\{256\}}(x) = ROTR^{6}(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$$

- After repeating steps one through four a total of N times (i.e., after processing M(N)), the resulting hash function is $H_0^{(N)} \| H_1^{(N)} \| H_2^{(N)} \| H_3^{(N)} \| H_4^{(N)} \| H_5^{(N)} \| H_6^{(N)} \| H_7^{(N)}$ [10].

2.2. MD5

MD5 processes a variable-length message into a fixed-length output of 128 bits. The input message is broken up into chunks of 512-bit blocks (sixteen 32-bit words).For a padded message M with multiples of l-bit length, the iterating process is as follows:

$H_{i+1} = f(H_i, M_i);$ $0 \le i \le t - 1$

Here $M = (M_0, M_2, \ldots, M_{t-1})$, and $H_0 = IV$ is the initial value for the hash function. The following steps describe the compression function for MD5. For each 512-bit block Mi of the padded message M, divide M_i into 32-bit words, $M_i = (m_0;m_1,\ldots,m_{15})$. The compression algorithm for M_i has four rounds, and each round has 16 operations. Four successive step operations are as follows:

$a = b + ((a + \Phi_i(b,c,d) + w_i + t_i) \lll s_i)$

$d = a + ((d + \Phi_{i+1}(a,b,c) + w_{i+1} + t_{i+1}) \lll s_{i+1})$

$c = d + ((c + \Phi_{i+2}(d,a,b) + w_{i+2} + t_{i+2}) \lll s_{i+2})$

$b = c + ((b + \Phi_{i+3}(c,d,a) + w_{i+3} + t_{i+3}) \lll s_{i+3})$

Where the operation + means ADD modulo 2^{32}. T_{i+j} and S_{i+j} (j = 0, 1, 2, 3) are step-dependent constants. W_{i+j} is a message word . $\lll s_{i+j}$ is circularly left- shift by S_{i+j} bit positions. Each round employs one nonlinear round function, which is given below.

$$\Phi_i(X,Y,Z) = (X \wedge Y) \vee (\neg X \wedge Z), \quad 0 \le i \le 15$$
$$\Phi_i(X,Y,Z) = (X \wedge Y) \vee (Y \wedge \neg Z), \quad 16 \le i \le 31$$
$$\Phi_i(X,Y,Z) = X \oplus Y \oplus Z, \quad 32 \le i \le 47$$
$$\Phi_i(X,Y,Z) = Y \oplus (X \vee \neg Z), \quad 48 \le i \le 63$$

Where X, Y, Z are 32-bit words. The chaining variables are initialized as: a = 0x67452301; b = 0xefcdab89; c = 0x98badcfe; d = 0x10325476 [4].

III. PROPOSED HASH ALGORITHM

The first part of our proposed algorithm is similar to SHA-256 process with message expansion modification but the second part looks like MD5 with extended compress function. The message expansion is not the same one as in SHA256 We use message expansion that helps the minimum hamming weight of the disturbance vector to be high. For a 512-bit message block M, we separate M into 16 words, $W_0 \| W_1 \| \cdots \| W_{15}$. From these 16 words we obtain W_i as follows.

$W_t = \sigma_1(W_{t-1}) + W_{t-9} + \sigma_2(W_{t-15}) + W_{t-16}$, $(16 \le t \le 63)$

Where

$$\sigma_1(x) = x \oplus x^{\lll 7} \oplus x^{\lll 22}$$
$$\sigma_2(x) = x \oplus x^{\lll 13} \oplus x^{\lll 27}$$

Rotation values of σ_1 and σ_2 is selected together by exhaustive search so as to make the hamming weight of the disturbance vector high [11,13,14]. The rest of the SHA256 steps are not changed, we use it as it is. We use two MD5 as SHA256 has 8 registers and MD5 has 4 registers so we use two MD5 .In each round the output of eight registers of SHA256 update the four registers of two MD5 and the output four registers of two MD5 update the eight registers of SHA256 in the next round. We propose Double-Davis-Mayer scheme which we call it SHA256∪MD5 as is shown in Figure 1.

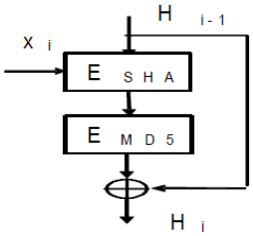


Figure 1. Double-Davis-Mayer scheme

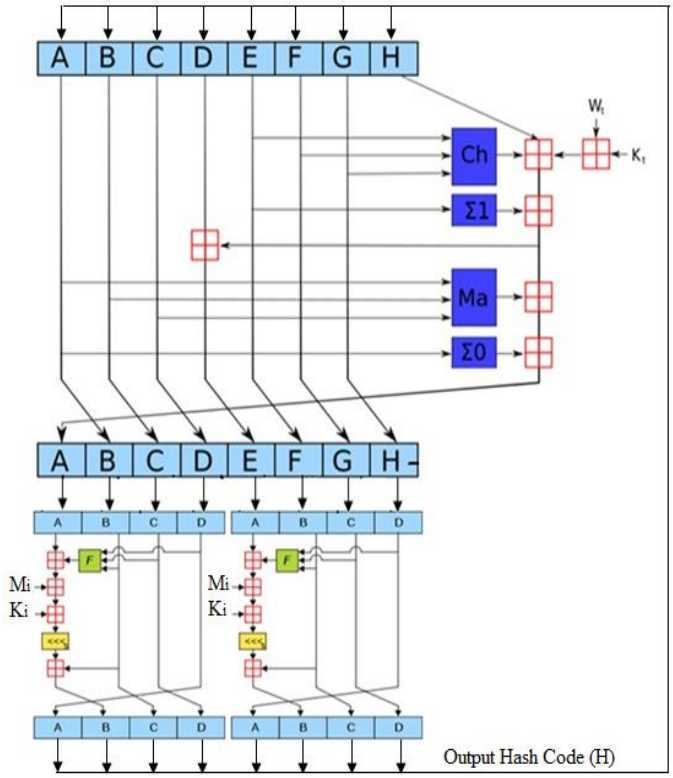


Figure 2. Diagram of the proposed Hash Algorithm

Where: F is a nonlinear function, M_i denotes a 32-bit block of the message input, K_t denotes a 32-bit constant, different for each operation, $\lll s$ denotes a left bit rotation by s places; s varies for each operation, $\boxplus$ denotes addition modulo 2^{32}.
The detailed diagram of algorithm is shown in figure 2 and outputs of certain messages are given in table 1. From table 1 we notice that MD5 gives hash code with length 128 bits while the proposed algorithm gives hash code with length 256 bits like SHA-256 hash code length but it totally differs from hash code of the SHA-256.

IV. RESULTS AND DISCUSSIONS

Like every cryptographic function, hashes are susceptible to brute-force attacks. The longer the hash length (L) is, the more work an attacker has to do to mount an attack. There are three important attacks on hashes:

- A "collision attack" allows an attacker to find two messages M1 and M2 that have the same hash value in fewer than 2^(L/2) attempts.
- A "first-preimage attack" allows an attacker who knows a desired hash value to find a message that results in that value in fewer than 2^L attempts.
- A "second-preimage attack" allows an attacker who has a desired message M1 to find another message M2 that has the same hash value in fewer than 2^L attempts [9,14].

Two tests were run on both proposed algorithm and SHA-256 for comparison between them.

4.1. Avalanche Test

The avalanche effect is evident if, when an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g., half the output bits flip).If hash function does not exhibit the avalanche effect to a significant degree, then it has poor randomization, and thus a cryptanalyst can make predictions about the input, being given only the output. This may be sufficient to partially or completely break the algorithm. Thus, the avalanche effect is a desirable condition from the point of view of the designer of the cryptographic algorithm or device.

Table 1. Comparison of Hash Function for Certain Messages

Message	**MD5 (128 bits)**	**SHA256(256 bits)**	**SHA256∪MD5 (proposed)**
""	D41D8CD98F00B204 E9800998ECF8427E	36A9E7F1C95B82FF B99743E0C5C4CE95 D83C9A430AAC59F8 4EF3CBFAB6145068	F4BE3F7D9C4D05D7 12462F84112EEB25 7A5AC222F56B7304 6E5BC8E32FC02A59
"a"	0CC175B9C0F1B6A8 31C399E269772661	CA978112CA1BBDC FAC231B39A23DC4D A786EFF8147C4E72 B9807785AFEE48BB	EAB4B390344CFAAA A9A62457CC88C8BF 3F04A962A4BEDE3B 1E4F341AE2E49B73
"message digest"	F96B697D7CB7938D 525A2F31AAF161D0	F7846F55CF23E14E EBEAB5B4E1550CAD 5B509E3348FBC4EF A3A1413D393CB650	64A2144BBE953C32 33CE65DF738F06F9 0845F7B138885D93 B138B37224A632B0
"abcdef ghijklm nopqrst uvwxyz"	C3FCD3D76192E400 7DFB496CCA67E13B	71C480DF93D6AE2F 1EFAD1447C66C952 5E316218CF51FC8D 9ED832F2DAF18B73	8EBBEFB78713FD4E FCED26FB022BC624 F401DC5143A0A6B7 BD30FC96AB65D70F
"ABCDEF GHIJKLM NOPQRST UVWXYZ abcdefghijklm nopqrstuvwxy z 0123456789"	D174AB98D277D9F5 A5611C2C9F419D9F	DB4BFCBD4DA0CD85 A60C3C37D3FBD880 5C77F15FC6B1FDFE 614EE0A7C8FDB4C0	1DF8246D8CC48BE4 F6AF9A5152EC9C46 29EDDF4316457325 8407ADC498287A4E
"12345678901 234567890123 456789012345 678901234567 890123456789 012345678901 234567890"	EB579EA6B659CA85 505543CDB391A0AB	F371BC4A311F2B00 9EEF952DD83CA80E 2B60026C8E935592 D0F9C308453C813E	1CC69324FD114353 667C51C033280139 F9ABCBD101E6ADE2 6E68E881707B72C1

We apply the avalanche test with 1 bit difference and more than one bit difference to both proposed algorithm and SHA256 for 20 different messages. From the figure 3 and 4 it is shown that the proposed algorithm has avalanche test with probability greater than SHA256.

4.2. Differential Attack Test

Differential attack makes use of the fact that by changing a small number of message bits of message expansion, it is possible to cancel the difference after a few rounds, or keep the Hamming distance low. Differences are usually defined as the XOR of the value in one run and the corresponding value in the other run (or alternatively additive or multiplicative difference) [9,13].
The new hash function proposed is based on Double- Davies-Meyer scheme which satisfies Merkle Damgard condition. The test is applied to the same message in the previous section.

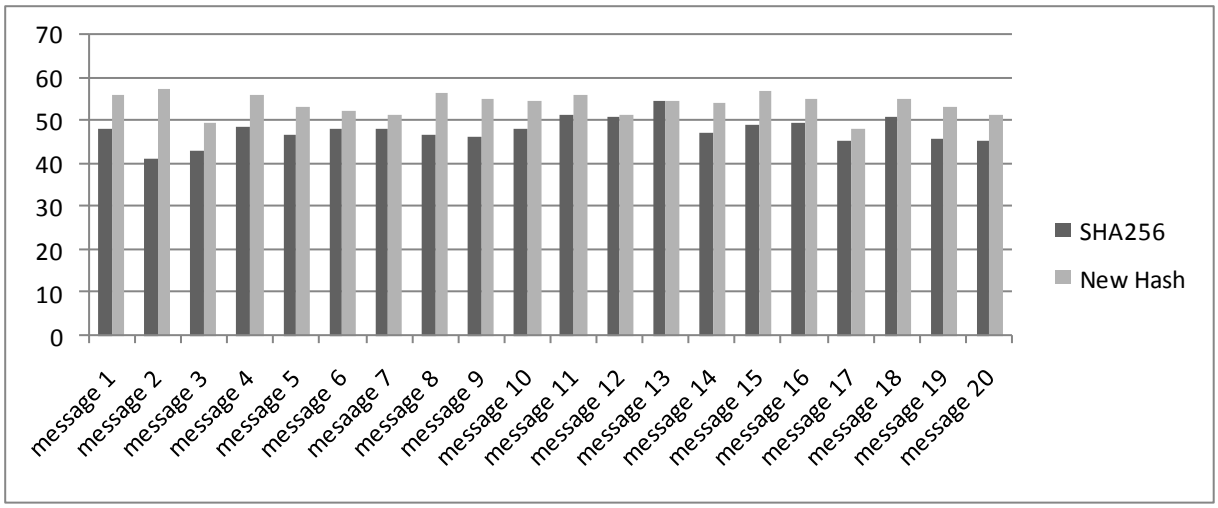


Figure 3. Avalanche test with one bit difference.

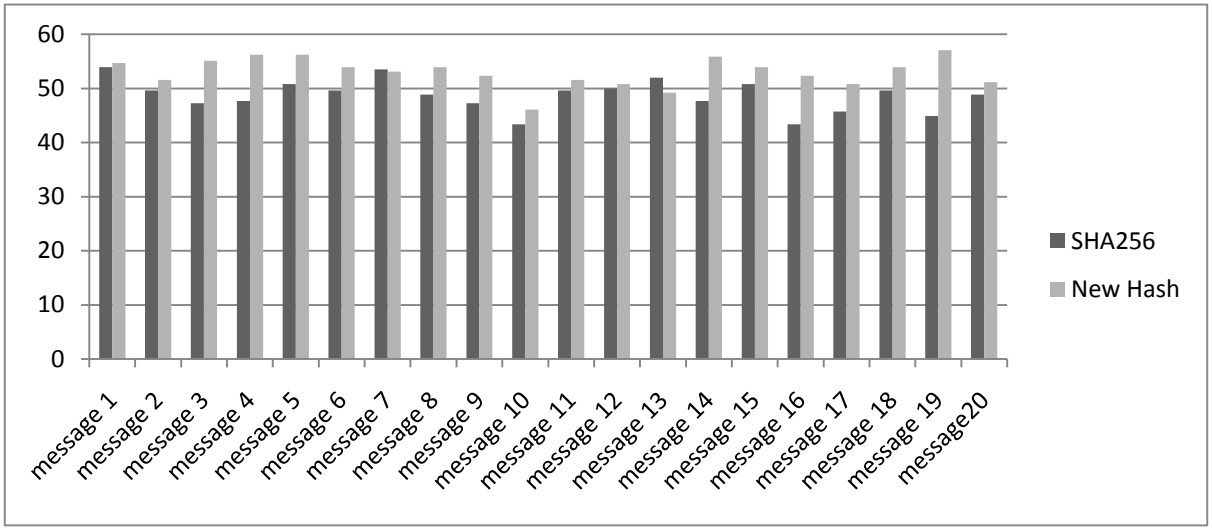


Figure 4. Avalanche test with small difference (more than one bit difference).

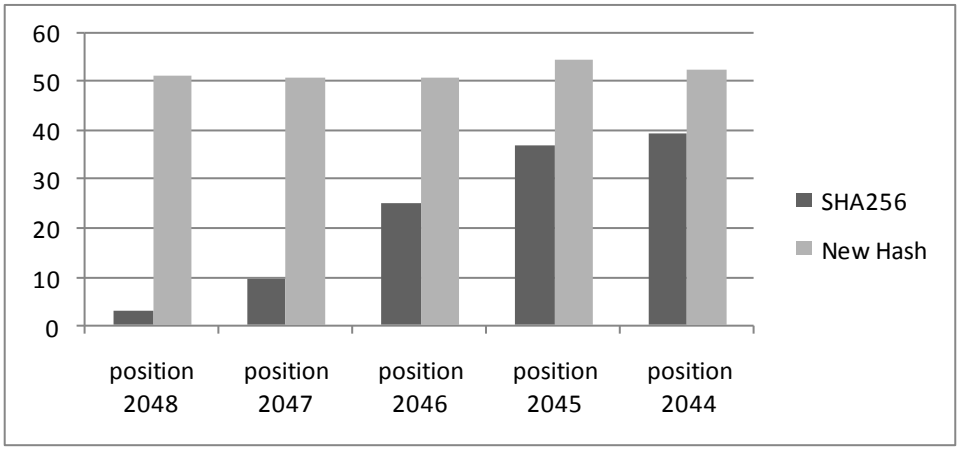


Figure 5. Bit positions difference that has a local collision in SHA256 but not found in New Hash.

Figure 5 illustrate the result of the test. Through the change of bits in message expansion of SHA256 we found that in bit position of 2048,2047,2046,2045,2044 the probability of hash value before and

after changing bits in previous position is less than 50% and from these Positions local collisions may happen but in the proposed algorithm the probability is greater than 50%.
It is also clear that the security of this new algorithm is higher than SHA-256 and MD5 because even if local collision happened in the middle of SHA second algorithm will fade it with an acceptable diffusion so at the start of next round there are no equal states with previous round. It means that $Hi - Hi\text{-}1 \neq 0$ [9].

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new secure hash algorithm based on the previous algorithms, MD5 and SHA-256 that can be used in any message integrity or signing applications where its hash code length is 256 bits. The complexity of the proposed hash algorithm is higher than that of SHA-256 and MD5. The test results of the proposed algorithm show that its security is higher than that of SHA-256 and MD5.The proposed algorithm passes the avalanche test and differential attack test with probability greater than SHA-256.The proposed algorithm is immune to differential attack since the probability of hash value before and after changing bits in previous position is greater than 50%.
We can extend our algorithm to have a bigger size of hash (512, 768 …) like SHAs by extending the block size of compression functions or increasing number of them.

REFERENCES

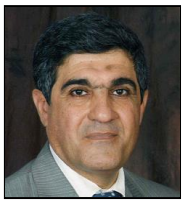
[1] Praveen_Garavaram, “Cryptographic Hash Functions: Cryptanalysis Design and Application", Ph.D thesis, Information Security Institute, Faculty of Information Technology, Queensland University of Technology, 2007.
[2] W. Stalling “Cryptography and Network Security Principles and Practices", Prentice Hall, Fourth Edition, 2005, P 353.
[3] DR.H.Handschub and Dr.H.Gilbert, “Evaluation Report Security Level of Cryptography – SHA-256”, Technical Report, Issy-les-Moulineaux, January 2002.
[4] X. Wang, H. Yu, “How to Break MD5 and Other Hash Functions”, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 19–35, 2005.
[5] R.L. Rivest. The MD5 Message Digest Algorithm. RFC 1321, 1992.
[6] M. Lamberger and F. Mendel, “Higher-order differential attack on reduced SHA-256”, Cryptology ePrint Archive, Report 2011/037, 2011.
[7] Y. Sasaki, L. Wang, and K. Aoki., “Preimage Attacks on 41-Step SHA-256 and 46-Step SHA-512”, IACR Cryptology ePrint Archive, Vol. 2009.
[8] H. Mirvaziri, K.Jumari and M.Ismail “A new Hash Function Based on Combination of Existing Digest Algorithms”, The 5th Student Conference on Research and Development, SCOReD 2007, December 2007.
[9] A. kasgar, J. Agrawal and S. Sahu “New Modified 256-bit MD5 Algorithm with SHA Compression Function”, International Journal of Computer Applications (0975 – 8887) ,Vol.42,No.12, March 2012.
[10] NIST, “Secure Hash Standard (SHS)”, FIPS PUB 180-2, 2002.
[11] J. Lee, D. Chang, E. Lee, H. Kim, D. Hong, J. Sung, S. Hong, and S. Lee, “A new 256-bit hash function DHA-256 − Enhancing the security of SHA-256,” Presented at NIST Cryptographic Hash ,Workshop, 2005.
[12] L. Chen and Gaithersburg, “Communication System Security”, CRC Press, 2012.
[13] M. Juliato and C. Gebotys, “A Quantitative Analysis of a Novel SEU-Resistant SHA-2 and HMAC Architecture for Space Missions Security”, IEEE Transactions on Aerospace and Electronic Systems, Vol. 49, pp. 1536-54, July 2013.
[14] G. Gupta, S. Sharma, “Enhanced SHA-192 Algorithm with Larger Bit Difference”, International Conference on Communication Systems and Network Technologies (CSNT), 2013.

AUTHORS

R. Roshdy received the B.Sc. with grade Very Good in Electronics and Communications Engineering from Faculty of Engineering, at Zagazig University, Zagazig, Egypt, in 2008. She achieved 8th position on her department. Graduation Project “Integrated Smart Home”, with grade Excellent. Currently, he is Assistant Lecturer in Electrical and Computer Engineering department (Higher Technological Institute) at 10th of Ramadan City.

M. Fouad received B.Sc. and M.Sc. degrees in Electronics and Communications Engineering Department from Menofia University, Menofia, Egypt in 1978 and 1984 respectively. He received Ph.D degree in Electronics and Communication Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt in 1991. Currently, he is professor in Electronics and Communications Department at Zagazig University. He published papers in international conferences

in the areas of Communications and Electronics. His research interests are in Digital Communication Systems, Mobile Communication System and Satellites.



M. Aboul-Dahab received B.Sc.in Electronics and Communications Engineering Department from Cairo University, Cairo, Egypt in 1973, received M.Sc. and Ph.D degree in Electronics and Communications Engineering Department from Alexandria University, Alexandria, Egypt in 1980 and 1986 respectively. Currently, he is professor of Antennas Engineering in Electronics and Communications Engineering department (Arab Academy for Science, Technology and Maritime Transport for Cairo Campus). He published papers in international conferences and journals in the areas of Electronics and Communications. His research interests are in Communication Security, Digital Communication Systems and Antennas.