Secure and Time Efficient Hash-based Message Authentication Algorithm for Wireless Sensor Networks

Haider M. Al-Mashhadi
Computer Science Dept.
University of Technology
Baghdad, Iraq
Mashhad01@gmail.com

Hala B. Abdul-Wahab
Computer Science Dept.
University of Technology
Baghdad, Iraq

Rehab F. Hassan
Computer Science Dept.
University of Technology
Baghdad, Iraq

***Abstract*— Wireless sensor networks (WSNs) are used in many applications to gather sensitive information which is then forwarded to an analysis center. Resource limitations have to be taken into account when designing a WSN infrastructure. Authentication in WSNs is critical, as they are often deployed unattended in hostile environments and must transmit information over unsecured mediums. However, the cost of performing cryptographic operations is an extremely limiting factor because sensor devices and related equipment are constrained by storage and computational ability. Hash functions are the most widespread among all Authentication primitives, and are currently used in multiple cryptographic schemes and in security protocols. This paper presents a new Secure Hash Algorithm called (2AMD-160) which uses a famous structure of hash algorithm given by the MIT Laboratory for Computer Science and RSA Data Security, Inc. To demonstrate the effectiveness of 2AMD-160 in terms of security and execution time, we compare our approach with two methods namely, MD5 and SHA1 hash function algorithms. Simulation results demonstrate that the execution time and the security achieved by the proposed method are more effective than the MD5 and SHA1.**

Keywords- **Authentication; digital signature; hash function; security; Wireless Sensor Network (WSN).**

I. Introduction

Wireless sensor network, WSN is an ad-hoc like network that work autonomously. It is implemented in various applications including monitoring and sensing activities drives by the versatility of the technology. It uses economical sensor node that housed multiple sensing units to monitor physical changes of the environment such as temperature, light intensity, level of radiation, pressure and more [1] [2]. It is also largely used to monitor enormous geographical area at low cost because the inexpensive equipment and ease of deployment. Sensor nodes typically have limited memory and processing capabilities, and also must strive to conserve power. Security protocols/algorithms are usually the opposite, requiring memory for key storage, processing overhead for encryption/authentication, and do not really consider power-scarce applications [3].

In order to secure the message transmission security measures such as encryption, decryption and authentication are implemented into the system. Encryption and decryption ensure confidentiality and conceal the message from outsiders [4]. On the other hand, authentication allows entities to validate the integrity of the message and also verify the authority of the communicating devices [5]. Moreover it has the ability to thwart replay attack and it use hash function as a mean of protection [6]. Besides hash function, Message Authentication Code, MAC, Digital Signature and Encryption function are other alternatives that are commonly used to create authenticator for authentication process [7].

Despite all the advantageous, WSN operation suffers from resource's limitation. Wireless sensor is a device that has limited storage, battery life and computational capability [8]. Unfortunately, due to the nature of wireless communication in WSNs, adversaries can easily eavesdrops the traffic, impersonate other users, inject bogus data or alter the contents of legitimate messages during the multi-hop forwarding. Hence, authentication mechanisms need to be implemented to protect messages from various malicious attacks. Here, authentication involves both source and message authentication. While source authentication ensures a receiver that the received data originates from the claimed source, message authentication guarantees that the data from the source is fresh and unchanged [9].

In the recent years much progress has been made in the design of practical one-way hashing algorithms which is efficient for implementation by both hardware and software. Noteworthy work includes the MD family which consist of three algorithms MD2, MD4, MD5 [10] [11], the federal information processing standards for secure hash proposed by NIST [12] for the past few years NIST designed the SHA family which produce 160, 256, 384 and 512bit [13] [14] [15] [16]. SHA-1 which produces message digest of 160bits long was the best established of existing SHA hash functions and employed in several widely uses security application and protocols.

The aim of this research is to design a secure one-way hashing algorithm of 160bit to enhance the security and energy consumption. Certain modifications are introduced in

978-1-4799-5627-2/14/$31.00 ©2014 IEEE

the existing MD5 algorithm to improve the strength of security with a less execution/run time. The maximum security depends on the length of message digest generated by the hash functions which is limited by the size of input to the algorithm. The simulation results show that proposed scheme provides better security than the existing one. To demonstrate the performance of the proposed technique, we performed a set of statistical tests on this method and on other methods. The detailed quantitative analysis and experiment show that our scheme is greatly superior to the traditional message authentication methods in terms of energy consumption and time execution.

The rest of this paper is organized as follows. Related works and related concepts of the cryptographic hash function algorithms are presented in Part 2. In Part 3, the details regarding the proposed enhancement are given. The analysis of the proposed enhancement is presented in Part 4. Finally, conclusion and discussion are presented in Part 5.

II. Related Works

Many of the issues with WSN security have been discussed by [17], [18], [19]. To distinguish legitimate data from intruder's data, authentication techniques are frequently used to verify the integrity of the received data in a communication system. There are several message authentication schemes in wireless sensor networks have been proposed.

Perrig *et al.* [20] proposed a well-known broadcast authentication scheme for WSNs, called µTESLA. µTESLA exploits symmetric key cryptographic operations to authenticate broadcast messages. Thus, it is efficient to WSNs. µTESLA, however, is resilient to node compromise attacks because of delayed disclosure of secret keys. Many variants of µTESLA are proposed for WSNs (e.g., [21] [22] [23] [24]) to improve its performance. The variants as well as µTESLA, however, are subject to the following defects. First, maintenance of time synchronization in WSNs is a complicated task. Second, distribution of the initial parameters introduces heavy overhead since it is implemented by the unicast transmission. Third, delayed authentication is inevitable.

Ren *et al.* [25] and Du et al. [26] propose to authenticate broadcast messages of WSNs with PKC since overhead of PKC for WSNs has significantly been reduced by using Elliptic Curve Cryptography (ECC). Nevertheless, the PKC based broadcast authentication schemes so far are not affordable by current generation of sensor nodes because of the intensive use of Elliptic Curve Digital Signature Algorithm (ECDSA).

In [27], the author introduced an enhanced hash function based on SHA-1 for message authentication implementation in WSN. The proposed SHA-1boolean function is substitute with pseudorandom equation thus provide simplicity and better security performance. In other hand, Arazi in [28] proposed a stream block cipher modification as a message authentication function to be used in resource constrained environment.

Moreover, number of research proposed the enhancement of message authentication function in hardware based. Khan in [29] proposed enhanced HMAC that achieve higher throughput with pipelining and parallelism. Michail in [30] further enhanced the hash function by modifying the hash function blocks using the technique called unrolled. Two steps operation is simplified into one step thus reduce the number of clock cycle and throughput. Lakshmanan et al [31] introduced an enhanced hash function based on SHA-1 for message authentication implementation in WSN called SHA-192. According to previous works and problems many ways are possible to improve the hash function and some of them are reviewed to give insight of the focus of this paper.

III. Proposed Message Authentication Function

The message authentication function is improved by enhancing the underlying cryptographic hash function used to compute the authenticator or digest. MD5 is selected as the hash function to be used in message authentication in WSNs. The aim is to provide better efficiency without compromising the security performance thus making it feasible for implementation in resource constraint environment.

MD5 is one of the hash standards that are recommended by Network Working Group under RFC 1321 [32]. The proposed hash function is called 2ArraysMD-160 (2AMD-160) where it inherits the architecture of original MD5 except that it has one an extra 32-bit word, with modification to improve its efficiency performance.

One of the desirable efficiency metrics selected for this paper is execution/run time. It is the amount of time that is required by a system to produce the message digest after it received the input. One of the ways to improve the time is to revise the structure of the hash function. The MD5 algorithm takes as input a message of arbitrary length and produces as output a 128-bit "message digest" of the input as shows in Fig. 1.

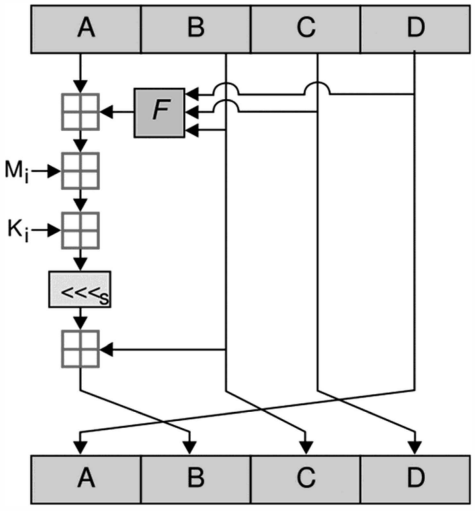

Figure. 1. Structure of the MD5 approach. [32].

The proposed method takes in account the rapid development in processors speed and memory for sensor nodes and the life of batteries that deals with these constrains so we take the length of message digest equal to 160-bit.

The proposed 2AMD-160 algorithm has three processing steps: pre-processing, iterated processing and output transformation. The pre-processing step involves padding, parsing the padded message into m bit block and setting initial values to be used in iterated processing. The iterating process has 64 steps in all and in each step there is an elementary function which calculates a message digest every time and sends it to the next step. The proposed hash algorithm gives us a message digest of length 160 bits.

The modified 2AMD-160 uses the padding algorithm, breaking the message into 512 blocks and adding the length as 64 bit number at end. The output transformation is used in a final step to map the n bit to variable length m bits results called the MD.

The word size and the number of rounds are same as that of Md5. The key characteristics of 2AMD-160 algorithm are shown in Table I.

In order to increase the security aspects of the algorithm the length of message digest should be increased. To achieve this first, number of chaining variables used initially is increased by one. Due to this number of bits generated by message digest is considerably increased, which makes 2AMD-160 method more complex in breaking. The structure of 2AMD-160 algorithm is given in Fig. 2.

The authenticity and integrity of transmitting messages must be secure and easy to compute. The following are the points that explain the proposed scheme.

At the beginning, we present a flow chart of the proposed algorithm as shown in Fig. 3. The process consists of three parts:

A. The Pre-Processing Step

Pre-processing is the step used to prepare the message before the 2AMD-160 processing step. This contains the three steps: padding, parsing the padded message into blocks and setting initial hash values.

- Padding the Message: The purpose of padding is to ensure that the padded message is multiple of 512.

  Padding is performed as follows: a single "1" bit is appended to the message, and then "0" bits are appended so that the length in bits of the padded message becomes congruent to 448, modulo 512. In all, at least one bit and at most 512 bits are appended. A 64-bit representation of b (the length of the message before the padding bits were added) is appended to the result of the previous step.
- Parsing the Padded Message: Parse the message M into N 512 bits of blocks M_1, M_2……, M_N. Each of the Mi parsed into 16, 32bit words Mi_0, Mi_1, …… ..Mi_{15}. The message blocks are processed one at a time, beginning with the initials hash values called message digest buffer.
- Setting Initial Hash Values: Before the hash function begins, the initial hash value H_0 must be set. The hash is

Table I. Key characteristics of 2AMD -160 algorithms.

Name	Block Size/Bits	Word Size/Bits	Output size/Bits	Rounds
MD5 [32]	512	32	128	64
SHA-1 [15]	512	32	160	80
2AMD-160 (Proposed)	512	32	160	64

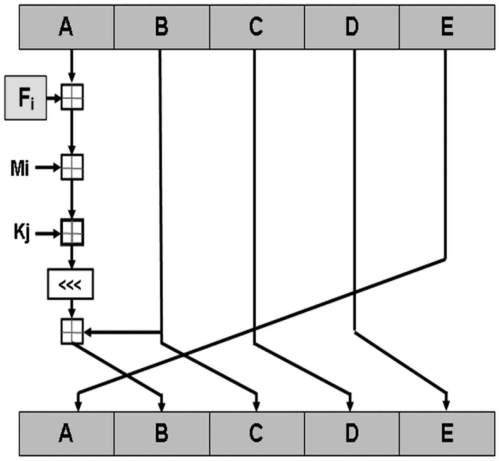

Figure. 2. Structure of the 2AMD-160 approach.

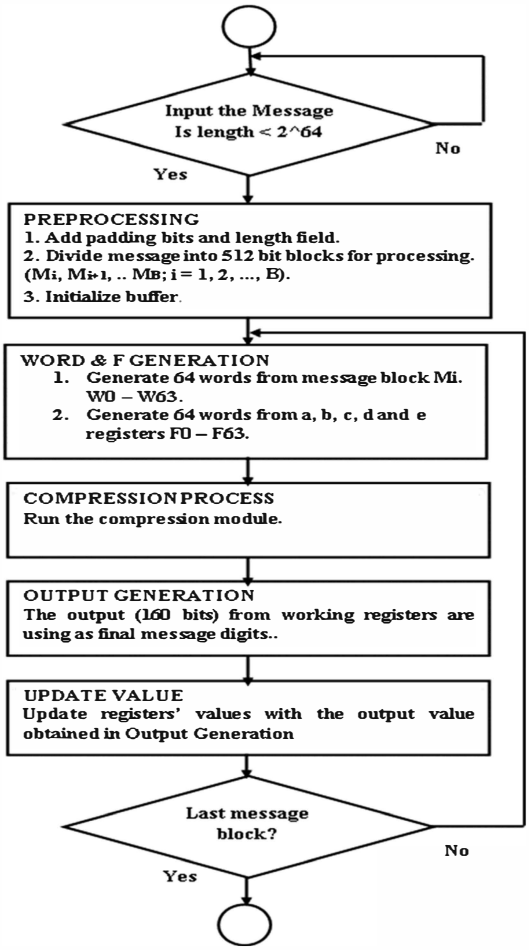

Figure. 3. Flowchart of the proposed algorithm.

160bits used to hold the intermediate and final results. The hash can be represented as five 32bit words registers A, B, C, D and E:
A: 0x67452301;
B: 0xefcdab89;
C: 0x98badcfe;
D: 0x10325476;
E: 0x 89ABCDEF;

B. The Processing Step

The processing step depends upon expanded message block and compression function. In order to increase the security level of the algorithm the size message digest produced should be increased .To achieve this first, number of chaining variables used initially is increased by 32bits. Due to increase in input value the number of bits generated as message digest is also considerably increased. Secondly, the changes have been introduced in substitutes the round function. In this the 64 words from each block message and 64 words from the five 32bits words (i.e. A, B, C, D, and E) are uses to generate the second array of F function as shown in the procedure 1. By using this procedure to replace F function that depending on five registers (A, B, C, D and E) to ensure the independent and unbiased for each bit in the new 64 words. The enhancement also includes add a new register (E) that can increase the security of the method.

Procedure 1. Creating the Wt and Ft words.

Input: Mi, a, b, c, d, e
Output: W, F

1. For i=17 to 64
2. W(i) ⟵ rotate(Xor(W_{i-3}, W_{i-8}, W_{i-14},W_{i-16}),3)
3. End
4. For i=6 to 64
5. F(i) ⟵ rotate(Xor(F_{i-1}, F_{i-2}, F_{i-3}, F_{i-4}) ,3)
6. End

2AMD-160 hash computation uses functions and constants previously defined mod operation is performed after pre-processing is completed each message block is processed in order using the following steps:

Then the computation for T is taken as follows:

```
For t=0 to 64
  {
   T=( A+Wt+Ft+Kt mod 2^32)
   T=circshift(t)
temp =E
E= D
D= C
C= B
B= (B+T mod 2^32)
A=E
}
```

where Wt, Ft is 64 words of 32 bits.
Kt is T[1 ... 64] constructed from the sine function.

C. Output Transformation Step

Compute the intermediate hash value H(i):

$$H_0(i) = Xor(Xor(Xor(A_{New},A_{old}),Wt),f) \quad (1)$$
$$H_1(i) = Xor(Xor(Xor(B_{New},B_{old}),Wt),f) \quad (2)$$
$$H_2(i) = Xor(Xor(Xor(C_{New},C_{old}),Wt),f) \quad (3)$$
$$H_3(i) = Xor(Xor(Xor(D_{New},D_{old}),Wt),f) \quad (4)$$
$$H_4(i) = Xor(Xor(Xor(E_{New},E_{old}),Wt),f) \quad (5)$$

The output transformation step is Xor operation used to map the final output of the single compression function of n bits to the output length.

IV. Simulation Results and Analysis

The hashing algorithms MD5, SHA1 and a newly proposed 2AMD-160 were tested based on the security and time needed to generate message digests for the data. The algorithms have been tested using a system with Intel Core 2 Duo, 2.00GHz processor with 4GB RAM running Microsoft Windows 7 with only one processor enabled for precise result. The efficiency performance is analyzed in software implementation using MATLAB R2010a (version 7.10).

Table II shows the results of message digest that obtained from the variety strings message for all approaches.

We compare the three methods by number of statistical tests on different types of text to explain the success of new method and these statistical tests are [33][34] [35][36]:

A. Frequency

The purpose of this test is to determine whether the number of ones and zeros in a sequence are approximately the same as would be expected for a truly random sequence. The test assesses the closeness of the fraction of ones to ½, that is, the number of ones and zeroes in a sequence should be about the same. Table III shows that the results of message digest for our method is much better than others.

B. Binary Derivation

The binary derivative has been used to measure the randomness of a binary string formed by a pseudorandom number generator for use in cipher systems. Table IV shows that the results of message digest for all methods are much closed.

Table II. Message and Message Digest Results for the three approaches.

Message	MD for MD5	MD for SHA1	MD for 2AMD-160
""	7215EE9C7D9D-C229D2921A40E899EC5F	FCB945BCF88B82999489508787 74C145DF7A6E8A	956C2E0B269F6FE01CF522C30 644A9930D92FCE8
"a"	0CC175E9C0F1B6A831C399E2 69772661	680AE-B77502C48B40C83C441F128FE 8A64B8DD6B	1AA4049F5D19F561EB0318EF-D036F8C480A077BE
"abc"	900150983CD24FB0D6963F7D2 8E17F72	6D1FB81F76AD-FA5FD116F1BF4BEF9578BDA A8586	46117ED54243253534E730BF54 4FBB11A91C5C9A
"Message Digest"	BBD9D8CC4AD8AD2599DBF6 23E7E5282E	1D0650C0870381B2E7922480A 32BCA3DCA2F159F	55307885C4C2EE41CF-F168D4CF9D9C156B70FA5F
"ABCDEFGHIJKLMNOPQ RSTUVWXYZabcdefghijkl mnopqrstuvwxyz0123456789 "	D174AB98D277D9F5A5611C2 C9F419D9F	635D44BDE234505B66A5E5F68 BC7DD209D652D4E	F86AF1C4BCFD6216B2B75765 CF42BD86357FAD1F
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	D5A01D2D92D9026419F2C4B-B5A35B08A	1244A3369C-C727AB7BE64E3318C9526DA3 CACD11	AF81A71412DF6FC326AAEC05 78DC724794E381E7

Table III. Results of the Frequency test of the all Approaches

Message	Frequency		
	MD5	SHA-1	2AMD-160
""	0.4023	0.4	0.3906
"a"	0.4219	0.3688	0.3750
"abc"	0.3984	0.3750	0.4031
"Message Digest"	0.3711	0.3750	0.4
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	0.4023	0.3875	0.3937
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	0.3555	0.4031	0.4

Table IV. Results of the Binary Derivative test of the all Approaches

Message=Message digits	Binary Derivative		
	MD5	SHA-1	2AMD-160
""	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=No
"a"	1st BT=Ok, 2nd BT=No	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=Ok
"abc"	1st BT=Ok, 2nd BT=No	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=No
"Message Digest"	1st BT=Ok, 2nd BT=OK	1st BT=Ok, 2nd BT=No	1st BT=Ok, 2nd BT=Ok
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=No
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	1st BT=Ok, 2nd BT=Ok	1st BT=Ok, 2nd BT=No	1st BT=Ok, 2nd BT=No

C. Change Point

Test for a significant change in the proportion of ones throughout the stream. At each bit position in the stream the proportion of ones to that point is compared to the proportion of ones in the remaining stream. The bit where the maximum change occurs is called the 'change point'. This test determines whether this 'change' is significant. Table V shows that the results of message digest for all methods are the same.

D. Sub-block

Test for the uniformity of non-overlapping sub-blocks of a chosen length. For sub-block sizes up to 16 the 'uniformity test' requires a sample of at least 5 * b * 2^(b) bits, where b is the sub-block size. For sub-block sizes bigger than 16 the 'repetition test' is applied. This test requires a sample of b * 2^(b/2+3) bits. Table VI shows that the results of message digest for our method is much better than others.

E. Runs

The purpose of the runs test is to determine whether the number of runs of ones and zeros of various lengths is as expected for a random sequence. In particular, this test determines whether the oscillation between such zeros and ones is too fast or too slow. Table VII shows that the results of message digest for all methods are the same.

F. Sequence Complexity

Tests that there is a sufficient number of new patterns encountered throughout the stream. A stream with a sequence complexity measure below a given 'threshold' value is considered non-random. An average value of sequence complexity for a stream of this length is also calculated. Table VIII shows that the results of message digest for our method is better than others.

Table V. Results of the Change Point test of the all Approaches

Message=Message digits	Change point		
	MD5	SHA-1	2AMD-160
""	Ok	Ok	Ok
"a"	Ok	Ok	Ok
"abc"	Ok	Ok	Ok
"Message Digest"	Ok	Ok	Ok
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	Ok	Ok	Ok
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	Ok	Ok	Ok

Table VI. Results of the Sub-Blocks test of the all Approaches

Message=Message digits	Sub-Blocks		
	MD5	SHA-1	2AMD-160
""	No	No	No
"a"	No	No	Ok
"abc"	No	No	No
"Message Digest"	No	Ok	Ok
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	No	Ok	Ok
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	No	No	Ok

Table VII. Results of the Runs test of the all Approaches

Message=Message digits	Runs		
	MD5	SHA-1	2AMD-160
""	No	No	No
"a"	No	No	No
"abc"	No	No	No
"Message Digest"	No	No	No
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	No	No	No
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	No	No	No

Table VIII. Results of the Sequence Complexity test of the all Approaches

Message=Message digits	Sequence complexity		
	MD5	SHA-1	2AMD-160
""	No	No	No
"a"	No	Ok	No
"abc"	No	No	No
"Message Digest"	No	No	Ok
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	No	No	No
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	No	No	Ok

G. Linear Complexity

The purpose of this test is to determine whether or not the sequence is complex enough to be considered random. Random sequences are characterized by longer linear feedback shift register (LFSR). Table IX shows that the results of message digest for all methods are the same.

The execution/run time of all approaches depends on the length of input message. Therefore, the run time of 2AMD-160 will improve in the range of 50 % as illustrated in Table X.

From the above tests, 2AMD-160 is success in terms of statistical tests and run time. Therefore, it could be used in the wireless sensor network to authenticate the transmitted message between nodes and base station.

Table IX. Results of the Linear Complexity test of the all Approaches

Message=Message digits	Liner complexity		
	MD5	SHA-1	2AMD-160
""	Ok	Ok	Ok
"a"	Ok	Ok	Ok
"abc"	Ok	Ok	Ok
"Message Digest"	Ok	Ok	Ok
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	Ok	Ok	Ok
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	Ok	Ok	Ok

Table X. Results of run time test

Message=Message digits	MD5 Time/ms	SHA-1 Time/ms	2AMD-160 Time/ms	The time ratio	
				MD5	SHA1
" "	0.1440	0.47	0.2	0.144/0.2=72%	0.2/0.47=42%
"a"	0.1460	0.469	0.2	0.146/0.2= 73%	0.2/0.469=43%
"abc"	0.1530	0.471	0.204	0.153/0.204= 75%	0.204/0.471=43%
"Message Digest"	0.1620	0.466	0.23	0.162/0.23= 70%	0.23/0.466=49%
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"	0.1740	0.7480	0.253	0.1740.253= 69%	0.253/0.748=34%
"12345678901234567890123456789012345678901234567890123456789012345678901234567890"	0.1750	0.744	0.245	0.175/0.245= 71%	0.245/0.744=33%

V. Conclusion

In this paper a 2AMD-160 hash function is proposed. The performance of 2AMD-160 is improves in increasing of security and time consuming without compromising the security. It is found that the number of message blocks influences the run time of the hash function while the increment in message size only slightly increase the run time.

The performance of the proposed method is evaluated and compared with other method under the same test results to demonstrate the effectiveness of the new approach with regards to enhancement of the run time and security of message in wireless sensor network nodes.

References

[1] I. S. Alshawi, L. Yan, W. Pan and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm," *IEEE Sensors J.*, vol. 12, no. 10, pp. 3010-3018, Oct. 2012.

[2] W. Dargie, and C. Poellabauer, "Network layer," in *Fundamentals of Wireless Sensor Networks Theory and Practice*, 1st. Chichester, UK: John *Wieley & Sons, Ltd,* Jul., 2010, pp. 163-204.

[3] Trevatha J., Ghodosi H., and Myers T., "Efficient batch authentication for hierarchical wireless sensor networks," in *IEEE ISSNIP* 2011, pp. 217–222.

[4] W. Stallings, "Classical encryption techniques," in *Cryptography and Network Security: Principles and Practice*, 5th Ed.: *Prentice Hall*, 2011, pp. 31-60.

[5] R. Sharma, et al., "Analysis of security protocols in wireless sensor network," *Int. J. Advanced Networking and Applications*, vol. 2, no. 3, pp. 707-713, 2010.

[6] M. A. Simplicio, et al., "Comparison of authenticated-encryption schemes in wireless sensor networks," in *IEEE LCN* 2011, pp. 450- 457.

[7] W. Stallings, "Cryptographic hash functions," in *Cryptography and Network Security: Principles and Practice*, 5th Ed.: *Pearson Hall*, 2011, pp. 327-356.

[8] Y. Shuai, et al., "A new design of security wireless sensor network using efficient key management scheme," in *IEEE NIDC* 2010, pp. 504-508.

[9] X. Fan and G. Gong, "Accelerating signature-based broadcast authentication for wireless sensor networks", *Ad Hoc Networks*, vol. 10, no. 4, pp. 723–736, Jun. 2012.

[10] S. Bruce, "Applied cryptography: protocols, algorithms and source code in C," John Wiley and Sons, Canada, 1996.

[11] S. Goldwasser, S. Micali and R. Rivest, "A digital signature scheme secure against adaptive chosen-message attacks," *J. Comput.*, vol. 17, no. 2, pp. 281-308, 1988.

[12] M. Ilya, "Hash functions: theory, attacks, and applications," in *Proc. Microsoft Research, Silicon Valley Campus*, pp. 1-22, 2005.

[13] National Institute of Science and Technology, "Secure hash standard (SHS)," Federal Information Processing Standard 180-1, available at: http:/http://www.itl.nist.gov/fipspubs/fip180-1.htm, last visited 1995.

[14] National Institute of Science and Technology, "Secure hash standard," Federal Information Processing Standard 180-2, available at: http://www.itl.nist.gov/fipspubs/fip180-1.htm, Aug. 2002.

[15] National Institute of Standards and Technology, "Secure hash standard," "FIPSPUB180, www.itl.nist.gov/fipspub/fips180-1.html, last visited 2003.

[16] National Institute of Science and Technology, "Implementing cryptography," NIST SP 800, available at: http://csrc.nist.gov/publications/nistpubs/800-21-1/sp800-21-1_Dec2005.pdf, lastvisited 2005.

[17] A. Perrig, J. Stankovic and D. Wagner, "Security in wireless sensor networks," *Commun. ACM*, vol. 47, no. 6, pp. 53–57, Jun. 2004.

[18] J.P. Walters, Z. Liang, W. Shi and V. Chaudhary, "Wireless sensor network security: A survey," *Security in Distributed, Grid, and Pervasive Computing, Auerbach Publications*, pp. 357–410, 2007.

[19] Shantala P., Vijaya B. P., Sonali S., Rashique J. "A survey on authentication techniques for wireless sensor networks", *Int. J. Applied Engineering Research*, vol. 7, no.11, 2012.

[20] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient authentication and signing of multicast streams over lossy channels," *Proc. IEEE Symp. Security Privacy* 2000, pp. 56–73.

[21] D. Liu and P. Ning, "Multilevel μTESLA: broadcast authentication for distributed sensor networks," *Trans on Embedded Computing Sys.*,vol. 3, no. 4, pp. 800–836, Nov. 2004.

[22] Y. Zhou and Y. Fang, "BABRA: batch-based broadcast authentication in wireless sensor networks," in *IEEE GLOBECOM* 2006, pp. 1–5.

[23] P. Ning, A. Liu, and W. Du, "Mitigating DoS attacks against broadcast authentication in wireless sensor networks," *ACM Trans. Sen. Netw.*, vol. 4, no. 1, pp. 1–35, Jan. 2008.

[24] T. Kwon and J. Hong, "Secure and efficient broadcast authentication in wireless sensor networks," *IEEE Trans. Comput.*, vol. 59, no. 8, pp. 1120–1133, Aug. 2010.

[25] K. Ren, W. Lou, K. Zeng, and P. Moran, "On broadcast authentication in wireless sensor networks," *IEEE Trans. Wireless Commun.*, vol. 6, no. 11, pp. 4136–4144, Nov. 2007.

[26] X. Du, M. Guizani, Y. Xiao, and H.-H. Chen, "Defending DoS attacks on broadcast authentication in wireless sensor networks," in *IEEE ICC* 2008, pp. 1653–1657.

[27] A. Abduvaliev, et al., "Simple hash based message authentication scheme for wireless sensor networks," in *IEEE ISCIT* 2009, pp. 982-986.

[28] B. Arazi, "Message Authentication in Computationally Constrained Environments," *IEEE Trans. on Mobile Comput.*, vol. 8, no. 7, pp. 968-974, Jul. 2009.

[29] E. Khan, et al., "Design and performance analysis of a unified, reconfigurable hmac-hash unit," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 12, pp. 2683-2695, Dec. 2007.

[30] H. Michail and C. Goutis, "Holistic methodology for designing ultra high-speed SHA-1 hashing cryptographic module in hardware," in *IEEE EDSSC* 2008, pp. 1-4.

[31] Thulasimani Lakshmanan and Madheswaran Muthusamy," A novel secure hash algorithm for public key digital signature schemes", *Int. Arab J. Infor. Techn.*, vol. 9, no. 3, May 2012.

[32] R. Rivest, "The MD5 message-digest algorithm", *MIT Laboratory for Computer Science and RSA Data Security, Inc. Network Working Group, Request for Comments: 1321*. Apr. 1992.

[33] Andrew Rukhin, et al. "A statistical test suite for random and pseudorandom number generators for cryptographic applications", *National Institute of Standards and Technology*, Apr. 2010.

[34] A. Menezes, P. van Oorschot, and S. Vanstone, "Pseudorandom bits and sequences," in *Handbook of Applied Cryptography*, 5th Ed., *CRC Press*, 2001, pp. 169-187.

[35] Neville Davies, Ed Dawson, Helen Gustafson, A. N. Pettitt ,"Testing for randomness in stream ciphers using the binary derivative", *Stat. Comput.*, vol. 5, no. 4, pp 307-310, Dec. 1995.

[36] Jhon M. Carroll, yonghua Sun, "The binary derivative test for the appearance of randomness and its use as a noise filter", Report no. 221, Nov. 1989. Available at, www.*sim.sagepub.com/content/53/3/129.*