International Journal of Security and Its Applications
Vol. 10, No. 6 (2016) pp.55-64
http://dx.doi.org/10.14257/ijsia.2016.10.6.07

More Reliable Approach for Detecting Data Alterations in Communication Networks

Vikas Dhawan[1] and Gurjot Singh Gaba[2*]

[1,2]Discipline of Electronics & Communication Engineering, Lovely Professional University, Jalandhar, India - 144411
[1]dhawanvicky18@gmail.com
[*]Corresponding Author – [2]er.gurjotgaba@gmail.com

Abstract

Security of precious data is the most critical design parameter of a communication network. Applications like online banking and military information exchange requires more concern over security regardless of other design issues like power and energy consumption. Data alterations in some applications can lead to severe irreparable losses. This paper presents a novel and reliable Hash algorithm which inherits the basic architecture of MD5. Performance of proposed technique is compared with existing MD5, SHA-160, SHA-256 and SHA-384 techniques through statistical test suite for random numbers. Results reveal that the suggested technique is more effective in terms of randomness than the MD5, SHA-160, SHA-256 and SHA-384. The proposed technique thus finds its applicability in the sensitive data environment.

Keywords*: Hash, Data Integrity, Message Digest, Message Authentication Code, Security.*

1. Introduction

In any communication network, different nodes communicate with each other in order to perform certain task; like in Wireless Sensor Network (WSN), thousands of nodes monitor's physical changes in the environment such as pressure, temperature, level of radiation etc. [1, 2]. Generally, each network has to collect some critical information which should not be disclosed to any unauthorized member, in order to prevent mismanagement of data. In recent years, the need of data assurance has been increased to great extent due to the nature of data and its usage in critical applications, where we cannot compromise with the security of data. Data verification schemes have been developed in order to verify that data received at receiver end has not been altered [3]. Different security measures are thus suggested in the literature [9] such as confidentiality, integrity, authenticity and availability. Authenticity is basically of two types: *Source Authentication* & *Message Authentication* where the former assures the identity of the user and the latter assures the reception of unaltered data. In order to ensure confidentiality and authenticity, various encryption and decryption algorithms were applied in the past [15].

In order to ensure data integrity, various hashing algorithms are available. Hash function simply maps an input of arbitrary length to a fixed output by using a noninvertible compression function. Hash functions are very hard to reverse, due to which they have been used for providing security services such as data integrity, origin authentication. They are widely used in applications such as secure email, digital signatures, VPN (Virtual private Network), electronic voting, e-commerce and digital cash. Hash functions are more efficient than cryptographic primitives such as symmetric and asymmetric ciphers [7]. Over the years, many changes have been proposed in the Hash algorithms in order to enhance the strength. There are basically two existing families used for calculating Hash: One-way Hash such as MD family, which constitutes of MD2,

ISSN: 1738-9976 IJSIA

MD4, and MD5 [9] and SHA family, which constitutes of SHA-160, SHA-256, SHA-384, and SHA-512 [13]. Subset of Hash algorithms incurs the use of key for the purpose of producing message digest. They are known as Message Authentication Code (MAC) and Digital Signatures etc. [13]. Message Digest produced by different algorithms has been appended with original message, then at receiver end value of message digest has been compared with initial value, if both values resembles then it assures data integrity [13]. Hashing algorithms are secure because for a given algorithm, it is computationally infeasible to find a message corresponds to a given message digest, or to find two different messages having same message digest. Any change in message will result in a different message digest with a very high probability [4]. In the past, SHA-512 hash algorithm has been used for enhancing the security [4] of MANETs.

In [5], CDF (Check Determinant Factor) is used in measuring data integrity assurance. It involves appending of DF for each data matrix before storing or transmitting the series of data. Now-a-days no security strategy is achieved without assuring data integrity Assurance of data integrity provides reliability which is a prerequisite for most communication network systems and applications.

A research to ensure data integrity is carried out on distributed system consisting of collection of independent nodes where each node stores data fragments. These nodes are connected through LAN or WAN depending upon requirement. When these nodes communicate with each other then there should be some mechanism present to ensure data integrity. In order to overcome this problem, a global hash store system has been added to the distributed system [6], which consists of hash function values of all the data fragments stored. This addition will confirm the integrity of data by calculating hash value and then comparing it with the stored hash value. Stored hash values are available to all authenticated users.

In [8], author illustrated an authenticator based data integrity verification techniques based on cloud and IoT (Internet of Things) data. Due to the adoption of cloud computing technique to a great extent for big data processing and due to the increasing need in analytics of big data such as data generated by IoT, need of data integrity has become most important concern.

Communication Network basically suffers from various types of attacks such as masquerade, disclosure, and traffic analysis, content modification, sequence modification, source repudiation and destination repudiation etc [9]. Therefore, in order to maintain secure transmission, there must be a system that ensures authenticity of the source and message in order to prevent the insertion of false data by adversaries. Different approaches have been derived in the past to maintain data integrity; like in [15], a new scheme of data integrity verification has been proposed for the WSN environment. Cluster Head in WSN not only aggregates the data but also owns the responsibility to assure the flow of legitimate data in the network to avoid disruption of services. A new Homomorphic MAC – based scheme, named E-SHM is also designed in the past to provide data integrity but for different application. Thus, MAC assures for the data integrity of original message and Extended - MAC assures for the data integrity of different MACs. Their dependability ensures the detection of alterations in the data around the whole network. In [14], a security check has been applied in DSDV (Destination Sequenced Distance Vector) routing protocol where all the nodes are assumed to be trustworthy which makes it more difficult to identify malicious attacks. The authors have then proposed the use of MD5 technique to detect the modification attack.

An IPDA (An Integrity Protecting Private Data Aggregation Scheme) has been suggested to provide data privacy and integrity in WSNs [12]. In order to maintain data integrity, it utilizes node-disjoint aggregation tree concept, where each node has its own tree; thus adversary cannot harm the integrity of entire region. Hence, by comparing the output of different trees, base station can easily verify the integrity of aggregated output.

2. Proposed Algorithm

The proposed algorithm is designed to secure one-way hashing algorithm of 128-bit by enhancing the complexity of the function. It is clearly observed from Figure 1 that proposed scheme inherits the architecture of MD5 algorithm. MD5 algorithm is one of the hash standards which are recommended by Network Working Group under [16] RFC 1321. To enhance the impact of MD5 in terms of randomness, certain blocks are appended such as Expansion, S-box, and Permutation into the MD5 architecture. Further, instead of using 2^{32} modulo addition in MD5, 2^{48} modulo addition used in proposed algorithm. The whole process is divided into three steps:

2.1. Pre-Processing Step

It consists of three steps: Padding the message, dividing of message, and initialization of hash values.

2.1.1. Step 1: Hash algorithms have a constraint of input length. Therefore padding has been done in order to ensure that padded message is multiple of 512. Padding is done by appending single '1' bit followed by '0' bits till the length of bits in the message becomes congruent to 448 modulo 512.

2.1.2. Step 2: The message is divided into blocks of specified length. In the recommended technique, a block of 512 bits is used. Then entire message is divided into blocks of 512 bits as seen in Figure 3, and each 512 bit block further divided into 16 blocks of 32 bits each.

2.1.3 Step 3: Before further processing, four 32 bit word registers A, B, C and D are initialized with following values mentioned in Table 1.

Table 1. Initial Hash Values

Registers	32-bit Words
A	01234567
B	89abcdef
C	fedcba98
D	76543210

2.2. 'F' Function

The strength of the proposed technique is due to the integration of 'F' function. It constitutes of expansion, permutation and S-box block as depicted in Fig.2. Expansion block is used to transform 32 bit input data to 48 bit data. Later it undergoes 2^{48} modulo addition followed by substitution box which converts the 48 bit data to 32 bit data. At last, permutation is done in order to achieve more complexity. Rest all the working is similar to MD5.

2.3. Generation of Message Digest

The word registers are updated after execution of 2^{48} modulo addition operation between the initial value and final output value of word register. After the generation of preliminary message digest, next 512 bit block of message and updated value of all four words register acts as a next input for compression function. The message digest of the complete message is obtained after the processing of the last block of the input message.

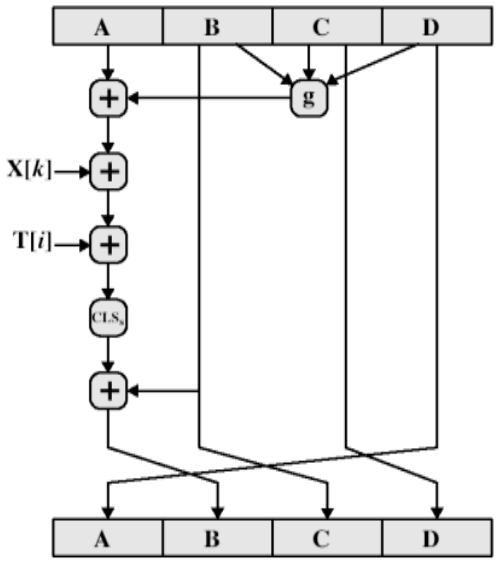

Figure 1. MD5 Compression Function [3]

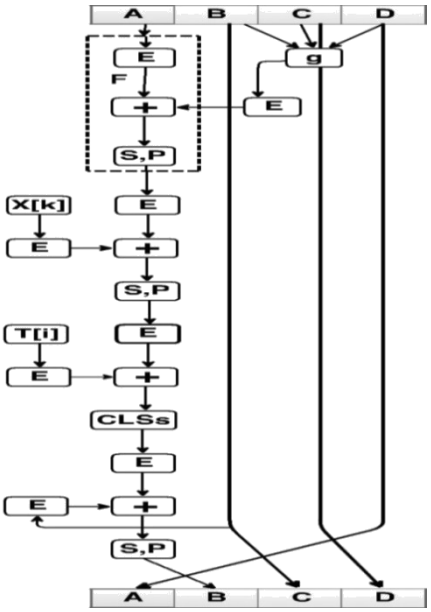

Figure 2. Structure of Proposed Algorithm

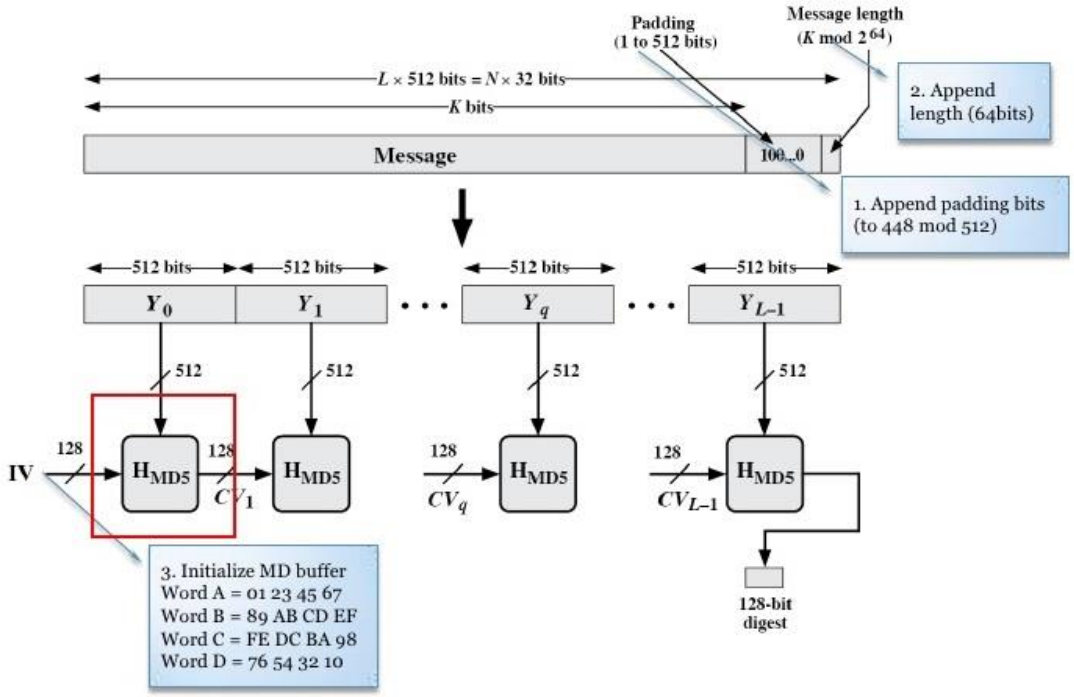

Figure 3. Formation of Blocks

2.4. Processing of Proposed Algorithm

2.4.1. Step 1: Computation of 'g' function, where g(b,c,d) is a different nonlinear function in each round (G,H,I,J):

$$G(B,C,D) = (B\ and\ C)\ or\ (not(B) and\ D) \quad (1)$$

$$H(B,C,D) = (B\ and\ D)\ or\ \Big(C\ and\ \big(not(D)\big)\Big) \quad (2)$$

$$I(B, C, D) = B\ xor\ C\ xor\ D \quad (3)$$

$$J(B, C, D) = C\ xor\ (B\ and\ (not(D)) \quad (4)$$

2.4.2. Step 2: 'X[k]' is used to access the sixteen message blocks, each consisting of 32 bits. For first four blocks equation 1 is used for computing 'g'. For next four blocks equation 2 is used. Similarly same process happens in next blocks which used equations 3 and 4.

2.4.3. Step 3: 'T[i]' is a constant value derived from *Sin* function:

$$for\ i = 0\ to\ 63$$
$$T[i] = floor\left(2^{32} * abs(\sin(i + 1))\right) \quad (5)$$
$$end$$

2.4.4. Step 4: 'CLSs' specify per-round shift amounts with fixed values for each round.

$$for\ i = 0\ to\ 15 => s = [7,12,17,22]$$
$$for\ i = 16\ to\ 31 => s = [5,9,14,20]$$
$$for\ i = 32\ to\ 47 => s = [4,11,16,23]$$
$$for\ i = 48\ to\ 63 => s = [6,10,15,21] \quad (6)$$

2.4.5. Step 5: 'F' function has to perform following tasks. In first instance expanded value of *register A* and *function g* containing 48 bit data has been provided as input to perform 2^{48} modulo operation as in equation 7 and then their output consisting of 48 bit has been provided to 'S' and 'P' block as in equation 8 and 9 in order to convert 48 bit data to 32 bit.

$$\mathrm{Y} = \mathrm{mod}\left(\left(\mathrm{E(A)} + \mathrm{E(g)}\right), 281474976710656\right) \quad (7)$$

$$\mathrm{Y} = \mathrm{S(Y)} \quad (8)$$

$$\mathrm{Y} = \mathrm{P(Y)} \quad (9)$$

3. Results and Discussions

To demonstrate the effectiveness of our technique, we performed a set of different statistical tests to show the randomness of the message digest for the different set of inputs. Message Digest values for different set of inputs are clearly mentioned in Table 2.

3.1. Frequency Test

Frequency test is used to determine the proportion of number of ones and zeros in entire sequence. It checks the closeness between the number of ones and number of zeros. A sequence is said to be random if the proportion of zeros and ones are close to each other [11]. It is clearly observed from Table 3 that proposed algorithm is much better than another mentioned techniques.

Table 2. Message Digest Values for Different Inputs

Hash Techniques	Inputs		
	'messageDigest'	**'Vikas'**	**'CheckWheatherTheTechniqueIsBetterForLongerInputValues'**
	Hash	**Hash**	**Hash**
MD5	'abd8ceec1abec18bffd5f706bd225e2c'	'400cd5bb4f6e6d54b72d8a04b67823e0'	'da4c40b7ae2936e004219d384dbb3a6e'
SHA-160	'5f85283307aedf592336acec10531bdfa67e630f'	'13aba701d4cafb2b12fc265280a7f1d470bad061'	'0f6df2eb03f48b1377c0dbce3638000e611f9059'
SHA-256	'f2f40967a64799dea6eaa2beff7ee2c6f94244f27f63dd510dd9cbd375ea0ef8'	'3a147a0643e7ad80a70963ab3153dd15c42d5f4587b7975debdc4d57f887a5ff'	'84a94730efd3ebb702588001613d06c848142c5591389c07d2e2b16954edcec0'
SHA-384	'03e517a4348a32143aa13e6dab366160f25360ebdfae1d43d801c12643a2c485f837f46afd0422eb2cd3e830438110a6'	'f7f1a00085ea823b388aead26d2ba5f7ffa9daeb17695c5c1122f0fb0ac4703ea7cb339a29cc6f0c2a13c51392384287'	'ffe910b1fa1adeb6d130bcb7393d84bf694206b8bc9d761f0d0499facce2cf5b3302d4035c5eae21ab45bec78c92fff5'
Proposed	'4f537ae87f15064b9e1128d33db588a6'	'07724c724a492cb885d8cf4d9176c4f7'	'7d1056de4ffaea8f80522f77a1456666'

Table 3. Frequency Test Analysis

Hash Techniques	Inputs								
	'messageDigest'			**'Vikas'**			**'CheckWheatherTheTechniqueIsBetterForLongerInputValues'**		
	Ns	**Zs**	**D**	**Ns**	**Zs**	**D**	**Ns**	**Zs**	**D**
MD5	73	55	18	61	67	6	60	68	8
SHA-160	84	76	8	76	84	8	77	83	6
SHA-256	146	110	36	137	119	18	111	145	34
SHA-384	174	210	36	189	195	6	204	180	24
Proposed	*64*	*64*	*0*	*62*	*66*	*4*	*67*	*61*	*6*

***Ns**: Number of Ones, **Zs**: Number of Zeros, **D**: Difference of Ns and Zs*

3.2. Run Test

The purpose of applying this test is to measure the number of runs in entire sequence, where run specifies the number of uninterrupted sequence of identical bits [10, 11]. Table 4 clearly depicts that proposed algorithm has higher rate of interruptions.

Table 4. Run Test Analysis

Hash Techniques	Inputs								
	'message Digest'			**'Vikas'**			**''CheckWheatherTheTechniqueIsBetterForLongerInputValues''**		
	R	**L**	**P**	**R**	**L**	**P**	**R**	**L**	**P**
MD5	62	128	48.43	67	128	47.65	64	128	50.00
SHA-160	76	160	47.50	82	160	51.25	64	160	40.00
SHA-256	126	256	49.21	128	256	50.00	126	256	49.21
SHA-384	183	384	47.65	189	384	49.21	189	384	49.21
Proposed	*69*	*128*	*53.90*	*68*	*128*	*53.21*	*65*	*128*	*50.78*

***R**: Number of Runs in message digest, **L**: Length of message digest (bits), **P**: Percentage of Runs*

3.3. Binary Derivative Test (BDT)

BDT is performed through exclusive-or operation between successive bits until we get only one bit left in the last and store it in different variables, followed by finding the ratio of number of ones to the length of entire sequence in each case. At last, find the average of ratio of all the sequences. If the value lies near to 0.5, then consider the sequence as random in nature [17]. Therefore from Table 5, it is clear that output of proposed algorithm is random in nature.

Table 5. Binary Derivative Test Analysis

Hash Techniques	Inputs		
	'messageDigest'	'Vikas'	'CheckWheatherTheTechniqueIsBetterForLongerInputValues'
	P-Average Value	P-Average Value	P-Average Value
MD5	0.5094	0.5007	0.4953
SHA-160	0.5114	0.5020	0.4848
SHA-256	0.4983	0.5085	0.5078
SHA-384	0.4938	0.4996	0.5044
Proposed	*0.5187*	*0.4895*	*0.5138*

3.4. Avalanche Test

A small change in the input value can cause a significant change in the output message digest value. Generally, we measure the number of flipped bits [9]. Hence, it is clear from Table 6 that rate of change in number of bits is more in proposed algorithm.

$$Avalanche\ effect = \frac{No.of\ flipped\ bits\ in\ the\ message\ digest}{Total\ no.of\ bits\ in\ the\ message\ digest} * 100 \quad (10)$$

Table 6. Avalanche Test Analysis

Hash Techniques	Inputs		
	'messageDigest'	'Vikas'	'CheckWheatherTheTechniqueIsBetterForLongerInputValues'
	P-Average Value	P-Average Value	P-Average Value
MD5	47.65%	46.09%	47.60%
SHA-160	48.75%	49.37%	47.50%
SHA-256	47.26%	49.00%	51.90%
SHA-384	51.56%	50.00%	51.56%
Proposed	*56.25%*	*50.00%*	*52.00%*

3.5. Random Excursion Variant Test

It uses the principle of cumulative sum in order to measure the randomness of the sequence. In this test, P-value is calculated using the error function (*erfc*) [11] as given in equation (11). If the P-value > 0.01, then the sequence will be consider as random sequence. Therefore from Table 7, it is clear that output of proposed algorithm is random in nature.

P-value in Random Excursion Variant test can be calculated using following formula:

$$P - value = erfc\left(\frac{|sigma(x) - j|}{\sqrt{\left(2 \times j \times \left((4 \times |x|) - 2\right)\right)}}\right) \quad (11)$$

3.6. Throughput

Cost of the network resources such as bandwidth is quite high. It is desirable to make the best use of resources by reducing the amount of redundant data transmission. Throughput can measure by using equation (12).

$$Throughput = \frac{Data\ without\ overheads}{Total\ data} \quad (12)$$

Proposed technique appends the message digest of 128 bits with the data which is quite less as compared to others, thus enhancing the throughput of the system.

Table 7. Random Excursion Variant Test Analysis

Hash Techniques	Inputs					
	'messageDigest'		'Vikas'		'CheckWheather TheTechniqueIs BetterFor LongerInput Values'	
	P-Value	Conclusion	P-Value	Conclusion	P-Value	Conclusion
MD5	0.723	Random	0.659	Random	0.753	Random
	1.000	Random	0.836	Random	0.926	Random
	0.838	Random	0.825	Random	0.923	Random
SHA-160	0.513	Random	0.490	Random	0.930	Random
	0.515	Random	0.567	Random	0.848	Random
	0.513	Random	0.539	Random	1.000	Random
SHA-256	0.238	Random	0.904	Random	0.851	Random
	0.813	Random	0.528	Random	0.899	Random
	0.637	Random	0.260	Random	0.923	Random
SHA-384	1.000	Random	0.804	Random	0.479	Random
	0.737	Random	0.932	Random	1.000	Random
	0.798	Random	1.000	Random	1.000	Random
Proposed	*0.246*	Random	*1.000*	Random	*0.698*	Random
	0.465	Random	*0.754*	Random	*0.867*	Random
	0.715	Random	*0.445*	Random	*0.772*	Random

*First three P-values are mentioned for individual inputs

4. Conclusion

In this paper, a new scheme is recommended which is formed by the combination of the basic architecture of MD5 with expansion, substitution and permutation function. The recommended technique is tested on a statistical test suite for random and pseudorandom number generators for cryptography applications introduced by NIST. After analyzing proposed algorithm on different tests, it is clearly observed that performance of the proposed algorithm has improved the performance over the existing MD5, SHA-160, SHA-256 and SHA-384 algorithms. Hence, proposed scheme finds its applicability in the data sensitive environment.

References

[1] W. W. Dargie and C. Poellabauer, "Fundamentals of Wireless Sensor Networks Theory and Practice", UK: JohnWiley & Sons; **(2010)**.
[2] I. S. Alshawi, L. Van, W. Pan and B. Luo, "Lifetime enhancement in wireless sensor networks using fuzzy approach and A-star algorithm", IEEE Sensors Journal, vol. 12, no. 10, **(2012)**, pp. 3010-3018.
[3] C. Malinowski and R. Noble, "Hashing and Data Integrity Reliability of Hashing and Granularity Size Reduction", Digital Investigation 4, (2007), pp: 98-104.
[4] D. Ravilla, C. Shekar, "Enhancing the Security of MANETs Using Hash Algorithm", Proceedings of Elsevier IMCIP., (2015), pp: 196-206.
[5] J. A. Ghaeb, MA. Smadi, J. Chebil, "A High Performance Data Integrity Assurance Based on Determinant Technique", Future Generation Computer Systems, (2011), 27, pp: 614-619.
[6] M. Mittal, R. Sangani, K. Srivastava, "Testing Data Integrity in Distributed Systems", ICACTA., (2015), pp: 446-456.
[7] A. Al-Riyami, N. Zhang, J. Keane, "Impact of Hash Value Truncation on ID Anonymity in WSN", Adhoc Networks., (2016): doi: 10.1016/j.adhoc.2016.02.019.
[8] C. Liu, C. Yang, X. Zhang, J. Chen, "External Integrity Verification for Outsourced Big Data in Cloud and IoT: A Big Picture", Future Generation Computer Systems, (2015), 49, pp: 58-67.
[9] W. Stallings, "Cryptography and Network Security: Principles & Practices", New York, NY: Pearson Education, (2006).
[10] A. Menezes, PV. Oorschot, SA. Vanstone, "Pseudorandom bits and sequence in Handbook of Applied Cryptography", 5th Ed. CRC Press., (2001), pp: 169-187.
[11] AL. Rukhin, LE. Bassham, J. Soto, JR. Nechvatal, ME. Smid, SD. Leigh, M. Levenson, M. Vangel, NA. Heckert, DL. Banks, "A statistical test suite for random and pseudorandom number generators for

cryptographic applications", National Institute of Standards and Technology, (2010).
[12] W. He, H. Nguyen, X. Liuy, K. Nahrstedt, T. Abdelzaher, "iPDA: AnIntegrity- Protecting Private Data Aggregation Scheme for Wireless Sensor Networks", Proceeding of IEEE, Military Communications Conference., (2008)., pp: 1-7.
[13] HM. Al-Mashhadi, HB. Abdul-Wahab, RF. Hassan, "Secure and Time Efficient Hash-Based Message Authentication Algorithm for Wireless Sensor Networks", Proceeding of IEEE, Global Summit on Computer and Information Technology., (2011), pp: 1-6.
[14] Ranjani, Shashikala, C. Kavita, "Secured Data Integrity Routing For Wireless Sensor Networks", Proceeding of IEEE, International Conference On Advances in Electronics, Computer and Communication., (2014), pp: 1-6.
[15] H. Hayouni, M. Hamdi, TH. Kim, "A Novel Efficient Approach for Protecting Integrity of Data Aggregation in Wireless Sensor Networks", Proceeding of IEEE, International Wireless Communications and Mobile Computing Conference., (2015), pp: 1193-1198.
[16] 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).
[17] J. M. Carroll, Y. 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.comlcontent/53/3/129.

Authors

Gurjot Singh Gaba, is currently pursuing Ph.D. in Electronics & Electrical Engineering with Spl. in *Cryptography and Network Security of WSN and IoT's*. He is working as an Asst. Prof. in Lovely Professional University since 2011. His research interest includes Wireless Sensor Networks, Computer Networks, Optical Communications and Cryptography. He is currently engaged in the project of 'Micro Satellite'. He is an author of six International books and more than two dozen research papers.

Vikas Dhawan, is currently pursuing his Masters in Electronics and Communication Engineering from Lovely Professional University. His research area of interest includes - '*Enhancing and Maintaining Security in Wireless Communication Systems' and 'Networks'*. He is working in this field since 2015 and has potential to resolve several problems of industry through his expertise.