International Journal of Security and Its Applications
Vol. 10, No. 8 (2016), pp.39-48
http://dx.doi.org/10.14257/ijsia.2016.10.8.05

A Novel Forgery Detection Mechanism for Sensitive Data

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, [2]er.gurjotgaba@gmail.com

Abstract

There are some applications of military and e-commerce networks, where security of transmitted data on unfriendly environment requires more concern as compared to other design issues like power and energy consumption. Data alterations in above mentioned applications can lead to irreparable losses; so to avoid such losses, we have to maintain Data Integrity in the communication network. This paper presents a novel and reliable Hash algorithm which inherits the basic architecture of SHA-1. Performance of proposed technique is compared with existing techniques through statistical test suite for random numbers. Results reveal that the suggested technique is more effective in terms of randomness than the existing algorithms. The proposed technique, thus, finds its applicability in the sensitive data environment.

Keywords*: Hash Algorithm, Data Integrity, Message Digest, Forgery, Security*

1. Data Integrity: Concept & Threats

In communication networks, a node has to collect some critical information which should not be disclosed to any unauthorized member in order to prevent misuse of data, 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]. Communication Network basically suffers from various types of attacks such as masquerade, disclosure, 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.

From the past few years, the need of data assurance has been increased to great extent due to the nature of data and its usage in critical applications. In case of applications, like military information exchange, and online transactions, 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 hash algorithms are developed in the past. Hash algorithm simply maps an input of arbitrary length to a fixed length output by using a non-invertible compression function. Hash functions are very hard to reverse, due to which they have been used for providing security services such as data integrity, and origin authentication. They are widely used in applications such as secure email, digital signatures, VPN (Virtual private Network), electronic voting, e-commerce, and digital

* Corresponding Author

ISSN: 1738-9976 IJSIA

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, MD4, and MD5 [9] and SHA family, which constitutes of SHA-160, SHA-256, SHA-384, and SHA-512. 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*. Message Digest produced by the algorithm is usually appended with original message during transmission which is cross verified at recipient end. Recipient repeats the process of message digest generation to ensure the resemblance of message digest amongst both parties which 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. In the past, SHA-512 hash algorithm has been used for enhancing the security of MANETs [4].

In the past, Check Determinant Factor (CDF) 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 [5]. 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 in [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.

To ensure reliability, an authenticator based data integrity verification technique based on Cloud and IoT (Internet of Things) is suggested. 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 [8]. Similar scheme for data integrity verification has been proposed for the Wireless Sensor Network (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 [15].

A unique 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 proposed the use of MD5 technique to detect the modification attack [14]. In order to maintain data integrity and data privacy in WSNs, an iPDA (Integrity Protecting Private Data Aggregation Scheme) scheme has been proposed [12]. 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.

The rest of the paper is organized as follows. Section 2 describes the design principle and working of proposed algorithm followed by the Results and Discussions in Section 3. Finally, Section 4 concludes the paper.

2. Proposed Methodology

The proposed algorithm is designed to secure one-way hashing algorithm of 160-bit by enhancing the complexity of the operating algorithm. It can be clearly seen from Figure 1 and 2 that proposed scheme inherits the architecture of SHA-1 algorithm. The SHA-1 belongs to the family of SHA (Secure Hash Algorithm) cryptographic hash functions [16]. It is proposed by the U.S National Security Agency in 1995 as an U.S Federal information processing standard (FIPS), which is published by the National Institute of Standards and Technology (NIST). Proposed algorithm takes an input of arbitrary length and then produce 160-bit message digest. It constitutes of 80 rounds. To enhance the impact of SHA-1 in terms of random output generation, a novel and reliable 'F' function is added into the SHA-1 architecture.

2.1. Working of SHA-1 Algorithm

The whole process is divided into three steps:

2.1.1. Pre-Processing Step: Pre-processing generally refers to: Padding, fragmentation of message, and initialization of hash values.

Step 1: Hash algorithms have a constraint of input length. Therefore padding has been done in order to ensure that padded message becomes multiple of 512 bits. 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.

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 illustrated in Figure 3, and each 512 bit block is further divided into 16 blocks of 32 bits each.

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

Table 1. Buffer Values for SHA-1

Registers	32-bit Words
A	67452301
B	efcdab89
C	98badcfe
D	10325476
E	c3d2e1f0

Step 4: Save these above values in different variables like

$$A_o = A \quad (1)$$

$$B_o = B \quad (2)$$

$$C_o = C \quad (3)$$

$$D_o = D \quad (4)$$

$$E_o = E \quad (5)$$

2.1.2. Processing Step: Each 512 bit block is further divided into sixteen 32-bit blocks and these 16 blocks are further expanded to eighty 32-bit blocks by using various mixing and shifting operations as follows:

$$\begin{gathered} for\ t = 17{:}80 \\ W(t) = W(t-3)\ XOR\ W(t-8)\ XOR\ W(t-14)\ XOR\ W(t-16) \ll 1 \\ end \end{gathered} \quad (6)$$

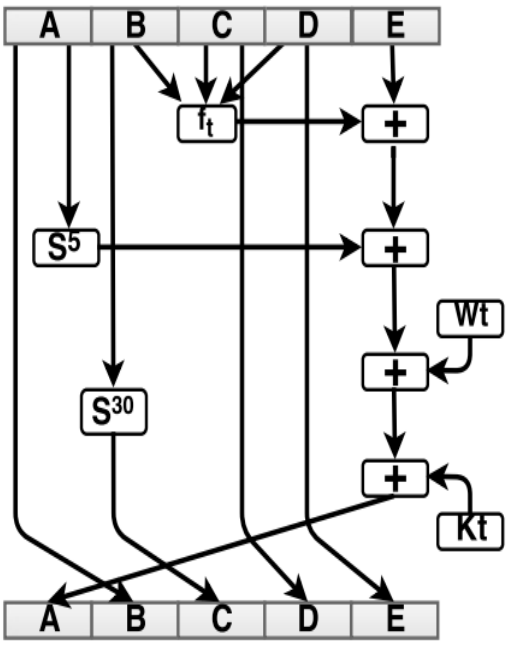

Figure 1. SHA-1 Compression Function

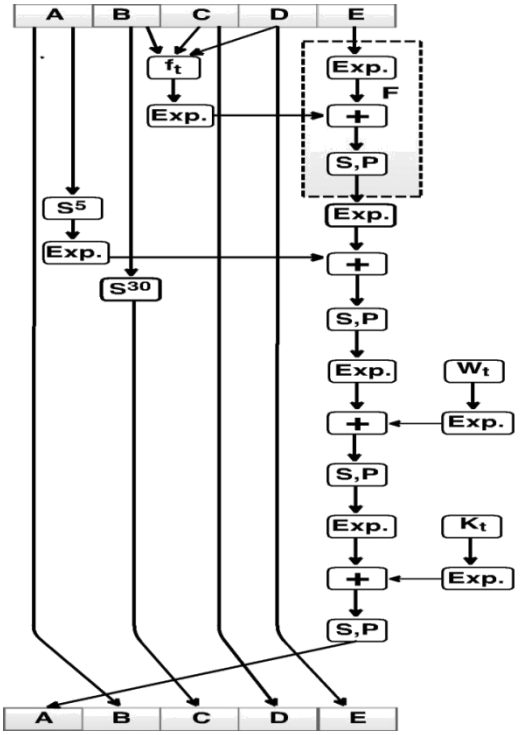

Figure 2. Architecture of Proposed Technique

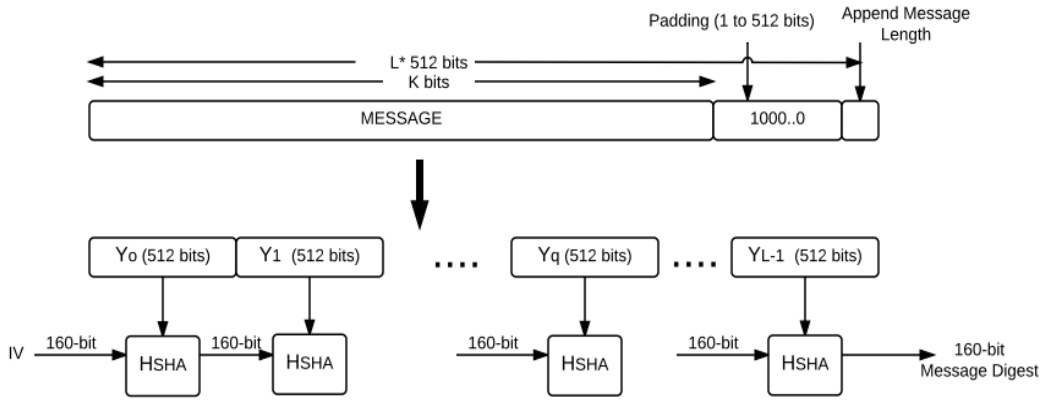

Figure 3. Padding, Fragmentation & Initialization of Buffers

● Computation of 'f_t' function, where f_t (B,C,D) exhibits nonlinear characteristics in each round (G,H,I,J):

$$for\ i = 1\ to\ 20 => \ f(B,C,D) = (B\ AND\ C)\ OR\ (NOT(B) AND\ D) \quad (7)$$

$$for\ i = 21\ to\ 40 => f(B,C,D) = B\ XOR\ C\ XOR\ D \quad (8)$$

$$for\ i = 41\ to\ 60 => f(B,C,D) = (B\ AND\ C)\ OR\ (B\ AND\ D)\ OR\ (C\ AND\ D) \quad (9)$$

$$for\ i = 61\ to\ 80 => f(B,C,D) = B\ XOR\ C\ XOR\ D \quad (10)$$

● W_t is derived from the input block of message.

● K_t is a constant value, derived from the Sine function as follows:

$$for\ i = 1\ to\ 20 => K = 5A827999 \quad (11)$$

$$for\ i = 21\ to\ 40 => K = 6ED9EBA1 \quad (12)$$

$$for\ i = 41\ to\ 60 => K = 8F1BBCDC \quad (13)$$

$$for\ i = 61\ to\ 80 => K = CA62C1D6 \quad (14)$$

- Pre-final values of Word registers are obtained through following operations:

$$Temp = E + f(B, C, D) + (A \ll 5) + W_t + K_t \quad (15)$$

$$E = D \quad (16)$$

$$D = C \quad (17)$$

$$C = (B \ll 30) \quad (18)$$

$$B = A \quad (19)$$

$$A = Temp \quad (20)$$

where '+' implies 2^{32} modulo addition.

2.1.3. Output Transformation Step:

$$A = mod((A_o + A), 4294967296) \quad (21)$$

$$B = mod((B_o + B), 4294967296) \quad (22)$$

$$C = mod\big((C_o + C), 4294967296\big) \quad (23)$$

$$D = mod\big((D_o + D), 4294967296\big) \quad (24)$$

$$E = mod((E_o + E), 4294967296) \quad (25)$$

The word registers are updated after execution of 2^{32} modulo addition operation between the initial values as given in equations (1), (2), (3), (4) and (5) with final output value of word registers as given in equations (21), (22), (23), (24), and (25). After the generation of preliminary message digest, next 512 bit block of message and updated value of all 5 word registers 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.

2.2. Principle of Proposed Methodology

Proposed work additionally incorporates Expansion, Permutation and S-box into the SHA-1 to design more complex and reliable Hash generation method. 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. 'F' function has to perform following tasks. In first instance expanded value of *register E* and *function 'f_t'* containing 48 bit data has been provided as input to perform 2^{48} modulo operation as in equation 26 and then their output consisting of 48 bit has been provided to 'S' and 'P' block as in equation 27 and 28 in order to convert 48 bit data to 32 bit.

$$\mathrm{Y_1} = \mathrm{mod}\Big(\big(\mathrm{Exp(E)} + \mathrm{Exp(f_t)}\big), 281474976710656\Big) \quad (26)$$

$$\mathrm{Y_2} = \mathrm{S(Y_1)} \quad (27)$$

$$\mathrm{Y_3} = \mathrm{P(Y_2)} \quad (28)$$

Further operations are carried out on the final produced value of Y_3. Rest of the working is similar to SHA-1 algorithm.

3. Result and Discussions

To demonstrate the effectiveness of proposed 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 shown in Table 2.

Table 2. Message Digest Values for Different Inputs

Hash Techniques	Inputs		
	'p'	'Vikas'	'messageDigest'
	Hash Values	Hash Values	Hash Values
MD2	3687e026b0a5f81a9fabd5205d804615	109ce46a1a570f332a12c99150148149	b0e9081a28f46f64a160665ae5b89d96
MD5	83878c91171338902e0fe0fb97a8c47a	400cd5bb4f6e6d54b72d8a04b67823e0	abd8ceec1abec18bffd5f706bd225e2c
SHA-224	e22bd066f428b7a77a1b936f99c1f4c117b856705d8f90379579f1e9	82f1662541876589ad967a6801adaeb46975fac4787030a4a6628a1a	54038adb8c9c4accd5767ca31972b521eac608aa091eedd8dd8d34bb
SHA-256	148de9c5a7a44d19e56cd9ae1a554bf67847afb0c58f6e12fa29ac7ddfca9940	3a147a0643e7ad80a70963ab3153dd15c42d5f4587b7975debdc4d57f887a5ff	f2f40967a64799dea6eaa2beff7ee2c6f94244f27f63dd510dd9cbd375ea0ef8
SHA-384	049e7caf67d83409ea363e89c09d67c7f1fd1bd679016ad9f422830ef105435e12a4c2dcad5a9e5a9602924d479574dc	f7f1a00085ea823b388aead26d2ba5f7ffa9daeb17695c5c1122f0fb0ac4703ea7cb339a29cc6f0c2a13c51392384287	03e517a4348a32143aa13e6dab366160f25360ebdfae1d43d801c12643a2c485f837f46afd0422eb2cd3e830438110a6
SHA-512	929872838cb9cfe6578e11f0a323438aee5ae7f61d41412d62db72b25dac52019de2d6a355eb2d033336fb70e73f0ec0afeca3ef36dd8a90d83f998fee23b78d	4c1b426de1061f70a7e0bd4e7a37be18e7ee4111588746df4d02063b058b06b7afe809810b0ef952d77ac700c70de43113c1398e2eb9a0fe7072f3521fbaa530	db252f661fd3af59715c625047342b55b0680c5f46cf29cedfe83601d2578d875d756afe89d88717f55ebfd8833554d74d08be21da8586b24d8e09fc7c3b3307
Proposed	8e1ab2734ddd3ab16041d59c65cd8106ea85ef31	73ea169c9636e2a727b46d15e967c5a1114caa37	602e8475e9e30162d7507875c28ead62f57d5e58

Frequency Test:

Frequency test is used to determine the proportion of number of ones and zeros in entire sequence. It inspects 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 produces better results than other conventional techniques.

Table 3. Frequency Test Analysis

Hash Techniques	Inputs								
	'p'			'Vikas'			'messageDigest'		
	Ns	Zs	D	Ns	Zs	D	Ns	Zs	D
MD2	58	70	12	50	78	28	59	69	10
MD5	59	69	10	61	67	6	73	55	18
SHA-224	118	106	12	102	122	20	110	114	4
SHA-256	134	122	12	137	119	18	146	110	36
SHA-384	189	195	6	189	195	6	174	210	36
SHA-512	268	244	24	247	265	18	265	247	18
Proposed	**78**	**82**	**4**	**81**	**79**	**2**	**79**	**81**	**2**

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

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 indicates that proposed algorithm has higher rate of interruptions.

Table 4. Run Test Analysis

Hash Techniques	Inputs								
	'p'			'Vikas'			'messageDigest'		
	R	L	P	R	L	P	R	L	P
MD2	62	128	48.93	70	128	54.68	66	128	51.56
MD5	54	128	42.18	67	128	47.65	62	128	48.43
SHA-224	105	224	46.87	120	224	53.57	116	224	51.78
SHA-256	132	256	51.56	128	256	50.00	126	256	49.21
SHA-384	195	384	50.78	189	384	49.21	183	384	47.65
SHA-512	253	512	49.41	229	512	44.72	259	512	50.58
Proposed	**83**	**160**	**51.87**	**92**	**160**	**57.50**	**83**	**160**	**51.87**

R: Number of Runs, **L:** Total Length, **P:** Percentage of Runs

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 5 that rate of change in number of bits is more in proposed algorithm.

$$Avalanche\ effect = \frac{No.of\ flipped\ bits\ in\ the\ message\ digest}{Total\ no.of\ bits\ in\ the\ message\ digest} \times 100 \quad (29)$$

Table 5. Avalanche Test Analysis

Hash Techniques	Inputs					
	'p'		'Vikas'		'messageDigest'	
	Flipped Bits	Avalanche Effect	Flipped Bits	Avalanche Effect	Flipped Bits	Avalanche Effect
MD2	67	52.34	62	48.43	56	43.75
MD5	58	45.53	59	46.09	61	47.65
SHA-224	118	52.67	115	51.33	108	48.21
SHA-256	123	48.04	136	53.12	121	47.26
SHA-384	198	51.56	195	50.78	198	51.56
SHA-512	242	47.26	263	51.36	231	45.11
Proposed	**87**	**54.37**	**84**	**52.50**	**91**	**56.87**

Binary Derivative Test (BDT):

BDT is performed through Exclusive-OR operation between successive bits until one bit is left in the last, followed by finding the ratio of number of ones to the length of entire sequence in each case. Thereafter, find the average of ratios. If the value lies near to 0.5, then consider the sequence as random in nature [17]. Therefore from Table 6, it is clear that output of proposed algorithm is random in nature.

Table 6. Binary Derivative Test Analysis

Hash Techniques	Inputs		
	'p'	'Vikas'	'messageDigest'
	$P_{Average\ Value}$	$P_{Average\ Value}$	$P_{Average\ Value}$
MD2	0.4849	0.5085	0.5050
MD5	0.5038	0.5087	0.5094
SHA-224	0.5009	0.5019	0.5057
SHA-256	0.5006	0.5085	0.4983
SHA-384	0.5023	0.4996	0.4938
SHA-512	0.4980	0.5066	0.5018
Proposed	**0.5058**	**0.5123**	**0.5022**

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]. If the P-value > 0.01, then the sequence will be consider as random sequence. Therefore from Table 7, it can be seen that output of proposed algorithm exhibit randomness. P-value can be calculated using following formula:

$$P - value = erfc\left(\frac{|sigma(x) - j|}{\sqrt{\left(2 \times j \times \left((4 \times |x|) - 2\right)\right)}}\right) \quad (30)$$

Table 7. Random Excursion Test Analysis

Hash Techniques	Inputs					
	'p'		'Vikas'		'messageDigest'	
	P-Value	Conclusion	P-Value	Conclusion	P-Value	Conclusion
MD2	0.754	Random	1.000	Random	0.881	Random
	0.779	Random	0.848	Random	0.918	Random
	0.696	Random	0.770	Random	0.914	Random
MD5	0.932	Random	0.659	Random	0.723	Random
	1.000	Random	0.836	Random	1.000	Random
	0.787	Random	0.825	Random	0.838	Random
SHA-224	0.468	Random	0.898	Random	0.825	Random
	0.633	Random	0.947	Random	0.864	Random
	0.805	Random	1.000	Random	0.906	Random
SHA-256	0.328	Random	0.904	Random	0.238	Random
	0.376	Random	0.528	Random	0.813	Random
	0.435	Random	0.260	Random	0.637	Random
SHA-384	0.375	Random	0.818	Random	1.000	Random
	0.380	Random	1.000	Random	0.737	Random
	0.486	Random	0.438	Random	0.798	Random
SHA-512	0.346	Random	0.804	Random	0.472	Random
	0.490	Random	0.932	Random	0.504	Random
	0.590	Random	1.000	Random	0.482	Random
Proposed	**0.560**	Random	**0.194**	Random	**0.715**	Random
	0.626	Random	**0.102**	Random	**0.827**	Random
	0..568	Random	**0.838**	Random	**0.688**	Random

Discrete Fourier Transform Test:

It uses the principle of Discrete Fourier Transform, where we have to find the peak heights present in the DFT of the sequence. The aim of this test is to find various periodic features that would give us the deviation from assumed randomness. The purpose is to find the number of peaks having threshold more than 95% and significantly different than other 5%. Table 8 give us output of DFT test for different hash algorithms. [11]

Table 8. DFT Test Analysis

Hash Techniques	Inputs		
	'p'	'Vikas'	'messageDigest'
	P-Value	P-Value	P-Value
MD2	0.5164	0.8711	0.1443
MD5	0.3304	0.8711	0.3304
SHA-224	0.8063	0.8063	0.1411
SHA-256	0.3588	0.4220	0.8185
SHA-384	0.0027	0.7787	0.4537
SHA-512	0.7456	0.3723	0.7456
Proposed	**0.1468**	**0.4682**	**1.0000**

Maurer's Statistical Test:

The main aim of the test is to find the sequence which can be compressed without any loss of information. The compressible sequence is considered to be non-random. It is clearly observed from Table 9, that proposed algorithm produces more random sequences comparable to others. [11]

Table 9. Maurer's Statistical Test Analysis

Hash Techniques	Inputs		
	'p'	'Vikas'	'messageDigest'
	P-Value	P-Value	P-Value
MD2	0.8527	0.9359	0.9355
MD5	0.9588	0.9053	0.8978
SHA-224	0.9845	0.9259	0.9437
SHA-256	0.9648	0.9466	0.9876
SHA-384	0.9468	0.9904	0.9742
SHA-512	0.9897	0.9777	0.9446
Proposed	**0.9497**	**0.9686**	**0.9930**

Binary Matrix Rank Test:

The focus of the test is the rank of disjoint sub-matrices of the entire sequence. The purpose of the test is to check for linear dependence among fixed length substrings of the original sequence. Table 10 highlights the random output sequences obtained from proposed algorithm. [11]

Table 10. Binary Matrix Rank Test Analysis

Proposed Algorithm		P-Value
Inputs	'p'	**0.5635**
	'Vikas'	**0.5344**
	'messageDigest'	**0.2507**

4. Conclusions

In this paper, a new scheme for generation of Hash is recommended which is formed by the combination of basic architecture of SHA-1 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 observed that performance of the proposed algorithm has outperformed in comparison to existing MD2, MD5, SHA-224, SHA-256, SHA-384 and SHA-512 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", John Wiley & Sons, UK, **(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. Malinowsk and R. Noble, "Hashing and Data Integrity Reliability of Hashing and Granularity Size Reduction", Digital Investigation, vol. 4, no. 2, **(2007)**, pp. 98-104.
[4] D. Ravilla and C. Shekar, "Enhancing the Security of MANETs Using Hash Algorithm", Proceedings of the 11th International Multi-Conference on Information Processing", Bangalore, India, **(2015)** August, pp. 196-206.
[5] J. A. Ghaeb, M. A. Smadi and J. Chebil, "A High Performance Data Integrity Assurance Based on Determinant Technique", Future Generation Computer Systems, vol. 27. no. 5, **(2011)**, pp. 614-619.
[6] M. Mittal, R. Sangani and K. Srivastava, "Testing Data Integrity in Distributed Systems", Proceedings

of the International Conference in Advanced Computing Technologies and Applications, vol. 45, **(2015)**, pp. 446-456.
[7] A. Al-Riyami, N. Zhang and J. Keane, "Impact of Hash Value Truncation on ID Anonymity in WSN", Adhoc Networks, vol. 45, **(2016)**, pp. 80-103. DOI: 10.1016/j.adhoc.2016.02.019.
[8] C. Liu, C. Yang, X. Zhang and J. Chen, "External Integrity Verification for Outsourced Big Data in Cloud and IoT: A Big Picture", Future Generation Computer Systems, vol. 45, **(2015)**, pp. 58-66.
[9] W. Stallings, "Cryptography and Network Security: Principles & Practices", New York, NY: Pearson Education, **(2006)**.
[10] A. Menezes, P. V. Oorschot and S. A. Vanstone, "Pseudorandom bits and sequence in Handbook of Applied Cryptography", 5th Ed. CRC Press, **(2001)**, pp. 169-187.
[11] A. L. Rukhin, L. E. Bassham, J. Soto, J. R. Nechvatal, M. E. Smid, S. D. Leigh, M. Levenson, M. Vangel, N. A. Heckert and D. L. 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 and T. Abdelzaher, "iPDA: An Integrity- Protecting Private Data Aggregation Scheme for Wireless Sensor Networks", Proceedings of the Military Communications Conference, San Diego, US, **(2008)**, pp. 1-7.
[13] H. M. Al-Mashhadi, H. B. Abdul-Wahab and R. F. Hassan, "Secure and Time Efficient Hash-Based Message Authentication Algorithm for Wireless Sensor Networks", Proceedings of the IEEE, Global Summit on Computer and Information Technology, Sousse, **(2011)**, pp. 1-6.
[14] S. Ranjani and C. Kavita, "Secured Data Integrity Routing For Wireless Sensor Networks", Proceedings of the International Conference On Advances in Electronics, Computer and Communication, Bangalore, India, **(2014)**, pp. 1-6.
[15] H. Hayouni, M. Hamdi and T. H. Kim, "A Novel Efficient Approach for Protecting Integrity of Data Aggregation in Wireless Sensor Networks", Proceedings of the International Wireless Communications and Mobile Computing Conference, Dubrovnik, **(2015)**, pp. 1193-1198.
[16] B. Schneier, "Opinion: Cryptanalysis of MD and SHA: Time for a new standard", Computer World, August **(2004)**.
[17] J. M. Carroll and Y. Sun, "The binary derivative test for the appearance of randomness and its use as a noise filter", Report no. 221, November **(1989)**. Available at www.sim.sagepub.com lcontent/53/3/129.

Authors

Gurjot Singh Gaba, He 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, India since 2011. His research interest includes Wireless Sensor Networks, Computer Networks, Optical Communications and Cryptography. He is a reviewer of SCIE and Scopus Indexed Journals. He has recently been appointed as Editor of IJEEE journal. He is a member of many technical bodies including ISCA, IAENG, IACSIT, CSI, and ISTE. He is an author of six International books and more than two dozen research papers.

Vikas Dhawan, He 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.

