# A Implementation of Modified Secure hash Algorithm-3 for High speed and Throughput

[1]Naveen Kumar Harsule, [2]Dr. Vikas Gupta, 3Prof. Neeraj Tiwari

[1]M.Tech Scholar, [2]Professor and HOD, 3Assistant Professor,
[1&2&3]Department of Electronics & Communication Engineering,
[1&2&3]Technocrats Institute of Technology, Bhopal, India.

***Abstract :*** A Secure Hash Algorithms belongs to cryptographic functions which are designed to keep data secured. It works by transforming the data using a hash function: an algorithm that consists of bitwise operations, modular additions, and compression functions. This paper presents implementation of secure hash algorithm-3 for password protection. Simulation is done using Xilinx ISE 14.7 software with verilog code. Result show that proposed SHA-3 gives better area and delay than previous.

***IndexTerms* – Secure, Hash-3. Password, Storage, Xilinx, ISE.**

## I. INTRODUCTION

Rounding Cryptographic hash function is a special class of hash function that has certain properties which make it suitable for use in cryptography. It is a mathematical algorithm that maps data of arbitrary size to a bit string of a fixed size (a hash) and is designed to be a one-way function, that is, a function which is infeasible to invert. The only way to recreate the input data from an ideal cryptographic hash function's output is to attempt a brute-force search of possible inputs to see if they produce a match, or use a rainbow table of matched hashes.

The ideal cryptographic hash function has five main properties:

•It is deterministic so the same message always results in the same hash

•It is quick to compute the hash value for any given message

•It is infeasible to generate a message from its hash value except by trying all possible messages

•A small change to a message should change the hash value so extensively that the new hash value appears uncorrelated with the old hash value.

•It is infeasible to find two different messages with the same hash value.

Hashing methods are categorized into two groups:

1. Data-oriented hashing versus security-oriented hashing

(i) Data-Oriented Hashing Data-oriented hashing refers to methods that intend to use hashing to speed up data retrieval or comparison, where a hash table is often maintained for a query.

(ii)Security-Oriented Hashing Security-oriented hashing refers to methods that use hashing for verification or validation. For example, a user may download software from a public web server but is worried whether the software has been modified by a third party.

### A. Message-Digest Algorithm (MD5)

MD5 algorithm uses four rounds, each applying one of four non-linear functions to each sixteen 32-bit segments of a 512-bit block source text. The result is a 128-bit digest. Figure 1 is a graph representation that illustrates the structure of the MD5 algorithm.

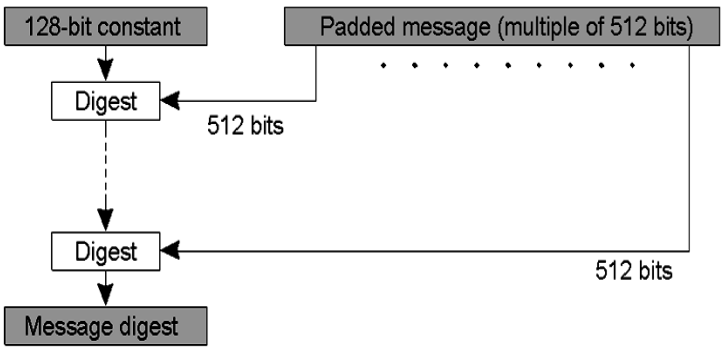


Figure 1: The structure of MD5 algorithm.

MD5 algorithm takes a b-bit message as input, where b is an arbitrary nonnegative integer. The following five steps are performed in C programming language to compute the message digest of the input message.

**B. SHA-3**

SHA-3 (Secure Hash Algorithm 3) was released by NIST on August 5, 2015. SHA-3 is a subset of the broader cryptographic primitive family Keccak. The Keccak algorithm is the work of Guido Bertoni, Joan Daemen, Michael Peeters, and Gilles Van Assche. Keccak is based on a sponge construction which can also be used to build other cryptographic primitives such as a stream cipher. SHA-3 provides the same output sizes as SHA-2: 224, 256, 384 and 512 bits.

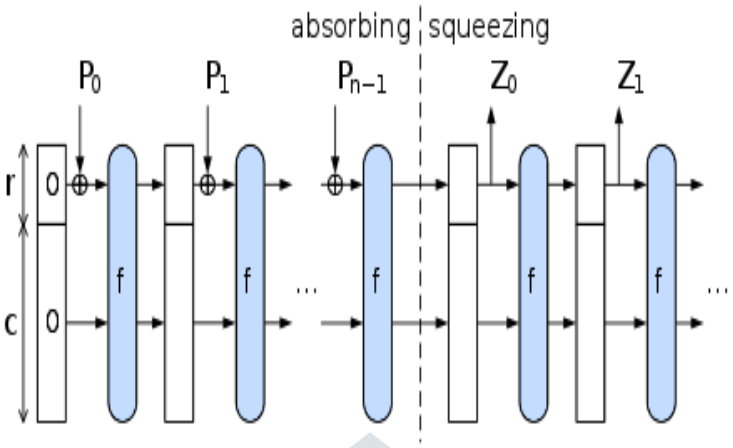


Figure 2: HASH-3

Configurable output sizes can also be obtained using the SHAKE-128 and SHAKE-256 functions. Here the -128 and -256 extensions to the name imply the security strength of the function rather than the output size in bits.

SHA-3 uses the sponge construction, in which data is "absorbed" into the sponge, then the result is "squeezed" out. In the absorbing phase, message blocks are XORed into a subset of the state, which is then transformed as a whole using a permutation function f. In the "squeeze" phase, output blocks are read from the same subset of the state, alternated with the state transformation function f.

A 160 bit buffer is used to hold intermediate value and final results of the hash function.

The buffer can be represented as five 32 bit registers (A, B, C, D, E)

A= 67452301, B = EFCDAB89

C= 98BADCFE, D= 10324576

E =C3D2E1F0

The values are stored in little-endian format, which is the most significant byte of a word in the low address byte position.

Word A= 01 23 45 67, Word B= 89 AB CD EF

Word C= FE DC BA 98, Word D= 76 45 32 10, Word E= F0 E1 D2 C3.

## II. BACKGROUND

A. Alzahrani et al., [1] Implanted multi-center systems are executed as systems-on-chip that depend on packet storeand- rely upon parcel storeand-forward frameworks on-chip for interchanges. These frameworks don't use transports or worldwide clock. Rather switches are used to move information between the focuses, and each inside uses its own neighborhood clock. This proposes simultaneous offbeat figuring. Completing algorithms in such frameworks is particularly encouraged using dataflow ideas. Right now, is propose a philosophy for realizing algorithms on dataflow stages. The system can be applied to multi-strung, multi-focus stages or a mix of these stages too.

J. Haj-Yahya et al., [2] Making sure about a large number of associated, asset compelled registering gadgets is a significant test these days. Adding to the test, outsider specialist organizations need standard access to the framework. To guarantee the trustworthiness of the framework and authenticity of the product merchant, secure boot is upheld by a few business processors. Be that as it may, the existing arrangements are either unpredictable, or have been compromised by decided aggressors.

A. Sengupta et al.,[3] proposed procedure utilizing lightweight secure hashing algorithm (SHA-512)- based key encryption custom equipment reconfigures the key-bits (resulting into basic reconfiguration) of the securing rationale a functionally jumbled DSP configuration increased with the total rationale synthesis of the plan. This thwarts the discovery of the securing rationale in the muddled plan due to disguising. The proposed mechanism incorporated with the functional obscurity system yielded lower power, lower door check, and upgraded security contrasted and an existing methodology. A normal decrease of 25.86 % in door

consider and power well as the normal improvement of 43.75 % in protection from evacuation assault was gotten in the proposed approach contrasted and an ongoing existing methodology.

H. Liu et al., [4] Secure hash function assumes a significant job in cryptography. This work builds a hash algorithm utilizing the hyperchaotic Lorenz framework, which fills in as a sponge function to retain input message by means of numerous parameters time-changing annoyance. To start with, the information message is partitioned into four 1D exhibits, to produce four irritation groupings by means of parameter refreshing guideline, the bothered parameters are still inside their noteworthy interims, to cause the framework to stay a hyperchaotic state. Every emphasis of the discrete hyperchaotic framework utilizes refreshed parameters, and the last state variable qualities are removed to create a len-bit hash an incentive by change algorithm.

C. Biswas et al.,[5] Right now cryptography has been applied utilizing AES and RSA. Right now, the symmetric key utilized for message encryption is likewise scrambled, which guarantees a superior security. An extra component of this work is to make a digital mark by encoding the hash estimation of message. At the accepting side this digital mark is utilized for respectability checking. Then the encoded message, scrambled symmetric key and scrambled digest are joined together to frame a total message. This total message again has been secured utilizing the steganography technique, LSB. Here half and half cryptography gives a superior security, steganography strengthens the security. Message uprightness checking is an exceptional element of this algorithm. Effective reproductions have been appeared to help the possibility of this algorithm.

I. Alfiansyah et al.,[6] Biometric unique finger impression authentication has numerous focal points, yet biometric unique finger impression authentication has a shortcoming in unique mark format that can be taken and the aggressor can recreates finger impression layout to unique finger impression picture. Right now present a usage of SHA-3 hash function for unique mark layout on biometric unique finger impression authentication based Arduino Mega 2560. This work depicts about how to secures unique finger impression format with SHA-3 from aggressor that need to reproduces unique mark layout to unique finger impression picture.

F. E. De Guzman et al., [7] In addition, this highlights the design of 8 registers of A, B, C, D, E, F, G, and H which consists of 64 bits out of the complete 512 bits. The testing of recurrence for Q 15 and Q 0 will demonstrate that the determination of crude function and the consistent utilized are not similarly distributed. Usage of broadened bits for hash message will give extra assets to lexicon assaults and the augmentation of its hash outputs will give an all-inclusive time to giving a stage of 512 hash bits.

S. Yakut et al., [8] In the examination, a post-processing algorithm is proposed for genuine irregular number generators. The last activity is the Keccak hash algorithm. Disorganized frameworks that pre-owned arbitrary characters for creation of crude irregular numbers were utilized. With this strategy, safe arbitrary numbers were produced by applying the proposed post-processing algorithm. The numbers created were indicated that they didn't contain statistical shortcomings by utilizing NIST and autocorrelation tests, Besides; the proposed finishing algorithm is more proficient than the other finishing techniques. As a result, a productive post-processing algorithm is devised which is ok for genuine arbitrary number generators.

A. Alzahrani et al., [9] This technique depends on a novel dataflow chart portrayal of the algorithm. We applied the proposed strategy to get a novel dataflow multi-center processing model for the secure hash algorithm-3. The resulting equipment was actualized in field-programmable entryway cluster to confirm the exhibition parameters. The proposed model of calculation has favorable circumstances, for example, adaptable I/O timing in term of planning strategy, execution of undertakings at the earliest opportunity, and self-coordinated occasion driven framework. In other words, I/O timing and accuracy of algorithm assessment are dissociated right now. The primary favorable position of this proposition is capacity to powerfully muddle algorithm assessment to upset side-channel assaults without updating the framework. This has significant ramifications for cryptographic applications.

A. Jabbar et al.,[10] proposed algorithm is a hybrid encryption algorithm that uses the concept of EC-RSA, AES algorithm and Blowfish algorithm along with SHA-256 for auditing purpose. Presented experiment results show that the proposed concept is reasonable, it enhancing efficiency about 40% in terms of execution time i.e. encryption as well as decryption time and security and providing confidentiality of cloud data at could end.

## III. PROPOSED WORK

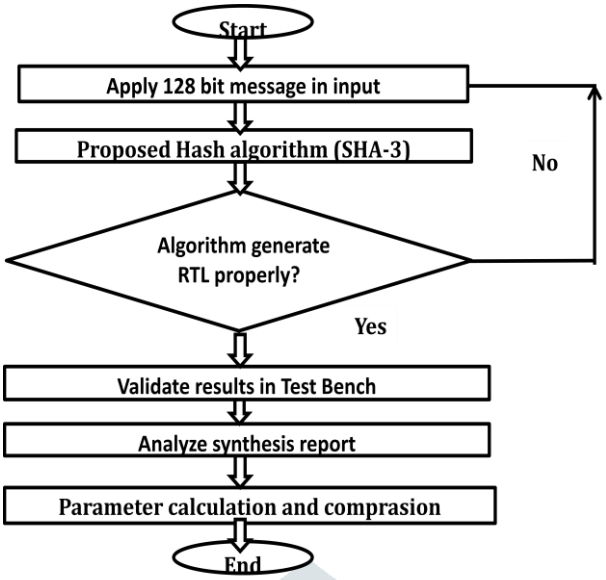


Figure 3: Flow Chart

Algorithm-

- Apply input bits upto 128 bit that may be password or any secure data.
- Now apply proposed hash-3 algorithm, it can generate hash function through hash table.
- Now it will be check from data base, if entered data match from database than user can be access.
- Now view RTL results.
- Now check all result in test bench using Isim simulator.

A cryptographic hash function is an algorithm that can be run on data such as an individual file or a password to produce a value called a checksum. The main use of a cryptographic hash function is to verify the authenticity of a piece of data. Hashing is a mathematical operation that is easy to perform, but extremely difficult to reverse. (The difference between hashing and encryption is that encryption can be reversed, or decrypted, using a specific key.)

## IV. SIMULATION RESULT

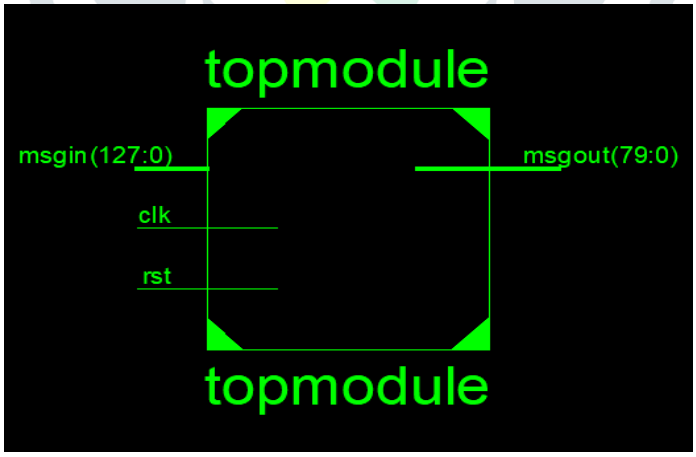


Figure 4: Top view of proposed model

This figure 4 is showing top level module of proposed secure hash algorithm-3. In which apply 128 bit data and it generate 80 bit hash output.

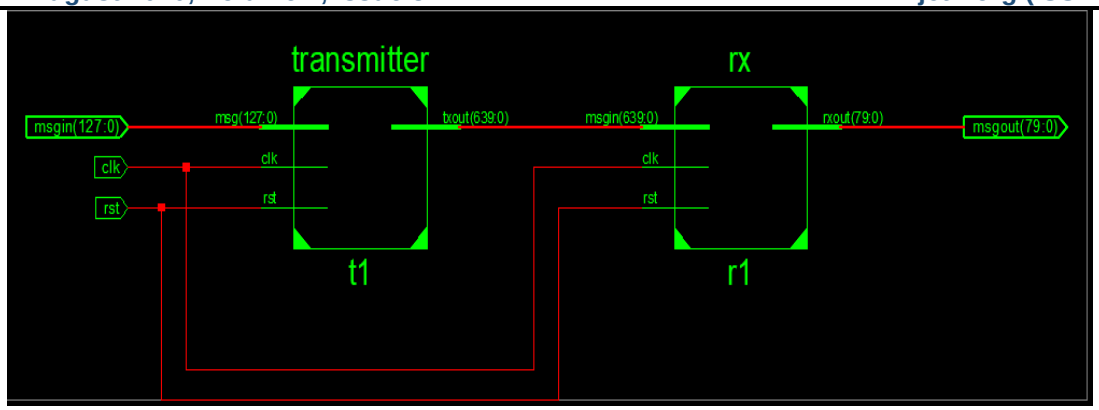


Figure 5: RTL view of proposed Block diagram

Figure 5 is presenting block RTL of sha-3 function. Here firstly apply 128 bit input then at transmitter stage it convert 640 bit. At the receiver end finally it generates 80 bit output.

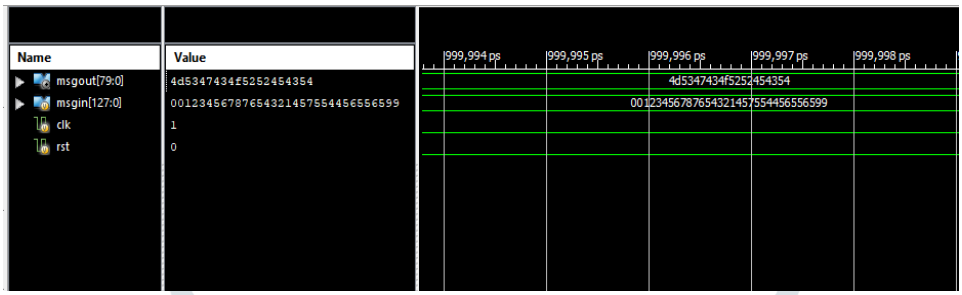


Figure 6: Result validation in test bench

Figure 6 showing 128 bit message in input and it generate 80 bit at output after reduction of bits in modified secure hash algorithm-3 function.

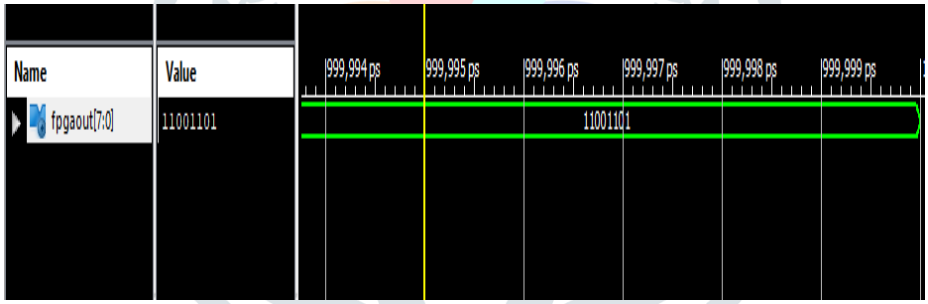


Figure 7: Test bench result for FPGA output

Table 1: Simulation Parameter and Comparison with previous work

| Sr No. | Parameter | Previous Work | Proposed Work |
|---|---|---|---|
| 1 | Method | SHA-3 | SHA-3 |
| 2 | Area ($mm^2$) | 57.6 | 7.5 |
| 3 | Delay(ns) | 24 | 3.259 |
| 4 | Power (mW) | 80 | 41 |
| 5 | Time(secs) | 87.31 | 42.48 |
| 6 | PDP | 1920 | 133.61 |
| 7 | Frequency (MHz) | 520 MHz | 307.69 |
| 8 | Throughput (GHz) | 0.251 | 2.4 |

Table 1 showing comparison of proposed work with previous work, so it can be seen that proposed work gives better result than existing work.

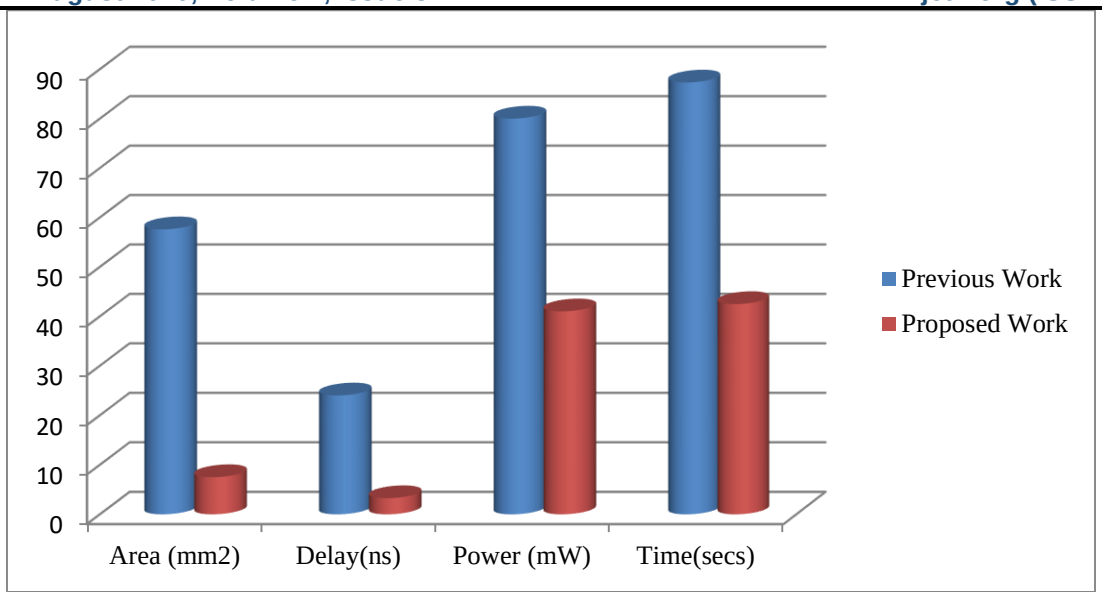


Figure 8: Comparison graph-I

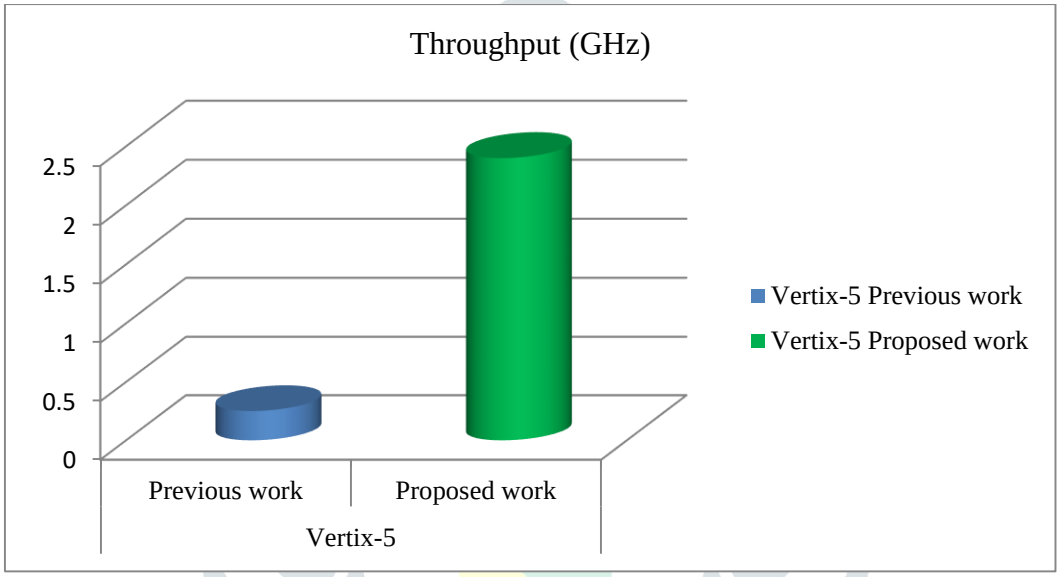


Figure 9: Comparison graph-II

Figure 8 and 9 present's comparison graph of area, delay, power, time and throughput of previous and proposed work. It is clear that proposed work achieve better performance than previous.

## V. CONCLUSION

This paper presents secure cryptographic algorithm, find SHA-3 is latest designed algorithm which is more suitable and useful for secure message in internet applications. Numerous scientists have proposed their own algorithms however none of them are time productive as SHA-3 and furthermore there are odds of enhancing the inward quality of these algorithms. Proposed sha-3 simulation result shows the significant achievement than previous work.

## REFERENCES

1. A. Alzahrani and F. Gebali, "Multi-Core Dataflow Design and Implementation of Secure Hash Algorithm-3," in *IEEE Access*, vol. 6, pp. 6092-6102, 2018.
2. J. Haj-Yahya, M. M. Wong, V. Pudi, S. Bhasin and A. Chattopadhyay, "Lightweight Secure-Boot Architecture for RISC-V System-on-Chip," *20th International Symposium on Quality Electronic Design (ISQED)*, Santa Clara, CA, USA, 2019, pp. 216-223.
3. Sengupta and M. Rathor, "Security of Functionally Obfuscated DSP Core Against Removal Attack Using SHA-512 Based Key Encryption Hardware," in *IEEE Access*, vol. 7, pp. 4598-4610, 2019.
4. H. Liu, A. Kadir and J. Liu, "Keyed Hash Function Using Hyper Chaotic System With Time-Varying Parameters Perturbation," in *IEEE Access*, vol. 7, pp. 37211-37219, 2019.
5. C. Biswas, U. D. Gupta and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography," *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, Cox'sBazar, Bangladesh, 2019, pp. 1-5.

6. Alfiansyah and R. W. Wardhani, "Implementation of Secure Hash Algorithm – 3 for Biometric Fingerprint Access Control Based on Arduino Mega 2560," *2018 International Conference on Applied Information Technology and Innovation (ICAITI)*, Padang, Indonesia, 2018, pp. 31-35.
7. F. E. De Guzman, B. D. Gerardo and R. P. Medina, "Enhanced Secure Hash Algorithm-512 based on Quadratic Function," 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology,Communication and Control, Environment and Management (HNICEM), Baguio City, Philippines, 2018, pp. 1-6.
8. S. Yakut and A. B. Özer, "Secure and Efficient Hash Based Finishing Algorithm for Real Random Numbers," *2018 International Conference on Artificial Intelligence and Data Processing (IDAP)*, Malatya, Turkey, 2018, pp. 1-5.
9. Alzahrani and F. Gebali, "Multi-Core Dataflow Design and Implementation of Secure Hash Algorithm-3," in *IEEE Access*, vol. 6, pp. 6092-6102, 2018.
10. A. Jabbar and P. U. Lilhore, "Design and Implementation of Hybrid EC-RSA Security Algorithm Based on TPA for Cloud Storage", IJOSCIENCE, vol. 3, no. 11, p. 6, Nov. 2017. DOI:https://doi.org/10.24113/ojsscience.v3i10.148.
11. Holmgren and A. Lombardi, "Cryptographic Hashing from Strong One-Way Functions (Or: One-Way Product Functions and Their Applications)," *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, Paris, 2018, pp. 850-858.
12. N. Mouha, M. S. Raunak, D. R. Kuhn and R. Kacker, "Finding Bugs in Cryptographic Hash Function Implementations," in *IEEE Transactions on Reliability*, vol. 67, no. 3, pp. 870-884, Sept. 2018.
13. D. Wang, Y. Jiang, H. Song, F. He, M. Gu and J. Sun, "Verification of Implementations of Cryptographic Hash Functions," in *IEEE Access*, vol. 5, pp. 7816-7825, 2017.
14. S. Chu, Y. Huang and W. Lin, "Authentication Protocol Design and Low-Cost Key Encryption Function Implementation for Wireless Sensor Networks," in *IEEE Systems Journal*, vol. 11, no. 4, pp. 2718-2725, Dec. 2017.
15. S. Koranne, "DÉJÀ VU: An Entropy Reduced Hash Function for VLSI Layout Databases," in *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 34, no. 11, pp. 1798-1807, Nov. 2015.