Reconfigurable FPGA Architecture for Cryptographic Hashing Algorithms

Madhukar M, Nagesh Kumar D N, M C Hanumantharju M C, B M Chandrashekar, Kajol R

Abstract: ***Nowadays, security is the most significant thing in the communication field. Most of the data transmitted over the communication channel are highly confidential so it needs more security. But this confidential data are easily stolen by hackers and it affects the users' privacy. Nowadays, so many encryption algorithms have been established to protect the original information of the users. But, Hash Function (HF) security is the most important primitive which used for data authentication and data integrity. The reconfigurable cryptography integrated with chip which used for cryptography. Hashing algorithm is used to generate the random number which also used as a key value for cryptography. In this paper, different kind of Secured Hash Algorithms (SHA) such as SHA-0, SHA-1, SHA-2, and SHA-3 in that the SHA-2 has different family (SHA-2F) such as SHA-224, SHA-256, SHA-384, and SHA-512 is studied.***

Keyword: Encryption algorithms, Decryption, Hash function, Reconfigurable cryptographic, Secured Hash Algorithms.

I. INTRODUCTION

In recent years, security services have concentrated more on e-transaction, high throughput design, and medical field to protect the confidential data. The authentication performed based on cryptographic HF [1]. This Hash based nearest neighbor is mainly used for visual search, object detection, image matching, and local sensitivity. High dimensional binary codes consider as a hash function which has high efficiency [2]. The SHA algorithm has been published in the National Institute of Standard Technology (NIST) [3-4].The Black algorithm based hash function used for secure the data which maps with binary data [5]. For security, Advanced Encryption Standard (AES) based algorithms used in many research papers. RFID based security is one of the major process to secure the data with high confidential [6]. In the past few years, so many existing algorithms have been used for hashing based security. Bayat-Sarmadi et al. [7], designed a concurrent reliable algorithm with SHA-3 which has effective reliability.

Revised Manuscript Received on December 12, 2019.

Madhukar M, Assistant Professor, Electronics and Communication Engineering , Jyothy Institute of Technology, Bengaluru.

Nagesh Kumar D N ,Assistant Professor of Electronics and Communication Engineering , Jyothy Institute of Technology.

Dr. M. C. Hanumantharaju, Professor of Electronics and Communication Engineering, BMS Institute of Technology and Management, Bengaluru, India.

B M Chandrashekar, Assistant Professor of Electronics and Communication Engineering, Jyothy Institute of Technology, Bengaluru.

Kajol R ,Electronics and Communication department, Jyothy Institute of Technology, Bengaluru.

Space of lightweight cryptographic hashing algorithm [8] used for permutation, pseudo random number generation and which also has random bit generator [9].The SHA processor with SHA-512, SHA-224, SHA-256 is used to get more clock frequency [10], and multi thresholding SHA-1 & SHA-2 algorithm has designed for padding process [11]. Many hashing algorithms have been designed based on SHA-0/1/2/3 [12-14], and these algorithms are briefly explained in below section.

This paper is organized as follows. Section 2, presents a discussion on Crypt-analytics Mechanism. Section 3, explains the application of the hashing algorithm. Section 4, explains the SHA algorithms. Section 4, explains some conventional literature papers. The conclusion is made in section 5.

II. CRYPT-ANALYTICS MECHANISM

Crypt-analytics is a strategy for sending the transmitter information in a specific frame. This transmitted information can also read by other users easily. To secure the information, cryptography is used in the transmission field. After encryption, the plain text converts into the Cipher Text (CT). With the help of key, the cipher text has is converted to original data [15].

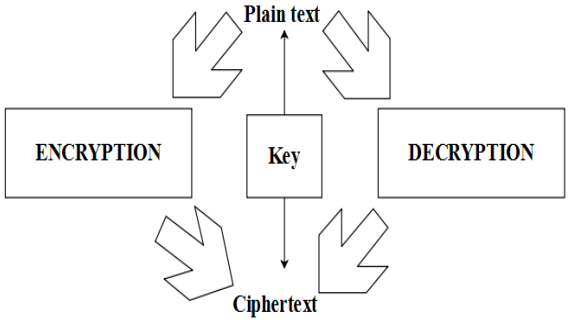

Fig. 1. Crypt-analytics Mechanism

2.1. Key:

This is used to perform the cryptography. This can be generate from manually or any source circuit.

2.2. Plain Text:

The input information is called as a plain text. If you can imagine, "hashing function" is a

plain text which is used to perform the encryption.

2.3. Encryption

After performing encryption, a text file is generated which denoted as CT. Assume, "Azx143@qz12+124" is a CT created for "hashing function". This CT is used to safe an input information. While seeing this CT, no one can understand the input information.

2.4. Decryption:

From the CT, the decryption has been performed. The person who has CT along with decrypted key, they can get the input information from the CT. The same key is required to decrypt the information.

III. APPLICATION OF HASH FUNCTION

- Data authentication is used to check whether the data has been modified or not.
- Data signatures are encrypted with a private key.
- The Password is stored in the memory. It is more difficult to retrieve the password form the memory.
- Key has been generated from passphrase which can be made to prevent the brute force attacks.
- A Pseudorandom number is used to generate the iterated value.
- Virus detection and check the hash file on a system

IV. SECURED HASHING ALGORITHM

In this section, different kind of SHA algorithm is described which is used to secure the data with high confidentiality.

4.1. SHA-0

This algorithm contains 160-bit hash function. Initially, this algorithm named as SHA-0 which generate the significant error. Hence, the SHA-0 is changed as SHA-1.

4.2. SHA-1

This algorithm produces the Message Digest (MD) based on MD4 and MD5 principles. The SHA-1 algorithm partially differs from SHA-0. Single bitwise rotation is the major difference between the SHA-0 and SHA-1. SHA-1 produces a 160-bit hash value which is called as MD. This hash value is represented as a hexadecimal value which has 40 digits long. This SHA-1 has 5 steps which are explained as follows.

Step 1: Bit padding process is performed.

Step 2: Appending length process is performed in this step which calculates the excluding length.

Step 3: The input text is segregated into 512-bit blocks.

Step 4: chaining variables should be initialized. Total 5 chaining variables have been initialized and each chaining variables has 32 bit.

Step 5: Process block

- Chaining variables should be copied
- 512 bits split into 16 sub blocks
- Process 4 rounds of 20 steps

SHA-1: The function H operates as follows,

- Each and every round contains 20 steps which is replaced the 5 buffer words such as F, G, I, J and K with $(K + f(t, G, H, I) + (F<<5) + X_t + K_t), F, (G<<30), I, J)$
- Step number is mentioned as t.
- Nonlinear function for a round is mentioned as f (t, G, H, I).
- From the message block, the X_t is derived and K_t constant value is derived from sine.

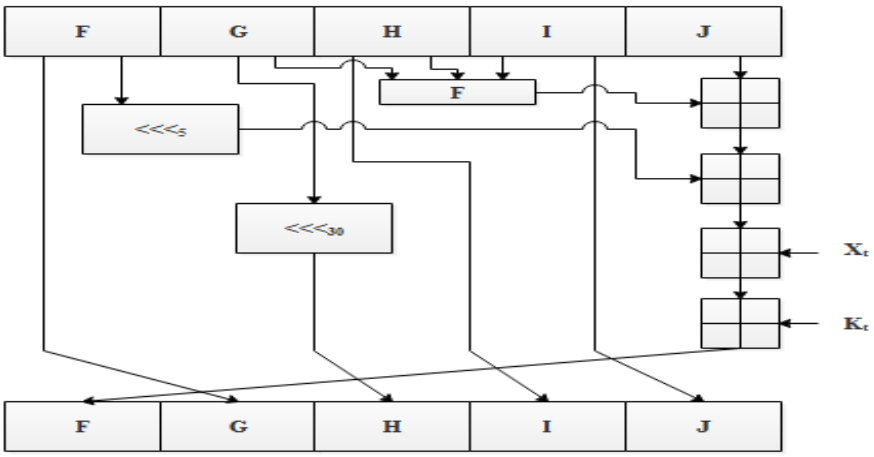

Fig. 2. SHA-1 operation

SHA-1 is most widely used applications which include the TLS, SSL, PGP, SSH, and IP.

4.3. SHA-2

TheSHA-2 algorithm is a modified version of SHA-1 algorithm. The SHA-2 has some family

which is represent as SHA-2F. These algorithms are computed with different bit words such as 64 and 32.But, the structures of the algorithms are identical except the number of rounds.

First, two algorithms have truncated and it produces the SHA-224 and SHA-384. SHA-512 is generating by truncating SHA-512/224 and SHA-512/256.

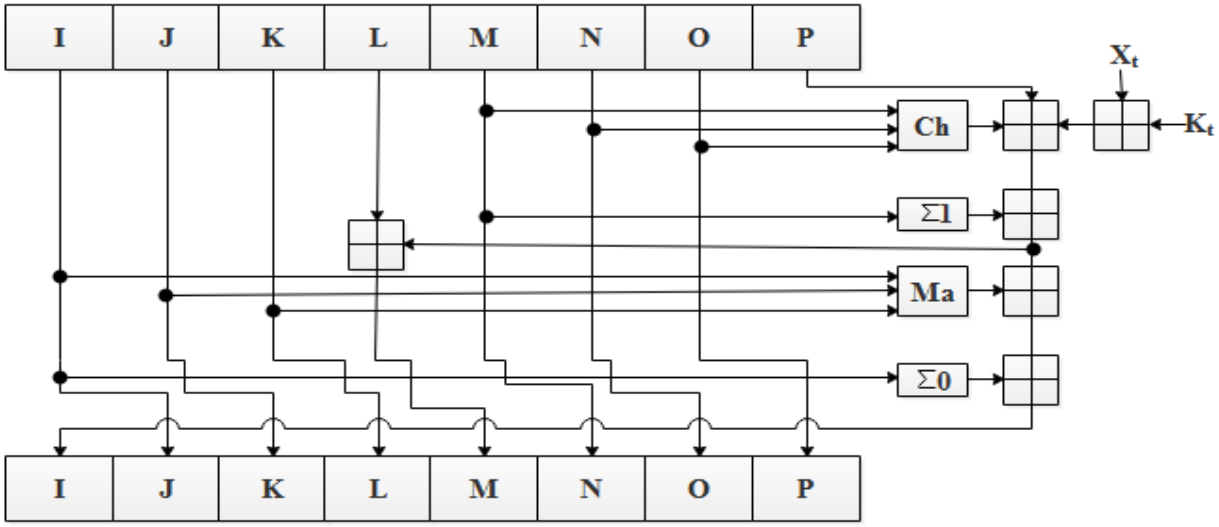

Fig. 3. SHA-2 operation

These families are iterative and it has more secured HF that can process the information without loss. The information digests range is mentioned from 160to 512 bits based on the algorithm. The hashing algorithms are used in some other cryptographic algorithms such as keyed hashed and digital signature functions.

This secured algorithm is very easy to predict the message digest or find the difference of two messages. If any variation occurs in transmission path it leads to the information loss in the message at the receiver section. This is the essential services in the electric mail, e-commerce, and financial transaction. Each algorithm is separated into two types such as pre-processing and hash computation. In preprocessing, padding and parsing process is performed. Hash computation is used to find out the data digest.

4.3.1. Reconfigurable design

The different kind of hashing algorithms are used to re-compute some module such as shift registers, basic operation, and constant storage. From the analysis of the constants K_t, four 32 bit SHA-1 and eighty 64 bit SHA-384/512 was designed. The entire algorithm should be used for both encryption and decryption.

4.3.1.1. X_t architecture

Based on the algorithm, in initial 16 steps, X_t is equal to M_t. For other steps it depends on algorithm process. For SHA-1 the remaining X_t is evaluated using Eq. (1).

$$X_t = (X_{t-3} \oplus X_{t-8} \oplus X_{t-14} \oplus X_{t-16}) <<< 1 \quad (1)$$

For remaining HA, the X_t is calculated using Eq.2.

$$X_t = \sigma_1(X_{t-2}) + X_{t-7} + \sigma_0(X_{t-15}) + X_{t-16} \quad (2)$$

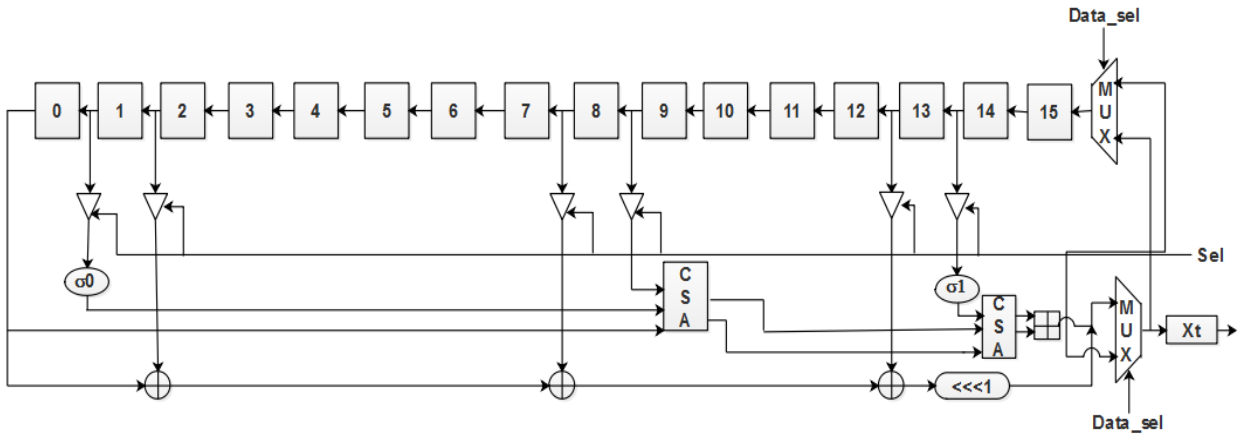

Fig. 4. Architecture of X_t

The architecture of X_t is shown in Fig. 4. The internal module of the X_t contains multiplexer, register, shifter, and XOP module. Reconfigurable

computation is realized by using of MUX. For Initial 16 step values, the MUX selects the 32/64-bit word from the 52/1024-bit data as input which is stored in the initial register and X_t register. The architecture of Carry Look Ahead Adder (CLA) and Carry Save Adder (CSA) is used to improve the performance of X_t architecture.

4.4. SHA - 3

This algorithm uses the sponge construction in which data absorbed into the sponge and the result is called as squeezed. The information is XORed into a subset of the state in absorption phase. In SHA-3, the state contains 5x5 array of 64-bits words. The permutation is performed by XOR operation.

4.4.1. Block Permutation (BP):

- Here, W is mentioned as 21 bit.
- State = 5x5xw bits array.
- BP function has (12+2x1) iterations of 5 sub rounds (f = l, α, β, μ, ʎ)
  - l: one word is created by XOR round constant
  - α: Bitwise rows
  - β: permute the 25 words
  - μ: bitwise rotation
  - ʎ: XOR with two 5xw columns neighbors.

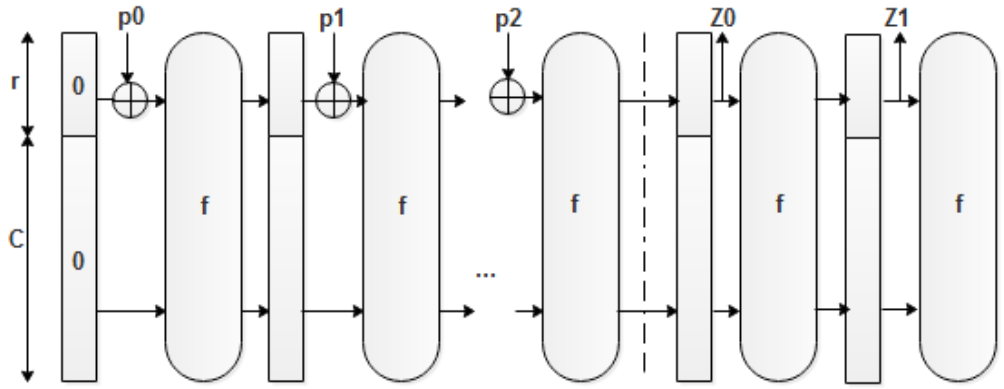

Fig. 5. SHA-3 operation

The SHA-3 family contains four different hash functions such as SHA3-224, SHA3-256, SHA3-384 and SHA3-512. This SHA3-512 has two different functions like SHAKE 256 and SHAKE 128. Overall, the SHA algorithms are used to prevent the data from the hackers.

V. LITERATURE SURVEY

Researchers suggested Several hashing algorithms for the security. In this scenario, a brief evaluation of some important contributions to the existing methods are presented below.

Author	Methodology	Advantage	Disadvantage	Performance Measure
Jaroshaw Sugier [15]	FPGA implementation of BLAKE hash algorithm with black memory resources. This block algorithm was developed for SHA-3.	Successfully qualified in the end stage and KECCAK also considered for suitable cryptographic solution.	The occupied memory is more, so its complex to store all the message information	Slices – 2604 (x1), 4638 (x2), 8382 (x4), 10657 (x5). LUT – 4961 (x1), 8684 (x2), 16142 (x4), 20448 (x5).
Pei Luo *et al*. [16]	Efficient Algebraic Fault Analysis (AFA) with SHA-3 under relaxed fault models. Keccak property is satisfied and suitable for fault analysis.	More efficient. The faults were created by single byte fault model which has very less fault.	The security of the entire algorithm is less.	16 bit fault model: 29.10s for SHA3-224 and 14.61s for SHA3-256

Rourab Paul *et al.* [17]	Pipeline architecture of NSP. SOC was implemented and it performed the encryption with hashing algorithm	This was dynamically executed to reduce the time complexity. Low power.	The key length is very easy to predict. The hash function can be hacked easily by the hackers.	In SHA-256 slices – 1385, power – 176mw and 3.85ns critical path (CP). In SHA-512, slices 2654, 278mw power, and 5.50ns CP.
Rommel Garcia *et al*. [18]	• This work presents Efficient and compact FPGA processor for the SHA-256 algorithm. • This is designed based on custom data path which has 4 input ALU.	• Virtex 5 devices has less hardware utilization. • The compact design saves around 60% of slices. • SHA suits for mobile applications.	• The operation of ALU took more time for execution. • For MUX design, normal adder only used which cause more power consumption.	Frequency – 64.45 Mhz Slices – 139 LUT – 527 Latensy – 280 Throughput – 117.85 Mbps
Jian Wang *et al.* [19]	Quantum Key Distribution (QKD) system introduced in this paper with hash based security.	• Key generation rate is more. • QKD is extended up to 10 times. • NIST test verifies the randomness test.	• This paper failed to explain the way of generate the key. • XOR encryption operation easy to predict the original data.	Frequency – 0.546 Runs – 0.639 Rank – 0.6724 FFT – 0.0937 Universal – 0.8846
H.E. Michail *et al.* [20]	In this paper, Area efficient and high throughput multi-mode architecture for SHA-1 and SHA-2 was designed.	• This system has exact systematic flow. • Better area and throughput. • High key security.	• Carry save adder occupy more power and more execution time. • Unbalanced registers are used to store the value.	In SHA-256/512, 40.3MHz frequency, 6911 slices, 5158 mbps throughput.
I.Algredo-badillo et al. [21]	FPGA based inner loop secure hash algorithm SHA-256 developed in this paper	• The mapping architecture helps to reduce the delay. • The throughput is more than conventional methods.	• This architecture has less clock frequency. So the speed of the entire architecture may be slow. • Normal adder was used in SHA-256 which causes more area.	In SHA-256a, frequency is 104.02Mhz, 819.20 throughput, and 0.728 efficiency.
Chenh-hung Lin *et al*. [22]	Perfect hashing based parallel algorithms.	• Memory architecture was widely adopted for flexibility and scalability. • DRAM was used to reduce the delay so; it is operated effectively.	Less throughput and more area is required for entire architecture. The secured data was easy to access by hackers.	Memory size – 1485KB Memory efficiency – 8.12B Throughput- 14.89Gbps

VI. CONCLUSION

In this paper, different SHA algorithms such as SHA-0/1/2/3, SHA-224, SHA-256, SHA-384, and SHA-512 are explained. In SHA, X_t architecture designed by using optimal CSA and CLA adder to reduce the hardware utilization. The overall application and requirements of the hashing algorithms have been detailed in this paper. The security systems in communication field deals with so many problems such as less memory storage, less efficiency, less key security, less efficiency in encryption and decryption also it requires more power and area for implementation. These all problems give an easy way for the hackers to steal/hack the users' information. In the future, a new hashing algorithm can be introduced with further optimal adder which will reduce both the area and power of the implementation.

REFERENCES

1. Athanasiou, G.S., Michail, H.E., Theodoridis, G. and Goutis, C.E., 2014. Optimising the SHA-512 cryptographic hash function on FPGAs. *IET Computers & Digital Techniques*, *8*(2), pp.70-82.
2. Liu, X., Deng, C., Lang, B., Tao, D. and Li, X., 2016. Query-adaptive reciprocal hash tables for nearest neighbor search. *IEEE Transactions on Image Processing*, *25*(2), pp.907-919.
3. Rote, M.D., Vijendran, N. and Selvakumar, D., 2015, July. High performance SHA-2 core using the Round Pipelined Technique. In *Electronics, Computing and Communication Technologies (CONECCT), 2015 IEEE International Conference on* (pp. 1-6). IEEE.
4. Yang, Y., He, D., Kumar, N. and Zeadally, S., 2018. Compact Hardware Implementation of a SHA-3 Core for Wireless Body Sensor Networks. *IEEE Access*, *6*, pp.40128-40136.
5. Sugier, J., 2017. Simplifying FPGA implementations of BLAKE hash algorithm with block memory resources. *Procedia Engineering*, *178*, pp.33-41.
6. At, N., Beuchat, J.L., Okamoto, E., San, I. and Yamazaki, T., 2017. A low-area unified hardware architecture for the AES and the cryptographic hash function Grøstl. *Journal of Parallel and Distributed Computing*, *106*, pp.106-120.

7. Bayat-Sarmadi, S., Mozaffari-Kermani, M. and Reyhani-Masoleh, A., 2014. Efficient and concurrent reliable realization of the secure cryptographic SHA-3 algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *33*(7), pp.1105-1109.
8. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K. and Verbauwhede, I., 2013. Spongent: The design space of lightweight cryptographic hashing. *IEEE Transactions on Computers*, *62*(10), pp.2041-2053.
9. Chen, S., Li, B. and Zhou, C., 2018. FPGA implementation of SRAM PUFs based cryptographically secure pseudo-random number generator. *Microprocessors and Microsystems*, *59*, pp.57-68.
10. Sang-Hyun Lee, Kyung-wook shin. An efficient implementation of SHA processor including three hash algorithms (SHA-512, SHA-512/224, SHA-512/256) International Conference on Electronics, Information, and Communication (ICEIC)2018
11. Omran, S.S. and Jumma, L.F., 2017, May. Design of multithreading SHA-1 & SHA-2 MIPS processor using FPGA. In *Information Technology (ICIT), 2017 8th International Conference on* (pp. 632-637). IEEE.
12. Li, M., Xu, J., Yang, X. and Yang, Z., 2009, July. Design and implementation of reconfigurable security Hash algorithms based on FPGA. In *Information Engineering, 2009. ICIE'09. WASE International Conference on* (Vol. 2, pp. 381-384). IEEE.
13. Aradhana, Dr.S.M. Ghosh, Review paper on secure hash algorithm with its variants, international journal of technical innovation in modern engineering and science, volume 3, issie 05, may 2017.
14. Ibrahim, R., Hussain, A. and Kadhim, R.A., 2015. Implementation Of Secure Hash Algorithm SHA-1 By Labview. International Journal of Computer Science and Mobile Computing, 4(3).
15. Sugier, J., 2017. Simplifying FPGA implementations of BLAKE hash algorithm with block memory resources. *Procedia Engineering*, *178*, pp.33-41.
16. Luo, P., Athanasiou, K., Fei, Y. and Wahl, T., 2018. Algebraic Fault Analysis of SHA-3 under Relaxed Fault Models. *IEEE Transactions on Information Forensics and Security*.vol13, no 7, pp1752-1761
17. Paul, R., Chakrabarti, A. and Ghosh, R., 2016. Multi core SSL/TLS security processor architecture and its FPGA prototype design with automated preferential algorithm. *Microprocessors and Microsystems*, *40*, pp.124-136.
18. García, R., Algredo-Badillo, I., Morales-Sandoval, M., Feregrino-Uribe, C. and Cumplido, R., 2014. A compact FPGA-based processor for the Secure Hash Algorithm SHA-256. *Computers & Electrical Engineering*, *40*(1), pp.194-202.
19. Wang, J., Luo, C.L., Lin, S.Z., Zhang, H.F., Cui, K., Liang, H., Jin, G., Zhou, L. and Chen, T.Y., 2013. Research of hash-based secure key expansion algorithm for practical QKD. *Optik-International Journal for Light and Electron Optics*, *124*(15), pp.2273-2276.
20. Michail, H.E., Athanasiou, G.S., Theodoridis, G. and Goutis, C.E., 2014. On the development of high-throughput and area-efficient multi-mode cryptographic hash designs in FPGAs. *Integration, the VLSI Journal*, *47*(4), pp.387-407.
21. Algredo-Badillo, I., Feregrino-Uribe, C., Cumplido, R. and Morales-Sandoval, M., 2013. FPGA-based implementation alternatives for the inner loop of the Secure Hash Algorithm SHA-256. *Microprocessors and Microsystems*, *37*(6-7), pp.750-757.
22. Lin, C.H., Li, J.C., Liu, C.H. and Chang, S.C., 2017. Perfect hashing based parallel algorithms for multiple string matching on graphic processing units. *IEEE Transactions on Parallel and Distributed Systems*, *28*(9), pp.2639-2650.

AUTHORS PROFILE

Madhukar M is an Assistant Professor of Electronics and Communication Engineering at Jyothy Institute of Technology, Bengaluru. His research interests include cryptography, image processing and FPGA.

Nagesh Kumar D N is an Assistant Professor of Electronics and Communication Engineering at Jyothy Institute of Technology, Bengaluru. His research interests include embedded systems, FPGA and machine learning.

Dr. M. C. Hanumantharaju is a Professor of Electronics and Communication Engineering at BMS Institute of Technology and Management, Bengaluru, India. He has authored two books and 50 technical articles in refereed journals and proceedings such as IEEE, Intelligent Systems, Particle Swarm Optimization, etc. He is currently serving as reviewer for IEEE Transaction on Industrial Electronics, Computers and Electrical Engineering Journal, Journal of Microscopy and Ultrastructure, etc. His research interests include, Design of hardware architectures for signal and image processing algorithms, computer vision, Register Transfer Level (RTL) verilog coding, synthesis and optimization of Integrated Circuits (ICs), FPGA/ASIC Design.

B M Chandrashekar is an Assistant Professor of Electronics and Communication Engineering at Jyothy Institute of Technology, Bengaluru. His research interests include energy audit and conservation, instrumentation and power electronics.

Kajol R was a student in Electronics and Communication department at Jyothy Institute of Technology, Bengaluru. Her research interests include Embedded Systems, Holography and cryptography.