Implementation of Secure Hash Function-3 Algorithm for High Speed and Throughput

Smriti[1] Prof. Anshuj Jain[2]
[1]M.Tech. Student [2]Associate Professor
[1,2]Department of Electronics & Communication Engineering
[1,2]Scope College of Engineering, Bhopal, (MP) India

Abstract— 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 present 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. attacks.

Keywords: Secure, Hash-3. Password, Storage, Xilinx, ISE

I. INTRODUCTION

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

1) 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.
2) 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.

Fig. 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.

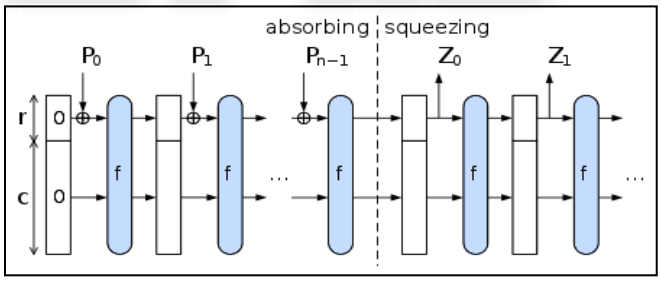


Fig. 2: HASH-3

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)

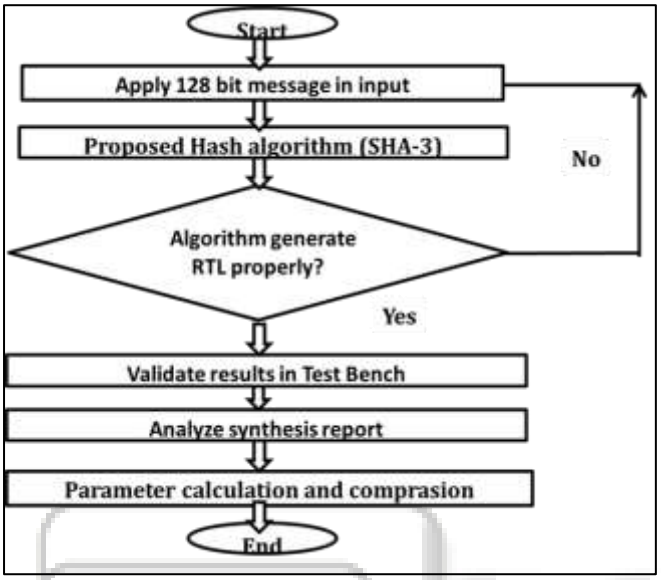
A= 67452301, B = EFCDAB89
C= 98BADCFE, D= 10324576

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. PROPOSED WORK



A. 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.

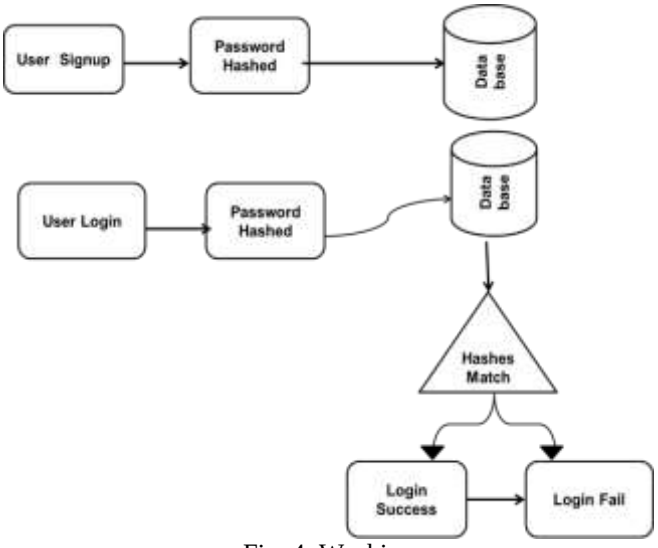


Fig. 4: Working

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.)

III. SIMULATION RESULT

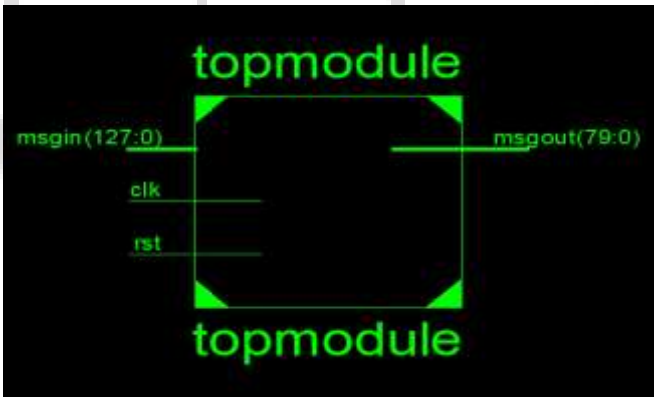


Fig. 5: Top view of proposed model

This figure 5 is showing top level module of proposed secure hash algorithm-3. In which apply 128 bit data and it generate 80 bit hash output.

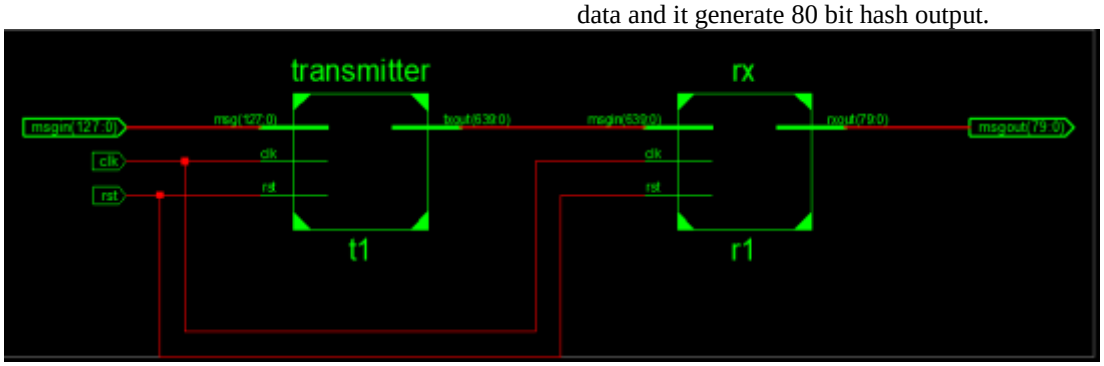


Fig. 6: RTL view of proposed Block diagram

Figure 6 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.

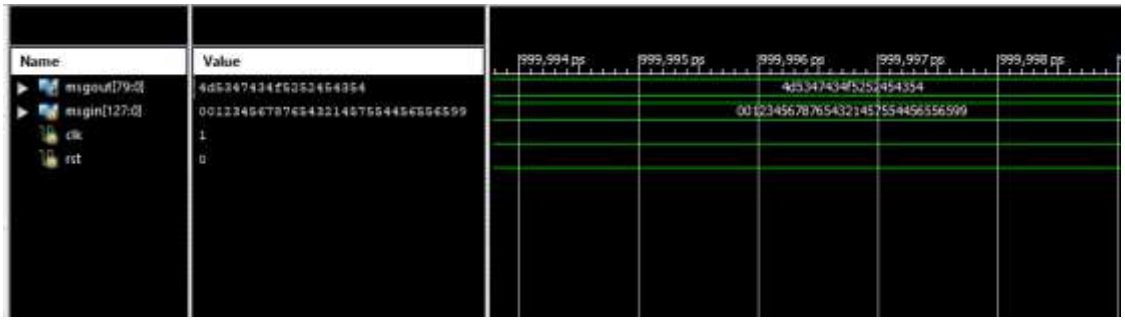


Fig. 7: Result validation in test bench

Figure 7 showing 128 bit message in input and it generate 80 bit at output after reduction of bits in modified secure hash algorithm-3 function.

Fig. 8: Test bench result for FPGA output

Sr No.	Parameter	Previous Work	Proposed Work	Improvement in Percentage
1	Method	SHA-3	Modified SHA-3	NA
2	Area (mm^2)	57.6	7.5	80%
3	Delay (ns)	24	3.259	70%
4	Power (mW)	80	41	50%
5	Time (secs)	87.31	42.48	55%
6	PDP	1920	133.61	NA
7	Frequency (MHz)	200	307.6	52%
8	Throughput (GHz)	4.8	4.92	2.43%

Table 1: Simulation Parameter and Comparison with previous work

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

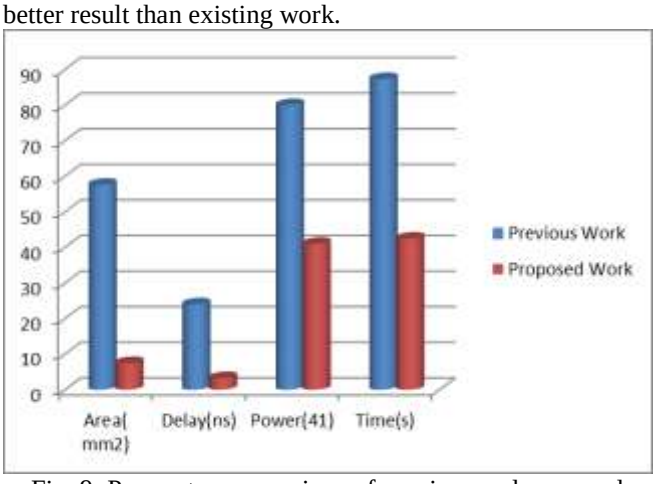


Fig. 9: Parameter comparison of previous and proposed work

Figure 9 presents comparison graph of area, delay, power and time of previous and proposed work. It i s clear that proposed work achieves better performance than previous work.

IV. 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] 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] A. Aghaie, M. M. Kermani and R. Azarderakhsh, "Design-for-Error-Detection in Implementations of Cryptographic Nonlinear Substitution Boxes Benchmarked on ASIC," 2018 IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS), Windsor, ON, Canada, 2018, pp. 574-577.

[3] J. Thesing and D. Kudithipudi, "Secure Neural Circuits to Mitigate Correlation Power Analysis on SHA-3 Hash Function," 2018 31st International Conference on VLSI Design and 2018 17th International Conference on Embedded Systems (VLSID), Pune, 2018, pp. 161-166.

[4] X. Qiuyun, H. Ligang, L. Qiming, G. Shuqin and W. Jinhui, "The Verification of SHA-256 IP using a semi-automatic UVM platform," 2017 13th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), Yangzhou, 2017, pp. 111-115.

[5] I. A. Landge and B. K. Mishra, "Hardware based MD5 implementation using VHDL for secured embedded and VLSI based designs," 2016 International Conference on Communication and Electronics Systems (ICCES), Coimbatore, 2016, pp. 1-6.

[6] I. Verbauwhede, "VLSI design methods for low power embedded encryption," 2016 International Great Lakes Symposium on VLSI (GLSVLSI), Boston, MA, 2016, pp. 7-7.

[7] M. D. Rote, Vijendran N and D. Selvakumar, "High performance SHA-2 core using the Round Pipelined Technique," 2015 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Bangalore, 2015, pp. 1-6.
[8] 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.
[9] B. Alomair and R. Poovendran, "E-MACs: Toward More Secure and More Efficient Constructions of Secure Channels," in IEEE Transactions on Computers, vol. 63, no. 1, pp. 204-217, Jan. 2014.
[10] M. Zhang, M. M. Kermani, A. Raghunathan and N. K. Jha, "Energy-efficient and Secure Sensor Data Transmission Using Encompression," 2013 26th International Conference on VLSI Design and 2013 12th International Conference on Embedded Systems, Pune, 2013, pp. 31-36.
[11] A. Bogdanov, M. Knezevic, G. Leander, D. Toz, K. Varici and I. Verbauwhede, "SPONGENT: The Design Space of Lightweight Cryptographic Hashing," in IEEE Transactions on Computers, vol. 62, no. 10, pp. 2041-2053, Oct. 2013.
[12] I. Algredo-Badillo, M. Morales-Sandoval, C. Feregrino-Uribe and R. Cumplido, "Throughput and Efficiency Analysis of Unrolled Hardware Architectures for the SHA-512 Hash Algorithm," 2012 IEEE Computer Society Annual Symposium on VLSI, Amherst, MA, 2012, pp. 63-68.
[13] N. Sklavos, "Multi-module Hashing System for SHA-3 & FPGA Integration," 2011 21st International Conference on Field Programmable Logic and Applications, Chania, 2011, pp. 162-166.
[14] A. Shahmoradi and M. Masoumi, "A new nanoelectronic based approach for efficient VLSI realization of SHA-512 algorithm," IEEE EUROCON 2009, St.-Petersburg, 2009, pp. 1206-1213.

