Design of FPGA Based Encryption Algorithm using KECCAK Hashing Functions

Deepthi Barbara Nickolas[#1], Mr. A. Sivasankar*[2]

 # PG Scholar, Department of ECE, Anna University: Regional Center, Madurai, Tamilnadu, India

*****Assistant professor, Department of ECE, Anna University: Regional Center, Madurai, Tamilnadu, India*

***Abstract* —Security makes the people to stay in the sense of vital modern technological improvements, especially focused in Cryptography process. We have to consider the high level of security, the speed of encryption and the level of hardware which decides the cost in trade basis. The work in this paper concentrates on KECCAK SHA-3 algorithm and the sponge construction encryption process with iterative permutation. Thus it leads to reduced amount of encryption time and more importantly provides the ultimate security level among the previous techniques followed so far. The algorithm uses the hashing function which is used for secured message authentication of data, digital signatures and password protection. The main advantage of this algorithm is it exhibits high level of parallelism. The hardware implementation process on FPGA is very fast and effective.**

***Keywords* — . Hashing, SHA-3 Algorithm, Sponge function, Encryption, Permutation.**

I. INTRODUCTION

In recent days security is a big threat in the transmission medium due to the development of the Internet and multimedia contents such as audio, image, video, etc. It enables us to easily purchase digital contents via the net. However, it causes several problems, such as violation of ownership and illegal distribution of the copy. The basic idea followed is based on cryptography technique. The technique is based on hashing function.

A cryptographic hash function is a deterministic procedure which takes an arbitrary size message and returns a fixed length bit string named message digest. This is a one way operation. It has been used in many security applications, such as digital signature and Message Authentication Codes (MACs) and password protection of files. The growing demand of secure high-speed digital communications has demanded the inclusion of cryptographic primitives into system design. One of the most important requirements to be satisfied in secure communications is data integrity and data origin authentication.

Cryptography is becoming an essential part of most electronic equipments that require data storing or manipulation. The algorithms used to enforce this security are too demanding to be implemented in software for the current required processing speeds. To achieve the required capability hardware components have to be used. In our analysis we are using FPGA for hardware implementation.

A cryptographic hash function should also be highly sensitive to the smallest change in the input message. A small change in single digit in the input message should produces a large change in the output hash value of the message. The message can be a binary text file, audio file, or executable program.

The security of the hash function does not originate in keeping the hash function itself secret but comes from its ability to produce one-way hash values with the property of being collision-free. When two or more different messages results in the exact same hash value collision happens. So far we have talked about hash functions used without a key, but hash functions can be used with a key; both symmetric and asymmetric keys can be used; in which case the function is called a message authentication code or MAC.

Data hiding is referred to as a process to hide data into cover media. This implies that the data hiding process links two sets of data, a set of embedded data and another set of cover media data. These two sets of data have different applications. As a popular and effective means of privacy protection, encryption process converts the ordinary signal into unintelligible data, so the traditional signal processing steps takes place before encryption or after decryption.

In [1], an interactive buyer–seller protocol for invisible watermarking, the seller does not know the exact watermarked copy that the buyer receives. So the seller cannot create copies of the original content containing the buyer's watermark. If the seller finds an unauthorized copy, the seller can identify the buyer from the watermark using the unauthorized copy, and hence the seller can prove this fact to a third party using a dispute resolution protocol. In [2] it is desired to transmit redundant data over an insecure and bandwidth-constrained channel, it is desired to first compress the data and then encrypt it. Traditional techniques used to compress the data first and then encrypt the data. But we are reversing the order of these steps, thereby first encrypting and then compressing the data. A significant compression ratio

can be achieved if compression is performed after encryption. Compression is performed by standard source coder and decryption by decompression. In [3] we propose a new fingerprinting protocol applying additive homomorphic property of Okamoto–Uchiyama encryption scheme. Exploiting the property, the enciphering rate of our fingerprinting scheme can be improved. The security can also be protected for both the buyer and the merchant, since the protocol is performed only between the buyer and the merchant which is similar to real-world market.

In [4] this scheme is proposed to implement commutative video encryption and watermarking during advanced video coding process. In [5] a new secure and fast hashing algorithm (SFHA) based on the generic 3C construction is used. The SFHA is an advanced version of SHA-256 which is more secure and faster than SHA-256. It is based on the generic 3C construction. The above construction is obtained by modifying the M-D iterated construction and it is more resistant against the recent differential multi-blocks attacks and the extension attacks. The generic 3C is a practical construction which preserves the collision resistance of the compression function in a much better way compared to the hash functions. SFHA-256 step function doesn't use Boolean functions but uses simple XOR, addition and shift rotation operations. These properties make it difficult for the attacker to analyze SFHA-256.

In [6] signal-processing modules works directly on encrypted data to provide a solution where valuable signals must be protected from a malicious processing device. Here, we investigate the implementation of the discrete Fourier transform (DFT) in the encrypted domain by using the homomorphic properties of underlying cryptosystem. In [7] this paper the hardware implementation of a symmetric-key authentication protocol in which a PUF is one of the relevant blocks. The second relevant block is a SHA-3 2nd round candidate, a Secure Hash Algorithm particularly KECCAK which has been proposed to replace the SHA-2 functions that have been broken. In [8] paper proposes a novel Coupled Integer Tent Mapping System-based cryptographic one-way hashing algorithm termed as THA. The integer tent map replaces the traditional logic functions as the major nonlinear component of compression function. The parallel iteration structure is adopted in the compression functions, which is helpful in high speed parallel operation of software and hardware.

In [9] we propose a novel hash-based approach for colour image steganography. The hash-based algorithms based approaches are considerably better in terms of providing better speed but these approaches are vulnerable in terms of providing security due to flaws caused by used checksum. In our approach, we propose the use of perfect hash function algorithm to provide a secure and fast approach for colour image steganography. In [10] efficient hardware implementation of cryptographic algorithms for a particular application Field Programmable Gate Array (FPGA) is used. In this paper the usage of the partial unrolling technique for designing and implementing hardware architectures for the SHA-512 hash algorithm, one of the most secure algorithm in the SHA-2 family is discussed. The main disadvantage of this technique is large amount of resource utilization even though throughput and efficiency are improved. Beyond a certain level of loop unrolling both throughput and efficiency are degraded. New improvements must be done to shorten the critical path to improve the efficiency and throughput.

In Advanced Encryption Standard the scheme uses AES encryption algorithm which uses the same keys for both encryption and decryption. AES is an iterated block cipher with a fixed block size of 128 and a variable key length. AES uses variable number of rounds, which are fixed. The key expansion algorithm is based on the assumption of a 128 bit key. AES calls for a larger number of rounds when we use a key length other than 128 bits.

Since this process has some disadvantages like usage of large power and area we are going to improve this process by a scheme known as keccak SHA-3 algorithm which can improve the security as well as throughput in terms of area and power.

II. PROPOSED METHODOLOGY

The proposed scheme uses Keccak SHA-3 algorithm which is based on hashing function. A cryptographic hash function is a deterministic procedure that can take strings of any length and return a string of a fixed size which should be ideally unique to the given input string. It also process arbitrary messages and produces fixed length output. The properties of cryptographic hash function are

1. Easy to compute.
2. Should be infeasible to compute a input string given a hash output string.
3. Should be infeasible to modify a input string without changing the hash output string.
4. Should be infeasible to find two different input strings that map to same hash output string.

A. Keccak algorithm

The keccak SHA-3 algorithm is based on 3rd round analysis of Keccak. The sponge construction is the heart of keccak SHA-3 algorithm. It is a sponge function with members KECCAK[r, c]. The parameters r = bit rate, c = capacity determine width of Keccak-f permutation, and is also applied to sponge construction.

The block diagram of keccak SHA-3 implementation is shown in figure.

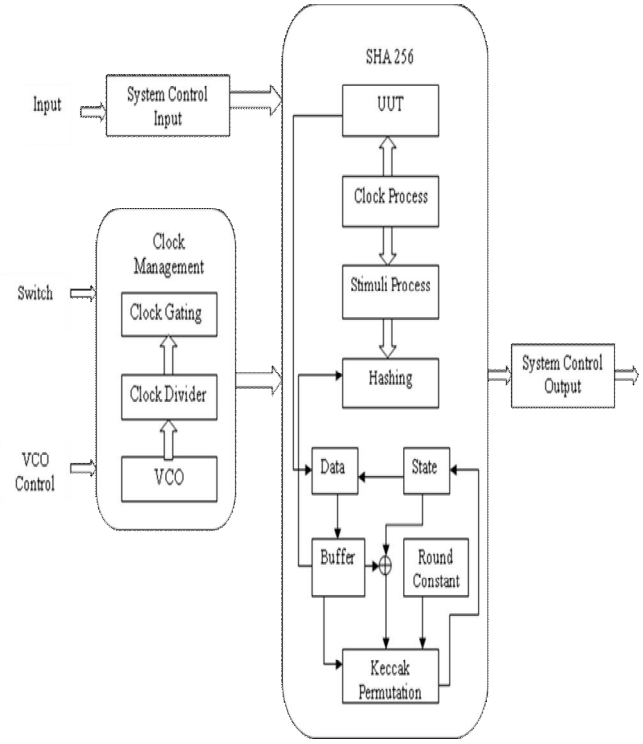

Fig. 1 Block diagram of SHA-3 encryption process

B. Sponge function

The security of a hashing algorithm is generally taken in terms of random oracle whose output has been truncated to desired n bits which implies that hash function has a security of $2^{n/2}$ for collision and 2^n for second preimage attacks. A sponge construction also referred as sponge function is closest approximation to a random oracle except for the side effects of finite memory or internal state collisions which are absent in a random oracle.

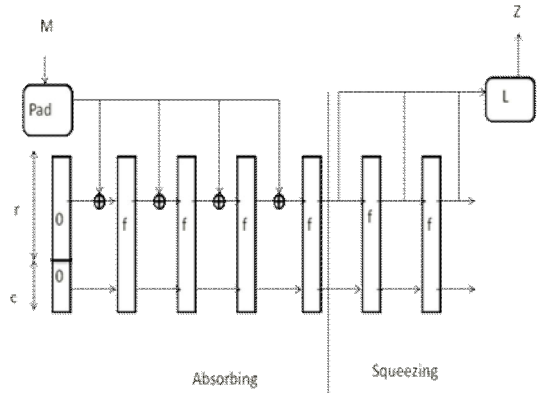

Fig 2. Sponge Construction

The sponge construction processes the message in two phases:

1) Absorption: The sponge state initially consists of all zeros. The first input block of length r is XORED with r bits of the state; and transform functions are applied on the state. Next input block is then XORED with this state like the previous one and transformed. This continues till all the input is consumed.

2) Squeezing: The first r blocks of the output are returned from this state and the transformations on the state are continued till all the blocks make for the output length required are obtained. The last c blocks are not directly output by the input or taken as output.

The sponge construction works on a state of b bits, which is split into two parts: the first part contains the first 'r' bits of the state called the outer part of the state and the second part contains the last c = b-r bits of the state called the inner part of the state.

Given a message, it is first padded and cut into r-bit blocks, and the b state bits are initialized to zero. The sponge construction processes the message in two phases: In the absorbing phase, all the message blocks are processed iteratively by XORing each block into the first r bits of the current state, and then applying a fixed permutation on the value of the b-bit state. After processing all the blocks, the sponge construction switches to the squeezing phase. First r bits of the state are returned as output, and then permutation is applied. This is repeated until n output bits are produced. The Keccak hash function uses multi-rate padding. Given a message, it first appends a single 1 bit and then it appends the minimum number of 0 bits followed by a single 1 bit, so the length of the result is a multiple of r. Thus, multi-rate padding appends at least 2 bits and at most r + 1 bits.

We can increase the security of the sponge transformations by increasing capacity and reducing the bit rate.

The Keccak permutation consists of 24 rounds, which operate on the 1600 state bits. Each round of the permutation consists of five mappings.

Keccak uses the following naming conventions, which are helpful in describing these mappings:

- A row is a set of 5 bits with constant y and z coordinates, i.e. a [*][y][z].
- A column is a set of 5 bits with constant x and z coordinates, i.e. a [x][*][z].
- A lane is a set of 64 bits with constant x and y coordinates, i.e. a [y][y][*].
- A slice is a set of 25 bits with constant z coordinate, i.e. a [*][*][z].

The main advantages of keccak are sponge function with thick safety margin. Reeseedable pseudo random bit generator can be used. Single permutation for all output length. On implementation software performance is good and excellent suitability in hardware in terms of area and speed. The hardware implementation is easier and cheaper which uses FPGA.

III. EXPERIMENTAL RESULTS

The simulation results for KECCAK SHA-3 algorithm is shown below

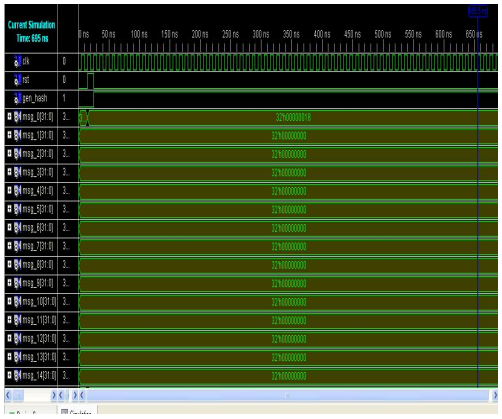

Fig. 3 Hash Generation

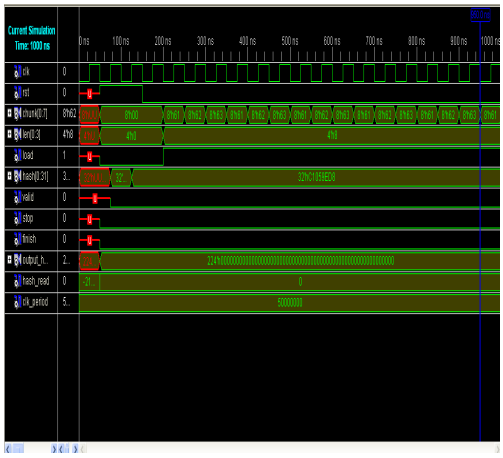

Fig. 4 Final Simulation output

IV. CONCLUSION

Using the symmetric algorithms for the encryption process generally increases the complexity of the hardware design. So with the use of advanced algorithms like KECCAK SHA3 not only improvise the secured cryptographic mechanisms but also reduces the hardware complexity. The sponge function implemented is of higher probability ratio and the analysis using the recurring iteration function is of higher functional value and has better encryption process. So the fast and effective mechanism using KECCAK hashing function gives better results in the cryptographic process.

REFERENCES

[1] N. Menon and P. W. Wong, “A buyer-seller watermarking protocol,” IEEE Trans. Signal Process.,vol.10,no.4,pp.643-649,Apr. 2001.

[2] M.Johnson, P. Ishawar, V.M. Prabhakaran, D. Schonberg, and K. Ramachandran, ”On compressing encrypted data,” IEEE Trans. Signal Process., vol. 52, no. 10, pp.2992-3006, Oct. 2004.

[3] M. Kuribayashi and H. Tanaka, ”Fingerprinting protocol for images based on additive homomorphic property,”IEEE Trans Image Process., vol. 14, no.12,pp. 2129-2139, Dec. 2005.

[4] S. Lian, Z. Liu, Z.Ren, and H. Wang,”Commutative encryption and watermarking in video compression,” IEE Trans. Circuit Syst. Video Technol., vol. 17,no. 6, pp. 774-778,Jun. 2007.

[5] Hassan. M. Elkamchouchi Mohamed E. Nasr Roayat Ismail Abdelfatah,” A new secure and fast hashing algorithm (SFHA-256)” 25th NATIONAL RADIO SCIENCE CONFERENCE (NRSC 2008)

[6] T. Bianchi, A. Piva, and M. Barni, “On the implementation of the discrete Fourier transform in the encrypted domain,” IEEE Trans. Inform. Forensics Security, vol. 4, no. 1, pp. 86-97, Feb. 2009.

[7] Susana Eiroa, Iluminada Baturone, “ Hardware authentication based on PUFs and SHA-3 2nd round candidates” IEEE ,2009.

[8] Liu Jian-dong, Tian Ye, Wang Shu-hong, Yang Kai, “A Fast New One-Way Cryptographic Hash Function”
IEEE 2010.

[9] Rubata Riasat, Imran Sarwar Bajwa, M. Zaman Ale, “A Hash-Based Approach for Colour Image
Steganography” IEEE 2011.

[10] Ignacio Algredo-Badillo, Miguel Morales-Sandoval, Claudia Feregrino-Uribe, Ren´e Cumplido, “Throughput and Efficiency Analysis of Unrolled Hardware Architectures for the SHA-512 Hash Algorithm” IEEE 2012.