# High Speed Architecture for KECCACK Secure Hash Function

Pasupuleti Sailaja
Department of ECE
Christ University
Bangalore, India

Mahendra Vucha
Department of ECE
Christ University
Bangalore, India

## ABSTRACT
Cryptography is a technique that protects the information, which is in transit or in storage, from unauthorized or unexpected reveals. This paper demonstrates a Secure Hash Algorithm-3(SHA-3) called Keccack, and also proposeda hardware architecture for the Keccack to support high speed security application. Since SHA-3 supports high level of parallelism, the proposed hardware architecture brings higher speed, in terms of bit rate and capacity, and also provides better security demanded by Internet of Things. This paper also demonstrates the architectural attributes of popular and real life cryptography techniques such as, Secure Hash Algorithm-1 (SHA-1), Secure Hash Algorithm-2 (SHA-2) and Advanced Encryption Standard (AES). In this research, the security techniques AES, SHA-1, SHA-2 and SHA-3 has been implemented on Virtex-5 FPGA device and their architectural attributes were captured. Finally, the proposed architecture of SHA-3 is compared with architecture of ontemporary security techniques(AES, SHA-1, and SHA-2) in terms of speed, area and power. The comparison results shown that the SHA-3 architecture brought optimum performance over its contemporary security techniques.

## Keywords
Internet of Things, Secure Hash Algorithm, Filed Programmable Gate Array.

## 1. INTRODUCTION
The cryptography is an art of twisting information into evident unintelligibility secret information. The essential service provided by cryptography is the capability to send information between participants in a way that avoid others from reading it [1] [2]. Initially, the cryptography includes techniques such as microdots, merging words with images, and other ways to hide information in transit or during storage. However, nowadays in computer centric world, the cryptography is most oftenly associates with scrambling plaintext into cipher text, a processcalled encryption, then back again known as decryption. Themoderncryptography systems are developed with the following constraints to support rapid development of internet technology and digital consumer product.

***Confidentiality:*** Information cannot be understood or recognized by anyone for whom it was unintended.

***Integrity:*** Information cannot be altered in storage or during transit between the sender and intended receiver without the alteration being detected

***Non-repudiation:*** The creator/sender of the information cannot deny at a later stage while creation or transmission of the information.

***Authentication:*** Authentication is a procedure in which the credentials offered are balanced to those in a database of endorsed users' information on a local functioning system or within an authentication server.

Cryptography is a science that relates complicated mathematics and logic to create robust encryption methods [3].ThrCryptography tolerates people to remain assurance in the electronic world and that helps the people to do their export on electronic channel without disturbance of presence and cheating. When people have started proceeding commerce and funds transfer through online or electronically, the purpose of cryptography for integrity start to exceed its use for confidentiality [4]. In today's world thousands of people interact electronically every day by various ways like E-mails, ATM machines, e-commerce and cellular phones. There are various cryptographic techniques presented in literature as stated in coming section.

The paper is organized as the literature review in the section 2, proposed architecture for cryptography technique SHA-3 is described in the section 3, implementation scheme of the methodology, experimental results and discussions in the section 4 and finally the paper is concluded in section 5.

## 2. LITERATURE REVIEW
This section demonstrates the state of art of cryptography techniques.

**Secret Key Cryptography** (SKC) is a cryptography technique uses single key for both encryption and decryption of the information, so the cryptography key should be known to both sender and receiver. The SKC also known as Symmetric Key Encryption, wherein the major risk is with distribution of the key to other end. Data Encryption Standard (DES) is a predominant SKC technique for encryption of electronic data [13] but the DES is considered as insecure for many applications due to 56-bit key size being too small. Advanced Encryption Standard (AES) is also a SKC technique populated as Rijndael, is a family of cipher with different key and block sizes. The AES has fixed block size of 128 bits and key sizes of 128, 192 or 256 bits. The variants of AES are demonstrated in [7] and the standards were implemented for hard disk security in [10].

**Public Key Cryptography** (PKC) is a cryptography technique [11] that uses asymmetric keys, one key for encryption of information at one end and different key for decryption of the information at other end. Because of the computational complexity of asymmetric encryption, it is typically used only for short messages i.e. transfer of a symmetric encryption key.

**Cryptographic Hash function** is a hash function [3] used to map the data of arbitrary size to data of fixed size. The values returned by hash function are called hash values or hash codes or simple hashes. The hash functions perform mathematical transformation iteratively to encrypt the information.

So, retrieving of input information from the hashes created by hash functions is practically impossible [4]. The ideal Cryptographic hash function would have the following four main properties.

1. Quick to compute the hash value for any given message.
2. Infeasible to generate a message from its hash.
3. Infeasible to modify the message without changing the hash.
4. Infeasible to find two different messages with the same hash.

National Institute of Standards and Technology (NIST) published the Secure Hash Algorithm (SHA) as a family of cryptographic hash functions and later it has released as SHA-0, SHA-1, SHA-2 and SHA-3. SHA initially published as 160 bit hash function SHA-0 and it has got withdrawn shortly due to an undisclosed significant flaw. Cryptographic SHA-0 weaknesses have been discovered in SHA-1 and the SHA-1 has become a part of digital signature algorithm but the SHA-1 has no longer applied for high secure cryptographic applications due to its short hash length. Later, SHA-2 has got released with similar hash functions with different block sizes 256 bits and 512 bits. These two SHA-2 versions differ in the word size i.e. SHA-256 uses 32-bit words whereas SHA-512 uses 64-bit words. A SHA-3 hash function formerly called *Keccak*has has released in 2012 after a public competition among non-NSA designers [12]. The cryptographic technique SHA-3 supports the same hash lengths as SHA-2 but its internal structure is differs significantly from the rest of the SHA family. The comparison chart of SHA cryptographic functions which was released by NSA is as described in table 1.

**Table 1: Comparison chart of SHA cryptograph techniques**

| Variants | Output size(bits) | Block Size(bits) | Number of Rounds |
|---|---|---|---|
| SHA-0 | 160 | 512 | 80 |
| SHA-1 | 160 | 512 | 80 |
| SHA-2 | 512 | 512 | 80 |
| SHA-3 | 512 | 1024 | 24 |

The SHA-3 algorithm use sponge construction encryption technique with iterative transformation. Thus, SHA -3 reduces number of encryption iterations and further significantly offers vital security level among the earlier techniques preceded so far [9][10]. The KECCAK algorithm employs the hashing function which is applied for secured message authentication of data, digital signatures and password protection. The major objective of offering the hardware estimation is to provide effective architecture that enhances the speed of the algorithm as proven in literature [16] [17] [18] [19] and also to find out the preeminent algorithm among them that will assure the new hashing algorithm principles. This kind of hardware implementations also provides assessment between each of the finalists with respect to security level, throughput, clock frequency, area, power consumption, and the cost.In this paper, we have demonstrated the behavior of SHA-3 and proposed architecture on reconfigurable device FPGA.

# 3. KECCACK (SHA-3) AND IT'S PROPOSED ARCHITECTURE

The SHA-3 is a family of sponge function characterized by two parameters called bit rate R and capacityC. The sum of R and C determines the hash width of the SHA-3 function permutation used in the sponge construction and are restricted to a maximum value of 1600 bits. Selection of R and C depends on the desired hash output value, i.e. for 256-bit hash output the bit rate R would be 1088 and capacity would be 512, similarly for 512-bit hash output R = 576 and C = 1024 is to be selected. The 1600 bit state 'A' of SHA-3 consists of a 5x5 matrix of 64-bit words as shown in Figure 1.

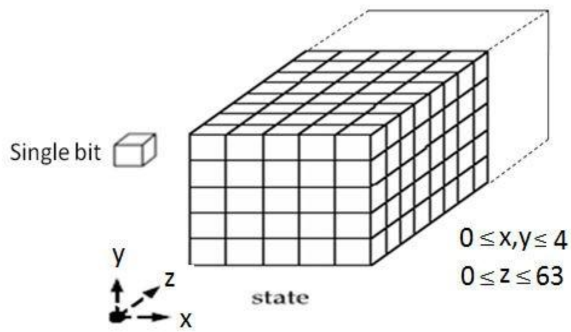


**Figure 1: State Matrix (A) of SHA-3**

The block diagram of SHA-3 consists of four functional blocks called state function, buffer function, and round constant and Keccack function as shown in figure 2.

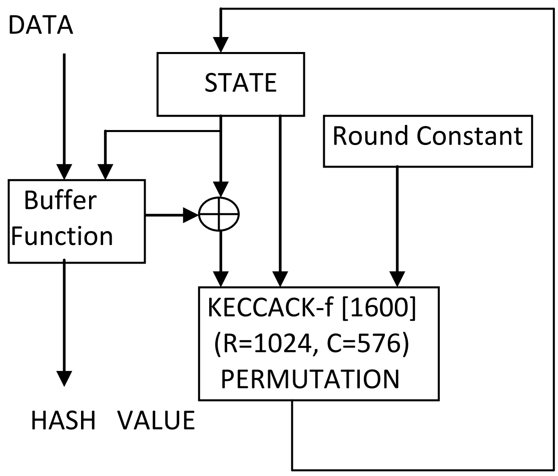


**Figure 2: Block diagram of SHA - 3**

The state function consists of 1600 state bits in the form of 5x5 matrix of 64-bit word in each element. The buffer function has two inputs one is data or information and other is from state matrix for generating random hash value. The buffer function takes 64 bits or 256 bits as data input and generate hash output based on sponge construction. The aims of using the sponge construction is to have a provable security against generic attacks and to make the use of compression function more simple, flexible, and functional. The sponge construction model contains two portions for the internal stage registers as shown in figure 3.

The operation of Keccak hash function comprised of three different stages i.e. first initialization followed by absorbing and finally squeezing stage as shown in Fig. 1, where M is the message and Z is the hash output. During initialization phase, every single bit of state matrix 'A' is initialized with zero.

In 2nd stage i.e absorbing stage, all r-bit wide block of messages is XORed with recent matrix state, in this way 24 rounds of Keccak permutation are accomplished sequentially.When absorbs every block of initial message taken as input block comes in that sequence.

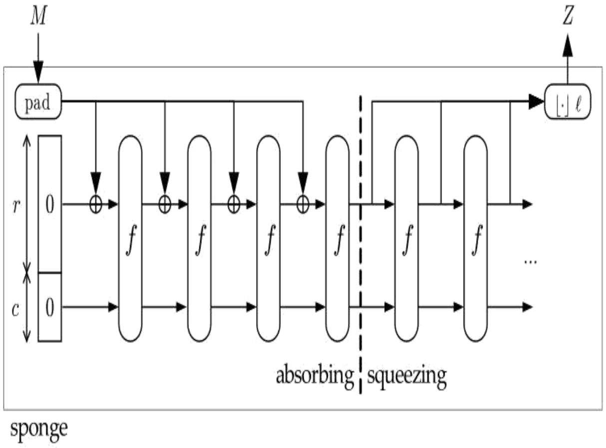


**Figure 3 : Sponge construction**

Sponge construction is an iterative construction designed to maps a variable length input to a fixed length output. The sponge function is a fixed-length transformation 'f' that operates on a fixed number of b = R + C bits. The data in sponge function can be processed in two phases called absorbed and squeezed.

**Absorbing phase:** The R-bit message blocks are XORed with the first R bits of the state of the function f and after processing all the message blocks, the squeezing phase starts.

**Squeezing phase:** The first R bits of the state are returned as output blocks of the function f.

The Round Constant (RC) function consists of 24 permutations values and that assign 64 bit data to KECCACK function. The RC values are represented in hexadecimal notation as shown table 2.

**Table 2: Round Constant values in hexadecimal**

| | | | |
|---|---|---|---|
| RC[0] | 0000000000000001 | RC[12] | 000000008000808B |
| RC[1] | 0000000000008082 | RC[13] | 800000000000008B |
| RC[2] | 800000000000808A | RC[14] | 8000000000008089 |
| RC[3] | 8000000080008000 | RC[15] | 8000000000008003 |
| RC[4] | 000000000000808B | RC[16] | 8000000000008002 |
| RC[5] | 0000000080000001 | RC[17] | 8000000000000080 |
| RC[6] | 8000000080008081 | RC[18] | 000000000000800A |
| RC[7] | 8000000000008009 | RC[19] | 800000008000000A |
| RC[8] | 000000000000008A | RC[20] | 8000000080008081 |
| RC[9] | 0000000000000088 | RC[21] | 8000000000008080 |
| RC[10] | 0000000080008009 | RC[22] | 0000000080000001 |
| RC[11] | 000000008000000A | RC[23] | 8000000080008008 |

The Kccack function KECCACK-f (1600) has 24 permutation rounds and each round consists of five sequential phases called ***Theta, Rho, Chi, Pi*** and ***Iota*** denoted as by $\boldsymbol{\theta}, \boldsymbol{\rho}, \boldsymbol{\pi}, \boldsymbol{\chi}$ *and* $\boldsymbol{\tau}$ respectively. In each round Round[b] (A, RC), the algorithm takes state matrix A and round constant as inputs and produce hash code output of size b for next round as shown in figure 2. The data flow of these five steps in KECCACK-f is demonstrated in the following expressions.

**Theta**

$C[x] = A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus A[x,3] \oplus A[x,4]$- (1)

$D[x] = C[x-1] \oplus rot(C[x+1],1), \forall\ in\ x = 0\ to\ 4$ - - (2)

Where $A[x,y] = A[x,y], \forall\ (x,y)\ in\ (0 \dots .4, 0 \dots .4)$

$A[x,y] = A[X,Y] \oplus D[x]\ \ \forall (x,y)\ in\ (0 \dots 4, 0 \dots 4)$ ----- (4)

**Rho and Pi**

$B[Y, 2X+3Y] = ROT(A[X,Y], r[X,Y]) 0 \leq X, Y \leq 4$ --- (3)

**Chi**

$A[x,y] = B[x,y] \oplus ((NOT\ B[x+1,y]\ AND$

$B[x+2,y] \forall (x,y) in (0 \dots 4, 0 \dots 4)$----------(4)

## Iota

$A[0,0] = A[0,0] \oplus RC\ , R[x]\ in\ 0 \dots .63$ -------------------- (5)

The hardware architecture of the function KECCACK is described in the following subsections.

**Theta:** The theta function reads the state matrix A as a two dimensional 5×5 state array and where each element in the array consists of a single word with w bits (in this paper the w is precisely equal to 64 bits). The architecture of θ function is shown in figure 4. The architecture of Theta function shown in figure 1 takes the state matrix A as input.

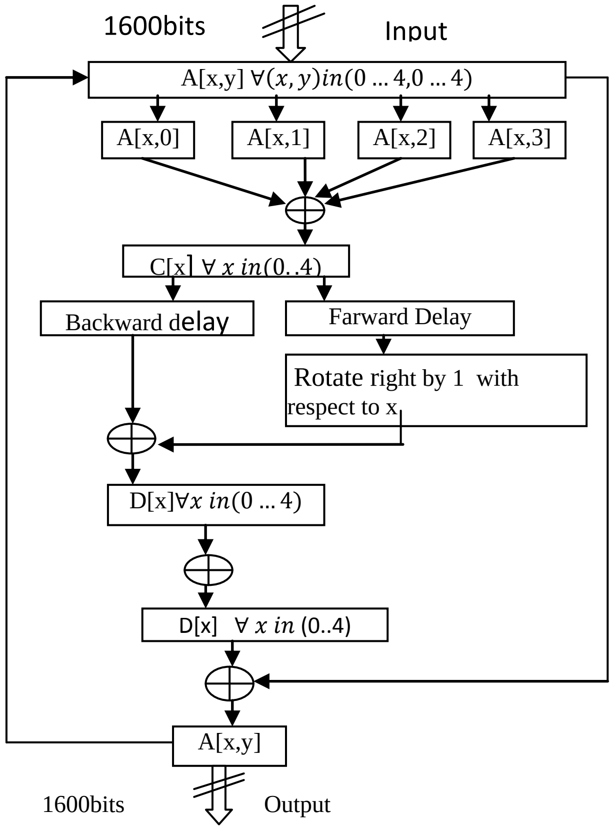


**Figure 4. Architecture of theta function**

Theta function comprises of three equations that involves simple XOR and bitwise cyclic shift operations. Equation (1) involves the XOR operation between lanes (set composed of 64-bits along the constant x and y coordinates) of each row of the state matrix A that results in five output lanes. Initially left circular shift will be applied on the five output lanes in such a way that last lane becomes first and second last lane becomes last lane and the behaviour is shown in equation (2). The resulted right circular shift will be carried out on the lanes so that first lane becomes the last and second lane becomes the first lane and then left circular shift will be applied on each lane in order to change the positions of the bits within each lane. The equation (3) of Theta just involves XORing between the input state matrix and output lanes obtained in equation (2).

**Rho and Pi :** The next two phases Rho (ρ) and Pi (π) are expressed jointly in equation (3) that computes an auxiliary 5x5 array, denoted as B, from the state array 'A'. The rchitecture of the phases Rho (ρ) and Pi (π) shown in figure 5.

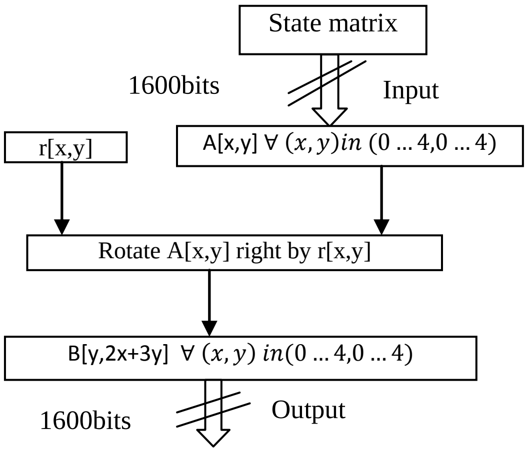


**Figure 5: Architecture of Rho and Pi functions**

The next two steps Rho ( ρ) and Pi (π) can be expressed jointly in equation (3) that compute an auxiliary 5 x 5 array B from the state array 'A'. The operation of Rho (ρ) and Pi (π) take each of the 25 lanes of the state array 'A',perform circular rotation on it by the fixed number of values depending upon the 'x' and 'y' co-ordinates i.e r[x, y] given in Table 2 (called Rho (ρ) step) and then place the above rotated lanes at the different location in the new array B . Note that all the indices are taken in modulo 5.

**Table 2: The Cyclic Shift Offsets "R(X,Y) for Keccak**

| r[x,y] | X=3 | X=4 | X=0 | X=1 | X=2 |
|---|---|---|---|---|---|
| Y=2 | 25 | 39 | 3 | 10 | 43 |
| Y=1 | 55 | 20 | 36 | 44 | 6 |
| Y=0 | 28 | 27 | 0 | 1 | 62 |
| Y=4 | 56 | 14 | 18 | 2 | 61 |
| Y=3 | 21 | 8 | 41 | 45 | 15 |

***Chi (χ) :*** The Chi (χ) phase perform the operations on the lanes in B array obtained in the Rho (ρ) and Pi ( π) phases and storeback the output into the state array A as shown in figure 6.

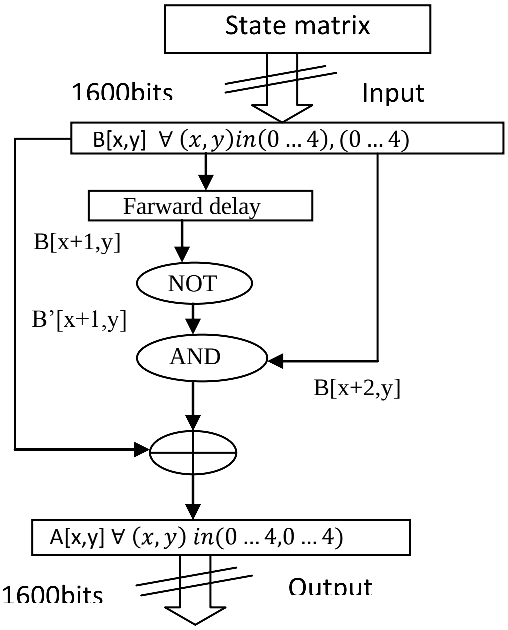


**Figure. 6. Architecture of Chi phase**

The Chi (χ) phase considers lane from state B[x,y] and perform the operations as stated in the expression (4). The Chi (χ) step operates on the lanes, i.e. words with 64-bits and manipulates the B array obtained in the previous Rho (ρ) and Pi ( π) step and replaces the result in the state array A. The Chi (χ) step takes the lane at location [x,y] and XOR it with the logical AND of the lane at address location of [x+1,y] and the complement at location [x+2,y].

**Iota:** The Iota is the last phase in Keccack function that computes the data as per expression (5) and its architecture is shown in figure 7.

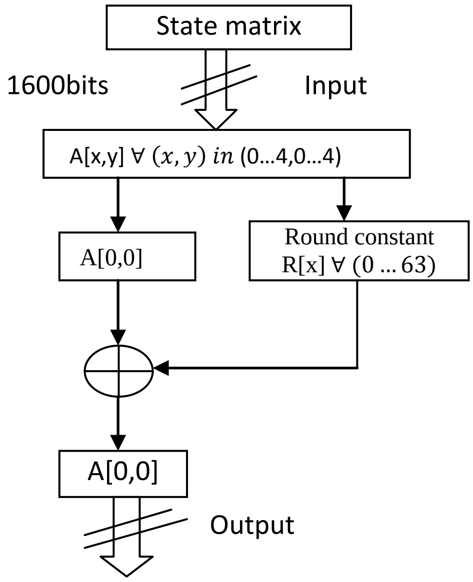


**Figure 7. Architecture of Iota phase**

These *Theta, Rho, Chi, Pi* and *Iota* in Keccack function provide the more security over its contemporary algorithm. In the coming section we would express the implementation scheme and experimental results for the architecture SHA-3.

## 4. RESULTS AND DISCUSSIONS

This section initially gives brief about implementation scheme and then discusses the experimental results.

*Implementation scheme:* The popular cryptography techniques in literature AES, SHA-1, SHA-2 and SHA-3 brought high security for the information in transit or in memory. Speed of these security techniques can be enhanced when they execute on FPGA. So, in this paper the proposed architecture for SHA-3 is implemented on FPGA by using Xilinx ISE hardware design process and that gives the hardware design parameters i.e. attributes for SHA-3. The Xilinx ISE is a tool set that supports hardware design from application behavior description to configure the target device with application. The behavior of the cryptography algorithms in literature are described in Very large scale integrated circuit Hardware Description Language (VHDL), synthesized using Xilinx Synthesis Tool (XST) that converts the VHDL into gate level netlist and analyzed for timing and area requirements and then configured on FPGA device to capture the power requirements.

*Experimental results:* The cryptographic techniques AES, SHA-1, SHA-2 and SHA-3 are implemented on Virtex-5 FPGA and the design parameters such as speed, area and power have been acquired. The design parameters for the chosen techniques are described in table 3.

**Table 3: Design parameters of AES, SHA-1, SHA-2 and SHA-3**

| Cryptographic technique | Number of bit slices | Speed (MHZ) | Power (Watt) |
|---|---|---|---|
| AES | 2208 | 115. 34 | 1.190 |
| SHA-1 | 1722 | 130. 70 | 1.084 |
| SHA-2 | 1687 | 111. 35 | 1.068 |
| SHA-3 | 1296 | 215. 14 | 1.047 |

z

The table 3 demonstrates the comparison chart of popular security techniques. From the comparison chart, the SHA-1 has better architecture in terms of area over contemporary techniques whereas the SHA-3 architecture brings superior performance in terms of speed and power consumption. The histogram of the area, speed and power of cryptographic techniques are shown in figure 8, figure 9 and figure 10 respectivelly.

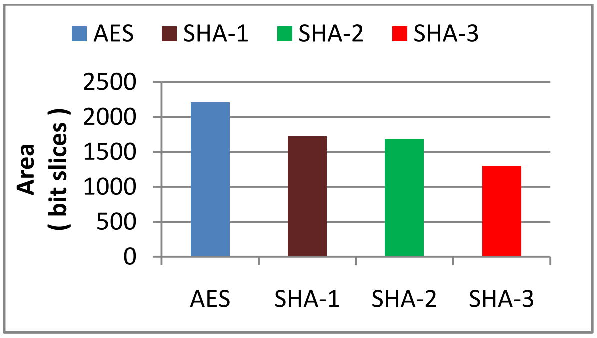


**Figure 8. Area requirement of cryptographic techniques**

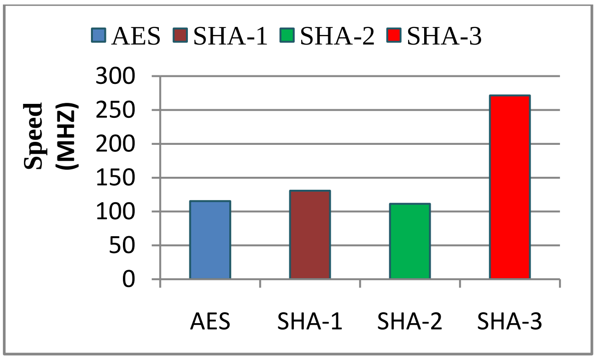


**Figure 9. Speed of cryptographic techniques**

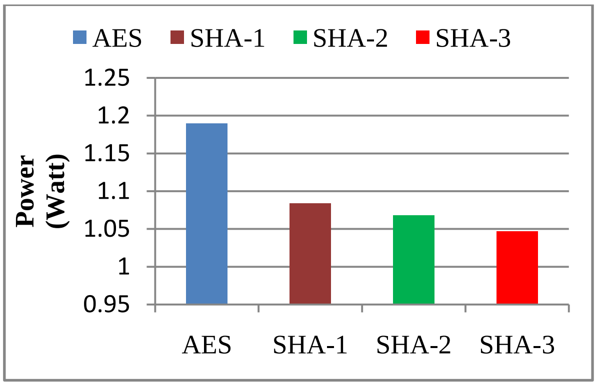


**Figure 10. Power consumption of cryptographic techniques**

The figure 8, figure 9 and figure 10 demonstrates that the Keccack cryptographic technique consumes less area and runs at optimized speed compared to contemporary cryptographic techniques but the Keccack algorithm consume almost same power of contemporary algorithms.

## 5. CONCLUSION

This paper presented a architecture for SHA-3 cryptography algorithm that runs at speed of 215.14MHz which is almost twice of the contemporary cryptography techniques (AES, SHA-1 and SHA-2) due to its parallelism nature and number of permutations. The proposed SHA-3 architecture also consumes half of the area consumed by contemporary cryptography techniques. Thus, the proposed architecture of SHA-3 technique brought optimum performance in terms of Speed, Area, and power.

## 6. REFERENCES

[1] Deepthi Barbara Nickolas and Mr. A. Sivasanka,"Design of FPGA Based Encryption Algorithm using KECCAK Hashing Functions**",**International Journal of Engineering Trends and Technology (IJETT) - Volume4 Issue6- June 2013

[2] S. Goldwasser, S. Micali and R. Rivest," A Digital Signature Scheme Secure against Adaptive Chosen-Message Attacks", SIAM Journal of Computing, vol 17, No. 2, pp. 281-308, April 1998.

[3] S. Haber and W. S. Stornetta. "How to timestampting a digital document". Journal of Cryptology 3(2), pp. 99-111, 1991.

[4] H. Krawczyk, M. Bellare and R. Canetti. "HMAC: Keyed- Hashing for Message Authentication". Internet RFC 2104, February 1997.

[5] V. Shoup. “Design and analysis of practical public-key encryption schemes secure against adaptive chosenCipher text attack”. SIAM Journal of Computing 33:167-226, 2003.

[6] Marc Stevens hash clash, “Framework for MD5 & SHA-1 Differential Path Construction and Chosen-Prefix Collisions for MD5”.

[7] X. Wang, H. Yu and Y.L. Yin, “Efficient Collision Search Attacks on SHA-0”, (Pub 2005)

[8] Xiaoyun Wang, X.L., Feng, D., Yu, H.: “Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD”. Cryptology ePrint Archive, Report 2004/199, pp. 1–4 (2004), http: // eprint.iacr.org/2004/ 199

[9] Szydlo, M.: “SHA-1 collisions can be found in 263 operations”. Crypto Bytes Technical Newsletter (2005).

[10] ChristofPaar, Jan Pelzl, "Introduction to Public-Key Cryptography", Chapter 6 of "Understanding Cryptography, A Textbook for Students and Practitioners". (companion web site contains online cryptography course that covers public-key cryptography), Springer, 2009.

[11] “Federal Register” / Vol. 72, No. 212 / Friday, November 2 (2007), Notices, http://csrc.nist.gov/groups/ST/hash/documents/FR_Notice_ Nov07.pdf

[12] National Institute of Standards and Technology (NIST).SHA-3 Winner announcement, http: // w ww. nis.gov /itl / csd / sha-100212.cfm.

[13] *Walter Tuchman "A brief history of the data encryption standard". Internet besieged: countering cyberspace scofflaws. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA. pp. 275–280.*

[14] J.-P. Aumasson, L. Henzen, W. Meier and R. C.-W. Phan, “SHA-3 Proposal BLAKE,” NIST (Round 3), University of California Santa Barbara, Santa Barbara, 2010.

[15] X. Guo, M. Srivastav, S. Huang, D. Ganta, M. B. Henry, L. Nazhandali and P. Schaumont, “Silicon Implementation of SHA-3 Finalists: BLAKE, Grøstl, JH, Keccak and Skein,” Center for Embedded Systems for Critical Applications (CESCA) Bradley Department of Electrical and Computer Engineering Virginia Tech, Blacksburg, 2010.

[16] Mahendra Vucha and Lincy Sara Varghese, “Design Space Exploration of DSP Techniques for Hardware Software Co-Design: An OFDM Transmitter Case Study,” International Journal of Computer Applications, Volume 116, Number 20, pp.29-33, April 2015.

[17] Sudeep M.C, Sharath Bimba M and Mahendra Vucha, “Design and FPGA Implementation of High Speed Vedic Multiplier,” International Journal of Computer Applications, Volume 116, Number 20, pp. 6-9, April 2014.

[18] Mahendra Vucha and ArvindRajawat, “Design and VLSI Implementation of Systolic Array Architecture for Matrix Multiplications,” International Journal of Computer Applications, Volume 26, Number 3, pp. 18-22, April 2011.

[19] Gurjar P., Solanki R., Kansliwal P, Vucha M., "VLSI implementation of adders for high speed ALU," 2011 Annual IEEE in India Conference (INDICON), , vol., no., pp.1-6, 16-18 Dec. 2011.

## 7. AUTHORS PROFILE

***Pasupuleti Sailaja*** received her MSc(Tech) in VLSI fromAndhra University in 2013. She is currently pursuing herM.Tech (Communication Systems) in Christ University,Bangalore. Her area of interests are VLSI and Cryptography.

***Mahendra Vucha*** received his B. Tech in Electronics & Communication Engineering from JNTU, Hyderabad in 2007 and M. Tech degree in VLSI and Embedded System Design from MANIT, Bhopal in 2009. He also received Ph. D degree in Electronic and Communication Engineering from MANIT, Bhopal (M.P), India. He is currently working as Asst. Prof in Department of Electronic and Communication Engineering at Faculty of Engineering, Christ University, Bangalore.His areas of interest are Hardware Software Co-Design, Analog Circuit design, Digital System Design and Embedded System Design.