FOREX Publication
Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

A Compact Hardware Design and Implementation on FPGA Based Hybrid of AES and Keccak SHA3-512 for Enhancing Data Security

K Janshi Lakshmi[1*] and G Sreenivasulu[2]

[1]*Research Scholar (Full Time), Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India; jansikaramala@gmail.com*
[2]*Professor, Department of Electronics and Communication Engineering, Sri Venkateswara University, Tirupati, Andhra Pradesh, India; gunapatieee@rediffmail.com*

***Correspondence:** K Janshi Lakshmi; jansikaramala@gmail.com

ABSTRACT- Data security means protecting important information from unauthorized persons. In a security system, cryptography is the most secure method. Cryptography has many kinds, but the Advanced Encryption Standard (AES) is the most secure system. If combined with AES and Secure Hash Algorithm-3-512Bits (SHA3-512), it becomes compact, more secure, and more authenticated for data communications. The proposed methodology is a hybrid cryptography technique that combines AES with the SHA3-512 algorithm. This system becomes a strong, secure system and produces a strong cipher text. The proposed method AES/SHA3-512 is Hardware implementation on the Artix-7 FPGA family yields the lowest cost and highest hardware efficiency with a reduction in area usage of LUT 18.49%, Flip Flops 0.83%, and IO 3.8%. The proposed architecture is synthesized and simulated using the Vivado 2017.2 version Tool.

Keywords: Data Security, Cryptography, AES, Keccak SHA3-512, FPGA Artix-7, Vivado

ARTICLE INFORMATION

Author(s): K Janshi Lakshmi and G Sreenivasulu
Received: 30/12/2023; **Accepted**: 22/02/2024; **Published**: 15/03/2024;
e-ISSN: 2347-470X;
Paper Id: IJEER 3012-18;
Citation: 10.37391/IJEER.120128
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-12/ijeer-120128.html

Publisher's Note: FOREX Publication stays neutral with regard to Jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Today, secure information transfer to third parties is necessary because hackers could steal the data. We use cryptography secure algorithms to prevent this kind of issue. In order to secure a network, cryptography algorithms are crucial. From *figure. 1*, In Cryptography network security technique has secure key and input data combined it become cipher text. This cipher text sends to channel. Due to this our information is securely transfer to others. The breadth of modern networks has expanded globally, and information has taken on a digital form. Data security has developed into a major issue as a result of the amount of data being communicated over the internet. Nowadays, sensitive information is hidden from an intrusive party using both symmetric (*Shown figure. 2.*) and asymmetric (*Shown figure. 3.)* Encryption techniques. Secret data's authenticity, non-repudiation, non-integrity, and confidentiality are all guaranteed by cryptography [4].

Hybrid cryptography is a combination of symmetric and asymmetric cryptography that aims to increase security by fusing the efficiency of symmetric encryption with the potential of asymmetric cryptography [4]. The combination of a hash function, an asymmetric algorithm and an AES, and symmetric algorithm is known as a hybrid technique.

Cryptography classifications: Two categories exist for contemporary cryptography: protocols for authentication and encryption, respectively. Two types of encryptions are standard: *(i)* public keys and *(ii)* private keys. There are two types of authentication protocols. Digital signatures and message authentication are two further ways to perform authentication. Two forms of message authentication *(I)* have a hash function: MD4, MD5, and SHA are a few examples. *(II)* MAC is an example of a message authentication code.

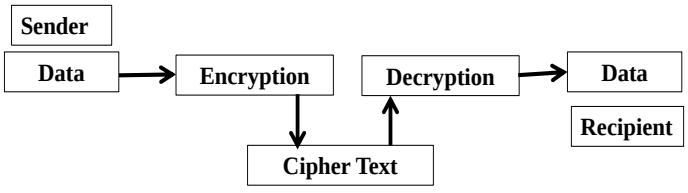

Figure 1. Cryptography Technique

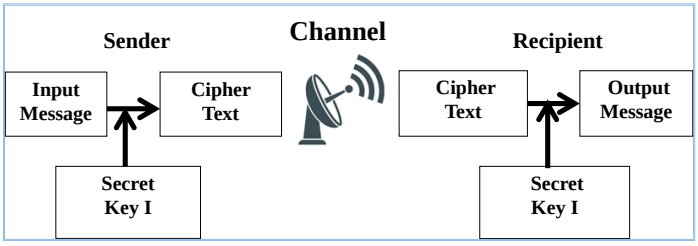

Figure 2. Symmetric Encryption Technique

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

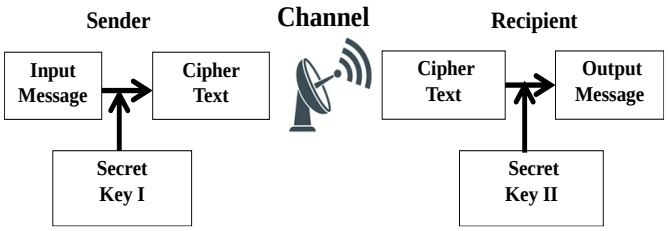

Figure 3. Asymmetric Encryption Technique

2. RELATED WORK

Sheng-Jung Yu et al. [4] were represented the first cryptographic processor for Internet of Things (IoT) devices that implements the double ratchet protocol with backward secrecy is presented in this study. The area is reduced by 39.5% and the energy consumption is decreased by 18.8% with the use of a pre-computation based constant modular divider. By taking advantage of the input's characteristics, the energy consumption of the length selector is proposed to be reduced by 89.8% and the energy consumption of the module by 35% by using a hash-based key derivative function (HKDF) module.

Noveline Aziz Fauziah et al. [8] to guarantee that ascend has also been thoroughly built, and its timing aspects are independent of the private data of its users. The security components (the ORAM controller, which consists of 12 AES rounds and one SHA-3 hash unit) impose a total area overhead of 0.51 mm2 in 32 nm silicon. At 857 MHz and 1.1 V upon tape-out, the security components of the Ascend chip have undergone satisfactory verification; at this frequency, they absorb 299 mW of power [15].

Huanyu Wang et al. [9] they recommended framework finds the flaws that might go against the design's SPs. This will lead to a significant reduction in overhead protection and increased effectiveness when using local countermeasures. The experimental findings implementations of RSA,SHA, and AES on the SP show that protecting the designated critical locations—which account for less than 0.6% of the design—is essential, can greatly reduce the security risk posed by fault-injection attacks.

William et al. [10] here, digital data is encrypted using a mathematical method called SHA-256. The integrity of the data is ensured by its use. It is claimed that the proposed hybrid approach is comparable to an existing method that encrypts text and images using the AES algorithm. When it comes to text encryption, the suggested strategy is more effective than the previously stated methods. When it comes to image encryption, the proposed method is not as effective as the existing one.

Wenjian Luo et al. [11] a password authentication framework that may be easily integrated into existing authentication systems is presented in this paper to enable secure password storage. First, in our architecture, the plain password received from a client is subjected to a cryptographic hash function (e.g., SHA-256). After that, a negative password is created by hashing the original password. Finally, an encrypted negative password (ENP) is produced by encrypting the negative password using a symmetric-key technique (like AES). To further boost security, multi-iteration encryption can be employed.

Kazim Yumbul, Erkay Savas [13] it was proposed that the most often used cryptographic approaches that are put into practice are elliptic curve cryptography, pairing-based cryptography, the AES block cypher, and the cryptographic hash algorithms SHA1 and SHA256. Their upgraded core software implementations of cryptographic algorithms, timing-wise, compare favorably to comparable software implementations on comparable CPUs that have been reported in earlier studies. Field-programmable gate array (FPGA) technology is used to implement the suggested protected zone improvements. The FPGA implementation results validate that its area overhead is generally small, meaning it may be applied to a wide range of embedded CPUs.

This work protects sensitive information from unauthorized access. It uses Advanced Encryption System (AES) and Secured Hash Algorithm (SHA) and combining them to form a hybrid cryptographic algorithm. SHA3-512 ensures data integrity and prevents tampering. The proposed system is later implemented in Artix-7 FPGA device. Through this methodology, significant reduction in resource usage particular in LUTs, flip-flops and IOs can be achieved.

3. PROPOSED METHODOLOGY

3.1 AES: Advanced Encryption Standard

One of cryptography's safest algorithms is AES. It is widely used in security system. The main applications are Military applications, Emails, Banks, Information technology of data base, secure vaccines data, etc. AES algorithm is Symmetric encryption. This block cipher is symmetric with configurable key sizes of 128bits or 192bits or 256 bits that allows data blocks of 128 bits. AES structures the information that it receives into states, which are two-dimensional arrays of 4x4 bytes containing an overall length of 16 bytes. AES has three categories based on key. 1) AES-128 has 128bits plain text, 128bits key and 10 rounds. 2) AES-192 has 128bits plain text, 192bits key and 12 rounds.3) AES-256 has 128bits plain text, 256bits key and 14 rounds displayed in *table. 1.*

Table. 1. AES Algorithm Specifications [12]

Specifications	AES-128	AES-192	AES-256
KeySize(Word#Bytes#Bits)	4#16#128	6#24#192	8#32#256
Block Size(Word-Bytes-Bits)	4#16#128	4#16#128	4#16#128
Rounds Number	10	12	14
Round keySize(Word#Bytes#Bits)	4#16#128	4#16#128	4#16#128
Expanded keySize(Word#Bytes)	44#176	52#208	60#240

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

3.1.1 AES Algorithm (Encryption and Decryption)

Sequences of 128 bits (digital values of 0s or 1s) are used for input and output. Blocks are another name for this. The secret key and input data are combined to form the encryption key. Sequences of 16bytes, 24bytes, and 32bytes make up the encryption keys.

Plain Text => 128bits/16Bytes
Key size => 128bits/16Bytes
Key size => 192bits/24Bytes
Key size => 256bits/32Bytes.

Table 2 shows the pattern arrangement of the binary and hexadecimal bits**.**

Table 2. Binary ~ Hexadecimal Bits Pattern

0000~0	0100~4	1000~8	1100~C
0001~1	0101~5	1001~9	1101~D
0010~2	0110~6	1010~A	1110~E
0011~3	0111~7	1011~B	1111~F

AES encryption typically involves four distinct byte-oriented transformations or processes, as seen in *figure 6.*

(i) SubByte Operation: A substitution table, or s-box, is used in this non-linear byte substitution step to operate independently on each byte in the state. The invertible s-box is generated using two transformations: an affine transform (AF) over GF and a multiplicative inverse (MI) in GF(28) with element (00) mapped to itself. Second, this stage of the decryption process's inverse transformation is called InvSubBytes.

(ii) ShiftRow Operation: The last three rows of the AES state are cyclically moved to the left by a predetermined number of bytes in Figure. 4 as part of a transposition step. The inverse of this is InvShiftRows, which is shifted cyclically to the right in the final three rows (*figure 5*) during the decryption process.

(iii) Mixcolumn Operation: Every column in the state is used by the randomization step called MixColumns. Each column is evaluated as a 4 term polynomial over GF(28) and multiplied by the constant polynomial d(x)=(03)x3 + (01)x2 +(01)x + 02.

$$\begin{bmatrix} s'0,d = ((02)\bullet s0,d) \oplus ((03)\bullet s1,d) \oplus s2,d \oplus s3,d \\ s'1,d = s0,d \oplus ((02)\bullet s1,d) \oplus ((03)\bullet s2,d) \oplus s3,d \\ s'2,d = s0,d \oplus s1,d \oplus ((02)\bullet s2,d) \oplus ((03)\bullet s3,d) \\ s'3,d = ((03)\bullet s0,d) \oplus s1,d \oplus s2,d\ ((02)\bullet s3,d) \end{bmatrix} \quad (1)$$

(iv)Addround Key Operation: A Key Scheduler generates each RoundKey from the input cypher Key, and AddRoundKeys adds the RoundKey to the state by bitwise XOR.

AES algorithm technique has 14 rounds for encryption and decryption shown in the *figure 4* and *figure 5*. Rounds number and rounds operations of both encryption and decryption shown in *table 3*.

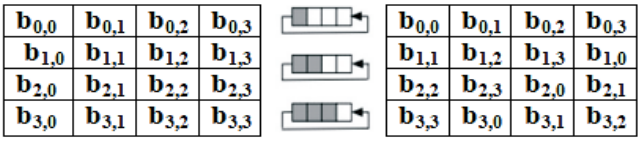

Figure 4. ShiftRow Operation

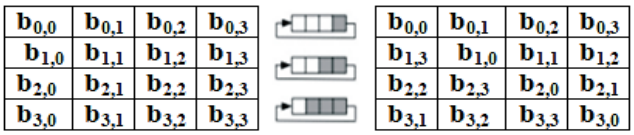

Figure 5. Inverse ShiftRow Operation

Table 3 depicts the round number operation of AES algorithm.

Table 3. Rounds operations of AES

Encryption		Decryption	
Round number	**Operations Name**	**Round number**	**Operations Name**
Round 1	Preround Subbyte Shift row Mixcoloumn Addroundkey	Round 1	Preround Inverse Subbyte Inverse Shift row Inverse Mixcoloumn Addroundkey
Round 2 to Round 13	Subbyte Shift row Mixcoloumn Addroundkey	Round 2 to Round 13	Inverse Subbyte Inverse Shift row Inverse Mixcoloumn Addroundkey
Round 14	Subbyte Shift row Addroundkey	Round 14	Inverse Subbyte Inverse Shift row Addroundkey

3.2 Secure Hash Algorithm-3(SHA-3)

In the field of cryptography, Secure Hash Algorithm-3, or SHA-3, is considered the standard. The Keccak algorithm using sponge function serves as its foundation. There are several SHA3 variations including the SHA3-224-bit, SHA3-256-bit, SHA3-384-bit and SHA3-512-bit, it is made up of several rounds, each of which contains certain logical processes. It basically produces the required output by employing the sponge function, which absorbs input first before pressing it to produce the intended output [6].

The sponge architecture serves as the foundation for the hash function family known as Keccak, which is also known as a

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

sponge function family. The underlying function of Keccak, termed Keccak-F[b], is a permutation selected from a set of seven Keccak-f permutations, where b is the permutation's breadth and values range from {25, 50, 100, 200, 400, 800, 1600}. The breadth of the state in the building of the sponge is equal to the width of the permutation [14]. The state is set up as an array of 5 x 5 lanes, each with a length of w 1, 2, 4, 8, 16, 32, 64, and b = 25w. A 64-bit CPU word can be used to represent a lane of Keccak-f[1600] when it is implemented on a 64-bit processor [3][14]. Figure 6 shows the encryption and decryption process of 128 bit plain text using a 256 bit key as given in [12].

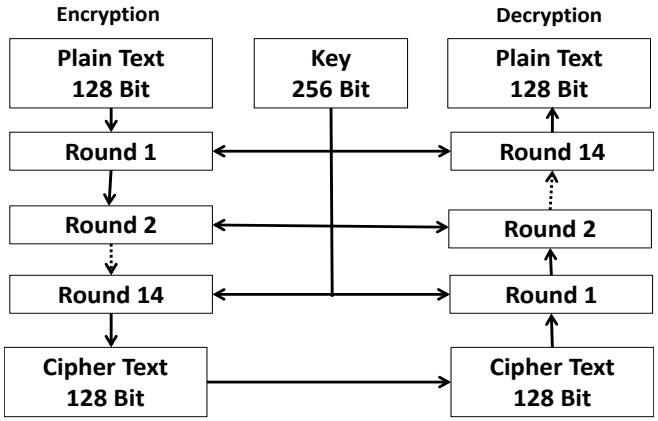

Figure 6. AES-256 Algorithm Encryption and Decryption [12]

Also, "b" = (r + c)-bit and is the result of combining two parameters, rate ("r") and capacity ("c"). The hash digest's intended length, or "h" bits, should be twice the value of "c," meaning that "c" = 2h. The three-dimensional matrix that makes up the Keccak-F[b] input state has dimensions of 5 5 w, where 'w' stands for lane length and is equivalent to 'b/25'. Additionally, it uses the round function (*figure. 7*, [5]) which is composed of five step mappings (θ, ρ, π, χ, ι).

SHA-3-224: hash is 224bits long.
SHA-3-256: hash is 256bits long.
SHA-3-384: hash is 384bits long.
SHA-3-512: hash is 512bits long.

Table 4: Different kinds of SHA Function Technique

Hash Function	Hash Function	Block (Bytes)	Rounds	Internal State Size(Bits)	Output (Bits)
SHA0	SHA0	64	80	160	160
SHA1	SHA1	64	80	160	160
SHA2	SHA224	64	64	512	224
	SHA256	64	64	512	256
	SHA384	128	80	1024	384
	SHA512	128	80	1024	512
SHA3	SHA224	144	24	1600	224
	SHA256	136	24	1600	256
	SHA384	104	24	1600	384
	SHA512	72	24	1600	512

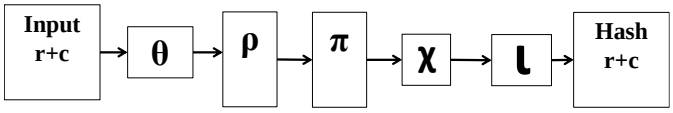

Figure 7. SHA-3(Secure Hash Algorithm-3) Round Function

M bits and *d* values are based on the assumption that the input to these functions consists of bytes shown in *table 5*.

Table 5. The parameters of SHA3 types [3]

Hash	*r*	*c*	Output length (bits)	Security level (bits)	M bits	d
SHA3-224	1152	448	224	112	1	00x06
SHA3-256	1088	512	256	128	1	0x06
SHA3-384	832	768	384	192	1	0x06
SHA3-512	576	1024	512	256	1	0x06

The algorithmic structure of SHA3 differs greatly from that of AES. $12+2(\log_2(b/25))$ is the formula used to determine how many rounds need to be run. Conversely, the intermediate results are B[m,n], C[x], and D[m], A[m,n,p] represents a specific state lane. The bitwise logical operations are XOR, NOT and AND, while with a value of r[x, y], Round constant (RC) is the same as bitwise cyclic shift operator (ROT) or rotate operator [5].

$\theta = (0 \le m, n \le 4, 0 \le p < w)$ (ii)
$C[m,n,p]=A[m,0,p]\oplus A[m,1,p]\oplus A[m,2,p](2)\oplus A[m,3,p]\oplus A[m,4,p];$ (iii)
$D[m,p] = C[(m-1),r] \oplus ROT(C[(m+1), 1]);$ (iv)
$A[m,n,p] = A[m,n,p] \oplus D[m,p];$ (v)
$\rho = (0 \le m, n \le 4)$ (vi)
$A[m,n,p] = ROT(A'[m,n,p],r[m,n,p]);$ (vii)
$\pi = (0 \le m, n \le 4, 0 \le p < w)$ (viii)
$B[n, (2m+3n),p] = A[m,n,p];$ (ix)
$\chi = (0 \le m, n \le 4, 0 \le p < w)$ (x)
$A[m,n,p] = B[m,n,p] \oplus (NOT(B[(m+1), n,p])$ (xi)
$AND(B[(m+2), n,p]));$ (xii)
$\iota = (0 \le z < w)$ (xiii)
$A^r[0, 0,p]=A[0, 0,p]\oplus RC[p];$ (xiv)

3.3 Proposed Method: hybrid and sharing of AES with SHA3-512

Look-Up-Table Section: Firstly, as illustrated in *figure. 8*, a merge AES encryption section and decryption section core is built. First, we reorganize and combine the ShiftRows and InvShiftRows transforms. Next, we create an Integrated LUT that combines the most computationally demanding transformations SubByte, MixColumn and InvSubByte, InvMixColumn of respective AES encryption section and decryption section into a single, unified Look-Up-Table see the *figure 8*. This LUT is intended for Block Rams (BRAMs) of FPGA.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

Unified XOR Section: After that, use of the Six Input Equation (SixIE) optimization method which is a logical optimization methodology applied to the SHA-3 algorithm see the *figure 8*. Where in the chosen step mappings (θ -3, ρ, π and χ; except ι) of SHA-3 are rationally merge into a single equation with '6' inputs, the first two θ transform equations, denoted as θ -1 & θ -2, remain unchanged.

SixIE Network: Later, the architectural similarities between SHA-3 and AES enc/dec in order to develop a resource shared hardware unit. These are the θ transform of SHA-3's first two equations, θ -1 & θ -2.

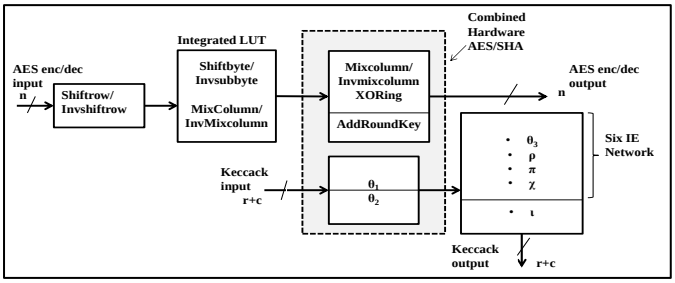

Figure 8. Combined or hybrid block diagram of AES/SHA-3-512

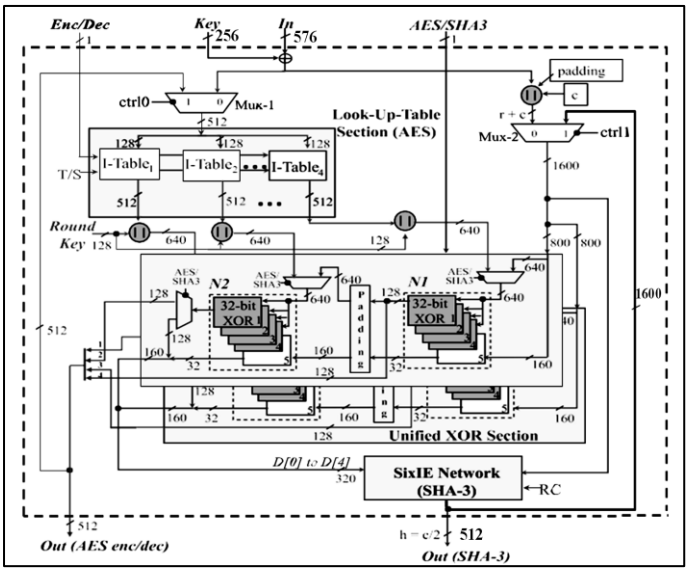

Figure 9. Hybrid Circuit diagram of AES/SHA-3-512 and Resource shared

The proposed methodology is Logical optimization of '6' Input Equation (SixIE) for SHA-3, integration of LUT (I Tables) for AES encryption/decryption, and a Unified EX-OR Section to perform key whitening in AES algorithm and SHA-3 conversions shows in *figure 9*. The proposed Module is easily extensible to SHA-3, which is the Keccak-f [1600] variation of the Keccak function. Depending on the desired security levels, it can support AES-256. The internal state wise sizes of 1600bits and number of rounds for SHA3-512 will also stay the same. The only things that will change are the 'c' bit and the resultant messages digest.

Table 6 is the input and control signals for AES/SHA3-512.

The architecture can process four simultaneous blocks of 128-bit inputs to provide an output of 512 bits for AES Encryption/decryption, or it can produce a 576-bit hash digest if SHA3 mode is used, such as in SHA3-512, which has a capacity of 512 bits/1024 bits. In the SHA-3-512 mode c-bit creates an internal state wise size of 1600bits for SHA-3-512 after being concatenated with the input.

When processing I-*table 4* data, N_1 of the first set of XOR networks generates a 128-bit output for AES, and N_2 processes I-*table 2* data to provide a 128-bit output, which together make up the final 256bit AES algorithm output. Analogously, for SHA-3-512 employing Keccak-f[1600], N_1 of the initial EX-OR network processes 50% of the SHA-3-512 state to get the 160-bit output of (2). The first 160-bit output of (3) is ultimately produced by applying the resulting 160-bit output to N2 following its passage through the Padding module. D[0] to D[4] see the *figure 9*, and the remaining 256bit AES algorithm outputs and 160bit θ-2 output of SHA-3-512, will be produced by the second set of XOR networks[5].

Table 6. Input and control signals for *figure 9*

Operation	In	Key	AES/SHA3	enc/dec
AES enc	128	256	1	1
AES dec	128	256	1	0
Keccak f[1600]	1600[r+c] r=in+32'b0 c=576'b0	256'b0	0	--

4. EXPERIMENTAL RESULTS

The AES/SHA-3-512 schematic diagram as shown in Figure 10, is designed and written in Verilog code, simulated, and synthesized in VIVADO 2017. 2 version, implemented on the Artix-7 edge A7 xc7a200tffv1156-1 FPGA Board. From Figure 11 is FPGA board hardware Implementation setup. AES/SHA3-512 in Hexcode

Input="4B2E4A414E534849204C414B53484D494B2E4A414E534849204C414B53484D494B2E4A414E534849204C414B53484D494B2E4A414E534849204C414B53484D49";

Key="5352492056454E4B41544553574152412055 4E49564552534954592C2054 5054".

When the FPGA board is conFigureured and connected to the system, LED output is produced. LED glow indicates logic "1," or one bit, and LED off indicates logic "0." Thus, it provides a 16-bit output display on the board at a time. 72 bytes, or 576 bits, are input; Key: 256bits; 512 bits are output.

The Output in Hexcode:"4B2E4A414E534849204C414B53484D494B2E4A414E534849204C414B3484D494B2E4A414E534849204C414B53484D494B2E4A414E534849204C414B53484D49".

The code on the FPGA board was implemented, and the result was a 16-bit output that appeared on the board 32 times like "0010010101100110", "0001011100110100". This output shown in *figure. 12* and graphical representation in *figure 13*.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

As indicated in *table 7*, the area utilisation of LUT 18.49%, Flip Flops 0.83%, IO 3.8%, and BUFG 3.13% is reduced by the suggested method. *Table 8* illustrates and graphical representation *figure 14*. The total power consumed for this method, which is 6883.953W. The dynamic power used is 6883.11W, while the static power is 0.8543W.

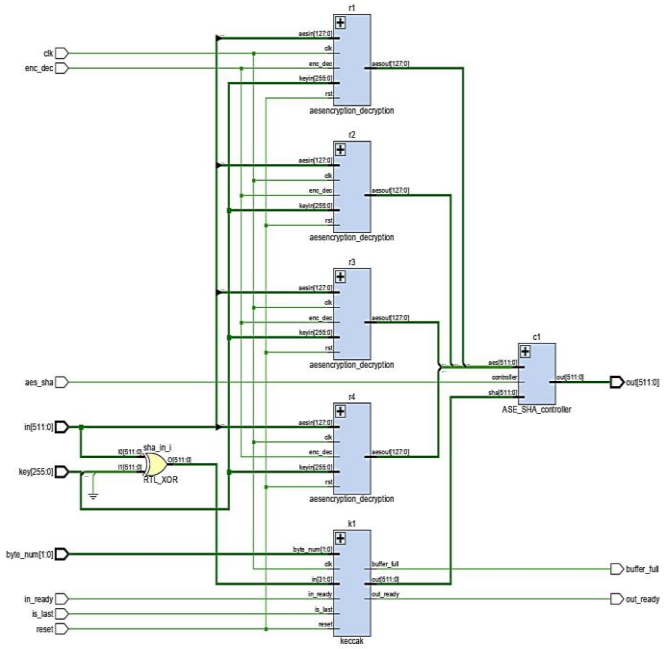

Figure 10. Schematic diagram of Hybrid AES/SHA3-512

Table 7. Proposed method Resource Utilization of FPGA ARTIX-7

Resource	Utilization	Available	Utilization%
LUT	24743	133800	18.49
FF	2245	269200	0.83
IO	19	500	3.8
BUFG	1	32	3.13

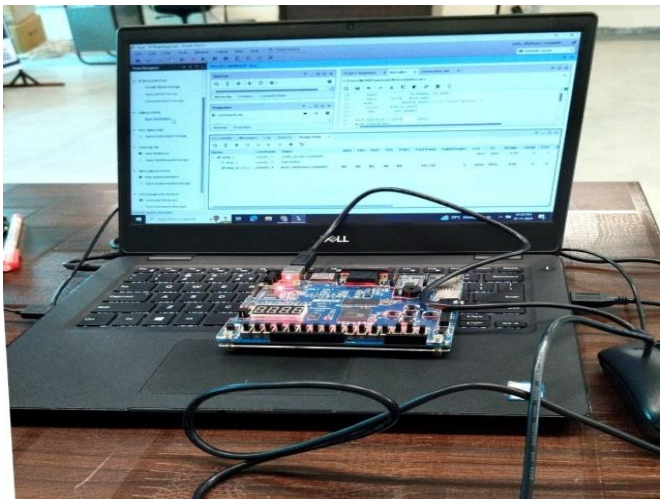

Figure 11. AES/SHA3-512 FPGA Output Implementation Display and hardware Setup

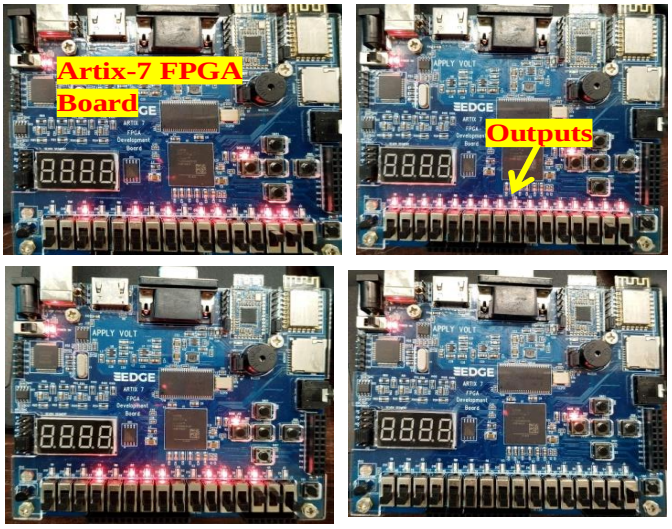

Figure 12. FPGA Board Output of proposed method AES/SHA3-512

Table 8. Proposed method Power Analysis of FPGA Artix-7

Components	Power (W)
Dynamic Power	6882.11
Signals:	3520.33
Logic:	2173.53
I/O:	1189.26
Static Power	1.8543
Total on chip power	6883.953

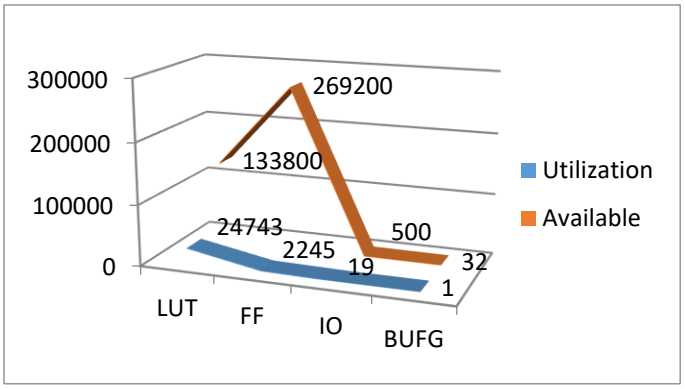

Figure 13. Area Utilization of FPGA ARTIX-7

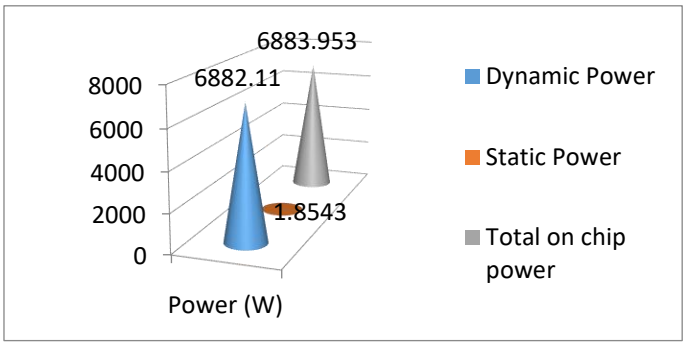

Figure 14. Power Utilization of FPGA ARTIX-7

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

4. RESULTS AND DISCUSSION

We compared our work—the hybrid cryptosystem AES/SHA3-512—with the five designs shown in *table 9* in the findings discussion. *Table 9* shows that all five designs, including Altera, Virtex 5 family, and Virtex 7 family, employ pricey FPGAs. *Figure 15* shown graphical representation of Area comparison different designs with AES/SHA3-512. But when we contrast our design with these five, we discover that it uses significantly less space and is far more cost-effective. When compared to other designs, space and power consumption also perform better.

Table 9. Area and power consumption comparison different designs with AES/SHA3-512 model

Design	FPGA	Area (%)	Power Consumption(W)
[5]	Virtex 7	49.37%	-
[4]	ALTERA	39.5%	243mW
[8]	Virtex 7	11%	299mW
[10]	Virtex 5	23%	7124.117W
[11]	Virtex 5	19%	-
Proposed Work	Artix 7	18.49%	6883.953W

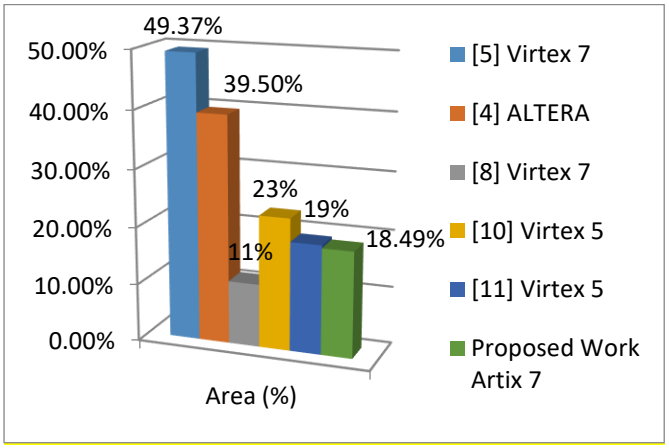

Figure 15. Area comparison different designs with AES/SHA3-512 model

5. CONCLUSIONS AND FUTURE WORK

The proposed method is a hybrid cryptosystem technique that combines Advanced Encryption Standard (AES) with the Secure Hash Algorithm-3-512Bits (SHA3-512). This system becomes a strong and secure system. Also produces a strong cipher text. The combined AES/SHA3-512 method offers a robust and secure solution for data encryption. When implemented on Artix-7 FPGAs, it achieves exceptional hardware efficiency, demonstrably reducing resource utilization by 18.49% for Look-Up Tables (LUTs), 0.83% for Flip-Flops, and 3.8% for I/O pins. This efficient implementation was successfully synthesized and simulated using the Vivado 2017.2 tool. The proposed cryptosystem is synthesized and simulated using the Vivado 2017.2 version Tool. In Future scope, for better performance will use another any two cryptosystem algorithm combined with AES or SHA.

Acknowledgement: The authors would like to thank the Indian Institute of Technology (IIT) - Tirupati, which granted permission for use of the VLSI Laboratory for hardware implementation on FPGA board.

Contributions: K. Janshi Lakshmi conducted the experiment and analyzed the data from the experimental results. Manuscript preparation, data collection and analysis was performed by K.Janshi Lakshmi. All authors read and approved the final manuscript.

Conflict of interest: The authors of this research declare that they have no conflicts of interest.

REFERENCES

[1] Advanced Encryption Standard, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA, Standard FIPS-197, Nov. 2001.

[2] SHA-3 Standard: Permutation-Based Hash Extendable-Output Functions, National Institute of Standards and Technology (NIST), Gaithers burg, MD, USA, Standard FIPS PUB 202, Aug. 2015.

[3] "Keccak Specification Summary", Keccak Team, [online] Available: https://keccak.team/keccak_specs_summary.html.

[4] Sheng-Jung Yu, Yu-Chi Lee, Liang-Hsin Lin, "An Energy-Efficient Double Ratchet Cryptographic Processor with Backward Secrecy for IoT Devices", IEEE Journal of Solid-State Circuits (Volume: 58, Issue: 6, June 2023), https://doi.org/10.1109/JSSC.2022.3220838

[5] Dur-e-Shahwar Kundi, Ayesha Khalid, Arshad Aziz, Chenghua Wang, Weiqiang Liu, "Resource-Shared Crypto-Coprocessor of AES Enc/Dec with SHA-3", IEEE transactions on circuits and systems–i: regular papers, 1549-8328 © 2020 IEEE https://doi.org/10.1109/TCSI.2020.2997916

[6] Jayanti Sharma, Deepali Koppad, "Low Power and Pipelined Secure hashing Algorithm3(SHA-3)", 2016 IEEE Annual India Conference (INDICON), 2016, Electronic ISSN: 2325-9418, https://doi.org/10.1109/INDICON.2016.7839059

[7] Sara al-shara'a, Raid Khalid ibraheem , Oguz Bayat ,"Implementation of cryptanalysis based on FPGA hardware using AES with SHA-1" 2019 International Conference on Smart Applications, Communications and Networking (SmartNets), ©2019 IEEE. https://doi.org/10.1109/SmartNets48225.2019.9069786

[8] Noveline Aziz Fauziah, Eko Hari Rachmawanto et al, "Design and Implementation of AES and SHA-256 Cryptography for Securing Multimedia File over Android Chat Application", IEEE Transactions on Dependable and Secure Computing (Volume: 16, Issue: 2, 01 March-April 2019), https://doi.org/10.1109/TDSC.2017.2687463

[9] Huanyu Wang, Henian Li et al., "SoFI: Security Property-Driven Vulnerability Assessments of ICs Against Fault-Injection Attacks", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (Volume: 41, Issue: 3, March 2022), https://doi.org/10.1109/TCAD.2021.3063998

[10] P. William, Abha Choubey, G. S. Chhabra, Riju Bhattacharya, K. Vengatesan, Siddhartha Choubey, "Assessment of Hybrid Cryptographic Algorithm for Secure Sharing of Textual and Pictorial Content", 2022 International Conference on Electronics and Renewable Systems (ICEARS), © 2022 IEEE, https://doi.org/10.1109/ICEARS53579.2022.9751932

[11] Wenjian Luo, Yamin Hu et al., "Authentication by Encrypted Negative Password", IEEE Transactions on Information Forensics and Security (Volume: 14, Issue: 1, January 2019), https://doi.org/10.1109/TIFS.2018.2844854

[12] K. Janshi Lakshmi and G. Sreenivasulu, "Design and Implementation of S-Box Using Galois Field Approach Based on LUT and Logic Gates for AES-256", Proceedings of the International Conference on Intelligent Computing,

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 12, Issue 1 | Pages 195-202 | e-ISSN: 2347-470X

Communication and Information Security, Algorithms for Intelligent Systems, Springer Nature, July 2023, https://doi.org/10.1007/978-981-99-1373-2_10

[13] Kazim Yumbul and Erkay Savas, "Enhancing an Embedded Processor Core for Efficient and Isolated Execution of Cryptographic Algorithms", The Computer Journal (Volume: 58, Issue: 10, October 2015), IEEE, https://doi.org/10.1093/comjnl/bxu040

[14] Jalel Ktari, Tarek Frikha, Mohamed Ali Yousfi,Mohamed Kadhem Belghith, Nessrine Sanei."Embedded Keccak implementation on FPGA", 2022 IEEE International Conference on Design & Test of Integrated Micro & Nano-Systems(DTS),2022 https://doi.org/10.1109/DTS55284.2022.9809847

[15] Ling Ren, Christopher W. Fletcher, AlbertKwon, Marten van Dijk, Srinivas Devadas."Design and Implementation of the AscendSecure Processor",IEEE Transactions on Dependable and Secure Computing (Volume: 16, Issue: 2, 01 March-April 2019)https://doi.org/10.1109/TDSC.2017.2687463.

K. JANSHI LAKSHMI is pursuing Ph.D. in the Department of Electronics and Communication Engineering at Sri Venkateswara University College of Engineering, S V University, Tirupati, Andhra Pradesh, India. She received her B. Tech degree in ECE from Aurora scientific and technological institute, Hyderabad, JNTUH in 2010, and received M.Tech degree in VLSI System Design as specialization from Siddhartha institute of engineering and technology, Puttur, JNTUA in 2012. She is a Life Member of ISTE (MISTE). She has presented papers in various National, International conferences and published papers in various reputed journals. Her research interest includes VLSI system Design, Network Security.

Prof. G. SREENIVASULU is received the Ph.D. degree in Process Control in 2007 and is currently working as Professor in the Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, S V University, Tirupati, Andhra Pradesh, India. He has 28 years of teaching experience in the Department of Electronics and Communication Engineering, Sri Venkateswara University College of Engineering, Tirupati, Andhra Pradesh, India. He has published 27 papers in reputed journals. He guided 28 PG and 08 Ph. D projects. He is a Life Member of ISTE (MISTE) and Fellow of IETE (FIETE). His research interests include Neural Networks and Fuzzy Logic Applications in Process Control, Process Instrumentation, Analog Electronics, and Digital Electronics.