Turkish Journal of Computer and Mathematics Education
DOI: https://doi.org/10.17762/turcomat.v9i3.13900 *Vol.09 No.03 (2018), 1107-1115*

Research Article

Design and Optimization of a FPGA Area Efficient Present Cryptographic Architecture for IoT Node

HimaniI Sivaraman

Asst. Professor, Department of CSE (Computer sc)

GEHU-Dehradun Campus

Abstract: Lightweight symmetric cyphers are crucial for securing constrained computing networks like the Internet of Things and wireless sensor networks. Traditional cryptographic algorithms are quite resource-intensive. IoT does not make use of these kinds of algorithms. Traditional cryptographic algorithms have to worry about the limited environment and the key size since they increase the complexity of the method. The data is hacked during the wireless transmission from one device to another. The LCS-PRESENT architecture suggested in this study is a mix of the LCS circuit and the PRESENT block. Utilisations of FPGA, including Lookup Tables (LUTs), flip-flops, slices, and frequency, are used in the assessment of the architecture's performance. Using a lookup table, we may construct the key module with less space requirements. By encrypting and decrypting an LCS-PRESENT block, the XILINX tool may be used to assess the security of a system. This study uses a hardware-level encryption technique with an 80-bit key for 64-bit input data. The primary actions of the PRESENT block are key rotation and key replacement. In order to encrypt a picture, it is necessary to switch keys, rotate keys, and generate a digital-based key for use in key rotation and replacement. It provides a lot of protection against threats, so your data is safer. In this study, we employ the Xilinx 14.2 ISE tool, the Verilog HDL programming language, and the FPGA tool to create the LCS-PRESENT architecture. This tool is used to evaluate the efficiency and precision of a building's design. Performance metrics for LUTs, flip-flops, and slices are compared to those of standard, cryptographic methods. Existing and prospective designs' FPGA performances are compared..

Keywords: Cryptography, Lightweight Symmetric Cipher, Xilinx, LCS, PRESENT, FPGA, IoT, Wireless Sensor Network, LUT, LCS-PRESENT, Key Rotation, Key Substitution, Flip-Flops, Slices.

Introduction

To safeguard data while it is being sent over the internet or kept in a database, or for any other reason, cryptology is utilised. Digital signatures is another name for this. The two main branches of the cryptosystem are cryptography and cryptanalysis, so keep that in mind. Cryptography is the study of developing secure, fast, and efficient cryptosystems via the use of encryption and decryption methods. Cryptanalysis is used in cryptography, and it is the study of finding and exploiting weaknesses in cryptographic systems. As was said before, the primary reason for using encryption is to safeguard the CIA of sent information. In the context of cryptography, a "cryptographic algorithm" is a mathematical function. The plaintext is encrypted using a key, and then decrypted after the procedure is complete. A solution to the puzzle might lie in certain words, numbers, or a catchy phrase. Only the secret key and the cryptographic technique used to encrypt the key may compromise its safety. Traditional cryptosystems have security flaws due to their reliance on static keys. The intricacy of the mathematical functions used to perform these strategies also affects how secure they are. Therefore, the encryption method would be compromised if a breakthrough were to occur in these areas. Concern for the safety of one's private data is shared across demographics. Throughout history, kings and queens have communicated with their troops and other rulers using covert methods to avoid having their communications decoded by the opposition. A more sophisticated method of decoding the secret plan is required in light of technological progress. In today's information-based and digital society, data is more valuable than gold. We must take measures to secure our data while it is in transit or storage; doing so is crucial and necessary if we are to deter data theft. Confidentiality, integrity, and availability are the three pillars of a secure data or information system. These three information security guidelines have not changed much throughout the centuries, from the Stone Age to the Internet Age. When moving data from one device to another, just as much care must be taken to ensure its security as it is being kept.

Cryptography in Internet of Things (IoT)

More than 20 billion IoT devices are predicted to be in use worldwide by 2020. It's a good indicator that prospects for corporate expansion will arise. A number of people have voiced worries about security and other possible threats. Wireless fidelity (wifi) and radio communications will be open to anybody and everyone if encryption isn't used. Private information, such as account numbers, will be broadcast over the airwaves and available to anybody who listens. The world's economy will suffer as a result if encryption is not widely used. Thus, care must be taken while using encryption methods. Information sent over the phone or during a credit card or ATM transaction is encrypted to prevent unauthorised access. All encryption methods work by taking the original, readable data and replacing it with an unreadable cypher string. This prevents any unauthorised parties from gaining access to the data. There is no way to read the ciphertext, with or without the use of decryption keys. These days, the Internet of Things is used by wireless sensor networks to allow for intelligent processing and reliable transmission through wireless channels. Therefore, the most important aspects of the Internet of Things are security and privacy. As the scope of the network expands, new security concerns arise. The potential for malicious parties to get access to private data may be greatly reduced by encrypting it. The use of consumer devices like cellphones and activity trackers is on the rise. A variety of sensors record and analyse vital signs including heart rate, blood pressure, and more. In addition to aiding in illness diagnosis and prevention, this information may be put to use in studies. Information submitted by the user should be encrypted so that it cannot be viewed by an unauthorised person. The future of the Internet of Things is bright if data is properly secured and safeguarded.

Chaotic Cryptography

Based on the definition of random sequence numbers produced by dynamic systems, the Chaos Encryption Principle is established. These sequences are used to encrypt the messages. If the initial circumstances are altered even little, a new will emerge. The standard cryptographic technique has more encryption rounds, making the algorithm's diffusion and confusion features more apparent. Metrics from the chaotic map are utilised to decipher the secret message. As may be seen in Figure 1's block diagram, information is sent through indirect coupling chaotic synchronisation. The structure consists of a network of chaotic oscillators. Two oscillators, one serving as a transmitter and the other as a receiver, may communicate with each other in a single direction. Both the sender and the receiver employ key stream generators, which consist of two oscillators connected in an indirect fashion for purposes of encryption and decryption.

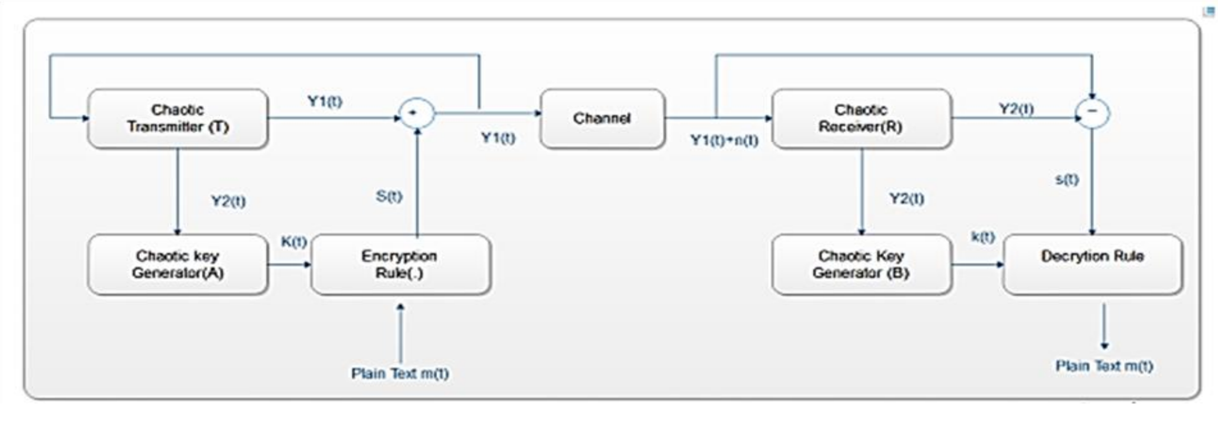


Figure 1: Communication techniques using indirect coupling chaotic synchronization

The N-shift cypher encryption technique is used to secure the data in this method, which is implemented using Lorenz and Chao's technology. The chaotic signals 4 produced by deterministic systems are affected by extraneous variables and random signals that cannot be created in the same manner. The study's conclusion was to use chaotic circuits for cryptographic and communication security. There are two ways to put chaotic cryptography into practice. Analogue circuits are the starting point for developing cryptosystems, with synchronization methods being the most often used approach. Second, unlike analogue systems, digital ones don't need to be synchronized.

FPGA implementation of chaotic systems

FPGAs are programmable semiconductor devices that have a matrix of configurable logic blocks (CLB). These CLB are linked together by a programmable connection. After an ASIC has been made, all changes to the design are impossible. However, FPGAs may be rewritten to meet the specific requirements of every given task. The Xilinx System Generator for Digital Signal Processing and the Integrated Software Environment (ISE) are prerequisites for running the Simulink Simulator. The mathematical programme Simulink is an integral part of MATLAB. Complex dynamical nonlinear systems may be modelled, simulated, and analysed in continuous and sampling time using the programme. The Xilinx System Generator includes a wide variety of Simulink blocks that may be used to implement a wide range of hardware-based tasks. Connecting to the MATLAB environment and the Simulink libraries is possible using a wide variety of Xilinx blocks, including but not limited to flip-flops, filters, memory, multipliers, and gateways. These blocks in the Simulink environment act like their hardware counterparts. It's simple to produce HDL (Hardware Description Language) using the given designs. Simulink's system generator for Xilinx FPGAs makes it easy to create high-performance DSP systems for these devices. Simulink modelling software may be used to automatically produce HDL code and a test bench, allowing for the construction of a realistic model of the FPGA circuits. When a Virtex or Spartan device from the Xilinx family is used to implement a VHDL design, an HDL netlist file is created that may be read by HDL simulators.

IoT Security and Privacy Issues

There are three parts to this section. The security elements of IoT are initially covered,

followed by security and privacy concerns in the following sections.

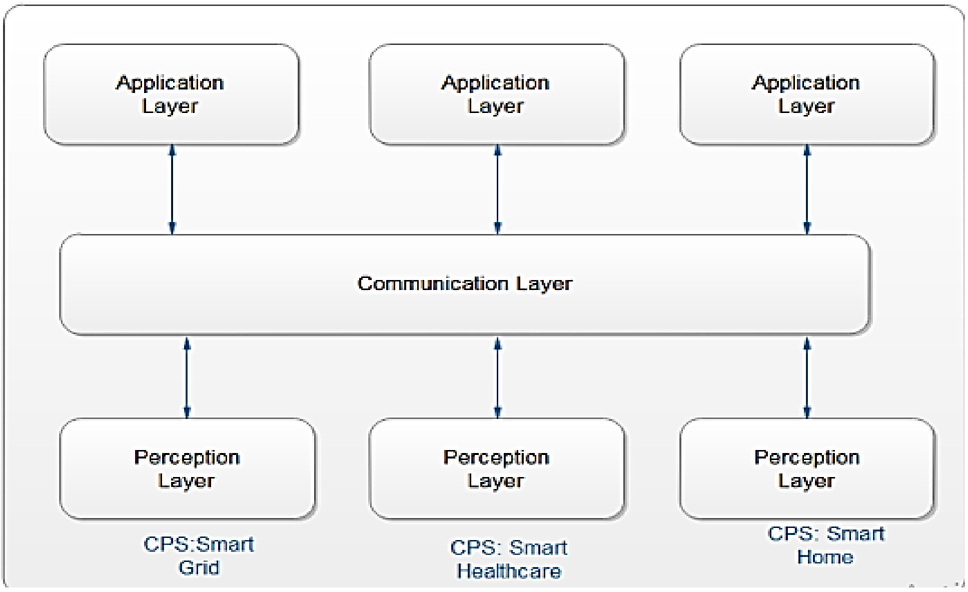


Figure 2: IoT and CPS Integration

Literature Review

While conventional compressed sensing strategies required more memory space for storing the matrix generation parameters, we discovered that Peng and colleagues' (2017) chaotic compressive sensing approaches were more successful at tackling challenging challenges. To do this, they resorted to a high-quality encryption technique and two distinct encryption strategies for the confuse and mask phases. A feasibility study demonstrated the efficacy of their methods by enhancing both energy efficiency and data security.

Gonzalez (2017) looked on using electrocardiogram (ECG) records for network user data encryption. To address these issues, an electrocardiogram (ECG)-based technique for creating the durable key has been developed.

Lara-Nino et al. (2017) created a lightweight hardware cryptography architecture that was implemented on an FPGA platform. This research approach necessitates the use of a round key, which may be generated in two different methods. A data route architecture using 16 bits of memory and a 128-bit key schedule was implemented. This design has a 16-bit data channel protected by an 80-bit key, making it suitable for situations when space and safety must be compromised. The results were the same for both LUT-4 FPGAs. Spartan-6 FPGA LUT-6 hardware platforms may be shown in action. The technique makes use of LUT components with few slices. Measurements of performance and assessment are recorded by the serial architecture, which also determines the size of the implementation.

Yamac et al. (2016) created a novel approach to data concealment by compressing it and then reconstructing it using a deflationary method. How the embedding process and susceptibility to noise impact signal sparsity and compression has been investigated.

Safwan El and Mousa (2016) built a unique cryptosystem architecture that they used to provide a safe and efficient means of data transfer. Their cryptosystem is not as secure as current cryptosystems, but it is quicker when dealing with fixed numbers than existing cryptosystems that operate with varying choas while ensuring high-level security.

An effective image encryption technique was developed by Zeinab et al. (2016) using two rounds of substitution and diffusion. Robustness against brute force attacks, statistics, selected plain text, and ciphertext, and minimising the temporal complexity of mobile application needs are some of the major aims that have guided their work. They constructed the replacement and diffusion processes, and also changed the input picture by dynamically adjusting the key. The encryption procedure was also carried out in line with the replacement and diffusion of each image matrix. Finally, after running a battery of tests on their method, they've determined that the image encryption section works as promised.

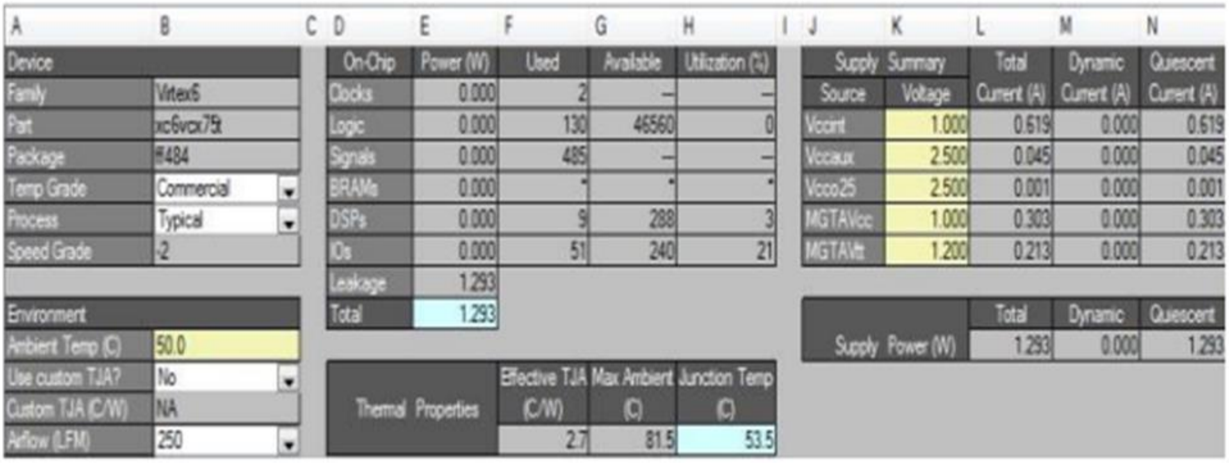
Experimental Results

This example shows how to use the RTL compiler for Virtex-6 FPGA devices to create Verilog code that implements the LCS-PRESENT, SRS, and MS-CLCG architectures on the Xilinx platform. The device utilisation in this node is being researched because of the high configuration devices in Virtex-6 of these architectures. In this research, the suggested PRESENT-made system is put through its paces using design support software like Xilinx and simulation software like ModelSim. The execution system has a substantial impact on the outcomes. Table 5.1 shows how the recommended PRESENT architecture makes use of various devices. This study explores the use of LUTs, flip-flops, slices, and frequency, among other methods, to encrypt and decode data using the PRESENT architectural cypher model with a 16-bit key. These three designs are all realised in an FPGA environment. These architectures are preferable to the ASIC platform for VLSI implementations because to their reduced power consumption, more flexibility, and upward compatibility.

Table 1: Xilinx FPGA device Virtex-6 (XC6VCX75T) utilization summary for LCS-PRESENT architecture

Elements	Available Resources	Used in count	Utilization in percentage
Number of slice registers	93,120	131	1%
Flip Flop	46,560	115	1%
Number of slice LUTs	46,560	130	1%
Number of used as a logic	46,560	125	1%
slices	11,640	47	1%

Bit-wise algorithms often need low-power, very-large-scale-integration (VLSI) circuits due to the need for high throughput and low latency. Due to the high data rates in communication networks, circuits are commonly needed. The proposed PRESENT architecture uses only 11,640 slices (0.1% of the total), 46,560 LUTs (0.1% of the total), and 46,560 flip-flops (0.1% of the total). The PRESENT architecture used in this research runs at 90.26 MHz with a delay of four clock cycles, both of which are within acceptable ranges. As can be seen in Fig.3, the total power used by the LCS-PRESENT layout is 1,293. The performance of the design has vastly increased as a consequence of better utilisation of device resources. The results clearly show that the suggested PRESENT architecture was a perfect match for data encryption.



Device	
Family	Virtex6
Part	xc6vcx75t
Package	ff484
Temp Grade	Commercial
Process	Typical
Speed Grade	-2

Environment	
Ambient Temp (C)	50.0
Use custom TJA?	No
Custom TJA (C/W)	NA
Airflow (LFM)	250

On-Chip	Power (W)	Used	Available	Utilization (%)
Clocks	0.000	2	---	---
Logic	0.000	130	46560	0
Signals	0.000	485	---	---
BRAMs	0.000	-	-	-
DSPs	0.000	9	288	3
IOs	0.000	51	240	21
Leakage	1.293			
Total	1.293			

Thermal Properties	Effective TJA (C/W)	Max Ambient (C)	Junction Temp (C)
	2.7	81.5	53.5

Supply Summary		Total	Dynamic	Quiescent
Source	Voltage	Current (A)	Current (A)	Current (A)
Vccint	1.000	0.619	0.000	0.619
Vccaux	2.500	0.045	0.000	0.045
Vcco25	2.500	0.001	0.000	0.001
MGTAVcc	1.000	0.303	0.000	0.303
MGTAVtt	1.200	0.213	0.000	0.213

	Total	Dynamic	Quiescent
Supply Power (W)	1.293	0.000	1.293

Figure.3 The power consumption outputs by LCS-PRESENT architecture.

Resource consumption on FPGA devices is compared between the present PRESENT design and the LCS-PRESENT architecture in Table 2. Both the 50-ton Virtex-5 and the 16-core Virtex-6 FPGAs, the two most common versions now available, have been used to test all of the existing design implementations. The LCS PRESENT architecture uses 31.57 percent of FPGA LUTs, 24.83% of flip flops, and 29.85 percent of slices. When compared to the architecture presented by Lara-Nino et al., the current design makes use of 92.32% more FPGA LUTs, 86.99% more flip flops, and 90.89% more slices. When compared to the design of Lara-Nino et al., the PRESENT architecture consumes 51.127% less FPGA LUTs and 31.88% fewer slices. Table 2 summarises the synthesis outcomes and implementation values. According to the data in the table, the LCC design may reduce the number of required FPGA chips by 90–92%. Key size reduction and efficient resource utilisation have contributed to a performance boost for the proposed LCS-PRESENT architecture. The suggested design has the potential to increase frequency performance over current designs by 90.26 MHz..

Authors	**YEAR**	**FPGA Devices**	**LUTs**	**Flipflops**	**Slices**	**Frequency (MHz)**
Lara-Nino et al. [47]	2017	Virtex-5XC5vlx 50t 3ff1136	239	201	73	431.78
			190	153	67	543.30
Lara-Nino et al [42]	2018	Virtex -6XC6LX1 6-CS324	1694	884	516	13.56
Pandey et al. [51]	2017	Virtex -5XC5VL X50	266	-	69	306.84
Proposed LCS-PRESENT	2019	Virtex-6XC6VC X75t	130	115	47	90.26

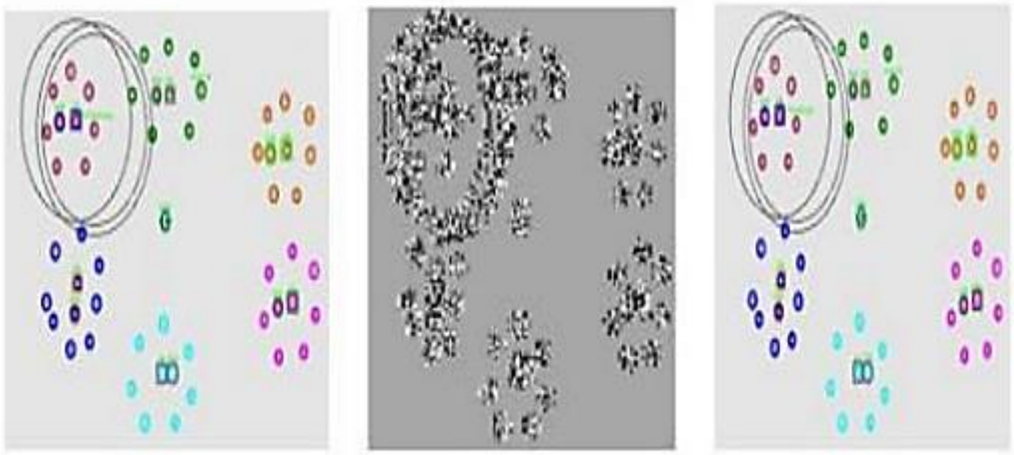


Figure 4: Sensor node input image, encrypted image and decrypted image.

Examples of sensor node deployment photographs, encrypted images, and decrypted images are shown in Figure 4. Images captured by the two sensor nodes are used to assess the LCS-PRESENT architecture. Figure 5.3 displays an encrypted version of the input picture that, as predicted by the suggested design, is identical to the original. Therefore, the encrypted picture retains all of its original qualities. This novel design allows for efficient encryption and decryption with a minimum of FPGA devices and power consumption.

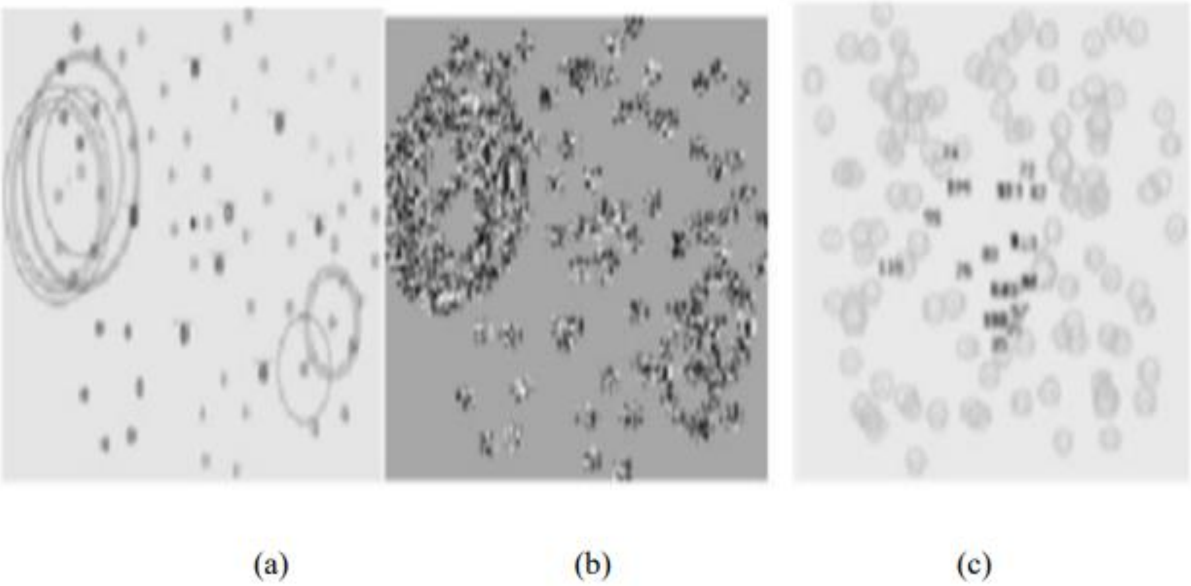


Figure 5: a) Input Plain text image b) Cipher image and c) Retrieved image.

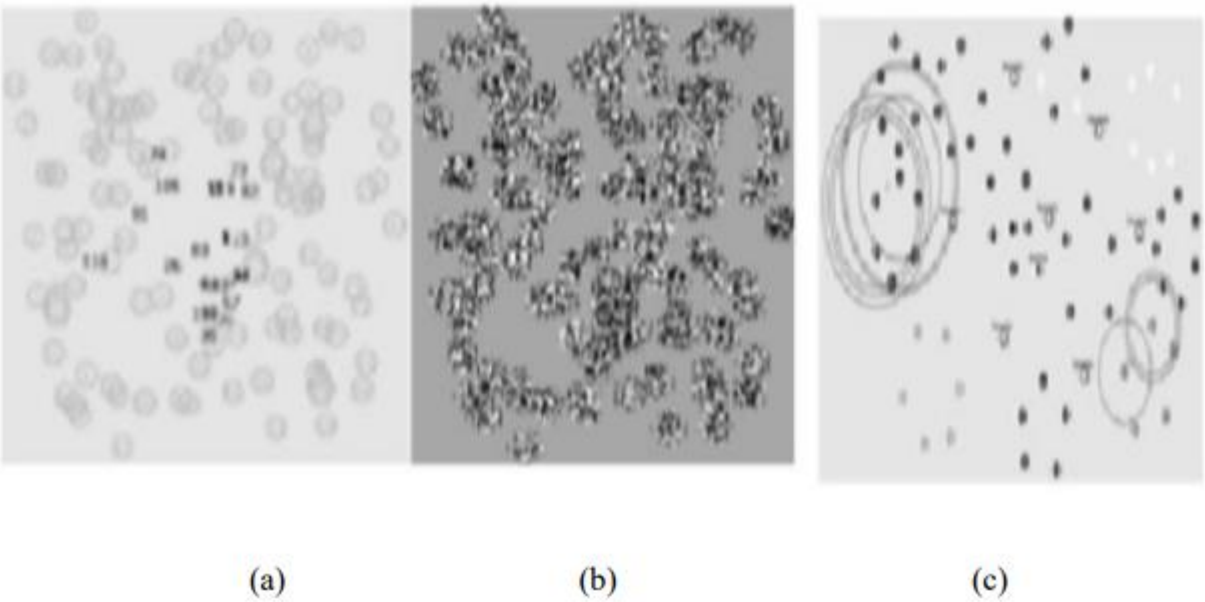


Figure 6: a) Input Plain text image, b) cipher image and c) retrieved images.

From the FPGA output, the encrypted text file is loaded into MATLAB. The system's binary values are converted to their corresponding pixel values using the binary to decimal (Base-10) conversion algorithm. Each 8-bit data represents a single pixel value. Encryption and decryption are used to generate text files, cyphers, and decrypted pictures, as seen in Figures 5. and 6. The unencrypted version of the method correctly retrieves the sensor deployment. This result seems to validate the effectiveness of the intended PRESENT design. The standard approach uses straightforward key scheduling, which may be broken by malicious actors. In contrast, the SRS-PRESENT design uses LUT-based key scheduling to generate a key for each cycle and, by extension, plain text. Therefore, the SRS-PRESENT architecture is more secure for storing information. The four LUTs create the 16-bit key and utilise it to carry out the scheduling of the keys.

Conclusion

The LCS-PRESENT architecture suggested in this study is a mix of the LCS circuit and the PRESENT block. Lookup Table (LUT), flip-flop, slice, and frequency utilisations are used to measure the performance of the design. Using a lookup table, we may construct the key module with less space requirements. By encrypting and decrypting an LCS-PRESENT block, the XILINX tool may be used to assess the security of a system. This study uses a hardware-level encryption technique with an 80-bit key for 64-bit input data. The primary actions of the PRESENT block are key rotation and key replacement. In order to encrypt a picture, it is necessary to switch keys, rotate keys, and generate a digital-based key for use in key rotation and replacement. It provides a lot of protection against threats, so your data is safer. The PRESENT architecture may be implemented using less energy and fewer materials.

References

1. Soltani and S. Sharifian, "An ultra-high throughput and fully pipelined implementation of AES algorithm on FPGA," Microprocess. Microsyst., vol. 39, no. 7, pp. 480–493, 2015.
2. Y. Wang and Y. Ha, "FPGA-based 40.9-gbits/s masked AES with area optimization for storage area network," IEEE Trans. Circuits Syst. II Express Briefs, vol. 60, no. 1, pp. 36–40, 2013.
3. H. Lee, Y. Paik, J. Jun, Y. Han, and S. W. Kim, "High-throughput low-area design of AES using constant binary matrix-vector multiplication," Microprocess. Microsyst., vol. 47, pp. 360–368, 2016
4. H. Lee, Y. Paik, J. Jun, Y. Han, and S. W. Kim, "High-throughput low-area design of AES using constant binary matrix-vector multiplication," Microprocess. Microsyst., vol. 47, pp. 360–368, 2016
5. F. X Standaert, G. Piret, N. Gershenfeld and J.-J. Quisquater. (2006), "SEA: a scalable encryption algorithm for small embedded application", Smart Card Research and Applications, Proceedings of CARDIS 2006, volume 3928 of LNCS, pages 222- 236, Springer-Verlag
6. X.Fan, G. Gong, K.Lauffenburger and T.Hicks. (2010), Design Space Exploration of Hummingbird Implementations on FPGAs, Techincal Report
7. Xinxin Fan, Honggang Hu, Guang Gong1, Eric M. Smith and Daniel Engels. (2009), "Lightweight Implementation of Hummingbird Cryptographic Algorithm on 4-Bit Microcontrollers", Institute of Electrical and Electronics Engineers
8. Xinxin Fan, Guang Gong, Ken Lauffenburger and Troy Hicks. (2010), "FPGA Implementations of the Hummingbird Cryptographic Algorithm", 978- 1- 4244-7812-5/10/, IEEE. Advances in Systems Science and Application (2015) Vol.15 No.4 365
9. Biao Min, Ray C.C. Cheung and Yan Han. (2011), "FPGA-based HighThroughput and Area-Efficient Architectures of the Hummingbird Cryptography", 978-1-61284-972-0/11/, IEEE.
10. Ismail San and Nuray At. (2011), "Compact Hardware Architecture for Hummingbird Cryptographic Algorithm", 21st International Conference on Field Programmable Logic and Applications, 978-0-7695-4529-5/11, IEEE
11. P. Yalla and J.P. Kaps. (2009), "Lightweight Cryptography for FPGAs", International Conference on Re-ConfigurableComputing and FPGAs ReConFig'09
12. F. Mace, F.X. Standaert and J.J. Quisquater. (2007), "FPGAimplementation(s) of a Scalable Encryption Algorithm", IEEETrans, Very Large Scale Integ, (VLSI) Syst.Vol.16, No.2, pp.212-216
13. Suzaki, T., Minematsu, K., Morioka, S. and Kobayashi, E., 2011, November. Twine: A lightweight, versatile block cipher. In ECRYPT Workshop on Lightweight Cryptography (Vol. 2011)
14. Suzaki, T., Minematsu, K., Morioka, S. and Kobayashi, E., 2011, November. Twine: A lightweight, versatile block cipher. In ECRYPT Workshop on Lightweight Cryptography (Vol. 2011)
15. Shibutani, K., Isobe, T., Hiwatari, H., Mitsuda, A., Akishita, T. and Shirai, T., 2011, September. Piccolo: An ultra-lightweight blockcipher. In CHES (Vol. 6917, pp. 342-357).
16. Gong, Z., Nikova, S. and Law, Y.W., 2011. KLEIN: A new family of lightweight block ciphers. RFIDSec. Springer, 7055, pp.1-18.

17. Guo J., Peyrin T., Poschmann A., and Matt Robshaw M.,Preneel B. and Takagi T., 2011. The LED Block Cipher. CHES 2011, In International Association for Cryptologic Research, LNCS 6917 (pp. 326–341).
18. Lim, C.H. and Korkishko, T., 2005, August. mCryptona lightweight block cipher for security of low-cost RFID tags and sensors. In WISA (Vol. 3786, pp. 243- 258).
19. Daemen, J. and Rijmen, V., 2001, December. The wide trail design strategy. In IMA International Conference on Cryptography and Coding (pp. 222-238). Springer, Berlin, Heidelberg