ISSN 2319-5991 www.ijerst.com
© 2026 IJERST. All Rights Reserved
Peer Reviewed, Referred & Indexed Journal

Research Paper

ADVANCE RC6 IMPLEMENTATION FOR SECURITY APPLIANCES

[1]Mohammed Aleem
aleemmohammed766@gmail.com

[2]Mohammad Gandhi Babu
gandhi.mohammad15@gmail.com

[3]Dr.N.Suresh
Ecehod@avniet.ac.in

[1]Scholar, Master of Technology,Dept of Electronics and Communication Engineering, AVN Institute of Engineering and Technology Koheda Road M.P.Patel Guda Post Ibrahimpatnam, Dist., Hyderabad, Telangana 501510

[2]Assistant Professor, Dept of Electronics and Communication Engineering, AVN Institute of Engineering and Technology Koheda Road M.P.Patel Guda Post Ibrahimpatnam, Dist., Hyderabad, Telangana 501510

[3] Professor& Hod, Dept of Electronics and Communication Engineering, AVN Institute of Engineering and Technology Koheda Road M.P.Patel Guda Post Ibrahimpatnam, Dist., Hyderabad, Telangana 501510

ABSTRACT

In the modern digital era, protecting confidential information from unauthorized access is an important requirement in communication systems, embedded devices, and hardware-based security applications. Cryptographic algorithms provide a mechanism for securing sensitive data by converting readable plaintext into an encrypted ciphertext using a secret key. The **Advance RC6 Implementation for Security Appliances** project focuses on the hardware implementation of the RC6 symmetric-key block cipher using Verilog Hardware Description Language (HDL). The primary objective is to design and simulate an encryption and decryption architecture that can process digital data securely and efficiently in dedicated hardware.The proposed implementation consists of essential functional modules, including a datapath, key expansion unit, key selector, round processing unit, and control unit. The key expansion module generates the round keys from the input cipher key, while the key selector supplies the appropriate round key according to the current processing stage and operation mode. The controller coordinates the encryption and decryption sequences using clock, reset, and control signals. During encryption, the input plaintext is processed through the required cryptographic operations to produce ciphertext. During decryption, the corresponding inverse operations recover the original plaintext. The architecture is designed to support controlled, sequential processing of data blocks and provide a data-ready signal when the operation is complete.The project uses Verilog HDL to describe the hardware architecture and employs Xilinx ISE 13.7 and Xilinx ISim for design synthesis and functional simulation, as described in the document. The simulation process helps verify the behavior of the encryption and decryption modules using suitable input data and cipher keys. The design can serve as a foundation for hardware security applications involving confidential messages, digital images, embedded systems, and communication devices. Overall, the project demonstrates the implementation and functional verification of a cryptographic processing architecture in hardware, with scope for further optimization of processing latency, resource utilization, and power consumption for security appliances.

Keywords:
RC6, cryptography, symmetric-key encryption, decryption, Verilog HDL, VLSI, key expansion, datapath, hardware security, Xilinx ISE, functional simulation.

I. INTRODUCTION

In the present digital world, information security has become an essential requirement for protecting confidential data from unauthorized access, modification, and misuse. The increasing use of digital communication, embedded systems, electronic devices, and network-based applications has created a need for reliable data protection mechanisms. Cryptography plays a significant role in information security by transforming readable information, known as plaintext, into an unreadable format called ciphertext through an encryption process. The original information can be recovered through decryption using the appropriate secret key. Cryptographic algorithms are widely used to protect sensitive information during storage, processing, and transmission.RC6 is a symmetric-key block cipher designed to provide encryption and decryption using the same secret key. It is based on a modified Feistel-like structure and uses operations such as modular addition, exclusive OR (XOR), integer multiplication, and data-dependent rotations. These operations contribute to the algorithm's cryptographic processing and make it suitable for implementation in software and dedicated hardware. RC6 supports a 128-bit data block and different key lengths, including 128, 192, and 256 bits under its standard parameter configurations. Its structured round-based design provides an opportunity to implement cryptographic operations through modular hardware components.The project, **Advance RC6 Implementation for Security Appliances**, focuses on designing and implementing the RC6 encryption and decryption architecture using Verilog Hardware Description Language (HDL). Verilog HDL is commonly used to describe digital circuits and their behavior before hardware synthesis and implementation. By expressing the cryptographic algorithm as interconnected hardware modules, the design can be simulated and evaluated to verify whether the required encryption and decryption operations function correctly.The proposed hardware architecture consists of a datapath, key expansion module, key selector, round-processing logic, and controller. The key expansion module generates the required round keys from the supplied secret key. The key selector provides the appropriate round key during each processing stage, while the datapath performs the cryptographic operations on the input data. The controller coordinates the sequence of operations using clock, reset, and control signals. During encryption, plaintext is transformed into ciphertext, whereas the decryption process reverses the corresponding operations to recover the original information.The implementation is developed using Verilog HDL and can be verified through simulation tools such as Xilinx ISE 13.7 and Xilinx ISim, as described in the project document. Functional simulation helps examine input signals, intermediate processing stages, output data, and control behavior. The implementation can also provide a foundation for further hardware evaluation, including processing latency, logic resource utilization, and power consumption.The primary goal of this project is to demonstrate the design and functional verification of an RC6-based cryptographic hardware architecture for security-oriented applications. Such an architecture may be relevant to embedded devices, secure communication systems, digital data protection, and other applications requiring encryption and decryption. The

project emphasizes modular hardware design, correct key scheduling, coordinated round processing, and simulation-based verification as important steps toward developing cryptographic components for security applianceIn the present digital world, information security has become an important concern due to the increasing use of digital communication, electronic devices, and network-based applications. Cryptography is widely used to protect confidential information from unauthorized access by converting readable data, known as plaintext, into an encrypted format called ciphertext. The original information can be recovered through decryption using the appropriate secret key. Symmetric-key cryptographic algorithms are particularly useful for data protection because the same secret key is used for both encryption and decryption. RC6 is a symmetric-key block cipher designed to perform these operations using a structured sequence of cryptographic transformations.

The project **Advance RC6 Implementation for Security Appliances** focuses on implementing the RC6 algorithm using Verilog Hardware Description Language (HDL). The design consists of important hardware modules, including the datapath, key expansion unit, key selector, round-processing logic, and controller. The key expansion unit generates the required round keys from the input cipher key, while the datapath performs the encryption and decryption operations. The controller coordinates the processing sequence using clock, reset, and control signals. These modules work together to transform plaintext into ciphertext during encryption and recover the original data during decryption.

The proposed implementation is described using Verilog HDL and evaluated through simulation using Xilinx ISE 13.7 and Xilinx ISim, as specified in the project document. Simulation helps verify the functional behavior of the encryption and decryption modules by examining the input data, secret key, and resulting output. The project demonstrates the application of cryptographic algorithms in digital hardware design and provides a foundation for security-oriented applications such as confidential data transmission, embedded systems, and digital information protection. Further hardware analysis can be conducted to evaluate processing speed, resource utilization, and power consumption.

II. LITERATURE REVIEW

Rivest et al. [1] introduced RC6 as an improvement over RC5, using data-dependent rotations, modular addition, multiplication, and XOR operations. Their work provides the algorithmic foundation for implementing RC6 encryption and decryption using Verilog HDL.Nechvatal et al. [2] evaluated AES candidate algorithms, including RC6, MARS, Rijndael, Serpent, and Twofish. Their research highlights the importance of comparing cryptographic algorithms based on security and implementation efficiency.Nechvatal et al. [3] documented the AES selection process, which resulted in Rijndael being selected as the AES standard. Their study provides historical context for RC6 and clarifies the distinction between RC6 and AES.

Contini et al. [4] analyzed the security of RC6 against differential and linear cryptanalysis. Their research emphasizes the importance of evaluating the security of the algorithm in addition to verifying its encryption and decryption functions.Beuchat [5] investigated RC6 implementations on Xilinx FPGA devices. The study examined hardware architectures and implementation efficiency, providing a foundation for developing RC6-based digital circuits.

Hashim et al. [6] proposed a modified RC6-Cascade encryption and decryption architecture on an Altera FPGA. Their work demonstrates how cryptographic hardware architectures can be adapted for different implementation requirements.Miyaji and Nonaka [7] analyzed

reduced-round RC6 without whitening operations. Their research highlights potential security weaknesses in modified cipher configurations and emphasizes the importance of correct algorithm implementation.Borst et al. [8] studied linear cryptanalysis techniques related to RC5 and RC6. Their work demonstrates the importance of analyzing cryptographic algorithms for vulnerabilities and preserving their intended operations.

Fernandes et al. [9] reviewed VLSI implementations of cryptographic algorithms. Their research explains how encryption techniques can be implemented in digital hardware and supports the use of HDL-based design and simulation.Al-Gailani [10] studied FPGA implementations of AES candidate algorithms, including RC6, MARS, and Twofish. The research considered hardware parameters such as throughput, operating frequency, resource utilization, and power consumption, which are useful for evaluating cryptographic hardware performance.

III. EXISTING SYSTEM

The existing system uses conventional cryptographic algorithms and hardware implementations to provide secure data encryption and decryption. These systems implement encryption operations using software or digital hardware, but their performance depends on the algorithm, architecture, and available resources. Existing implementations may require optimization to achieve better throughput, reduced hardware resource utilization, and efficient key management. Therefore, an improved RC6 implementation using Verilog HDL can be developed and evaluated through simulation to verify its functional correctness and hardware design.

IV. PROPOSED SYSTEM

The proposed system implements the RC6 encryption and decryption algorithm using Verilog Hardware Description Language (HDL). The design consists of a datapath, key expander, key selector, and controller to manage data processing and encryption operations. The system is simulated using Xilinx ISE 13.7 and ISim to verify the functionality of the design. This modular architecture supports structured implementation and testing of cryptographic operations. Further synthesis and FPGA testing can be performed to evaluate hardware parameters such as speed, resource utilization, and power consumption.

V. METHODOLOGY

The proposed system begins with the study of the RC6 symmetric-key encryption algorithm and its mathematical operations. The input plaintext and secret key are provided to the system, and the key expansion module generates the round keys required for encryption and decryption. The datapath performs arithmetic and logical operations, including modular addition, XOR, multiplication, and data-dependent rotation. The key selector supplies the appropriate round keys to the datapath, while the controller manages the sequence of operations.

The design is implemented using Verilog Hardware Description Language (HDL). The system is divided into functional modules, including the datapath, key expander, key selector, and controller, to simplify design and verification. During encryption, the plaintext is divided into four 32-bit registers represented by (A), (B), (C), and (D). Modular addition is represented as (X=(A+B)\bmod 2^{32}), while the XOR operation is expressed as (X=A\oplus B). These operations help process the input data during encryption.

Circular left rotation is another important operation used in RC6. It is represented by the formula

(\operatorname{ROTL}_{32}(X,n)=(X\lln)\lo

r(X\gg(32-n))), where (X) is a 32-bit value and (n) is the rotation amount. This operation rotates the bits within a 32-bit register and contributes to the internal transformation of data. The algorithm combines these operations with multiplication and round-key additions to produce the encrypted output.

For the standard RC6-32/20/16 configuration, the algorithm uses 20 rounds and generates 44 round-key words of 32 bits each. The initial register values are updated using the expanded key according to (B=(B+S[0])\bmod 2^{32}) and (D=(D+S[1])\bmod 2^{32}), where (S) represents the expanded key array. During each round, intermediate values are calculated as (t=\operatorname{ROTL}{32}(B(2B+1),5)) *and* (u=\operatorname{ROTL}{32}(D(2D+1),5)). The register updates are then performed using (A=\operatorname{ROTL}{32}(A\oplus t,u)+S[2i]) *and* (C=\operatorname{ROTL}{32}(C\oplus u,t)+S[2i+1]), with register arithmetic performed modulo (2^{32}). These equations describe the main round transformations of the standard RC6 algorithm.

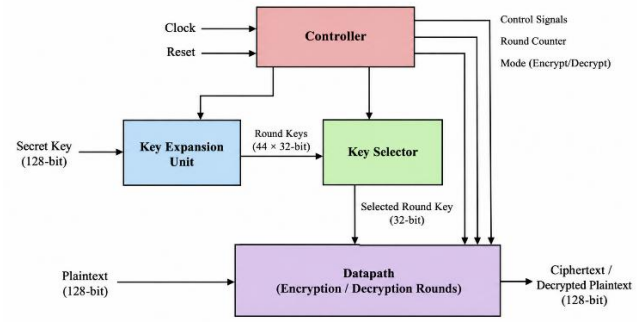
The key expansion process generates the round keys required for encryption and decryption. For RC6-32/20/16, the initial key schedule uses the constants (P_{32}=\texttt{0xB7E15163}) and (Q_{32}=\texttt{0x9E3779B9}). The key array is initialized using (S[0]=P_{32}) and (S[i]=(S[i-1]+Q_{32})\bmod 2^{32}), followed by mixing operations involving the secret key words. After the encryption rounds, the final register values are updated using the corresponding round keys, and the ciphertext is formed by concatenating the four registers: (C_{\text{cipher}}=A\parallel B\parallel C\parallel D).

During decryption, the inverse RC6 operations are applied in reverse order using the same secret key and the corresponding expanded round keys. The main objective is to recover the original plaintext from the ciphertext. The functional correctness of the encryption and decryption process is expressed as (D_K(E_K(P))=P), where (P) is the original plaintext, (E_K) represents encryption using key (K), and (D_K) represents decryption using the same key.

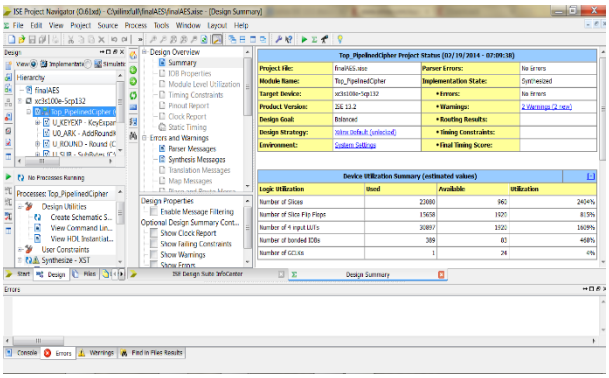
The complete design is simulated using Xilinx ISE 13.7 and ISim. Test inputs, secret keys, ciphertext, and decrypted outputs are examined using simulation waveforms. The correctness of the design is verified by checking whether decryption reproduces the original plaintext when the same valid secret key is used. Further synthesis and FPGA implementation can be performed to measure hardware resource utilization, operating frequency, throughput, and power consumption.

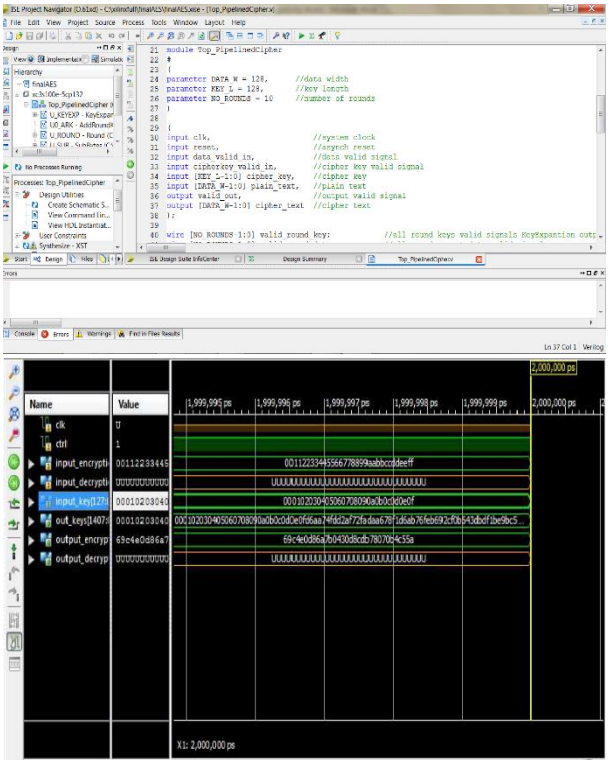
VI. SYSTEM MODEL

System Architecture



VII. RESULTS AND DISCUSSIONS





VIII. CONCLUSION

The proposed system, "Advance RC6 Implementation for Security Appliances," focuses on implementing the RC6 encryption and decryption algorithm using Verilog Hardware Description Language (HDL). The system consists of a datapath, key expansion unit, key selector, and controller to perform cryptographic operations in a structured manner. The design is simulated using Xilinx ISE 13.7 and ISim to verify the functionality of encryption and decryption. The correctness of the system can be evaluated by checking whether the decrypted output matches the original plaintext. This implementation provides a foundation for hardware-based data security applications. Further improvements can be achieved through FPGA implementation and evaluation of hardware parameters such as speed, power consumption, and resource utilization.

IX. FUTURE WORK

In the future, the proposed RC6 implementation can be improved by implementing the design on FPGA hardware and evaluating its real-time performance. The system can be optimized to reduce hardware resource utilization, power consumption, and encryption latency while improving throughput. Additional testing can be performed with different secret keys and input data to verify the correctness and reliability of encryption and decryption. The design can also be integrated into security appliances, embedded systems, and secure communication applications. Further work may focus on hardware optimization, improved key management, and comprehensive security analysis to enhance the efficiency and reliability of the proposed system.

X. REFERENCES

[1] R. L. Rivest, M. J. B. Robshaw, R. Sidney, and Y. L. Yin, "The RC6 Block Cipher," 1998. Available: https://people.csail.mit.edu/rivest/pubs/RRSY98.AES1.slides.pdf

[2] J. Nechvatal et al., "Report on the Development of the Advanced Encryption Standard (AES)," *Journal of Research of the National Institute of Standards and Technology*, vol. 106, no. 3, 2001. Available: https://doi.org/10.6028/jres.106.023

[3] J. Nechvatal et al., "Status Report on the First Round of the Development of the Advanced Encryption Standard," *Journal of Research of the National Institute of Standards and Technology*, vol. 104, no. 5, 1999. Available: https://doi.org/10.6028/jres.104.027

[4] S. Contini, R. L. Rivest, M. J. B. Robshaw, and Y. L. Yin, "The Security of the RC6 Block Cipher," RSA Laboratories, 1998. Available: https://people.csail.mit.edu/rivest/pubs.html

[5] J.-L. Beuchat, "FPGA Implementations of the RC6 Block Cipher," in *Proceedings of the 13th International Conference on Field Programmable Logic and Applications*, 2003, pp. 101–110. Available: https://doi.org/10.1007/978-3-540-45234-8_11

[6] A. Hashim, H. Hasan, and H. Abbas, "Design and Implementation of Proposed 320 Bit RC6-Cascaded Encryption/Decryption Cores on Altera FPGA," *International Journal of Electrical and Computer Engineering*, vol. 10, no. 6, pp. 6370–6379, 2020. Available: https://doi.org/10.11591/ijece.v10i6.pp6370-6379

[7] A. Miyaji and M. Nonaka, "Cryptanalysis of Reduced-Round RC6 without Whitening," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E86-A, no. 1, pp. 19–30, 2003.

[8] J. Borst, B. Preneel, and J. Vandewalle, "Linear Cryptanalysis of RC5 and RC6," in *Fast Software Encryption*, Lecture Notes in Computer Science, vol. 1636, pp. 16–30, 1999. Available: https://doi.org/10.1007/3-540-48519-8_2

[9] D. Fernandes, A. Dungarwal, S. Gaikwad, A. Kareliya, and S. Shilaskar, "VLSI Implementation of Cryptographic Algorithms and Techniques: A Literature Review," arXiv:2110.13407, 2021. Available: https://arxiv.org/abs/2110.13407

[10] M. A. Al-Gailani, "AES Cipher's Candidates: Design and FPGA Implementation," *Journal of Internet Services and Information Security*, vol. 15, no. 1, pp. 51–66, 2025. Available: https://doi.org/10.58346/JISIS.2025.I1.004