Analog Integrated Circuits and Signal Processing
https://doi.org/10.1007/s10470-024-02290-z

FPGA-based implementation and verification of hybrid security algorithm for NoC architecture

T. Nagalaxmi[1] · E. Sreenivasa Rao[2] · P. ChandraSekhar[3]

Received: 12 May 2023 / Revised: 19 April 2024 / Accepted: 1 September 2024

Abstract
Networks on Chip (NoCs) are a crucial component in modern System on Chips (SoCs), which provide the communication infrastructure for various processing elements such as CPUs, GPUs, DSPs, and other IPs. As a result, security is a critical aspect of NoCs, and it is essential to protect them from various security threats such as information leakage, denial of service attacks, and unauthorized access. The communication over NoCs carries sensitive and confidential information, which needs to be protected from unauthorized access, interception, or tampering. A Hybrid Secure technique is proposed in this research paper to protect the data during NoC transmission. The Noekeon and RSA algorithms are combined to create the hybrid secure algorithm for NoC architecture. The Noekeon algorithm provides a high level of security, efficiency, flexibility, and resistance to side-channel attacks, making it an ideal choice for securing communication in NoC and other applications. The RSA encryption algorithm is modified to minimize the number of calculations. The proposed hybrid secure algorithm is tested on 4×4 2D mesh NoC architecture. The average throughput of the proposed algorithm is increased to 64% and 51% latency is reduced when compared to existing research work.

Keywords NoC · Noekeon · RSA · Encryption · Low latency

1 Introduction

A Network on Chip (NoC) is a communication architecture that provides a scalable and efficient way of interconnecting the various components of a system-on-chip (SoC) [1]. It is a packet-switched, many-to-many communication infrastructure that uses network protocols to move data between different processing elements and memory blocks within the chip. NoCs are used in complex SoCs, such as those found in high-performance computing, embedded systems, and multimedia applications, to improve system performance, reduce power consumption, and simplify chip design [2].

NoC technology is used in a wide range of applications that require high-performance and scalable communication between various components within an SoC [3]. Some common applications of NoCs include; High-performance computing (HPC). NoCs are used in HPC systems to connect multiple processing elements and memory blocks, such as CPUs, GPUs, and FPGAs, to improve system performance and reduce latency [4]. Embedded systems; NoCs are used in embedded systems to interconnect various functional blocks, such as processors, memories, and peripherals, to reduce power consumption and improve performance. Multimedia applications; NoCs are used in multimedia applications, such as video decoding and image processing, to connect different processing elements, such as DSPs, to improve system performance and reduce power consumption [5]. Automotive systems; NoCs are used in automotive systems, such as advanced driver assistance systems (ADAS) and autonomous vehicles, to connect various sensors and processing elements to enable real-time processing and

✉ T. Nagalaxmi
tnagalaxmi@stanley.edu.in

E. Sreenivasa Rao
e.sreenivasarao@staff.vce.ac.in

P. ChandraSekhar
sekharpaidimarry@gmail.com

[1] Department of Electronics and Communication Engineering, Stanley College of Engineering and Technology for Women(A), Abids, Hyderabad, Telangana 500001, India

[2] Department of Electronics and Communication Engineering, Vasavi College of Engineering, Hyderabad, Telangana 500031, India

[3] Department of Electronics and Communication Engineering, Osmania University, Hyderabad, Telangana 500007, India

Published online: 14 September 2024

decision-making. Internet of Things (IoT); NoCs are used in IoT devices to interconnect various sensors, processors, and wireless communication modules, to enable real-time data processing and decision-making.

Security is critically important in NoC design because NoCs are used in a wide range of applications where sensitive information is being processed and transmitted [6]. NoCs connect multiple processing elements and memory blocks within an SoC, making them vulnerable to various security threats, such as unauthorized access, data interception, data tampering, and Denial of Service (DoS) attacks [7]. A security breach in a NoC can lead to a variety of consequences, depending on the application. In some cases, it can lead to the theft of sensitive data, such as personal information, financial data, or confidential business information. In other cases, it can lead to the disruption or failure of critical systems, such as medical devices, aerospace systems, or industrial control systems.

Therefore, designing secure NoCs is essential to protect against these security threats and ensure the reliability and safety of the systems in which they are deployed [8]. This includes implementing security measures, such as access control, encryption, traffic monitoring, secure booting, and secure communication protocols, as well as testing and verifying the security of the NoC design to ensure that it meets the required security standards [9].

Encryption is an important security measure for Network on Chip (NoC) designs [10]. It can protect the confidentiality and integrity of the data being transmitted over the NoC, making it more difficult for attackers to intercept or tamper with the data. Traditional security methods, such as firewalls and intrusion detection systems, are designed to protect against external attacks on a network. However, these methods may not be effective in securing NoC designs because NoCs are typically deployed within a single chip or SoC, and the security threats may come from within the chip or SoC. Conventional encryption methods may not be suitable for NoC designs because NoCs typically involve communication between multiple cores [11] or processing elements, and traditional encryption schemes may introduce significant overhead in terms of latency, area, and power consumption [12]. This is because encrypting and decrypting large amounts of data requires significant computational resources and can result in significant delays, especially in high-performance NoC designs. As a result, alternative encryption approaches, such as lightweight encryption schemes or hardware-based security solutions, may be more suitable for NoC designs.

In addressing the critical security needs of NoC architectures, this study emphasizes the implementation of advanced encryption methods to safeguard sensitive data. A hybrid security algorithm that integrates the robustness of the Noekeon and RSA algorithms is employed to optimize for minimal computational overhead while improving security against unauthorized access and data breaches. Noekeon provides rapid encryption with high resistance to common cryptographic attacks, making it suitable for real-time applications, while the modified RSA algorithm employed in this approach reduces the computational complexity traditionally associated with such strong encryption methods. This strategic combination of encryption technologies underpins the focus of our research, leading seamlessly into the detailed investigation and verification of these techniques within an FPGA-based 4×4 2D mesh NoC architecture. Thus, this research work focuses on FPGA-based implementation and verification of a hybrid security algorithm for NoC architecture to provide real-time data security in the network in a real-time environment.

2 Related studies

A tier-based, reconfigurable security architecture that is capable of adapting to a variety of use-case situations is proposed by Charles and Mishra [13]. The authors investigated how to design an effective reconfigurable architecture that is capable of supporting three common NoC security mechanisms (encryption, authentication, and denial-of-service attack detection and localization), and the authors implemented suitable techniques for dynamic architecture reconfiguration. The authors assessed the proposed framework by running common benchmarks that enabled varying tiers of security. They also provided a complete study of how different degrees of security might affect application performance as well as energy efficiency and area overhead.

Harttung et al. [14] proposed three innovative and lightweight techniques to protect communication in NoCs. Encryption, authentication, and network coding are the three components that are included in the provided solutions to assure resilience, secrecy, and integrity. As a result of the importance placed on performance in NoC settings, the proposed solutions place a special emphasis on having short latencies and a small chip area. The effectiveness of the proposed methods was assessed using exhaustive software simulations.

A data encryption framework for NoC was presented by Ayachi et al. [15] and was based on an algorithm for a light encryption device (LED). The primary benefits offered by the proposed algorithm are a reduction in the implementation area as well as an increase in speed alongside a reduction in the amount of power that is used. The proposed encryption system was tested using Verilog/VHDL simulations run on the Xilinx ISE, and it was tested in Xilinx Virtex 5 XC5VFX200T device. The findings that were gathered demonstrated that the proposed structure has a smaller area and a greater speed in comparison to the works that are already in place. The new technique has improved network

performance in terms of both speed and security, while simultaneously decreasing the amount of NI implementation space required.

Singh et al. [16] proposed a Proffering Secure Energy-Aware Network-on-Chip (NoC) employing Incremental Cryptogine for secure data transmission and power and space reduction to improve performance. Incremental Cryptogine uses network interface NoC features for lightweight encryption and safe data transmission. Aberrant Congestion Pattern Detection detects unexpected data transmission traffic. Spiking neural networks classify traffic and decode encrypted data into the original packet at the destination node for safe data transport. Cognizest Bipartite Buffer creates a unified buffer with input ports on NoC and saves under-utilized buffer capacity by prioritizing switch traversals to provide competent performance for power and space savings. Hence, utilizing an incremental cryptogine to protect data in Network-on-chip (NoC) reduces power and space while improving attack detection.

Charles and Mishra [17] offered a simple encryption method that may be carried out on the network interface. By using incremental cryptography, which makes use of the particular qualities that NoC traffic has, the proposed method can boost the efficiency of encryption without lowering its level of protection. The results of experiments show that our suggested method may save the time required for encryption by up to 57%, or by 30% on average, compared to more conventional methods, all while having a small influence (less than 2%) on the amount of area overhead.

Haase et al. [18] provided a communication protocol that establishes secure end-to-end communication between the NoC nodes. The protocol is based on authenticated encryption and includes recovery methods. In addition to this, the chosen key agreement strategy that is necessary for secure communication has been put into action. Each processor element has a network adapter, which houses the functionality necessary for maintaining security. Recovery procedures guarantee that faulty data is retransmitted by the network adapter without the need for intervention from the processing element if data is altered or erased while being transmitted. This occurs if the data is corrupted. The authors simulated the whole system using SystemC TLM and implemented it on the NoC simulation platform PANACA.

Kumar et al. [19] reported on the proposed work, which is meant to be used in the process of developing an NoC architecture via the incorporation of an enhanced TACIT security algorithm in Virtex-5 FPGA. The authors of this work made use of a hash function that is classified as a 4-H scheme hash function. The main advantage of this key generation scheme is it is applicable for block size and key size up to "n ‘’ bit. Thus, this TACIT security technique allows “n” bit by using the software VHDL programming language in Xilinx ISE 14.2 and Modelsim 10.1 b. These two are appropriate for 1024-bit and "N " bits of block size on Virtex-5 FPGA devices. This design system requires improvement in areas such the timing parameters, supporting memory, higher frequencies, and used summaries.

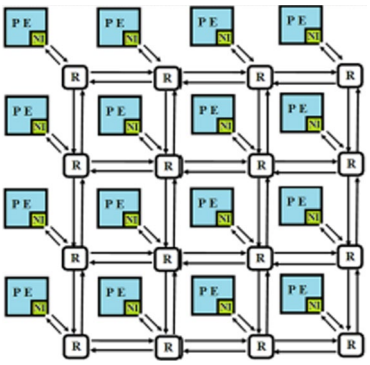
Harttung et al. [14] described three unique lightweight ways for securing communication in NoCs. Encryption, authentication, and network coding are the three components that are integrated into the proposed solutions to assure resilience, secrecy, and integrity. As a result of the importance placed on performance in NoC settings, the suggested solutions place a special emphasis on having short latencies and a small chip area. The effectiveness of the suggested methods was assessed using exhaustive software simulations. The findings have shown that the aforementioned advantages exceed the performance decrease that is caused by protective measures. In addition, the space increase necessitated by the new components is not a very significant one.

3 Proposed method

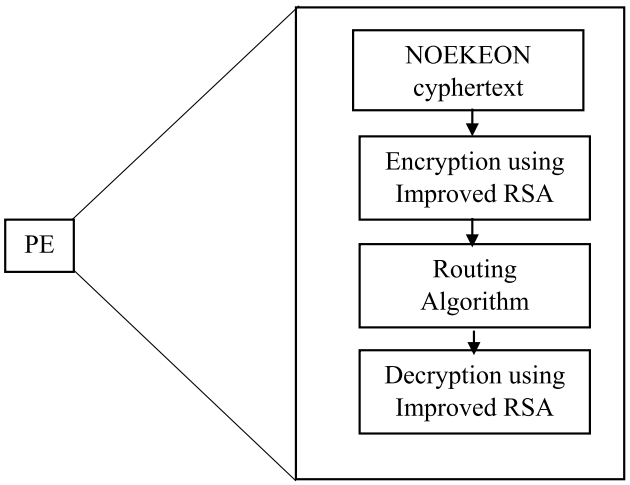
This section describes the proposed NoC system architecture with an encryption algorithm. Taking into account the specifications of the system, a 4×4 mesh topology will be constructed. The following is a description of the architecture: it would be composed of three primary parts, such as links, network interface modules, and routers. The links will function as the physical connecting medium that sets up the efficient communication system between all of the processing parts and the routers. This will be accomplished by connecting the links to each other.

The Network interface module is the second most important component in the design of the architecture for the NoC. This module will consist of all of the processing elements in which all of the logical operations will be designed and carried out with IP cores, which is where the network protocols will be called and used. The encryption operation will be performed by each network interface, and their respective network protocols will be created to route data quickly and effectively to the intended target port without causing any blocking difficulties on the network.

In terms of the design of the NoC architecture, the routers will serve as the last building piece. The data will be directed by these routers from the source to the destination that you choose. The processing components will be used to build the control logic for the routers, and the data will be routed by that logic. The source port address and the destination port address are sent to these routers through shared networks. The routers then forward data packets with the port address that is specified. The Network on chip architecture with the proposed Hybrid Security algorithm is shown in Fig. 1.



(a) 4x4 2D Mesh NoC architecture



(b) Proposed Hybrid Secure Algorithm

Fig. 1 Proposed NoC Hybrid Secure Architecture **a** 4×4 2D Mesh NoC architecture, **b** Proposed Hybrid Secure Algorithm

The interior perspective of each processing element is shown in Fig. 1b., which may be seen here. It is made up of three main sections, which are as follows:

- Block of encryption using the NOEKEON algorithm
- The block for the routing algorithm
- Block decryption for the NOEKEON algorithm

Plain text will be encrypted using the NOEKEON algorithm in the encryption block, and then it will be sent to the route analyzer block. This encrypted data will be successfully sent to the target port with no loss of data packets occurring at any point in the process. In the end, this will be handed over to the NOEKEON decryption block, which is where the encrypted content will be decoded and turned back into plain text.

3.1 Noekeon

Noekeon has a block size of 128 bits and supports key sizes of 128, 192, and 256 bits. It is based on the AES structure, but with a different round function that is designed to be more resistant to certain types of cryptanalytic attacks. The Noekeon round function consists of several operations, including a key addition step, a substitution step using an S-box, a linear diffusion step, and a permutation step. The algorithm uses a Feistel network structure with four rounds for the 128-bit key size, six rounds for the 192-bit key size, and eight rounds for the 256-bit key size. Noekeon has been extensively analyzed by the cryptographic community and is secure against various types of attacks, including differential and linear cryptanalysis.

Encrypting the data sent over a network using the Noekeon algorithm does to make the transmission of network data more secure. The proposed Noekeon algorithm is made up of a total of sixteen SP nets. Each cycle incorporates 32 concurrent 4-bit-4-bit S-boxes, in addition to linear transformation and byte rotation. Since each fundamental component is paired with its counterpart, the system's encryptor and decryptor are quite comparable to one another. Data on a network may be encrypted by each basic module if it uses the Noekeon algorithm. The Gamma module of the Noekeon algorithm is the first component of the method to be put into action when the encryption is carried out.

The value of the Boolean function $f(x, y)$ is set to be equal to zero and one, the verification of the Boolean function is carried out, and the probability of 0.7 or 0.2 is linearly infinitely approximated to $f(x, y)$. This allows for the Gamma module of the Noekeon algorithm to produce a linear approximation result with a higher probability. Based on this, the assumption is made in the diffusion layer of the Noekeon algorithm that an input does not need to consider the sub-key. The following is a representation of the operational connection that exists between the input and the output in the form of a matrix:

$$\begin{bmatrix} Y_0 \\ Y_1 \\ Y_2 \end{bmatrix} = M \otimes \begin{bmatrix} X_0 \\ X_1 \\ X_2 \end{bmatrix} \tag{1}$$

In the formula, Y0, Y1, and Y2 represent the output results of the Noekeon algorithm's diffusion layer; X0, X1, and X2 represent the input values of the Noekeon algorithm's diffusion layer; and M represents the linear approximation coefficient. The data on the network is encrypted using the matrix that was built. The Noekeon algorithm needs to be combined with the RSA mode for this process to work. Additionally, the Noekeon algorithm needs to be encrypted and decrypted twice before it can be used. This effectively resolves the security issues that were caused by the ease of

use of the Noekeon algorithm in the arrangement. According to what was said before, the round constants are chosen in the following order: from left to right for the Noekeon algorithm, and from right to left for the Noekeon algorithm. There is also a nonlinear function known as "Gamma," a key function known as "Theta," and mobile operation modules known as "Pi1" and "Pi2," all of which employ this structure to finish encrypting network data.

3.2 Improved RSA algorithm

The RSA technique is mostly used for the subsequent encryption of the Noekeon key, and the decomposition level of the integer component is the primary determinant in determining the extent of its security. The RSA algorithm primarily consists of three links when it comes to the structure of its design and its execution, and those connections are key generation, information encryption, and information decryption. The RSA algorithm needs a significant number of prime numbers, each of which must be 2048 bits long. As a result, to encrypt information using the RSA method, it is necessary to carry out a significant number of complicated and extensive arithmetic operations. In this manner, while the RSA method may successfully increase the security of network data, the efficiency with which it operates is unavoidably on a lower level in comparison to that of other encryption algorithms. As a result, the data encryption transmission method needs to continue to be based on the Noekeon algorithm, and the RSA technique should be the sole one used for the encryption of the Noekeon key.

The RSA algorithm works as follows:

1. **Key generation:** Choose two large prime numbers p and q. Calculate their product $n = p * q$. Choose a positive integer e that is relatively prime to $(p-1)(q-1)$, and calculated such that $d * e = 1\ (mod(p-1)(q-1))$. The public key is (n, e) and the private key is (n, d).
2. **Encryption:** To encrypt a message m, convert it to an integer value. Calculate $c = m\text{^}e\ (mod\ n)$. The encrypted message is c.
3. **Decryption:** To decrypt the message c, calculate $m = c\text{^}d\ (mod\ n)$. decrypted message is m.

The security of the RSA algorithm is based on the difficulty of factoring large numbers. The security of the algorithm relies on the fact that it is computationally infeasible to find the prime factors of n. If an attacker can factor n, then they can easily calculate the private key and decrypt messages. To enhance the security of the RSA algorithm, a larger key size is used. The larger the key size, the more difficult it is to factor the product n. A typical key size used today is 2048 bits, which provides a high level of security against attacks. The security of the RSA algorithm relies on the difficulty of factoring large composite numbers into their prime factors. However, with the advent of powerful computing resources, the original RSA algorithm has become vulnerable to attacks. The formation and evaluation of large prime numbers as well as the computation of modular power are the procedures that take the most time during the operating process of the RSA algorithm. In the improved RSA algorithm, the fast calculation of the modular power is discussed below:

The RSA algorithm makes use of the sliding window approach, which was chosen because it allows for quick computation of $c\text{^}d(modn)$. The modularization of the index e in c^e is the fundamental idea behind the sliding window approach.

For instance, $e = (154548551155)_{10} = (00111001100100101110010111111100101 1)_2$ indicates that the preceding binary integers are grouped from the left and that the length of the grouping is 3 bits. This is the case if the length of the setting window is three. The concept of grouping is to make sure that the first bit of each group is a 1, so that a portion of the 0 in the center may be passed over. In the very end, when the length is less than three digits, the numbers that are still present come together to create a group apart from 0 and 1. After the grouping step, the various methods of grouping that may be used to enhance the effectiveness of the modular

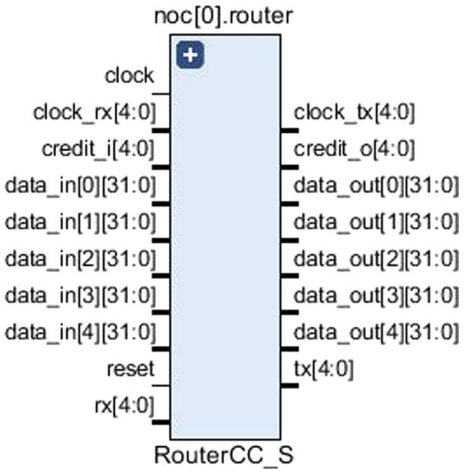


Fig. 2 The RTL schematic of NoC router

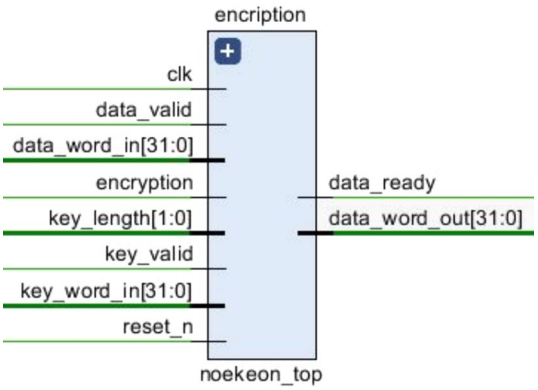


Fig. 3 RTL schematic of Noekeon encryption algorithm

exponentiation process are explored. The sliding window method's primary mode of operation, as determined by the calculations and analysis above, is to pre-process the index to minimise the number of calculations.

4 Results and discussions

The proposed Noekeon encryption method for NoC is created using Xilinx Vivado 2019.2. To test the functionality and performance of the proposed design in terms of space and power, it is built on the Zed board. The proposed NoC architecture works for 16 and 32-bit flit and tested for depth buffer considered is 4. The 4×4 mesh topology is considered for the realization of the proposed encryption algorithm. The routing algorithm is responsible for determining the path a packet should follow from the source to the destination. The routing algorithm can be deterministic or adaptive, and it should ensure efficient packet delivery, low latency, and high throughput. In this NoC architecture, a deterministic shortest-path routing algorithm is used. Flow control is a mechanism to manage the packet flow in the network and prevent congestion or deadlock. Flow control can be implemented using credit-based schemes, buffer-based schemes, or hybrid schemes. In this work a credit-based scheme is incorporated. The Register Transfer Level (RTL) schematic of NoC architecture is depicted in Fig. 2.

In Fig. 2, the RTL schematic of NoC router consists seven I/O pins. The clock_rx and clock_tx signals are used for the purpose of reception and transmission of data packets. The data_in and data_out signals are a packed array consists of 32-bit width and 5 depths. These five data signals are used for North, West, South, East and local ports respectively. Credi_i and credit_o are the flow control signals. Based on this NoC configuration, the Noekeon encryption algorithm is added to the router module. The RTL schematic of Noekeon encryption block is shown in Fig. 3.

From Fig. 3, clk and reset_n are the input signals to the Noekeon block. The data_valid represents flag to enable the data input and data_word_in signal is used to send 32-bit word at a time. The encryption signal is acting as a switch that enable encryption or decryption. If encryption signal asserted with 1, the encryption will be performed. Similarly, if encryption signal is asserted with 0, the decryption will be performed. The key_valid represents flag to enable key input having 2-bit word length configured by key_length signal. The key_word_in signal used as key input having 32-bit word at a time. The data_ready signal is an output signal which acting as flag to indicate the beginning of ciphertext output. The data_word_out signal also an output signal is used for the ciphertext output has 32-bit word length at a time. The packet transmission packet format is defined in a constant parameter. The packet transmission structure is described below:

Start Size src tgt Encryption

$$tpacket = \begin{matrix} ((0, & 2*FB, & 0, & 8, & '1'), \\ (0, & FB, & 2, & 6, & '1'), \\ (20, & 4*FB, & 0, & 7, & '1'), \\ (12, & 5*FB, & 8, & 0, & '1'), \\ (400, & 20*FB, & 2, & 3, & '1')); \end{matrix}$$

In the transmission packet, Start is the expected time to inject the packet, size is a multiple of FB (Flits per Block). The src and tgt are the source and target of the flit travels. The encryption defines the encryption or decryption process. Based on this configuration, the proposed NoC architecture with Noekeon algorithm is simulated in Xilinx Vivado tool.

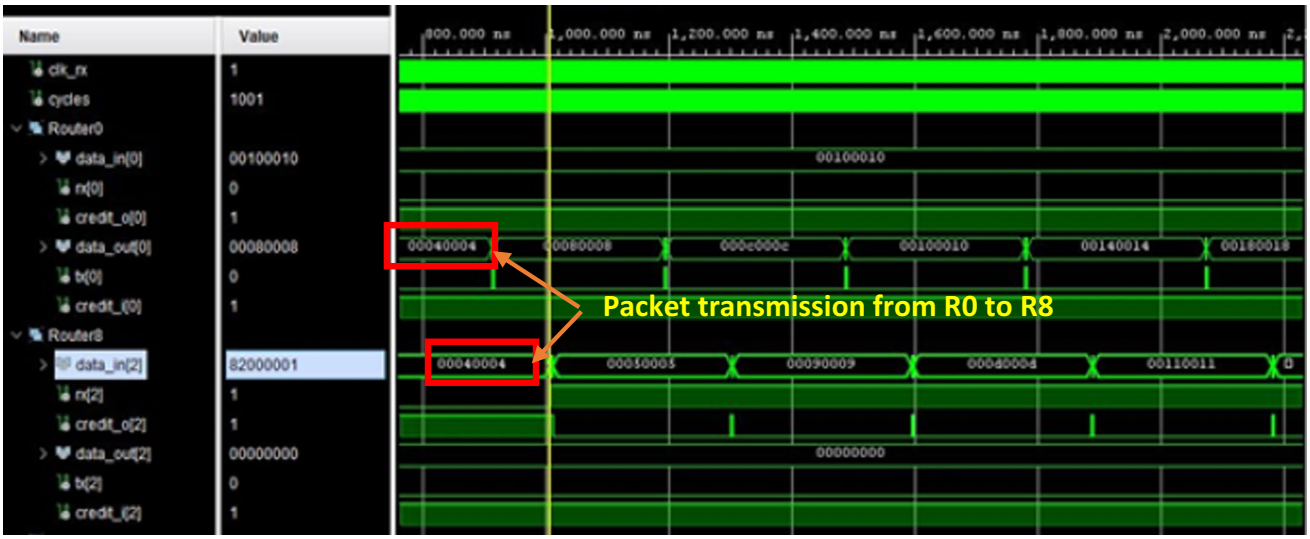


Fig. 4 The packet transmission from Router R0 to R8

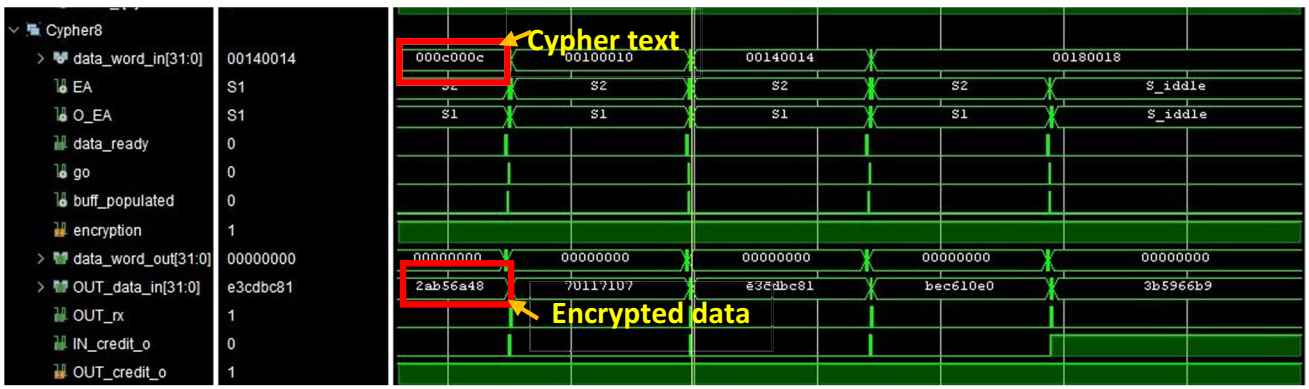


Fig. 5 The cyphertext and encryption results of the proposed model

Fig. 6 The Decryption simulation results

The packet transmission from Router-0 to Router-8 is shown simulation waveform which is depicted in Fig. 4.

From Fig. 4, the data_out signal of t Router-0 is ejecting the packet (0004004) to the destination Router-8. The data_in signal of Router-8 is receiving the packet which is transmitted from Router-0. This is the initial transmission from Router-0 to Router-8 and the size of this transmission is $2 * FB$ which means two flits per block. In between the

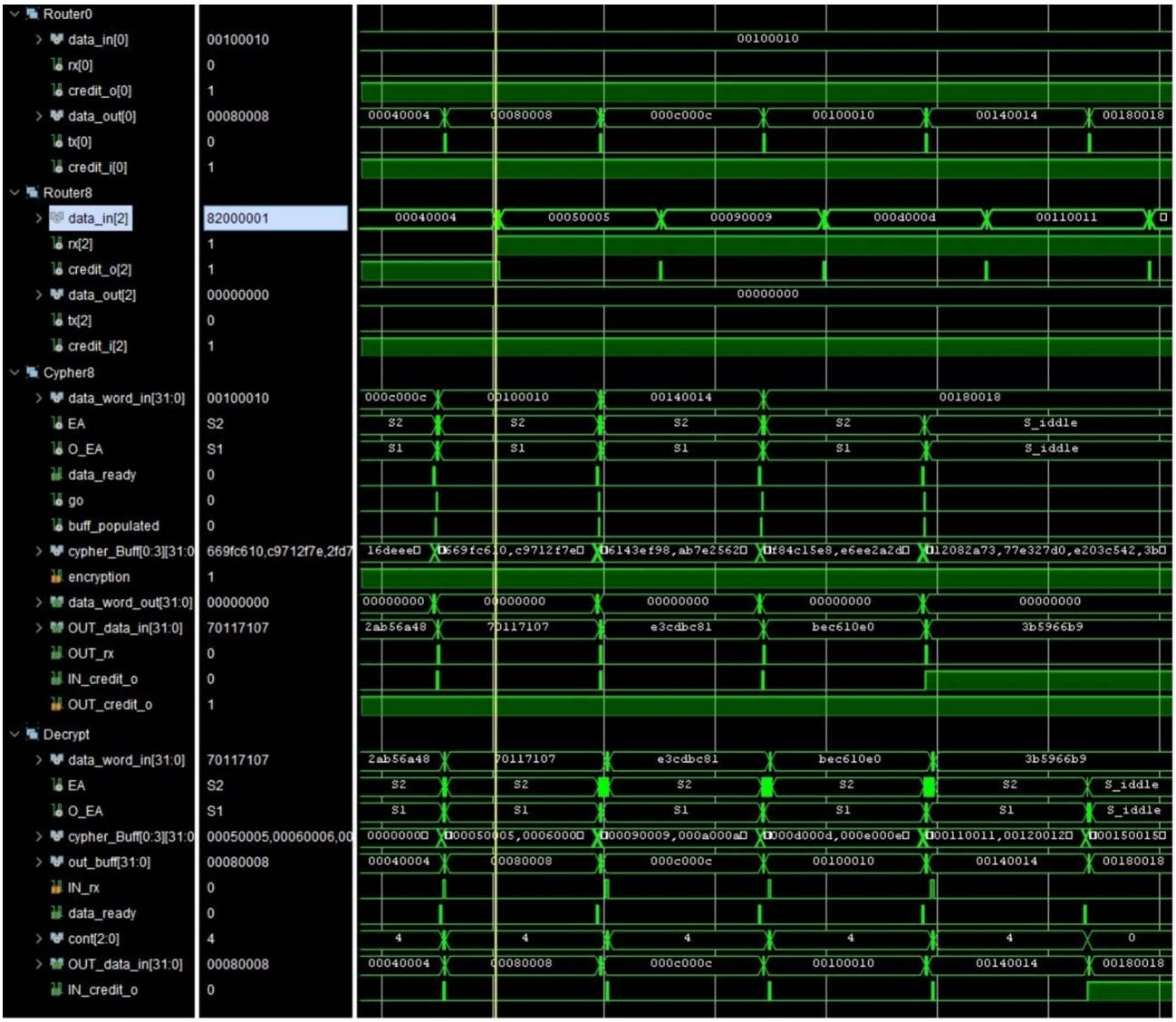


Fig. 7 ILA simulation results of proposed NoC architecture in Zynq-SoC

transmission of the cypher text conversion, Encryption and Decryption is performed. The cypher text conversion is done by using Noekeon algorithm. The Encryption is performed by using improved RSA algorithm. The corresponding cyphertext conversion and encryption simulated results are shown in Fig. 5.

From Fig. 5, the cypher block consists data_word_in signal which converts the actual data into to cyphertext. The converted cyphertext is further encrypted in the encryption block and which is denoted by OUT_data_in signal. After performing the encryption, the decryption operation is performed. The simulation waveform of the decryption is shown in Fig. 6.

From Fig. 6, the Decrypt block receives the encrypted data which can be represented by using data_word_in signal. The encryption and decryption are performed by using an improved RSA algorithm. After a successful decryption operation, the original data is obtained from OUT_data_in signal. The proposed NoC architecture is ported on Zynq-7000 all programmable SoC devices. The results of SoC implementation is observed in Integrated Logic Analyzer (ILA) core and shown in Fig. 7. The Fig. 7 represents the results which obtained from SoC device which are previously tested in the simulation environment. The corresponding simulation results are explained with by using Figs. 4, 5 and 6. The same simulations are tested in ILA core and verified the results.

Along with this secure NoC, the latency is also estimated. While performing the simulations, the latency and

Table 1 Comparison of Latency of the Proposed NoC architecture without and with Encryption and Conventional method

Packet transaction	Latency (ns) without encryptionproposed model	Latency (ns) with encryptionproposed model	Latency (ns) with encryptionmodel [20]
R2-R6	62	271	316
R0-R8	63	342	360
R8-R0	72	576	620
R0-R7	84	492	509
R2-R3	183	1740	5244

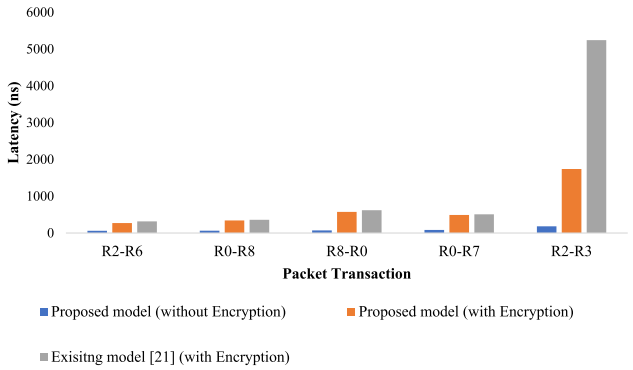


Fig. 8 Graphical Representation of Latency of proposed method without, with congestion and Conventional method

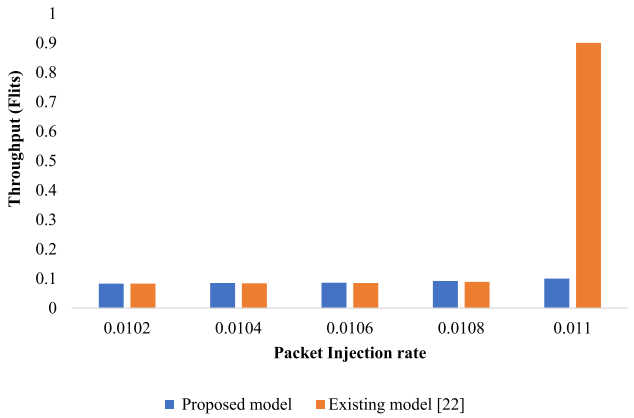


Fig. 9 Graphical Representation of Throughput of the proposed with Conventional method

Table 2 Comparison of Throughput of the proposed NoC architecture with Conventional method

packet injection rate	Throughput (Flits) of the proposed model	Throughput (Flits) of the model [21]
0.0102	0.083	0.083
0.0104	0.085	0.084
0.0106	0.086	0.085
0.0108	0.092	0.089
0.011	0.1	0.9

throughput of the proposed secure NoC architecture are captured in a text file for each transmission of the packet.

The performance of the NoC can be evaluated in terms of latency and throughput. Generally, NoCs are designed to have low latency and high throughput to meet the performance requirements of modern SoCs. The exact values of latency and throughput will depend on the specific design of the NoC and the applications it is intended to support. Latency refers to the time delay between the initiation of a communication and the time at which it is completed. In the case of the NoC, latency is the time it takes for a packet of data to travel from its source to its destination. The latency of the NoC is affected by various factors such as the distance between the source and destination, the number of nodes that the packet must traverse, and the routing algorithm used. The latency of the proposed NoC architecture is compared with state of art work [20] and the comparison results are reported in Table 1 and graphically presented in Fig. 8.

Throughput refers to the amount of data that can be transmitted over the network in a given period of time. In the case of the NoC, throughput is the amount of data that can be transmitted per unit time. The throughput of the NoC is affected by various factors such as the network topology, the bandwidth of the channels, and the routing algorithm used. The throughput of the proposed NoC architecture is compared with state of art work [21] and the comparison throughput results are reported in Table 2 and graphically presented in Fig. 9.

From Table 1 and 2, the proposed NoC architecture exhibits better results in terms of latency and throughput. The worst-case latency for a package is 140 cycles per block of 128 flits. The Noekeon with improved RSA algorithm takes 70 cycles to encrypt/decrypt (68 rounds). The latency has been decreased by 49% when compared to the Congestion aware router for Network-on-Chip [20] and 50% of throughput is increased when compared to the new

adaptive selection strategy for reducing latency in Network-on-Chip [21].

5 Conclusion

Security in Network on Chip (NoC) is crucial to ensure that the communication between processing elements such as CPUs, GPUs, and DSPs is secure and protected from various security threats. This paper aims to provide low latency based secure algorithm for NoC architectures. A Hybrid secure algorithm is implemented and incorporated into the 2D mesh network. The proposed algorithm is tested with different router transactions with different packet injection rate. The worst-case latency of the proposed algorithm is 140 cycles per block of 128 flits. The proposed method will require 70 cycles to encrypt and decrypt the data. As compared to state-of-the-artwork, throughput has grown by 50%, while latency has decreased by 49%.

Acknowledgements The authors wish to thank the Research Centre, Department of Electronics and Communication Engineering, Vasavi College of Engineering (Autonomous), Hyderabad for providing a supporting environment for carrying out this work.

Author contribution T. Nagalaxmi—writing and implementation, E. Sreenivasa Rao—conceptualization, P. Chandrasekhar—Proof reading

Funding There was no funding received for this study.

Data availability There was no data used in this study.

Declarations

Conflict of interests The authors declare no competing interests.

References

1. Oveis-Gharan, M., & Khan, G. N. (2020). Reconfigurable on-chip interconnection networks for high performance embedded SoC design. *Journal of Systems Architecture, 106*, 101711. https://doi.org/10.1016/j.sysarc.2020.101711
2. Arulananth, T. S., Baskar, M., & SM, U. S., Thiagarajan, R., Rajeshwari, P. R., Kumar, A. S., & Suresh, A. (2021). Evaluation of low power consumption network on chip routing architecture. *Microprocessors and Microsystems, 82*, 103809. https://doi.org/10.1016/j.micpro.2020.103809
3. Venkataraman, N. L., & Kumar, R. (2019). Design and analysis of application specific network on chip for reliable custom topology. *Computer Networks, 158*, 69–76. https://doi.org/10.1016/j.comnet.2019.03.014
4. Fotouhi, P., Fariborz, M., Proietti, R., Lowe-Power, J., VenkateshAkella, S. J., & Yoo, Ben. (2021). Hta: A scalable high-throughput accelerator for irregular hpc workloads. In Bradford L. Chamberlain, Ana-Lucia. Varbanescu, Hatem Ltaief, & Piotr Luszczek (Eds.), *High Performance Computing: 36th International Conference, ISC High Performance 2021, Virtual Event, June 24 – July 2, 2021, Proceedings* (pp. 176–194). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-78713-4_10
5. Kumar, A. S., & Rao, T. H. (2020). Scalable benchmark synthesis for performance evaluation of NoC core mapping. *Microprocessors and Microsystems, 79*, 103272. https://doi.org/10.1016/j.micpro.2020.103272
6. Mishra, P., & Charles, S. (2021). Trustworthy system-on-chip design using secure on-chip communication architectures. *Network-on-Chip Security and Privacy* (pp. 3–30). Cham: Springer International Publishing.
7. Sharma, G., Bousdras, G., Ellinidou, S., Markowitch, O., Dricot, J. M., & Milojevic, D. (2021). Exploring the security landscape: NoC-based MPSoC to Cloud-of-Chips. *Microprocessors and Microsystems, 84*, 103963. https://doi.org/10.1016/j.micpro.2021.103963
8. Mishra, P., & Charles, S. (2021). The Future of Secure and Trustworthy Network-on-Chip Architectures. In Prabhat Mishra & Subodha Charles (Eds.), *Network-on-Chip Security and Privacy* (pp. 483–489). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-69131-8_18
9. Sepúlveda, J. (2021). Secure Cryptography Integration: NoC-Based Microarchitectural Attacks and Countermeasures. In Prabhat Mishra & Subodha Charles (Eds.), *Network-on-Chip Security and Privacy* (pp. 153–179). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-69131-8_7
10. Indrusiak, L. S., Harbin, J., Reinbrecht, C., & Sepúlveda, J. (2019). Side-channel protected MPSoC through secure real-time networks-on-chip. *Microprocessors and Microsystems, 68*, 34–46. https://doi.org/10.1016/j.micpro.2019.04.004
11 Charles, S., & Mishra, P. (2021). Lightweight Encryption Using Incremental Cryptography. In Prabhat Mishra & Subodha Charles (Eds.), *Network-on-Chip Security and Privacy* (pp. 79–99). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-69131-8_4
12 Mishra, P., & Charles, S. (Eds.). (2021). *Network-on-chip security and privacy*. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-69131-8
13. Charles, S., & Mishra, P. (2020). Reconfigurable network-on-chip security architecture. *ACM Transactions on Design Automation of Electronic Systems (TODAES), 25*(6), 1–25. https://doi.org/10.1145/3406661
14. Harttung, J., Franz, E., Moriam, S., & Walther, P. (2019, May). Lightweight authenticated encryption for network-on-chip communications. In *Proceedings of the 2019 on Great Lakes Symposium on VLSI* (pp. 33–38). https://doi.org/10.1145/3299874.3317990
15. Ayachi, R., Mhaouch, A., & Ben Abdelali, A. (2021). Lightweight cryptography for network-on-chip data encryption. *Security and Communication Networks, 2021*, 1–10. https://doi.org/10.1155/2021/9943713
16. Singh, S., Ravindra, J. V. R., & Naik, B. R. (2022). Proffering secure energy aware network-on-chip (Noc) using incremental cryptogine. *Sustainable Computing: Informatics and Systems, 35*, 100682. https://doi.org/10.1016/j.suscom.2022.100682
17. Charles, S., & Mishra, P. (2020, July). Securing network-on-chip using incremental cryptography. In *2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)* (pp. 168–175). IEEE. https://doi.org/10.1109/ISVLSI49217.2020.00039.
18. Haase, J., Jaster, S., Franz, E., & Göhringer, D. (2022, July). Secure communication protocol for network-on-chip with authenticated encryption and recovery mechanism. In *2022 IEEE 33rd International Conference on Application-specific Systems, Architectures and Processors (ASAP)* (pp. 156–160). IEEE. https://doi.org/10.1109/ASAP54787.2022.00033.
19. Kumar, N. A., Shyni, G., Peter, G., Stonier, A. A., & Ganji, V. (2022). Architecture of network-on-chip (NoC) for secure data routing using 4-H function of improved TACIT

security algorithm. *Wireless Communications and Mobile Computing*, *2022*. https://doi.org/10.1155/2022/4737569
20. Balakrishnan, M. T., Venkatesh, T. G., & Bhaskar, A. V. (2023). Design and implementation of congestion aware router for network-on-chip. *Integration, 88*, 43–57. https://doi.org/10.1016/j.vlsi.2022.08.012
21. Trik, M., Akhavan, H., Bidgoli, A. M., Molk, A. M. N. G., Vashani, H., & Mozaffari, S. P. (2023). A new adaptive selection strategy for reducing latency in networks on chip. *Integration, 89*, 9–24. https://doi.org/10.1016/j.vlsi.2022.11.004

T. Nagalaxmi obtained B.E degree in Electronics and Communication Engineering from J.N.T University, M.Tech degree in Embedded Systems from J.N.T University, and Ph.D. degree from Osmania University, Hyderabad. She joined in ECE Department, Stanley College of Engineering and Technology for Women as Assistant Professor in 2013. She is presently working as Associate Professor in the ECE Department, Stanley College of Engineering and Technology for Women (Autonomous), Hyderabad. She has over 16 years of experience in teaching, training and research. Her areas of interest include VLSI, Embedded Systems, Computer Organization and Architecture, Communications, Computers, Circuits and Systems. She is a life member of IETE.

E. Sreenivasa Rao obtained B.E degree in Electronics and Communication Engineering from Andhra University in 1989, M.Tech degree in Opto-Electronics from Cochin University of Science and Technology in 1992, M.Tech (Computer Science) and Ph.D. degrees from J.N.T University, Hyderabad in 1998 and 2012 respectively. He joined ECE Department, Vasavi College of Engineering, as Lecturer in 1993. He is presently working as Professor & Head of the ECE Department, Vasavi College of Engineering (Autonomous), Hyderabad. He has over 30 years of experience in teaching, training and research. His areas of interest include Communications, Computers, Circuits and Systems. He is a life member of ISTE and fellow IETE.

P. ChandraSekhar received BE degree from Nagpur University, M.Tech degree from JNTU Hyderabad and PhD from Osmania University in 1991, 1999 and 2009 respectively. He had been awarded with Post Doctoral Fellowship by Shizuoka University, Japan for one year. Prior to joining in teaching, he has eight years of industrial experience of design and development of Embedded Systems. He has been working in the Department of Electronics and Communication Engineering, University College of Engineering, Osmania University, Hyderabad from 2001. He has been elevated as Professor of ECE in 2015. He is served as Head of Department, ECE, and Osmania University. He served as Chairman BOS in ECE Department for two years. He is actively involved in establishing the state of art Laboratories in the Department. He has more than 50 research publications to his credit. He delivered more than 15 invited talks and guest lecturers in various conference and events. Presently eight Ph.D. students are pursuing their research under his guidance. UGC sanctioned a Major Research Project on GNSS Receiver: Baseband algorithms in FPGA, worth of Rs.15 Lakh. He received a consultancy project from DLRL worth of 10 Lakh. He Received another Consultancy project from RCI, "Multi Communication protocols for SDR Appliactions" worth of Rs.10 lakh. He is currently Principal Investigator for CSIR SRF scheme. He is currently serving as Peer review committee member of DLRL projects and Member, System Engineering, BDL. He is member of Board of Studies in several Engineering colleges. His research interests include Development of high performance Computational Electro-magnetic and efficient FPGA based signal processing algorithms and Design Automation.