Strengthening NoC Security: Leveraging Hybrid Encryption for Data Packet Protection

Thejaswini P[1], Sahana A R[1], Shankar Singh C[1], John Jose[2]
[1]Department of ECE, JSS Academy of Technical Education, Benagluru, Karnataka, India
[2]Department of CSE, Indian Institute of Technology Guwahati, Guwahati, Assam, India

***Abstract*—The increasing complexity and scale of modern computing systems have led to the emergence of System-on-Chip (SoC) architectures , with Network-on-Chip (NoC) architectures serving as a communication infrastructure within these systems. Due to rigid time-to-market constraints, recent SoC designs involve using third-party IPs. This outsourcing can lead to security vulnerabilities in SoC, especially in NoC packet transmission. Considering security attacks, this paper addresses the need for cost-effective secure packet transmission in NoC. We propose a hybrid encryption framework by discussing the challenges associated with symmetric and asymmetric encryption techniques to achieve a balance between security and efficient data transmission in NoC. By combining the strengths of both encryption methods, our approach aims to provide enhanced protection against security threats while minimizing performance overhead.**

***Index Terms*—Hybrid Encryption, Packet Latency, Key Generation, Security**

I. Introduction

In the era of advanced computing systems, where integration of multiple components onto a single chip has become a necessity, SoC architectures [1] have emerged as a solution to address the increasing complexity and scope of advanced computing systems. However, efficient communication among these integrated components is crucial, leading to the prominence of NoC architectures [2]. NoC serves as a communication infrastructure within the SoC, enabling the exchange of data and control signals among different components. NoC, a packet-switched distributed communication system, replaces traditional bus-based architectures, enhancing scalability, performance, and power efficiency within the SoC.

Integrating components onto a single chip introduces security risks, especially during packet transmission in NoC architecture [3]. A packet consists of a head-flit, containing transit-critical information and a body-flit, carrying actual data that needs encryption for protection against threats like Trojans. Therefore, encrypting the body-flit has become the preferred method for securing NoC data. Traditionally, data transmission in NoC occurs in raw text for security reasons. To enhance protection, encryption methods like Advanced Encryption Standard (AES) [4], Rivet Cipher 4 (RC4) [5] and Hummingbird2 [12] are now used to safeguard information. However, these techniques introduce performance overhead and key length dependency, challenging efficient packet transmission.

To secure body-flit in NoC, traditional encryption methods like public and private key algorithms are commonly used. However, public key encryption (asymmetric key encryption) introduces additional overhead due to the necessity for distinct keys for transmission and reception, elongating process times. Moreover, each packet must be encrypted at the source node, adding to overall processing delays and contributing to increased delay overhead. Alternatively, private key encryption (symmetric key encryption) offers a speed advantage by using a uniform key for both encryption and decryption. Its primary drawback is the potential for inadvertent third-party interception each time the key is exchanged, which constitutes a security vulnerability in case of prolonged key exposure [6] [7]. To strike a balance between security and efficiency, we propose a hybrid encryption security framework that combines both symmetric and asymmetric key encryption techniques. The proposed methodology mitigates the limitations associated with each technique. Our key contributions in this paper are as follows:

1) We analyze limitations of conventional asymmetric and symmetric encryption in the NoC network.
2) We propose a cost-effective mechanism by combining existing encryption techniques for secure communication in NoC.
3) We experimentally analyze the latency overhead of the proposed encryption approach.
4) We suggest cost-effective alternatives through slight modifications to existing encryption methods.

The paper is organized as follows: Section II addresses existing security issues and related work in NoC. Section III discusses existing solutions and motivation. Section IV presents the architecture of our proposed scheme. Section V presents the simulation setup for performance analysis. Finally, Section VI concludes the paper.

II. Related Works

In recent decades, the security of NoC has emerged as a prominent research area, drawing significant attention from researchers and experts. A survey has been conducted to address the unique challenges encountered in securing NoC based SoCs [3].

Detection and mitigation of Hardware Trojans (HTs) in NoC have become increasingly important due to globalization of semiconductor industry and the potential risks posed by malevolent modifications made to hardware circuitry. A countermeasure was implemented by researchers to strengthen the security of NoC hardware designs and prevent unauthorized modifications [8]. A collaborative method was proposed, which utilized flit integrity and dynamic flit permutation to eliminate HTs inserted into the NoC router. Various approaches have been explored to ensure the integrity and confidentiality of IP cores, preventing unauthorized access and tampering. At the network level, a security wrapper is equipped for each IP core and a key-keeper core is

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

incorporated into NoC [9]. Encrypted private and public keys are safeguarded by this key-keeper core, ensuring that unencrypted keys cannot be accessed by untrusted software within or outside NoC. A security framework based on Authenticated Encryption (AE) specifically the Galois Counter Mode (GCM) algorithm was proposed within the Network Interface (NI) of secure IP cores [10] to avoid leakage of information and DOS attack.

Although existing authentication schemes can verify the data integrity of packets, they can impose unacceptable overhead on resource-constrained NoCs. In addition to securing IP cores themselves, researchers have also proposed routing mechanism that is lightweight in terms of computational resources and considers the trustworthiness of components involved in communication process [11]. ARNoC [13] and LEARN [12] are designed specifically for lightweight communication between IP cores in NoC based SoCs. Existing encryption-based protection methods in the literature typically leave certain parts of the packet unencrypted to allow routers to process and forward packets correctly. This exposes source and destination information of the packet to malicious routers, making them vulnerable to various attacks. To mitigate this issue, a secure anonymous routing [14] approach with minimal hardware overhead is proposed.

III. Motivation

NoC architectures are a reliable on-chip communication infrastructure, but their security can be compromised by HTs leading to breaches like data theft and operations disruption. Current security measures mostly secure routers, overlooking the transmitted packet's security. To combat these risks, an effective encryption system in NoC architectures is crucial. While strong algorithms like AES provide good encryption, they are computationally costly. Conversely, faster algorithms like Hummingbird2 may be vulnerable to brute force attacks. To address these issues, we propose a hybrid encryption mechanism, blending the benefits of smaller keys with frequent secure key changes, without sacrificing system performance. This enhances security without neglecting the safety of transmitted packets.

IV. Proposed Technique

This section presents a novel hybrid encryption security framework designed to facilitate efficient and secure packet transmission over a router. The section begins with an overview of the architecture of proposed scheme. Subsequently, the Elliptic Curve Diffie-Hellman (ECDH) algorithm for session key generation is discussed in detail. Following that, the process of packet encryption using session key and our proposed modified mCrypton encryption technique is explained. Finally, the Elliptic Curve Cryptograpghy (ECC) technique for encrypting the session key is discussed.

A. NoC Architecture

In the general NoC architecture of an 8 x 8 mesh topology, each node in the network is comprised of an IP core and a router, interconnected through a Network Interface . To strengthen the security of this existing framework, we propose the integration of Security Packetization Module (SPM) into router, as depicted in Fig. 1. This SPM encompass the following key components:

1) Key Management System: This dedicated component is embedded within each node and assumes a vital role in fortifying the network's security. It comprises a session key generator, which is responsible for generating unique session keys. These session keys are utilized to establish secure communication channels between nodes. Additionally, it incorporates a Key Bank to securely store session keys, private keys and public keys.
2) Symmetric Encryption and Decryption Circuit: This component encrypts and decrypts session key packets using a symmetric encryption algorithm, ensuring data confidentiality and integrity during transmission.
3) Asymmetric Encryption and Decryption Circuit: This component enhances network security by encrypting and decrypting session keys using an asymmetric encryption algorithm.

During the fabrication process, the public and private keys of all routers are pre-stored within each router ensuring their availability for cryptographic operations. Each node in the architecture possesses its own unique private key and public keys of the remaining $N - 1$ routers in the network, facilitating secure communication between nodes. The Key Bank within each router contains $N - 1$ session keys, generated through an asymmetric key exchange process involving other routers. A single router within the 8 x 8 mesh topology possesses its own private key and public keys of remaining 63 routers in the network are stored by this router. The key bank of this router is equipped with session keys that are generated using the session key generator. This mechanism guarantees that each router possesses necessary session keys to establish secure communication channels with other routers within the network.

B. Session Key Generation

The ECDH key exchange scheme holds significant importance within our proposed technique. It enables the establishment of session key between two communicating entities without the need to transmit their private keys over the network. By utilizing the properties of ECC, the ECDH scheme ensures secure and efficient generation of session key. This key exchange process is seamlessly integrated into the session key generator.

In the context of our proposed technique, we consider a cryptographic elliptic curve denoted as E, which is defined over a finite field Fp. This curve is mathematically represented by an Affine Weierstrass equation, expressed in equation (1).

$$y^2 = x^3 + ax + b \tag{1}$$

The coefficients a and b in equation (1) belongs to finite field Fp, where p is a prime number greater than 3. The process initiates with the selection of a generator point G from the elliptic curve $E(Fp)$. This chosen generator point serves as a foundation for performing specific steps that ultimately determine the session key. The following steps are used to calculate the session key using the ECDH algorithm:

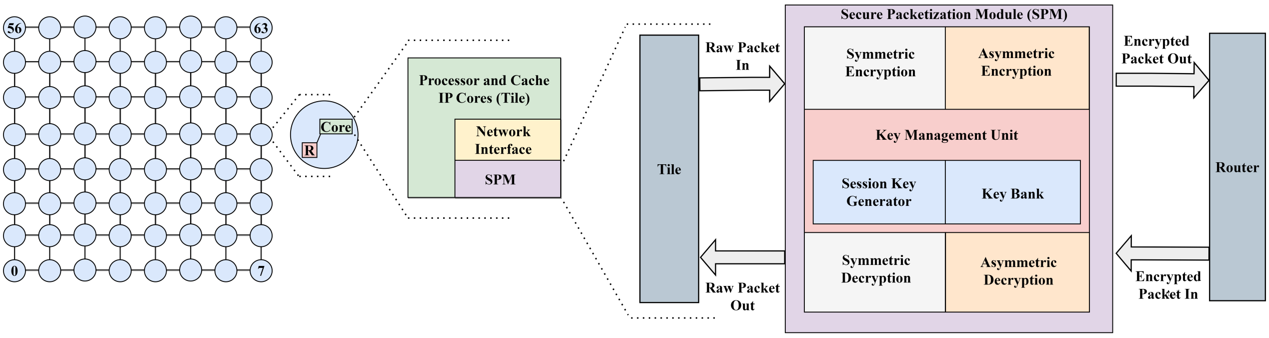

Fig. 1: Proposed General NoC architecture of an 8 x 8 mesh topology

1) The private key of the source router and public key of rest other routers is pre-stored in router's key bank.
2) Utilization of generator point G, private key of the source router and public key of the destination router to perform the ECDH operation.
3) Calculation of session key:

Session Key = Public Key of Destination Router $*$ Private Key of Source Router.

Upon generation, the session key is securely stored within the key bank of the source router. This storage mechanism ensures the safekeeping and accessibility of the session key for future utilization in the encryption and decryption processes of the session key packet. By securely storing the session key, the source router can readily access and utilize it whenever a packet arrives, enabling the seamless and protected transmission of data within the network.

C. Symmetric Key Encryption Algorithm - Packet's Body-flit Encryption

Upon the arrival of a session key packet at the source router, the session key stored in key bank of source router is retrieved and employed for symmetric encryption. Our proposed technique introduces a lightweight block cipher encryption method specifically designed for 32-bit systems, which is a modified version of the mCrypton 64-bit algorithm [19]. By optimizing the encryption process to suit 32-bit architecture, we enhance the efficiency and effectiveness of the encryption algorithm while maintaining a high level of security. Our modified symmetric encryption technique follows a similar structure to mCrypton 64-bit, which involves conducting round operations on a 4x4 bit array, utilizing 2 bits within each block. The encryption process comprises a total of 12 rounds, each consisting of four main functions: Column-to-Row Transformation, Bit Permutation, Nonlinear Substitution and Key Addition.

The Column-to-Row Transposition rearranges the bits in a 4x4 array by swapping the bits at position (i, j) with the bits at position (j, i). In other words, it exchanges the elements between rows and columns. The bit permutation operation plays a crucial role in reordering the bits within each column. It uses column permutations for each column, where each bit of the resulting column is determined by XORing specific bits from the original column according to a predefined pattern. The pattern is defined by masking bits $m0 = 01_2$, $m1 = 11_2$, $m2 = 00_2$, $m3 = 10_2$. The Nonlinear Substitution operates on a 4x4 bit array using four 2-bit S-boxes. Each S-box performs bit-wise substitutions denoted as S0, S1, S2 and S3. The S-boxes are designed such that S2 is the inverse of S0, and S3 is the inverse of S1 which is as shown in Table 1. The key addition operation performed by taking the bit-wise exclusive OR (XOR) of transformed array and round keys which are obtained from key scheduling algorithm of modified mCrypton.

```
Input (P) : 32-Bit Plain Text
Session Key (K): 32-Bits
Output (C): 32-Bit Encrypted Text
    R ←  P
    RK← Generate Round Keys (K)
    R ←  Add Round Key (R, RK)
    For i = 1 to 12 do
        R ←  Add Round Key (R, RK)
        R ←  Column-to-Row Transposition (R)
        R ←  Bit Permutation (R)
        R ←  Nonlinear Substitution (R)
    End for
    R ←  Column-to-Row Transposition (R)
    R ←  Bit Permutation (R)
    R ←  Column-to-Row Transposition (R)
    C ←  R
Return C
```

Fig. 2: Pseudo Code for Modified mCrypton Encryption

TABLE I: S-Boxses for 32-bit Modified mCrypton.

	0	1	2	3
S_0	0	2	1	3
S_1	1	3	0	2
S_2	3	1	2	0
S_3	2	0	3	1

The key scheduling algorithm in modified mCrypton generates round keys by applying nonlinear substitutions, word-wise rotations and bit-wise rotations to the initial session key. This process is repeated for all 12 rounds, resulting in the generation of 12 distinct round keys. The transformed array adds confusion and diffusion to the data, enhancing the security of the session key packet. The encryption process of the 32-bit modified mCrypton is defined by the pseudo-code in Fig. 2. Encryption process for 128-bit data packet using 32-bit modified mCrypton algorithm is depicted in Fig. 3. After the packet has been encrypted using the session key, an extra layer of security is added by encrypting the session key itself using asymmetric encryption. This additional encryption step protects against unauthorized access or tampering.

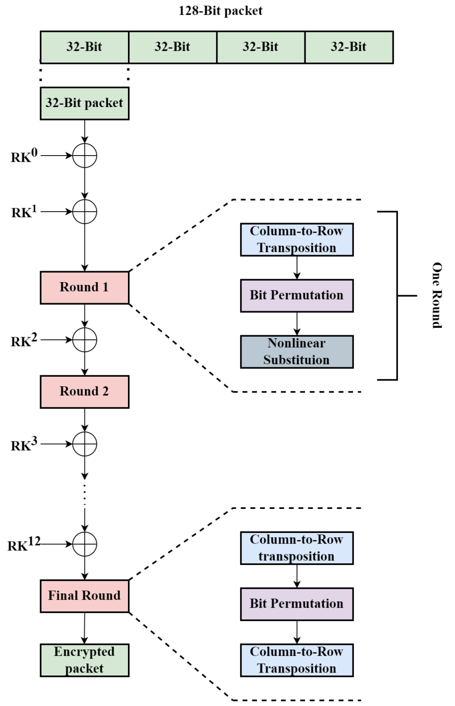

Fig. 3: Data Packet Encryption using Modified mCrypton

D. Asymmetric Key Encryption Algorithm - Session Key Encryption

The session key is encrypted using Elliptic Curve asymmetric key encryption by considering the elliptic curve E which is used in session key generator. The session key is encoded and mapped into an elliptic curve point $P_k(x_m, y_m)$. Then a random integer k is chosen by the source router and the encrypted key point is calculated. Then the session key point P_k is encrypted by choosing a random integer m at the source router and public key P_B of the destination router present in storage register of the source router. The encrypted point of the session key is obtained as in equation (2).

$$P_E = [([k]G), (P_k + [k]P_B)] \tag{2}$$

Following the encryption of both the data packet and the session key, they are interleaved and transmitted to the destination router. At the destination router, asymmetric session key decryption stage is responsible for decrypting the session key. This decryption process is performed using equation (3), which reverses the encryption operation and retrieves the original session key.

$$P_k = [(P_k + [k]P_B) - (y[k]G)] \tag{3}$$

where, y is the private key of the destination router which is multiplied with $[k]G$ and is subtracted from y_m co-ordinate.

After decrypting the session key, it is utilized to decrypt the body flit of the data packet. This decryption process employs the modified mCrypton decryption algorithm, which reverses the encryption operation applied during the encryption phase. By applying the session key to the decryption algorithm, the original content of the body flit is recovered and made available for further processing or delivery.

E. Session Key Regeneration

To enhance security and prevent potential attacks, a session key refresh mechanism is implemented. After a certain number of cycles or when the data packet is encrypted as sent across the router and before the packet the reaches the destination, a new session key is generated using the ECDH algorithm. This process ensures that even if an attacker manages to break the key within the estimated number of cycles, the session key will be changed before any significant damage can occur.

V. Experimental Analysis and Discussion

The proposed hybrid encryption technique is validated based on comprehensive performance analysis, security analysis, and overhead analysis. Through performance analysis, we evaluate the data movement rate of packets that involve the initial queuing as well as network latency of packets in UNoC and SNoC designs. Through security analysis, various potential security vulnerabilities and attack vectors are thoroughly examined. The overhead analysis focuses on evaluating the impact of the proposed hybrid encryption approach on key aspects such as area utilization, power consumption, critical path delay and cycle time.

A. Experimental Framework

We use the popular gem5 simulator [16] for modeling the SoC that houses the multi-core processor design similar to Intel KNL, AMD Ryzen etc. The garnet framework of gem5 is used to model the micro-architectural details of the underlying NoC interconnect. We model both 4x4 and 8x8 mesh NoC-based SoC systems. For experimental evaluation, we use uniform random synthetic traffic patterns. The packet injection rates are varied from zero load to saturation so as to study the impact of the proposed hybrid encryption module under different network load. We study the average latency of packets in UNoC and SNoC framework and understand the impact of hybrid encryption.

We implement the proposed hybrid encryption technique on a standard 2-stage pipelined input buffered NoC router using Verilog HDL. The RTL code generated is then successfully synthesized using the Cadence Genus Tool to analyze the timing constraints. The synthesis process is done using the TSMC90 (90nm) technology library, which provides accurate models and specifications for the hardware components. To validate our technique's applicability and effectiveness, we use an EDA tool, ProNoC [17], that facilitates the prototyping of custom NoC-based SoC.

B. Security Analysis

When evaluating the security aspects of our approach, we consider two main components: (A) Session key generation and transmission of the session key in an NoC packet using asymmetric encryption and (B) Symmetric encryption for NoC packets using the session key generated.

We use ECDH algorithm for session key generation at periodic intervals. This generated session key is sent to the target destination as a key packet using ECC based asymmetric encryption. The security of these algorithms relies on the hardness of the Discrete Logarithm Problem in elliptic curves. The Discrete Logarithm Problem states that given a point P on an elliptic curve and the result of scalar multiplication kP, it is computationally infeasible to determine the value of k. In the context of ECDH and ECC encryption, this means that without knowledge of the private keys involved, it is extremely difficult to calculate the shared session key generated during the key exchange process. Since the security of our approach is based on these widely studied and accepted mathematical problems, there is no specific need to conduct additional tests or evaluations to validate its resistance against known security attacks. Hence the session keys generated using ECDH and encrypted using ECC are secure and resistant to key-breaking attacks.

TABLE II: Results of Randomness Test

Parameter	Modified mCrypton for 32-Bit	Expected Value
Entropy	0.989921	1
Optimum Compression	1%	0
Arithmetic Mean	0.441	0.5
Monte Carlo Value of Pi Error	4 and 27.32%	3.14 and 13%
Serial Correlation Coefficient	-0.016481	0

TABLE III: Latency of various operations in the Proposed Hybrid Encryption Technique

Stage	Critical Path Delay (*ns*)	Latency (*cycles*)
ECDH Session Key Generation	8.3	14
Symmetric Encryption of Data Packet	5.9	10
Symmetric Decryption of Data Packet	5.6	9
Asymmetric Encryption of Session Key	12.5	21
Asymmetric Decryption of Session key	11.3	19

TABLE IV: Area and Power Overhead for Proposed Hybrid Encryption Technique.

Stage	Area (um^2)	Power (uW)
ECDH Session Key Generation	135.8	23516.6
Symmetric Encryption of Data Packet	13.1	88.8
Symmetric Decryption of Data Packet	11.7	74.6
Asymmetric Encryption of Session Key	14380.8	77316.6
Asymmetric Decryption of Session key	21571.2	51544.4

To assess the strength of our symmetric encryption of NoC Packets, we employ a comprehensive testing approach. We perform randomness test to verify the desired level of unpredictability in the output of the algorithm. This test examines the statistical property of the encrypted data to ensure its randomness and overall security. The widely adopted NIST test suite [18] is employed as a standard method to validate the random number generators. Parameters such as entropy, optimum compression, arithmetic mean, Monte Carlo value of Pi and Serial Correlation Coefficient are analyzed within the NIST Test Suite for Pseudo-Random Number Generators (PRNG). In Table II, we summarise the results of the randomness test and affirming the algorithm's security.

C. Overhead Analysis

To assess the overhead of our proposed hybrid encryption technique, we conduct a comprehensive analysis focusing on two crucial aspects: (1) the number additional cycles required for encryption and decryption operations and (2) parametric analysis. The total cycle time for our hybrid encryption technique is calculated by considering the critical path delay of each algorithm involved in the encryption process. This calculation determines the initial latency required to complete the encryption of a data packet. We compute the latency for (i) ECDH session key generation, (ii) Symmetric encryption of data packet and (iii) Asymmetric encryption of session key. The maximum circuit delay of each stage in our hybrid encryption technique is computed using HDL synthesis process. Table III depicts the critical path delay and latency for each component of our hybrid encryption technique.

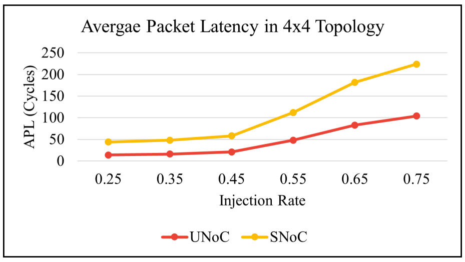

Fig. 4: Average Packet Latency in 4x4 Topology

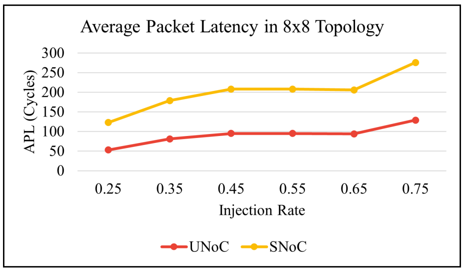

Fig. 5: Average Packet Latency in 8x8 Topology

We compute the area and power overhead of our proposed hybrid encryption technique in terms of area, power, and timing measurements, referencing a virtual clock. To ensure optimal performance, we fine-tune the virtual clock by con-

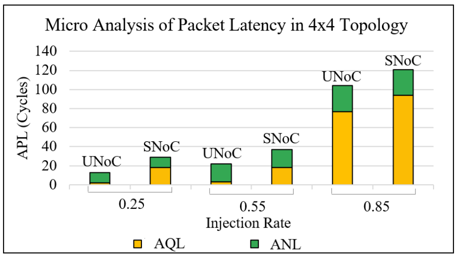

Fig. 6: Micro Analysis of Packet Latency in 4x4 Topology

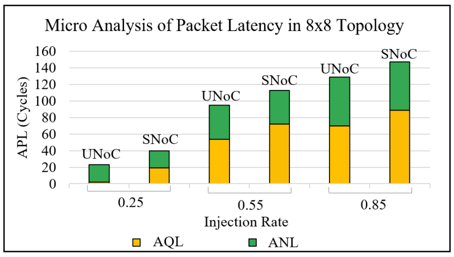

Fig. 7: Micro Analysis of Packet Latency in 8x8 Topology

straining its time period to achieve zero slack during static timing analysis. We summarize the overhead in Table IV. While analyzing the slack and critical path, we observe that the proposed router architecture can operate at the same clock as that of an UNoC design as the encryption modules are not operating on the critical path of the pipelined router. This is due to the activation of the asymmetric encryption module which works in parallel with the normal operations of the router. The new key generation and asymmetric encryption of the session key packet are not in the router pipeline. This justifies that the proposed architecture does not vary the operating frequency of the NoC pipeline.

D. Latency Analysis

We compute the Average Packet Latency (APL) of all NoC packets in UNoC as well as SNoC designs. As shown in Table II we obtain the latency for each component of our hybrid encryption technique. These latency values are integrated into garnet to create the necessary queuing delay for NoC packet before injecting them into the router. As discussed before, at regular intervals new session keys are generated and the same is sent in a key packet using asymmetric encryption. This operation is done parallel to the router pipeline and will not impact average packet latency. However once the session key is exchanged between a pair of source and destination, the subsequent packets are symmetrically encrypted with them. This takes 10 cycles of additional queuing latency at the source router and another 9 cycles at the destination router for obtaining the raw data packet back. Fig. 4 and Fig. 5 show the average packet latency in UNoC and SNoC in 4x4 mesh and 8xx8 mesh NoC, respectively.

The APL encompasses the Average Queuing Latency (AQL) and Average Network Latency (ANL). We also conduct microanalysis of packet latency to understand the fraction on AQL which consists of encryption and packetization time of a packet. The results of this analysis are depicted in Fig. 6 and Fig. 7 showcasing the split up of APL for the 4x4 and 8x8 mesh NoC configurations, respectively. We can clearly see that the ANL is not impacted by the hybrid encryption process and as expected the AQL is increased due to the additional latency of the encryption process. Across all injection rates, we see AQL and proportional variations in ANL. This can be observed from low injection rate to high injection rate.

VI. Conclusion

The exploration of hybrid encryption systems in NoC architectures focuses on aiming to address security vulnerabilities during packet transmission. By combining symmetric and asymmetric encryption techniques, the proposed framework achieved a good balance between robust security measures and efficient data transmission. Through a comprehensive evaluation, the effectiveness of the hybrid encryption approach is demonstrated in mitigating the limitations of traditional encryption algorithms. This integration of hybrid encryption in NoC architectures provides a secure and seamless communication environment for integrated components in modern computing systems. The empirical evaluation conducted in this study demonstrates that our hybrid encryption system maintains efficient data transmission while significantly improving security in NoC architectures.

VII. Acknowledgement

We thank JSSMVP, Mysuru, India for supporting and funding this research work. The experimental work is done using the facility created by the SERB Project CRG/2021/007400, DST, Government of India.

References

[1] Zorian, Yervant, Sujit Dey, and Michael J. Rodgers. ”Test of future system-on-chips.” In IEEE/ACM International Conference on Computer Aided Design. ICCAD-2000. IEEE/ACM Digest of Technical Papers (Cat. No. 00CH37140), pp. 392-398. IEEE, 2000.

[2] Agarwal, Ankur, Cyril Iskander, and Ravi Shankar. ”Survey of network on chip (noc) architectures and contributions.” Journal of engineering, Computing and Architecture 3, no. 1 (2009): 21-27.

[3] Charles, Subodha, and Prabhat Mishra. ”A survey of network-on-chip security attacks and countermeasures.” ACM Computing Surveys (CSUR) 54, no. 5 (2021): 1-36.

[4] Dworkin, Morris J., Elaine B. Barker, James R. Nechvatal, James Foti, Lawrence E. Bassham, E. Roback, and James F. Dray Jr. ”Advanced encryption standard (AES).” (2001).

[5] Klein, Andreas. ”Attacks on the RC4 stream cipher.” Designs, codes and cryptography 48 (2008): 269-286.

[6] Chandra, Sourabh, Smita Paira, Sk Safikul Alam, and Goutam Sanyal. ”A comparative survey of symmetric and asymmetric key cryptography.” In 2014 international conference on electronics, communication and computational engineering (ICECCE), pp. 83-93. IEEE, 2014.

[7] Yassein, Muneer Bani, Shadi Aljawarneh, Ethar Qawasmeh, Wail Mardini, and Yaser Khamayseh. ”Comprehensive study of symmetric key and asymmetric key encryption algorithms.” In 2017 international conference on engineering and technology (ICET), pp. 1-7. IEEE, 2017.

[8] Hussain, Musharraf, Naveed Khan Baloach, Gauhar Ali, Mohammed ElAffendi, Imed Ben Dhaou, Syed Sajid Ullah, and Mueen Uddin. ”Hardware Trojan Mitigation Technique in Network-on-Chip (NoC).” Micromachines 14, no. 4 (2023): 828.

[9] Gebotys, Catherine H., and Robert J. Gebotys. ”A framework for security on NoC technologies.” In IEEE Computer Society Annual Symposium on VLSI, 2003. Proceedings., pp. 113-117. IEEE, 2003.

[10] Sajeesh, K., and Hemangee K. Kapoor. ”An authenticated encryption based security framework for NoC architectures.” In 2011 International Symposium on Electronic System Design, pp. 134-139. IEEE, 2011.

[11] Charles, Subodha, and Prabhat Mishra. ”Lightweight and trust-aware routing in NoC-based SoCs.” In 2020 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), pp. 160-167. IEEE, 2020.

[12] Charles, Thelijjagoda SN, and Prabhat Kumar Mishra. ”Lightweight Encryption and Anonymous Routing in NoC based SoCs.” U.S. Patent Application 16/937,882, filed February 4, 2021.

[13] Charles, Subodha, Megan Logan, and Prabhat Mishra. ”Lightweight anonymous routing in NoC based SoCs.” In 2020 Design, Automation amd Test in Europe Conference and Exhibition (DATE), pp. 334-337. IEEE, 2020.

[14] Sarihi, Amin, Ahmad Patooghy, Mahdi Hasanzadeh, Mostafa Abdelrehim, and Abdel-Hameed A. Badawy. ”Securing network-on-chips via novel anonymous routing.” In Proceedings of the 15th IEEE/ACM International Symposium on Networks-on-Chip, pp. 29-34. 2021.

[15] Shi, Zhenqing, Bin Zhang, and Dengguo Feng. ”Practical-time related-key attack on Hummingbird-2.” IET Information Security 9, no. 6 (2015): 321-327.

[16] Nathan Binkert, Bradford Beckmann, Gabriel Black, Steven K. Reinhardt, Ali Saidi, Arkaprava Basu, Joel Hestness, Derek R. Hower, Tushar Krishna, Somayeh Sardashti, Rathijit Sen, Korey Sewell, Muhammad Shoaib, Nilay Vaish, Mark D. Hill, and David A. Wood. 2011. The Gem5 Simulator. ACM SIGARCH Comput. Archit. News 39, 2 (aug 2011), 1–7

[17] Alireza Monemi, Jia Wei Tang, Maurizio Palesi, and Muhammad N. Marsono. 2017. ProNoC: A low latency network-on-chip based many-core system-on-chip prototyping platform. Microprocessors and Microsystems 54 (2017),60–74.

[18] Lawrence Bassham, Andrew Rukhin, Juan Soto, James Nechvatal, Miles Smid, Stefan Leigh, M Levenson, M Vangel, Nathanael Heckert, and D Banks. A statistical test suite for random and pseudorandom number generators for cryptographic applications, 2010-09-16 2010.

[19] Lim, Chae Hoon, and Tymur Korkishko. ”mCrypton–a lightweight block cipher for security of low-cost RFID tags and sensors.” In International workshop on information security applications, pp. 243-258. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005.