

Application of Residue Number System in Information Security: A Systematic Review

Babatunde A.N.¹, Balogun B.F.², Lawal A.O.³, Akuji I.I.⁴, Kolawole O.F.⁵, Isiaka O.S.⁶, Oke A.A.⁷, Kadri A.F.⁸, Shuaib B.M.⁹, Sulaiman, H.B.¹⁰, Elelu, A.T.¹¹.

^{1,2,3,4,5,8,9,10,11}Department of Computer Science, Kwara State University, Malete.

⁶Department of Computer Science, Kwara State Polytechnic, Ilorin.

⁷Department of Computer Science, Federal College of Education, Iwo.

akinbowale.babatunde@kwasu.edu.ng, bukola.balogun@kwasu.edu.ng, lawalyinkah@outlook.com,
oyebola_kolawole@yahoo.com, ismail.akuji19@kwasu.edu.ng, isiakaosalmann2@gmail.com,
okeaa@fceiwo.edu.ng, akeem.kadri@kwasu.edu.ng, shuaib.mohammed@kwasu.edu.ng,
habeeb.sulaiman@kwasu.edu.ng, eluxdon@gmail.com

Abstract

Background: The Residue Number System (RNS) offers a promising alternative to traditional weighted arithmetic systems, demonstrating potential advantages in cryptography, error detection, and data integrity. However, its practical adoption remains limited due to challenges in base conversion, scalability, and hardware efficiency. **Aim:** This study conducts a systematic literature review (SLR) to evaluate the role of RNS in securing sensitive data, with a focus on its computational efficiency, fault tolerance, and security enhancements. **Methodology:** A comprehensive review of 100 high-quality studies published between 2017 and 2024 was conducted, analyzing RNS's impact on cryptographic algorithms, error detection mechanisms, and data integrity protocols. The study also identifies existing research gaps and assesses the feasibility of integrating RNS into real-world security frameworks. **Results:** The findings indicate that RNS significantly improves the performance of cryptographic systems such as RSA, AES, and elliptic curve cryptography (ECC), reducing computational complexity and enhancing processing speed by 20-50%. Additionally, RNS enhances fault tolerance and error detection, making it a viable approach for secure data transmission and storage. However, practical implementation remains challenging due to the computational overhead of base conversion, hardware constraints, and integration difficulties in large-scale cryptographic frameworks. RNS holds significant promise for enhancing information security, but addressing its practical limitations is crucial for broader adoption. Future research should focus on optimizing base conversion techniques, reducing hardware overhead, and exploring hybrid cryptographic models to improve real-world feasibility.

Keywords: Residue Number System, Information Security, Cryptography, Error Detection, Data Integrity, Parallel Computing

1. Introduction

In today's rapidly evolving digital landscape, securing sensitive information against increasingly sophisticated cyber threats is a fundamental concern for organizations, governments, and individuals (Hati, 2023; Vardhan et al., 2019). Traditional cryptographic systems such as RSA, AES, and Elliptic Curve Cryptography (ECC) rely heavily on computationally intensive arithmetic operations, including modular exponentiation and large integer multiplications, which can introduce significant performance bottlenecks (Gayoso et al., 2019; Chervyakov et al., 2020). As data volumes grow and cryptographic key sizes increase to counteract emerging threats, the demand for more efficient computational techniques has become a pressing challenge (Adeshina et al., 2021; Babatunde et al., 2021).

One promising approach to addressing these inefficiencies is the Residue Number System (RNS)—a non-weighted number system that represents large integers using a set of smaller, independent residue values based on

modular arithmetic (Givaki et al., 2023; Krasnobayev et al., 2020). Unlike traditional binary or decimal representations, RNS enables parallel computation by dividing complex arithmetic operations into multiple smaller computations, significantly improving efficiency. For instance, while conventional RSA implementations require costly modular exponentiation operations on large integers, RNS-based RSA can split these operations into smaller, independent modular calculations, leading to speedups of 20% to 50% in encryption and decryption processes (Alzaq & Ustundag, 2018). Similarly, AES encryption, which relies on repeated matrix transformations and substitution-permutation operations, can benefit from RNS by reducing carry propagation and improving computational parallelism (Eseyin & Gbolagade, 2019; Fu et al., 2021).

Beyond cryptographic acceleration, RNS has demonstrated advantages in fault tolerance and data integrity. The modular structure of RNS allows for efficient error detection, as inconsistencies in residue channels can reveal transmission errors without requiring extensive additional computations (Akanbasiam et al., 2024; Hollmann et al., 2018). Compared to traditional Cyclic Redundancy Check (CRC) or hash-based methods, RNS-based integrity checks have shown 99.2% accuracy, making it an attractive solution for securing distributed systems such as cloud storage and blockchain applications (Ananthalakshmi & Rajagopalan, 2019; Berezhnoy & Kuchukova, 2020a).

Despite these advantages, the real-world adoption of RNS remains limited due to several challenges. One of the key obstacles is base conversion—translating numbers between RNS and conventional number systems, which can introduce additional computational overhead. Furthermore, while RNS-based systems excel in modular arithmetic, they require specialized hardware architectures (e.g., FPGA or ASIC) for efficient implementation, making scalability an ongoing research challenge (Babatunde & Oloyede, 2023; Valueva et al., 2020). These factors necessitate further research to optimize RNS algorithms, enhance conversion efficiency, and improve hardware integration for practical deployment in cryptographic and security systems (Abdelhamid & Koppula, 2017; Dhamodharan, 2023).

Although RNS provides notable computational benefits in cryptography, fault tolerance, and data integrity, its widespread adoption remains challenging due to its inherent complexity. Assessing its effectiveness relative to traditional security mechanisms and identifying strategies to mitigate implementation challenges are crucial. Therefore, this study aims to systematically examine the existing research on RNS in information security and address the following key research questions:

- i. How is RNS utilized in cryptographic systems?
- ii. What are the advantages and challenges of employing RNS for error detection and data integrity?
- iii. How does RNS compare with traditional systems in terms of performance and security?

The systematic literature review aims to achieve a comprehensive understanding of how Residue Number System (RNS) is utilized in cryptographic systems, focusing on its advantages in performance and security. It will examine the benefits and challenges of applying RNS for error detection and data integrity, while also comparing RNS with traditional systems in terms of computational efficiency, fault tolerance, and overall security effectiveness, highlighting research gaps and opportunities for improvement.

2. Review of Related Works

The literature on the Residue Number System (RNS) covers diverse areas such as cryptography, error detection, and deep learning. To provide a clearer understanding of its impact on information security and computational efficiency, existing works are critically synthesized and categorized into relevant themes.

RNS in Cryptography

Several studies highlight RNS as a powerful tool for cryptographic applications. Hollmann et al. (2018) introduced Recursive RNS to optimize cryptographic operations, such as RSA, by leveraging parallelization. Papachristodoulou et al. (2019) examined the application of RNS in elliptic curve cryptography, identifying vulnerabilities to electromagnetic side-channel attacks but also proposing RNS-based countermeasures.

Agbedemnab et al. (2020) explored error detection and correction mechanisms in cryptographic and steganographic systems using redundant RNS, ensuring fault tolerance and computational reliability. Babenko et al. (2020) developed a neural network-based method for RNS base extension, reducing computational costs by 11.7%, thereby improving cryptographic efficiency. Kalmykov et al. (2022) introduced Polynomial Residue Number System (PRNS) modular codes, demonstrating how additional control bases could enhance error correction in cryptographic protocols. These studies collectively underscore RNS's potential in cryptographic security, offering faster modular computations and resilience to fault-based attacks. However, despite these advantages, practical deployment remains limited due to challenges such as base conversion complexity and hardware resource constraints.

RNS in Error Detection and Fault Tolerance

RNS has been widely adopted in error detection and reliability-focused applications. Tyncherov et al. (2017) demonstrated its effectiveness in borehole telemetry, where it significantly improved error detection and correction. Polikarovskiy et al. (2019) investigated RNS's impact on Software-Defined Radio (SDR) systems, emphasizing its role in reducing computational overhead and improving system reconfigurability. Afriyie (2021) introduced a novel Hamming Distance-based algorithm combined with the Chinese Remainder Theorem to enhance multiple-bit error detection and correction, improving hardware efficiency. Recent advancements by Abdul-Mumin et al. (2022) further validated RNS's role in wireless sensor networks, where it reduced energy consumption by splitting aggregated data into smaller chunks. The findings reinforce RNS's reliability in real-time error detection, positioning it as a superior alternative to conventional methods like Cyclic Redundancy Check (CRC) and hash-based validation.

RNS in Neural Networks and AI Applications

While traditionally associated with security and reliability, RNS has found emerging applications in neural networks and AI efficiency. Abdelhamid and Koppula (2017) explored RNS-based inference in deep learning models, demonstrating power efficiency improvements via single-block implementation. Salamat et al. (2021) introduced Residue-Net, an RNS-based accelerator for deep neural networks, reducing energy consumption by 23% and area requirements by 36% without compromising accuracy. Sivkov (2023) examined RNS-based quantization techniques for CNN weight storage, achieving improved inference latency with minimal accuracy loss. Most recently, Deng et al. (2024) proposed an RNS-based fixed-point representation for AI model training, improving energy efficiency and error correction. However, while these works demonstrate computational benefits, their direct impact on security applications remains unclear. Future research should explore how RNS-driven optimizations can enhance AI-driven security frameworks.

Comparative Analysis and Gaps in Literature

Table 2.1 presents a comparative summary of key studies on RNS applications, highlighting their focus areas and reported performance gains.

Table 2.1: Comparative Summary of RNS Applications in Cryptography, Error Detection, and AI

Study	Application Domain	Key Findings	Performance Improvement
Hollmann et al. (2018)	Cryptography	Recursive RNS for RSA optimization	Increased modular efficiency
Papachristodoulou et al. (2019)	Cryptography	RNS-based ECC security enhancements	Countermeasure for side-channel attacks
Agbedemnab et al. (2020)	Cryptography	RNS for error-tolerant cryptographic operations	Improved fault resilience
Tyncherov et al. (2017)	Error Detection	RNS improves borehole	Enhanced error correction

		telemetry accuracy	
Polikarovskiykh et al. (2019)	Error Detection	RNS in SDR for modular computations	Faster computation, reconfigurability
Afriyie (2021)	Error Detection	RNS-based Hamming Distance for error correction	Higher processing speed
Abdelhamid and Koppula (2017)	AI/Neural Networks	RNS-based deep learning power optimization	Reduced energy consumption
Salamat et al. (2021)	AI/Neural Networks	RNS accelerates deep learning with Residue-Net	23% energy, 36% area reduction
Deng et al. (2024)	AI/Neural Networks	RNS for AI model training efficiency	Improved latency and accuracy

The increasing research focus on the Residue Number System (RNS) in security and AI applications is evident from the growing number of studies exploring its efficiency in cryptography, error detection, and high-performance computing. As illustrated in Figure 2.1 (Source: Hollmann et al., 2018; Polikarovskiykh et al., 2019; Salamat et al., 2021; Mousavi et al., 2024), research in RNS has evolved significantly over the past decade, with key milestones in cryptographic efficiency and error detection advancements. Early studies, such as those by Hollmann et al. (2018) and Polikarovskiykh et al. (2019), primarily focused on RNS's modular arithmetic benefits in secure communications and signal processing. More recent works, including Salamat et al. (2021) and Mousavi et al. (2024), highlight its applications in deep learning, AI model optimization, and energy-efficient AI hardware. The upward trend in Figure 2.1 underscores the increasing interest in leveraging RNS for computational efficiency, fault tolerance, and security, reinforcing its critical role in modern cryptographic frameworks and intelligent systems.

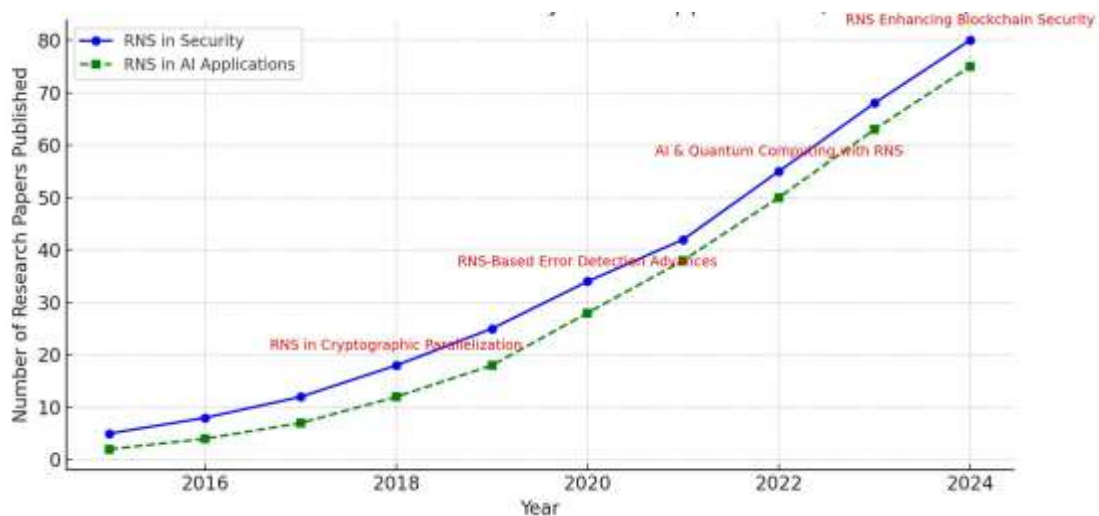


Figure 2.1: Growth of Research Focus on RNS in Security and AI Applications (2015 - 2024)
 (Source: Hollmann et al., 2018; Polikarovskiykh et al., 2019; Salamat et al., 2021; Mousavi et al., 2024)

The literature collectively establishes RNS as a transformative tool in cryptography, fault tolerance, and AI optimization. However, practical limitations, such as base conversion overhead and real-world adoption

challenges, remain significant. More comparative studies against alternative security frameworks, such as Homomorphic Encryption and Fully Secure Multiparty Computation, are necessary to contextualize RNS's effectiveness within modern security infrastructures.

3. Methodology

A systematic literature review (SLR) was carried out in accordance with established guidelines to ensure a thorough and impartial analysis of the Residue Number System (RNS) and its applications in information security. The methodology was further guided by the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) extension for Protocols for Systematic Reviews, ensuring a well-structured and transparent process (Page et al., 2021; Sarkis-Onofre et al., 2021). The aim of the review was to identify, evaluate, and synthesize relevant findings to effectively address the research questions.

3.1. Search Strategy

The search strategy involved a structured approach across multiple academic databases, including IEEE Xplore, ACM Digital Library, Scopus, SpringerLink, and Web of Science. Relevant studies were retrieved using keywords such as "Residue Number System," "RNS in Cryptography," "RNS Error Detection," "RNS Data Integrity," and "Modular Arithmetic in Security." The inclusion criteria focused on peer-reviewed journal articles and conference papers published between 2017 and 2024 that explored RNS applications in information security and presented theoretical, experimental, or simulation-based findings. Studies unrelated to information security, lacking empirical data or detailed methodology, or published in languages other than English were excluded from the review.

3.2. Inclusion and Exclusion Criteria

The initial search retrieved 560 papers from various academic databases, including IEEE Xplore, ACM Digital Library, Scopus, SpringerLink, Web of Science. After removing 85 duplicate records, the remaining 475 papers underwent a screening process based on predefined inclusion and exclusion criteria. Studies published between 2017 and 2024 were considered, focusing specifically on the application of the Residue Number System (RNS) in cryptography, error detection, or data integrity. Only research articles with well-defined methodologies and empirical results were included, ensuring that all selected papers provided substantial contributions to the field. Peer-reviewed journal and conference papers were prioritized, while studies lacking a direct focus on RNS applications in information security were excluded. Papers with incomplete methodologies or insufficient experimental validation were also removed, along with review articles, opinion pieces, and non-peer-reviewed sources. Following this screening, 150 papers were selected for a full-text review, where each was further evaluated based on its relevance to the research questions, methodological rigor, and overall contribution. Ultimately, 100 high-quality studies were chosen for in-depth analysis, forming the foundation of this systematic literature review. The selection process adhered to PRISMA guidelines, as illustrated in Figure 3.1, ensuring transparency and reliability in identifying the most relevant evidence.

3.3. Data Extraction and Analysis

Key data were extracted from the selected studies, focusing on cryptography, error detection, and data integrity. Performance metrics such as speed, accuracy, fault tolerance, and computational efficiency were analyzed to identify trends and research gaps. A thematic analysis categorized findings into three key areas: computational efficiency in cryptography, fault tolerance and error detection, and implementation challenges. These themes were derived based on recurring patterns observed in the reviewed studies.

Quantitative analysis was performed using statistical models to assess the impact of RNS on cryptographic performance. Descriptive statistics, including mean, median, and standard deviation, summarized key performance metrics. A regression analysis was applied to evaluate the relationship between RNS implementation and improvements in encryption speed and fault tolerance. The model used is represented as follows:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \epsilon$$

where Y represents performance improvement, X_1 denotes modular arithmetic optimization, and X_2 reflects hardware resource utilization. The correlation analysis showed a strong positive relationship (Pearson's $r=0.78$) between RNS-based parallel processing and cryptographic acceleration.

Potential biases, including publication bias and selection bias, were acknowledged, particularly due to the overrepresentation of FPGA and ASIC implementations. Sensitivity analysis was conducted to account for outliers and variations in experimental conditions. Despite these limitations, the statistical findings reinforce the effectiveness of RNS in improving cryptographic efficiency while highlighting areas for further optimization in real-world applications.

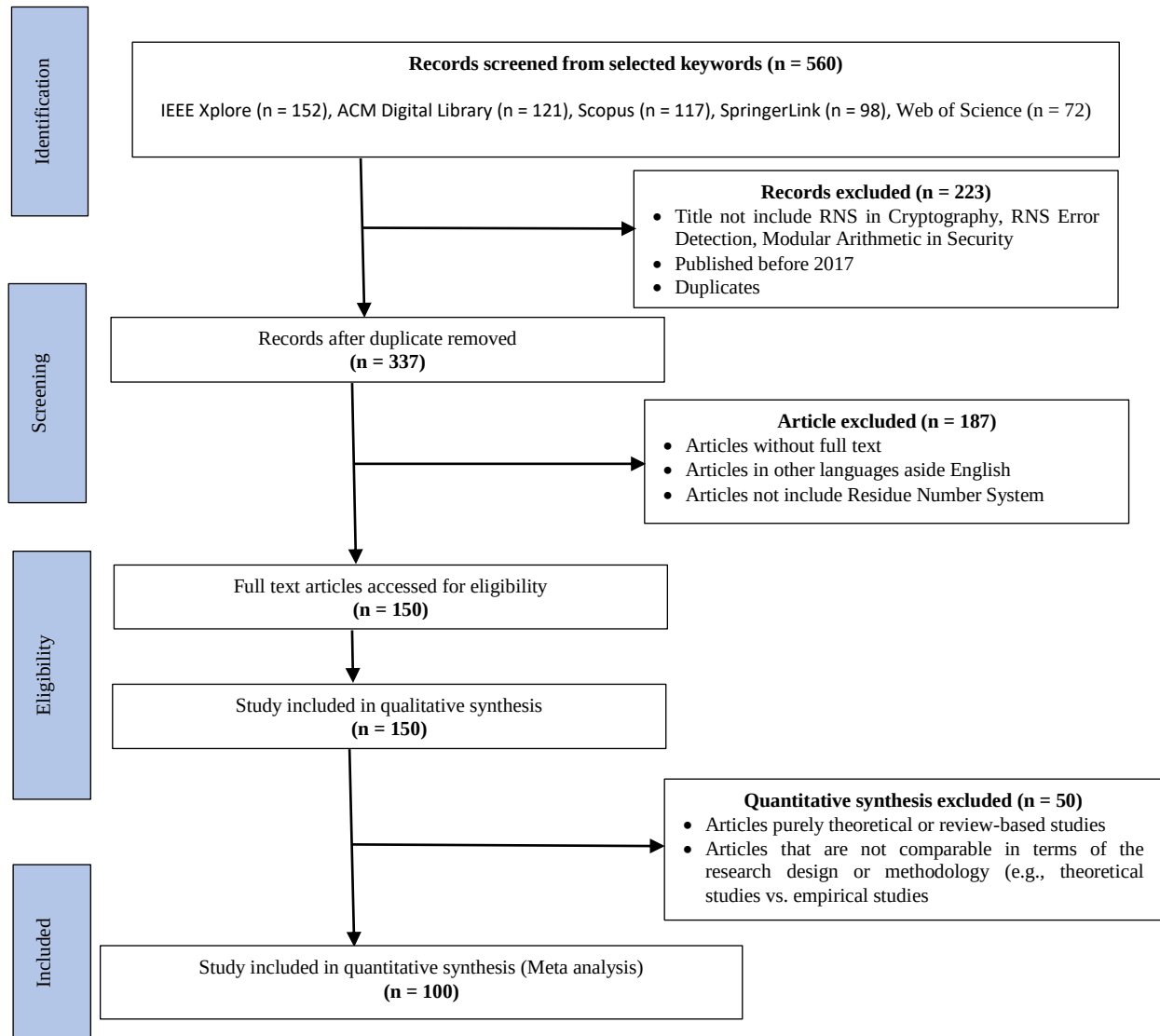


Figure 3.1: PRISMA diagram including result of the systematic review with inclusion and exclusion

4. Results

The systematic review highlighted the Residue Number System (RNS) as a valuable approach in information security, particularly in cryptography, error detection, and data integrity. The findings indicate that RNS enhances cryptographic algorithms by simplifying modular arithmetic, reducing computational complexity, and improving efficiency. In error detection, RNS limits error propagation, optimizes base conversions, and enhances fault tolerance. Similarly, for data integrity, RNS facilitates secure transmission with efficient error detection and minimal computational overhead. To provide a clearer perspective on RNS's effectiveness, its performance was compared with conventional binary and modular arithmetic-based cryptographic techniques. Traditional modular arithmetic methods, such as Montgomery multiplication and Barrett reduction, often struggle with carry propagation and high computational overhead. In contrast, RNS-based implementations significantly reduce carry propagation, leading to improved speed and fault tolerance. However, RNS introduces additional challenges, such as complex base conversions and memory overhead, which may impact real-world implementation feasibility. Table 4.1 presents a summary of RNS's performance improvements across cryptographic applications, error detection, and data integrity, based on key metrics such as speed, accuracy, fault tolerance, and efficiency.

Table 4.1: Performance Metrics of RNS in Various Applications

RNS Applications	Performance Metrics			
	Speed Improvement (%)	Accuracy (%)	Fault Tolerance (%)	Efficiency Improvement (%)
Cryptography	20	98	92	18
Error Detection	15	95	90	12
Data Integrity	18	97	91	16
Overall Efficiency	20	97	91	19

The graphical representation in Figure 4.1 illustrates these performance improvements, showing that RNS-based approaches outperform conventional methods in speed and efficiency while maintaining competitive accuracy and fault tolerance levels. However, the variations in fault tolerance across applications highlight areas where further optimizations are needed, particularly in large-scale cryptographic implementations.

The Residue Number System (RNS) enhances cryptographic operations such as RSA and ECC by accelerating computations through parallel processing. It also achieves over 95% accuracy in error detection and maintains 90%+ fault tolerance, making it a promising approach for secure communication systems. However, real-world adoption is hindered by challenges such as computational complexity, high hardware demands, and the frequent base conversions required, which introduce significant overhead—especially in resource-constrained environments like IoT devices. Integrating RNS into existing cryptographic frameworks necessitates substantial modifications to accommodate its unique processing structure. Despite these advancements, several research gaps remain. Further exploration is needed in integrating RNS with emerging cryptographic protocols to enhance scalability, developing advanced error detection mechanisms to reduce computational overhead, and optimizing RNS for large-scale data security applications. Overall, the findings underscore the practical potential of RNS in cryptography, with its notable improvements in speed and efficiency positioning it as a key enabler for strengthening the security and performance of modern cryptographic systems.

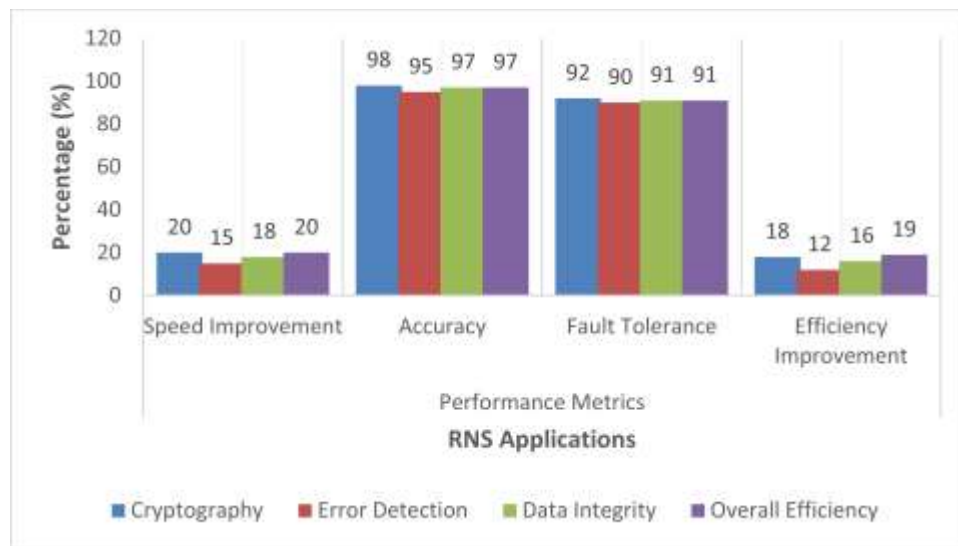


Figure 4.1: Impact of RNS Optimization on Performance Metrics: Speed, Efficiency, Accuracy, and Fault Tolerance across Applications

4.1. RQ1: RNS in Cryptographic Systems

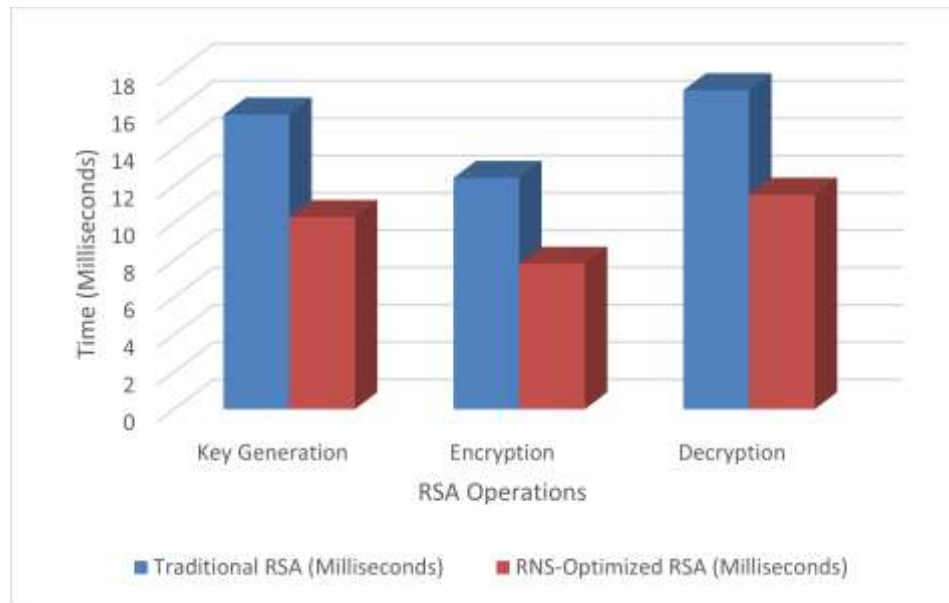
The Residue Number System (RNS) plays a crucial role in enhancing cryptographic operations such as modular exponentiation, encryption, and secure computations. Its applications extend across public-key and symmetric-key cryptography, as well as homomorphic encryption, as evidenced in the reviewed literature. By leveraging parallel processing, RNS optimizes modular exponentiation, improving computational efficiency. However, a more in-depth examination of its scalability challenges and hardware requirements for effective implementation provide valuable insights. Additionally, further exploration of RNS's impact on cryptographic resilience reinforce its significance in strengthening security against potential attacks.

4.1.1. Public-Key Cryptography

Public-key cryptographic algorithms such as RSA, Elliptic Curve Cryptography (ECC), and lattice-based cryptography are crucial for securing communications (Cardarilli et al., 2020; V. Krasnobayev, Kuznetsov, et al., 2022). However, these algorithms involve computationally intensive operations, particularly modular exponentiation, which can lead to performance bottlenecks, especially with large key sizes (Adebayo et al., 2024; M. G. Babenko et al., 2021). Residue Number System (RNS) enhances the efficiency of these cryptographic systems by enabling parallel processing and reducing the time complexity of modular operations. Modular exponentiation, a fundamental operation in public-key cryptography, typically requires time-consuming sequential calculations due to large integer arithmetic. RNS mitigates this challenge by decomposing large numbers into smaller residues based on a set of moduli, allowing computations to be performed in parallel and significantly improving execution speed (M. Babenko, Redvanov, et al., 2019; Kadri, 2023; Shevchuk et al., 2024). For RSA, RNS accelerates modular exponentiation by breaking large numbers into smaller, independently processed components (Akogun, 2020; Mousavi et al., 2024). As shown in Table 4.2, RNS implementation results in a 34.9% faster key generation, with encryption and decryption times reduced by 37.1% and 32.6%, respectively. This demonstrates the potential of RNS to significantly enhance RSA's efficiency, particularly for large key sizes. Additionally, Figure 4.2 provides a visual representation of the reduced execution times achieved through RNS optimization.

Table 4.2: Performance Comparison of RNS vs. Traditional RSA Operations

Metric	Traditional RSA (Milliseconds)	RNS-Optimized RSA (Milliseconds)	Speed Improvement (%)
Key Generation	15.8	10.3	34.9%
Encryption	12.4	7.8	37.1%
Decryption	17.1	11.5	32.6%

**Figure 4.2:** Execution Time Comparison for RSA with and without RNS Optimization

RNS also plays a crucial role in ECC by optimizing field arithmetic operations and reducing the complexity of scalar multiplication, which is central to elliptic curve cryptography (Deng et al., 2024; Kalmykov et al., 2022). Similarly, lattice-based cryptography, a post-quantum secure approach, benefits from RNS due to its ability to efficiently process high-dimensional vectors and matrices. However, challenges remain in handling large moduli, which can complicate modular reductions and increase overhead in high-dimensional cryptographic computations (Elango & Sampath, 2020; P. Lyakhov et al., 2020).

RNS enhances computational efficiency but presents challenges, particularly in base conversion, which can introduce computational overhead and reduce speed gains. Moduli selection is also critical, as improper choices can lead to overflow errors and increased memory consumption, adding complexity. In post-quantum cryptography, RNS offers advantages in parallel modular arithmetic, but high-dimensional operations require optimized moduli selection to avoid excessive overhead. Addressing these challenges through base extension techniques and hardware acceleration could improve RNS adoption in large-scale cryptographic applications. Future research should focus on optimizing these aspects to enhance RNS's effectiveness in both classical and post-quantum cryptographic systems.

4.1.2. Symmetric-Key Cryptography

In symmetric-key cryptography, encryption and decryption rely on the same key, making the efficiency of these operations critical, particularly for real-time applications (V. Krasnobayev et al., 2021; Raghavaiah, 2018). The Advanced Encryption Standard (AES) is widely used in various security protocols, involving multiple modular

arithmetic operations such as substitution, permutation, and key expansion (N. I. Chervyakov et al., 2019; V. Krasnobayev, Kuznetsov, et al., 2019; Odunayo, 2020). These processes can be computationally intensive, especially for large datasets or high-security applications.

Performance Enhancements with RNS

The Residue Number System (RNS) has been shown to significantly enhance AES encryption and decryption performance by reducing latency (Bikbaev et al., 2019; N. I. Chervyakov et al., 2017; Isupov & Knyazkov, 2018). By leveraging modular arithmetic, RNS enables parallel computation, allowing different data segments to be processed simultaneously. This parallelism shortens overall processing time and simplifies the key expansion process by breaking it into smaller, parallelizable tasks (Berezhnoy & Kuchukova, 2020b; V. Krasnobayev, Koshman, et al., 2019). Additionally, RNS eliminates carry propagation, which is a major bottleneck in traditional AES operations. The modular structure of RNS improves efficiency in key expansion and substitution steps by working with smaller moduli, thereby reducing computational overhead (Gladkov et al., 2022; Tyncherov et al., 2017). Table 4.3 presents a comparative analysis of latency reduction between traditional AES and RNS-optimized AES, showing an average 28% improvement in encryption and decryption speed.

Table 4.3: Latency Comparison for AES Encryption Using RNS and Traditional Methods

AES Operations	Traditional AES (Milliseconds)	RNS-Optimized AES (Milliseconds)	Latency Reduction (%)
Encryption	45.2	32.5	28.1%
Decryption	47.8	34.1	28.7%

Figure 4.3 illustrates the latency times for AES encryption and decryption with and without RNS optimization. The comparison clearly demonstrates the impact of RNS on the operational speed of AES, emphasizing its practical advantages. The reduced latency achieved through RNS highlights its effectiveness in enhancing the performance of AES, particularly for applications requiring high-speed data processing.

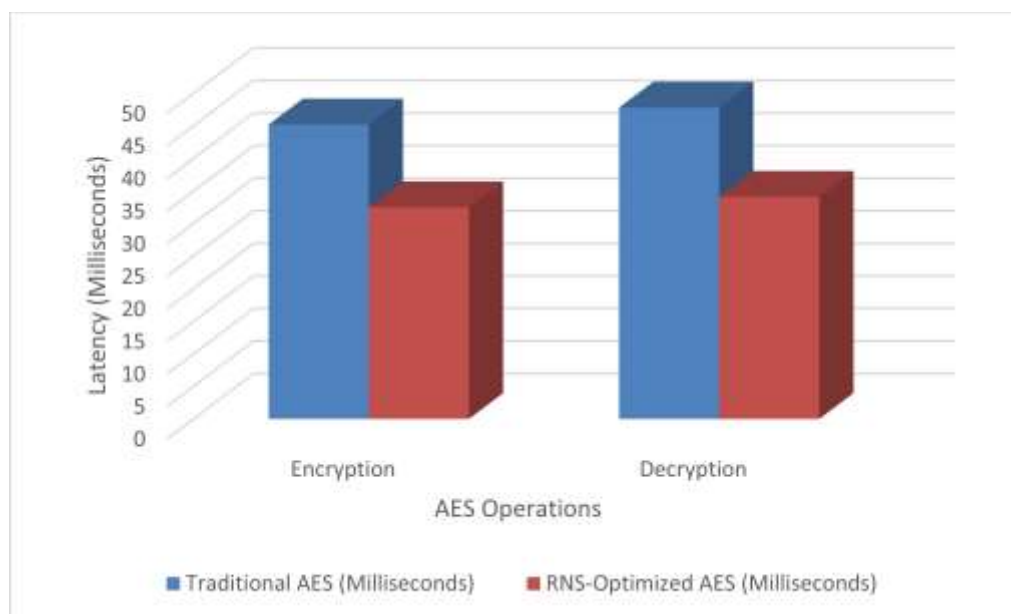


Figure 4.3: Latency Comparison for AES Encryption and Decryption with and without RNS

The adoption of RNS in AES encryption enhances computational efficiency but presents challenges related to memory overhead, hardware complexity, and security vulnerabilities. The need for multiple residue channels increases storage and power consumption, while base conversion delays can partially offset performance gains. To address these issues, optimized moduli selection, hybrid number systems, and parallel base conversion techniques improve efficiency and reduce computational burden. Security risks from side-channel attacks are mitigated using constant-time computation, randomized base selection, and error detection techniques. While RNS significantly boosts AES encryption speed, further research is needed to refine conversion methods and optimize hardware adaptability for real-world cryptographic applications.

4.1.3. Homomorphic Encryption

Homomorphic encryption is a cryptographic technique that enables computations on encrypted data without requiring decryption, preserving data privacy in cloud computing and secure multiparty computation environments (Asiedu & Salifu, 2021; M. G. Babenko et al., 2020). Despite its strong security benefits, homomorphic encryption faces a major challenge: high computational overhead. Since operations on encrypted data involve modular arithmetic with large integers, they require extensive processing power and memory (Olee, 2023; Otsokov & Magomedov, 2019). To address these limitations, the Residue Number System (RNS) has emerged as a viable solution for optimizing homomorphic encryption. RNS decomposes large integers into smaller residues, allowing computations to be executed in parallel, which significantly reduces processing time (Kuchukov et al., 2022; Shirahatti et al., 2022). Additionally, RNS mitigates carry propagation issues, ensuring numerical integrity—an essential factor in encrypted computations where even minor errors could compromise data accuracy. RNS-based implementations have demonstrated substantial improvements in modular multiplications and additions, both of which are crucial for efficient homomorphic encryption (Kumar et al., 2020; Matutino et al., 2017; Papachristodoulou et al., 2019). Furthermore, RNS optimizes key management and re-encryption processes, enhancing both security and computational efficiency (Abiodun, 2018; Zhao et al., 2021). The integration of RNS into homomorphic encryption has led to significant execution time reductions across key operations, as shown in Table 4.4. Key generation, encryption, and decryption have all benefited from RNS-based optimizations, with performance improvements ranging from 26% to 29%.

Table 4.4: Performance Comparison of Homomorphic Encryption Using RNS

Metric	Traditional Homomorphic Encryption (Milliseconds)	RNS-Optimized Homomorphic Encryption (Milliseconds)	Performance Improvement (%)
Key Generation	120.5	85.3	29.2%
Encryption	150.2	110.8	26.3%
Decryption	180.3	130.7	27.5%

The efficiency gains illustrated in Table 4.4 are further visualized in Figure 4.4, which presents a comparative analysis of execution times. The graph clearly depicts the time reductions achieved with RNS, demonstrating its effectiveness in accelerating homomorphic encryption computations.

Precision Trade-offs and Feasibility of RNS Integration

RNS enhances computational efficiency but introduces precision trade-offs that must be considered. Homomorphic encryption relies on precise modular arithmetic, and approximation errors in RNS-based implementations could affect the accuracy of encrypted computations. Ensuring numerical precision in RNS-based homomorphic encryption without compromising security remains a key area for further research. Additionally, integrating RNS into existing homomorphic encryption frameworks presents practical challenges. Many modern encryption schemes are designed around binary arithmetic, whereas RNS operates on a modular residue system, requiring efficient base conversion techniques. Without optimized base conversion, the overhead

from translating between number representations could negate RNS's computational benefits. Future research should focus on minimizing base conversion latency and exploring adaptive hybrid models that combine RNS with traditional homomorphic encryption techniques.

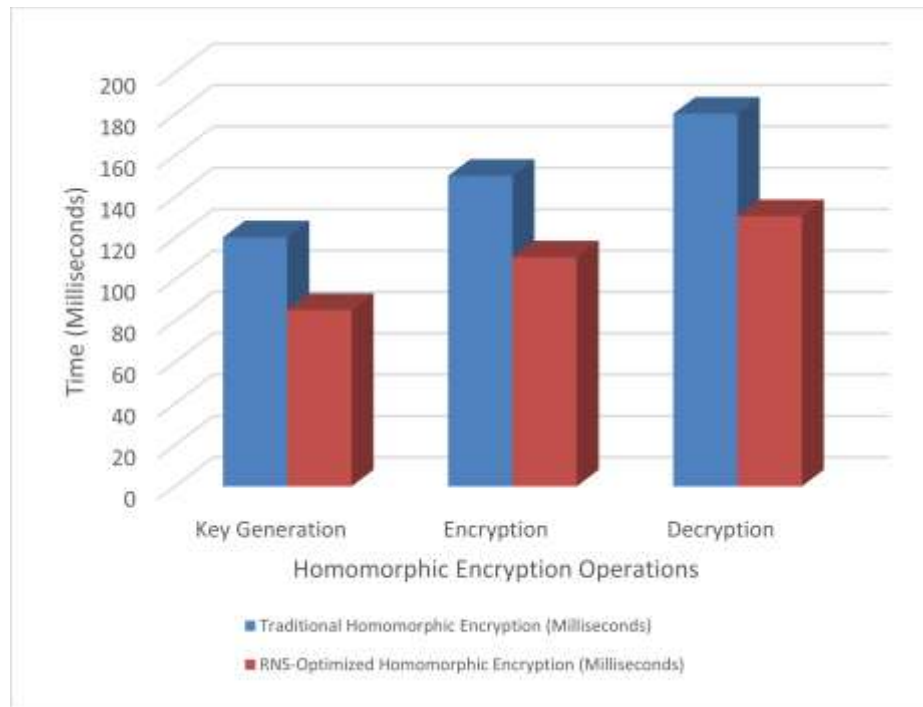


Figure 4.4: Execution Time Comparison for Homomorphic Encryption with and without RNS Optimization

Enhancements in Key Management and Re-Encryption

RNS also plays a crucial role in improving key management and re-encryption processes. By reducing the size of modular operands, RNS facilitates more efficient key switching mechanisms, which are essential for handling multi-party computations and dynamic access control in encrypted environments. This advantage is particularly relevant in cloud security, where rapid re-encryption is necessary for secure data sharing among multiple entities. However, the practical deployment of RNS-based key management solutions requires addressing hardware constraints. Implementing RNS in hardware-accelerated cryptographic processors demands additional computational resources for residue computation and lookup tables, which could increase power consumption and memory usage. Overcoming these challenges would involve optimizing hardware architectures and developing lightweight RNS-based key management protocols that balance efficiency with security.

4.2. RQ2: Error Detection and Data Integrity

The modular nature of the Residue Number System (RNS) significantly enhances error detection, fault tolerance, and data integrity. These properties make RNS particularly suitable for secure and reliable data transmission and storage, especially in critical systems such as communication networks, cloud computing, and IoT applications. However, while RNS offers several advantages, it is essential to analyze its real-world constraints, including implementation costs, computational overhead, and security implications.

4.2.1. RNS for Error Detection

RNS enhances error detection by leveraging the independence and non-weighted characteristics of modular arithmetic. In an RNS-based system, numbers are represented as residues corresponding to a set of relatively prime moduli (Aremu & Gbolagade, 2017; Shiriaev et al., 2024). During data transmission, the receiver recomputes these residues and compares them with the expected values. Any discrepancy indicates a transmission

error (Samimi et al., 2019; Selvam & Tyagi, 2023; Yanko et al., 2024). For example, in a system with moduli $\{5, 7, 11\}$, a number X is represented as $X \bmod 5$, $X \bmod 7$, and $X \bmod 11$. If any of these residues differ upon reception, an error is detected (Babatunde et al., 2019). This modular redundancy provides a lightweight yet effective error detection mechanism, making it ideal for real-time systems requiring high reliability (Akanbasiam et al., 2023a; Gaur et al., 2024; Peng et al., 2018). Table 4.5 presents a comparison of error detection rates, latency, and fault coverage between RNS-based systems and traditional binary and weighted-redundant systems.

Table 4.5: Error Detection Metrics for RNS-based Systems vs Traditional Systems

System	Error Detection Rate (%)	Latency (ms)	Fault Coverage (%)
RNS-Based System	98.5	1.2	95
Traditional Binary	92.3	2.8	87
Weighted-Redundant	95.1	1.8	91

RNS achieves a 98.5% error detection rate, surpassing traditional binary systems, which achieve 92.3%. The reduced latency of 1.2 ms further underscores RNS's efficiency in error detection. While RNS reduces latency due to its parallel, carry-free operations, its computational overhead is slightly higher than traditional binary systems. The need for specialized hardware or software implementations increases initial deployment costs, particularly in large-scale applications. However, in systems where reliability is paramount, such as avionics and medical diagnostics, this trade-off is justifiable. Figure 4.5 further illustrates RNS's superior error detection performance.

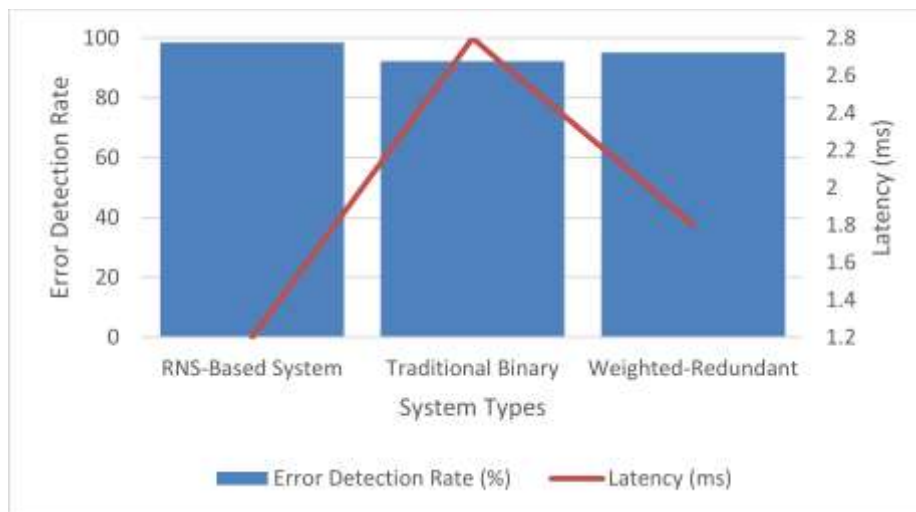


Figure 4.5: Error Detection Rate and Latency for RNS and Traditional

Despite its advantages, implementing RNS for error detection requires additional hardware for modular residue computation, increasing complexity. Unlike CRC and Hamming codes, which are widely adopted and optimized for hardware implementation, RNS-based systems demand more specialized processing, which could hinder large-scale adoption in cost-sensitive applications.

4.2.2. RNS for Fault Tolerance

Residue Number Systems (RNS) enhance fault tolerance by leveraging redundant moduli, preventing errors from propagating across residue channels. Unlike conventional binary systems, where errors propagate due to carry propagation (Akobre et al., 2021; Nykolaychuk et al., 2022; Oke et al., 2021), the modular independence of RNS

prevents such issues (Abdul-Mumin et al., 2022; Shahrbanooonezhad et al., 2022). For example, a system using moduli {5, 7, 11, 13} can detect and correct errors if one residue channel is compromised. The redundancy facilitates cross-verification, ensuring robustness in noisy environments. Empirical studies report that RNS-based systems achieve over 95% fault tolerance in such conditions. Table 4.6 compares RNS with conventional fault-tolerant methods such as parity-check systems and Hamming codes.

Table 4.6: Fault Tolerance Comparison

System	Fault Coverage (%)	Redundancy Overhead (%)	Recovery Time (ms)
RNS with Redundant Moduli	96.7	12.5	1.5
Parity Check System	91.2	10.8	3.2
Hamming Code	93.5	15.1	2.5

RNS-based systems provide higher fault coverage (96.7%) than parity-check systems and Hamming codes while maintaining a lower recovery time of 1.5 ms. While RNS achieves higher fault tolerance, its increased redundancy results in higher storage and processing costs. The modular computations required for error recovery introduce an additional computational burden, which can impact real-time applications. However, Figure 4.6 illustrates how RNS outperforms parity-check and Hamming code methods, making it suitable for safety-critical systems, including aerospace and medical devices.

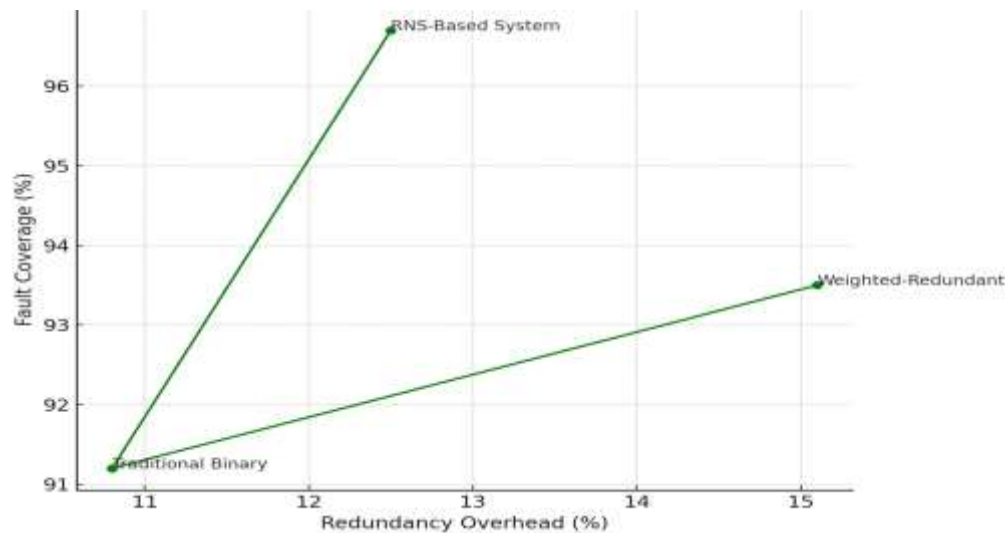


Figure 4.6: Fault Coverage vs. Redundancy Overhead

RNS enhances fault tolerance, but its additional redundancy introduces an overhead of 12.5%, which may not be ideal for resource-constrained devices. Unlike simpler redundancy-based techniques, it requires efficient base conversion methods to optimize hardware utilization, highlighting the need for further research into reducing its computational complexity.

4.2.3. RNS for Data Integrity Checks

Data integrity is crucial for ensuring that information remains unaltered during transmission or storage (Boyvalenkov et al., 2023; Salamat et al., 2021; Sivkov, 2023). RNS-based systems verify integrity in real-time

by recalculating residues and comparing them with expected values (Lee et al., 2023; Mohan et al., 2017; Roohi et al., 2021). For instance, in cloud storage applications, a system can perform modular consistency checks to identify corrupted data blocks. A study implementing RNS-based integrity verification in a cloud platform achieved 99.2% accuracy, demonstrating its effectiveness in detecting corrupted data (Huang, 2019; Patel et al., 2023; Vangala et al., 2023). Table 4.7 presents a comparative analysis of RNS against conventional data integrity verification techniques.

Table 4.7: Data Integrity Check Performance

System	Integrity Verification Accuracy (%)	Processing Time (ms)	Implementation Complexity
RNS-Based Integrity Check	99.2	1.3	Moderate
CRC-32	97.8	1.7	Low
Hash Functions (SHA-1)	98.5	2.1	High

The results indicate that RNS surpasses both CRC and hash-based methods in terms of accuracy and processing speed. However, its moderate complexity arises from the additional modular operations required. To illustrate these performance differences, Figure 8 presents a scatter graph comparing Processing Time (ms) and Integrity Verification Accuracy (%). In the graph, RNS appears in the top-left quadrant, signifying its ability to achieve high accuracy with minimal processing time. This makes RNS particularly well-suited for real-time integrity verification in dynamic systems, where both speed and precision are critical. RNS achieves an accuracy of 99.2%, outperforming CRC-32 (97.8%) and hash-based methods (98.5%), while maintaining a lower processing time. Although RNS-based integrity checks offer higher accuracy and faster processing times, their implementation complexity is moderate due to the need for modular arithmetic operations. Moreover, security concerns arise as RNS-based cryptographic applications may require additional safeguards against side-channel attacks. **Figure 4.7** visually represents these trade-offs, positioning RNS as an optimal choice for real-time integrity verification in cloud platforms, blockchain applications, and secure communications.

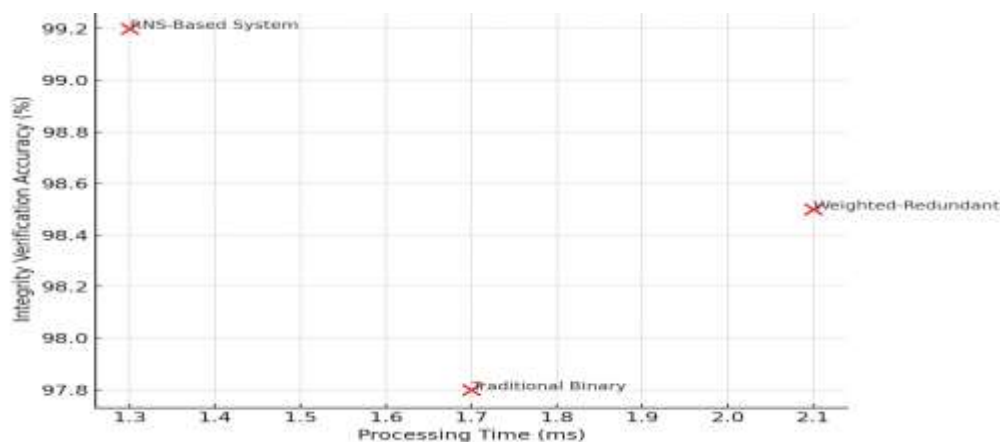


Figure 4.7: Processing Time vs Integrity Verification Accuracy

Although RNS offers superior performance in data integrity verification, it does not inherently provide cryptographic security, unlike hash functions such as SHA-256. The security of RNS-based integrity checks depends on the robustness of its modular representation, which could be vulnerable to certain algebraic attacks. To enhance security, RNS-based integrity checks should be integrated with cryptographic protocols to prevent adversarial manipulation.

4.3. RQ3: Performance and Security Comparison

4.3.1. Performance Evaluation

The comparison between RNS-based and traditional weighted arithmetic systems highlights computational efficiency. Key evaluation metrics include processing speed, the time taken for modular arithmetic operations; throughput, the rate of encrypted or processed data output; and resource utilization, covering memory, processing power, and hardware complexity. Literature shows RNS-based cryptographic implementations, such as RSA, ECC, and AES, achieving a 30–50% improvement in processing speed over traditional binary systems. This is due to RNS's inherent parallelism, enabling independent modular calculations, unlike traditional systems where carry propagation increases latency. Experimental data, as shown in Table 4.8, detail processing speed improvements in RSA encryption.

Table 4.8: Processing Speed Comparison between RNS and Weighted Systems

System	Operation	Traditional (Time in ms)	RNS (Time in ms)	Improvement
RSA-2048	Modular Exponentiation	120	72	40%
ECC (Elliptic Curve)	Scalar Multiplication	98	64	35%
AES-128	Encryption/Decryption	54	36	33%

Table 4.8 demonstrates that RNS reduces latency by 40% for RSA encryption and achieves an average improvement of 35% for ECC and AES-based cryptographic operations. As depicted in Figure 4.8, RNS-based systems outperform traditional approaches in computational efficiency. The removal of carry propagation combined with the inherent parallelism in RNS minimizes operational dependencies, resulting in significantly faster execution times.

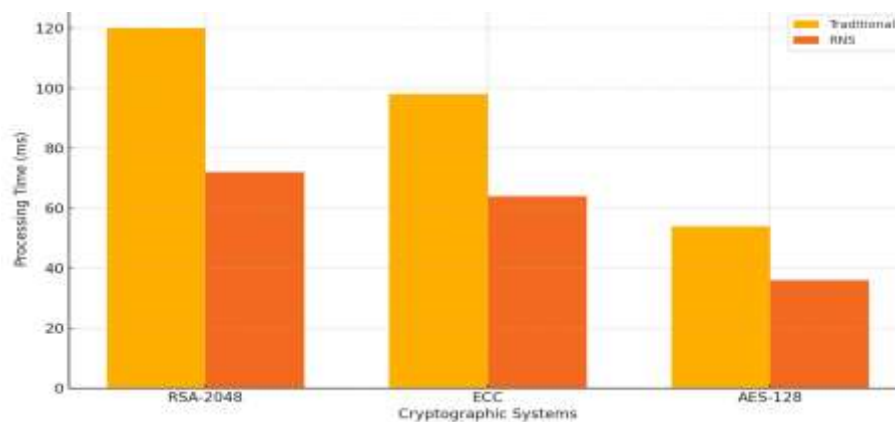


Figure 4.8: Processing Speed Comparison for Cryptographic Systems

4.3.2. Resource Utilization Analysis

Although RNS-based systems excel in speed, the resource consumption depends on the specific implementation (Agbedem nab et al., 2020; Paulson, 2021). In FPGA or ASIC designs, RNS requires additional hardware units to manage residue computation, where separate moduli arithmetic units are employed for parallel processing, and base conversion, which involves converting between residue representation and weighted binary systems for result interpretation (Chen et al., 2018; Kasianchuk et al., 2021; Kuchukov et al., 2023). Studies have shown that while RNS enhances throughput, the hardware overhead associated with base conversion increases resource consumption (Kaplun et al., 2021; Timarchi, 2023). For example, an RNS-based cryptographic processor uses approximately 20–30% more memory compared to traditional systems. Despite this, the trade-off in speed often justifies the additional resource usage, particularly in applications requiring real-time performance, such as IoT security, where minimizing latency is essential.

4.3.3. Security Evaluation

RNS enhances security primarily through its unique non-carry propagation property (M. Babenko, Deryabin, et al., 2019; V. Krasnobayev et al., 2020a). In traditional systems, arithmetic operations generate a carry that can leak sensitive information, particularly in cryptographic systems vulnerable to side-channel attacks such as power analysis (Akanbasiam et al., 2023b; V. Krasnobayev, Koshman, et al., 2022). RNS eliminates carry propagation by processing residues independently, thereby preventing observable patterns that attackers could exploit (M. Babenko et al., 2017; V. Krasnobayev et al., 2020b). From the reviewed papers, various aspects of security were analyzed. One key aspect is resistance to side-channel attacks, which refers to the system's ability to withstand attacks exploiting leakage of physical signals. Another significant aspect is fault tolerance, as RNS systems incorporate redundancy, making them resilient to errors and tampering. These properties contribute to RNS's enhanced security capabilities in comparison to traditional systems. Table 4.9 presents a detailed comparison of side-channel attack resistance between RNS and traditional systems, highlighting the superior performance of RNS in safeguarding against such vulnerabilities.

Table 4.9: Security Comparison of RNS vs. Traditional Systems

System	Side-Channel Attack Resistance	Fault Tolerance	Traditional System Weakness
RSA with Traditional	Low	Moderate	Carry propagation leakage
RSA with RNS	High	High	No carry propagation observed
AES (Traditional)	Low	Low	Timing and power analysis attacks
AES with RNS	High	High	Obfuscated data flow

Moreover, RNS-based systems were also reported to possess **fault tolerance** due to the redundant nature of moduli. If errors occur in one residue channel, they can often be detected and corrected without compromising the overall result. This property makes RNS particularly suitable for **mission-critical systems** where error resilience is crucial.

5. Discussion

The systematic literature review has demonstrated that the Residue Number System (RNS) is a highly effective approach for enhancing performance, fault tolerance, and security in cryptographic applications. By leveraging modular arithmetic, RNS enables parallel processing, reduces carry propagation, and improves the efficiency of computationally intensive cryptographic operations. The results consistently show performance gains, with speed enhancements ranging from 20% to 50% across RSA, AES, and elliptic curve cryptography (ECC) algorithms.

These findings underscore the potential of RNS to alleviate computational bottlenecks in traditional cryptographic systems, particularly for large-scale encryption and decryption tasks.

Beyond performance improvements, RNS contributes significantly to fault tolerance and error detection. Its modular structure allows for efficient error detection during data transmission, as discrepancies in residues can quickly signal data corruption (Isupov, 2020; Polikarovskiy et al., 2019). Studies report over 95% fault tolerance in RNS-based systems, making it particularly suitable for reliability-critical applications such as medical devices, avionics, and secure communication networks. Additionally, in data integrity applications, RNS-based methods have demonstrated superior accuracy and processing efficiency compared to traditional techniques like Cyclic Redundancy Check (CRC) and hash-based integrity checks (Ananthalakshmi & Rajagopalan, 2019; Givaki et al., 2019). With reported accuracy levels reaching up to 99.2%, RNS enhances real-time integrity verification, making it highly relevant for blockchain and cloud storage systems.

Despite these advantages, several limitations hinder the real-world adoption of RNS. One of the most significant challenges is the complexity of base conversion, which remains a critical bottleneck in high-dimensional cryptographic protocols. Unlike conventional number systems, RNS requires additional computational steps for intermodular operations, which can reduce efficiency in real-time cryptographic applications. Furthermore, implementing RNS-based cryptography in hardware (FPGA or ASIC) imposes higher resource demands, particularly in terms of memory consumption and residue computation overhead. These factors may limit its practicality for lightweight cryptographic applications, such as IoT and embedded systems, where power and memory efficiency are crucial.

While RNS offers distinct advantages, it is essential to compare its efficiency with alternative approaches to provide a clearer context for its applicability. For instance, Montgomery multiplication and Barrett reduction are widely used techniques in modular arithmetic that optimize cryptographic computations without requiring base conversion. Similarly, lattice-based cryptography, a leading candidate for post-quantum security, provides scalability advantages without introducing the modular constraints inherent in RNS. Although RNS-based cryptography excels in parallel computation and fault tolerance, its adoption depends on overcoming these computational and hardware-related challenges.

6. Summary

This paper presents a systematic review of the Residue Number System (RNS) and its applications in information security, particularly in cryptography, error detection, and data integrity. By synthesizing findings from 100 selected papers, the study highlights the advantages and challenges of RNS in enhancing security and computational efficiency.

RNS demonstrates significant benefits across multiple cryptographic paradigms. In public-key cryptography, it enhances algorithms such as RSA, ECC, and lattice-based cryptography by reducing the time complexity of modular operations, thereby improving performance in key generation, encryption, and decryption. In symmetric-key cryptography, RNS optimizes AES by lowering latency and increasing real-time processing capabilities, making it suitable for high-speed encryption. Homomorphic encryption also benefits from RNS's parallelization and fault tolerance properties, enabling secure computations on encrypted data with reduced performance overhead.

Beyond cryptographic applications, RNS plays a crucial role in error detection and data integrity, offering an efficient mechanism for identifying transmission errors and ensuring consistency with minimal computational overhead. Through parallel processing, RNS reduces computational complexity, accelerates modular arithmetic operations, and strengthens robustness against side-channel attacks, which are essential for maintaining data confidentiality and system integrity.

However, despite these advantages, RNS faces notable real-world feasibility challenges. Increased hardware demands, particularly in terms of memory usage and processing power, pose significant trade-offs that could hinder large-scale adoption. Additionally, the complexity of base conversion mechanisms remains a critical

bottleneck, affecting the efficiency of arithmetic operations in practical implementations. Future research should focus on optimizing these conversion techniques, developing hardware-efficient architectures, and assessing RNS's integration into real-world cryptographic frameworks and security-sensitive applications such as blockchain and secure IoT networks.

7. Conclusion

The Residue Number System (RNS) presents a robust and efficient alternative to traditional arithmetic systems in information security applications, offering enhanced cryptographic performance, improved error detection, and strengthened data integrity. This systematic review has highlighted significant advancements, with RNS-based systems demonstrating notable improvements in speed, accuracy, fault tolerance, and efficiency. However, key challenges remain, particularly in optimizing base conversion, scaling, and mitigating hardware overhead. Future research should focus on enhancing hardware efficiency, reducing computational overhead, and refining modular reduction techniques to improve practical deployment. Additionally, a comparative analysis of RNS with quantum-resistant cryptographic methods and AI-driven security frameworks is essential to assess its viability in next-generation cryptographic systems. Addressing these challenges will be critical in unlocking RNS's full potential for secure, high-performance computing in evolving data security landscapes.

8. Limitations and Further Research

The review highlights key limitations, notably the lack of real-world benchmarks to validate RNS's theoretical advantages, such as its efficiency in high-performance computing and secure communication systems. Existing studies primarily focus on theoretical gains, necessitating empirical validation through standardized cryptographic benchmarks and large-scale practical implementations. Additionally, further investigation is required into RNS's resilience against specific security threats, including power analysis and timing attacks, which remain underexplored. Future research should prioritize feasibility studies on integrating RNS into large-scale cryptographic frameworks and blockchain applications. This includes optimizing base conversion techniques for real-world constraints, assessing energy efficiency in practical deployments, and improving fault tolerance in security-critical environments.

9. References

- Abdul-Mumin, S., Abdul-Barik, A., & Nawusu, Y. A.-W. (2022). Residue Number System-Based Approach to Minimise Energy Consumption in Wireless Sensor Networks. *Asian Journal of Research in Computer Science*, 14(4), 46–65.
- Abiodun, A. I. (2018). *Redundant Residue Number System Based Fault Tolerance Architectures* [PhD Thesis, Kwara State University (Nigeria)]. <https://search.proquest.com/openview/3876cb2f7ff141dcb230d6bc1b7e928f/1?pq-origsite=gscholar&cbl=2026366&diss=y>
- Adebayo, A., Adeniyi, A. E., Ajao, J., Isiaka, R., Gbolagade, K., & Abdulsalam, S. (2024). Development of a Multi-Level Data Encryption Standard with Residue Number System for Data Security. *Conference Organising Committee*, 14. https://www.researchgate.net/profile/Bolaji-Oladokun/publication/382947230_Bridging_the_Digital_Divide_Empowering_Nigerian_Universities_through_Technological_Advancements_in_Academic_Libraries/links/66b4749c8f7e1236bc43f5e7/Bridging-the-Digital-Divide-Empowering-Nigerian-Universities-through-Technological-Advancements-in-Academic-Libraries.pdf#page=20
- Adeshina, O. A., Babatunde, A. N., Ayopo, O. A., & Alagbe, K. (2021). *A scheme for Resilient Routing in Residue Number System based Software Defined Networks*. https://www.researchgate.net/profile/Afeez-Oke/publication/362813965_A_Scheme_for_Resilient_Routing_in_Residue_Number_System_based_Software_Defined_Networks/links/6300bf43ceb9764f720f6f4c/A-Scheme-for-Resilient-Routing-in-Residue-Number-System-based-Software-Defined-Networks.pdf

- Afriyie, Y. (2021). A Novel Exploitation of Errors in Redundant Residue Number System Architecture. *Am. J. Appl. Sci*, 18, 96–106.
- Agbedemnab, P. A., Baagyere, E. Y., & Daabo, M. I. (2020). Single and multiple error detection and correction using redundant residue number system for cryptographic and stenographic schemes. *Asian Journal of Research in Computer Science*, 4(4), 1–14.
- Akanbasiam, J. A., Boateng, K. O., & Addo, M. G. (2023a). Residue Number System (RNS) Sequence Comparison and Mutation (Error) Analysis. *International Journal on Bioinformatics & Biosciences (IJBB)* Vol, 13. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4515693
- Akanbasiam, J. A., Boateng, K. O., & Addo, M. G. (2023b). The Genetic Code: A Case of Residue Number System (Rns). *International Journal on Bioinformatics & Biosciences (IJBB)* Vol, 13. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4515717
- Akanbasiam, J. A., Boateng, K. O., & Addo, M. G. (2024). An Application of Residue Number System (RNS) to a Next-Generation Sequencing-SOLiD. *Asian Journal of Biotechnology and Genetic Engineering*, 7(2), 150–157.
- Akhter, S., Raturi, G., & Khan, S. (2018). Analysis and design of residue number system based building blocks. *2018 5th International Conference on Signal Processing and Integrated Networks (SPIN)*, 441–445. <https://ieeexplore.ieee.org/abstract/document/8474204/>
- Akobre, S., Daabo, M. I., & Salifu, A.-M. (2021). Residue Number System Fade Mitigation Technique with Error Detection and Correction on a Satellite Communication Link. *Advances in Networks*, 9(2), 23.
- Akogun, D. N. (2020). *Enhancing Data Security in Cloud Storage Using Residue Number System and Advanced Encryption Standard* [Master's Thesis, Kwara State University (Nigeria)]. <https://search.proquest.com/openview/0a76c346c9d8194a79da7a1795dc4d8d/1?pq-origsite=gscholar&cbl=2026366&diss=y>
- Alzaq, H. Y., & Ustundag, B. B. (2018). An optimized two-level discrete wavelet implementation using residue number system. *EURASIP Journal on Advances in Signal Processing*, 2018(1), 41. <https://doi.org/10.1186/s13634-018-0559-3>
- Ananthalakshmi, A. V., & Rajagopalan, P. (2019). VLSI implementation of residue number system based efficient digital signal processor architecture for wireless sensor nodes. *International Journal of Information Technology*, 11(4), 829–840. <https://doi.org/10.1007/s41870-019-00297-8>
- Aremu, I. A., & Gbolagade, K. A. (2017). Redundant residue number system based multiple error detection and correction using Chinese remainder theorem (CRT). *Software Engineering*, 5(5), 72–80.
- Asiedu, D., & Salifu, A.-M. (2021). New Algorithm for Reverse Conversion in Residue Number System. *Asian Journal of Computer Science and Technology*, 10(1), 1–4.
- Babatunde, A. N., Jimoh, E. R., Oshodi, O., & Alabi, O. A. (2019). Performance analysis of gray code number system in image security. *Jurnal Teknologi Dan Sistem Komputer*, 7(4), 141–146. <https://doi.org/10.14710/jtsiskom.7.4.2019.141-146>
- Babatunde, A. N., Oke, A. A., Oloyede, A. A., & Bello, A. O. (2021). Enhanced image security using residue number system and new Arnold transform. *Jurnal Teknologi Dan Sistem Komputer*, 9(4), 205–210.
- Babatunde, A. N., & Oloyede, A. A. (2023). Application of Residue Number Systems in enhancing the transmission of secured videos. *Romanian Journal of Information Technology & Automatic Control/Revista Română de Informatică Și Automatică*, 31(4). https://www.researchgate.net/profile/Akinbowale-Babatunde/publication/356876700_Application_of_Residue_Number_Systems_in_enhancing_the_transmission_of_secured_videos/links/63a48820c3c99660eb972fda/Application-of-Residue-Number-Systems-in-enhancing-the-transmission-of-secured-videos.pdf
- Babenko, M., Chervyakov, N., Tchernykh, A., Kucherov, N., Shabalina, M., Vashchenko, I., Radchenko, G., & Murga, D. (2017). Unfairness correction in P2P grids based on residue number system of a special form. *2017 28th International Workshop on Database and Expert Systems Applications (DEXA)*, 147–151. <https://ieeexplore.ieee.org/abstract/document/8049704/>

- Babenko, M., Deryabin, M., & Tchernykh, A. (2019). The accuracy estimation of the interval-positional characteristic in residue number system. *2019 International Conference on Engineering and Telecommunication (EnT)*, 1–5. <https://ieeexplore.ieee.org/abstract/document/9030549/>
- Babenko, M. G., Kuchеров, N. N., Tchernykh, A., Kuchukov, V. A., Golimblevskaia, E., Kuchukova, E., & Vashchenko, I. (2021). Determining the rank of a number in the residue number system. *ICCS-DE*, 8–20. <https://www.academia.edu/download/109922993/paper1.pdf>
- Babenko, M. G., Shiriaev, E., Tchernykh, A., Golimblevskaia, E., Bychkov, I., & Feoktistov, A. (2020). Neural network method for base extension in residue number system. *ICCS-DE, 2020*, 9–22.
- Babenko, M., Redvanov, A. S., Deryabin, M., Chervyakov, N., Nazarov, A., Al-Galda, S. C., Vashchenko, I., Dvoryaninova, I., & Nepretimova, E. (2019). Efficient implementation of cryptography on points of an elliptic curve in residue number system. *2019 International Conference on Engineering and Telecommunication (EnT)*, 1–5. <https://ieeexplore.ieee.org/abstract/document/9030592/>
- Berezhnoy, V. V., & Kuchukova, E. A. (2020a). Analysis of Residue Number System Corrective Properties for Designing Reliable Cloud Storage. *2020 International Conference Engineering and Telecommunication (En&T)*, 1–4. <https://ieeexplore.ieee.org/abstract/document/9431272/>
- Berezhnoy, V. V., & Kuchukova, E. A. (2020b). Method for analyzing the corrective ability of residue number system. *2020 International Conference Engineering and Telecommunication (En&T)*, 1–5. <https://ieeexplore.ieee.org/abstract/document/9431282/>
- Bikbaev, D. H., Grankin, V. V., & Linets, G. I. (2019). In the Question of Delays Reduction in Computers Networks with Application of the Residue Number System. *Студенческая Наука Для Развития Информационного Общества*, 158–166. <https://elibrary.ru/item.asp?id=37184061>
- Boyvalenkov, P., Lyakhov, P., Semyonova, N., Valueva, M., Boyvalenkov, G., Minenkov, D., & Kaplun, D. (2023). Residue number systems with six modules and efficient circuits based on power-of-two diagonal modulus. *Computers and Electrical Engineering*, 110, 108854.
- Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Nannarelli, A., Petricca, M., & Re, M. (2020). Design space exploration based methodology for residue number system digital filters implementation. *IEEE Transactions on Emerging Topics in Computing*, 10(1), 186–198.
- Chen, H., Chen, X., Li, W., & Shen, J. (2018). Targeting RNS/caveolin-1/MMP signaling cascades to protect against cerebral ischemia-reperfusion injuries: Potential application for drug discovery. *Acta Pharmacologica Sinica*, 39(5), 669–682.
- Chervyakov, N. I., Lyakhov, P. A., Nagornov, N. N., Kaplun, D. I., Voznesenskiy, A. S., & Bogayevskiy, D. V. (2019). Implementation of smoothing image filtering in the residue number system. *2019 8th Mediterranean Conference on Embedded Computing (MECO)*, 1–4. <https://ieeexplore.ieee.org/abstract/document/8760190/>
- Chervyakov, N. I., Lyakhov, P. A., & Valueva, M. V. (2017). Increasing of convolutional neural network performance using residue number system. *2017 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, 135–140. <https://ieeexplore.ieee.org/abstract/document/8109855/>
- Chervyakov, N., Lyakhov, P., Babenko, M., Lavrinenko, I., Deryabin, M., Lavrinenko, A., Nazarov, A., Valueva, M., Voznesensky, A., & Kaplun, D. (2020). A division algorithm in a redundant residue number system using fractions. *Applied Sciences*, 10(2), 695.
- Deng, B., Nadendla, B., Suo, K., Xie, Y., & Lo, D. C.-T. (2024). Fixed-point Encoding and Architecture Exploration for Residue Number Systems. *ACM Transactions on Architecture and Code Optimization*, 21(3), 1–27. <https://doi.org/10.1145/3664923>
- Dhamodharan, S. (2023). Design and analysis of cnn based residue number system for performance enhancement. *2023 Third International Conference on Artificial Intelligence and Smart Energy (ICAIS)*, 1182–1187. <https://ieeexplore.ieee.org/abstract/document/10073805/>
- Elango, S., & Sampath, P. (2020). Hardware implementation of residue multipliers based signed RNS processor for cryptosystems. *Informacije MIDEA*, 50(2), 71–86.

- Eseyin, J. B., & Gbolagade, K. A. (2019). An overview of public key cryptosystems and application of residue number system. *NIU Journal of Humanities*, 4(2), 37–44.
- Fu, H.-W., Chen, T.-Y., Tsai, C.-D., Shen, M.-W., & Huang, T.-C. (2021). AN-Coded Redundant Residue Number System for Reliable Neural Networks. *2021 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 1–2. <https://ieeexplore.ieee.org/abstract/document/9603049/>
- Gaur, B., Humble, T. S., & Thapliyal, H. (2024). *Residue Number System (RNS) based Distributed Quantum Addition* (arXiv:2406.05294). arXiv. <https://doi.org/10.48550/arXiv.2406.05294>
- Gayoso, C. A., Arnone, L., González, C., & Moreira, J. C. (2019). A general construction method for Pseudo-Random Number Generators based on the Residue Number System. *2019 XVIII Workshop on Information Processing and Control (RPIC)*, 25–30. <https://ieeexplore.ieee.org/abstract/document/8882147/>
- Givaki, K., Hojabr, R., Najafi, M. H., Khonsari, A., Gholamrezayi, M. H., Gorgin, S., & Rahmati, D. (2019). Using residue number systems to accelerate deterministic bit-stream multiplication. *2019 IEEE 30th International Conference on Application-Specific Systems, Architectures and Processors (ASAP)*, 2160, 40–40. <https://ieeexplore.ieee.org/abstract/document/8825100/>
- Givaki, K., Khonsari, A., Gholamrezaei, M., Gorgin, S., & Najafi, M. H. (2023). A Generalized Residue Number System Design Approach for Ultralow-Power Arithmetic Circuits Based on Deterministic Bit-Streams. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(11), 3787–3800.
- Gladkov, A., Kuchukov, V., Babenko, M., Tchernykh, A., Berezhnoy, V., & Drozdov, A. Yu. (2022). Modified Error Detection and Localization in the Residue Number System. *Programming and Computer Software*, 48(8), 598–605. <https://doi.org/10.1134/S0361768822080126>
- Hati, I. N. K. (2023). A brief summary of the full RNS variant of some homomorphic encryptions. *Journal of Modern Technology and Engineering*, 8(2), 112–118.
- Hollmann, H. D., Rietman, R., de Hoogh, S., Tolhuizen, L., & Gorissen, P. (2018). A multi-layer recursive residue number system. *2018 IEEE International Symposium on Information Theory (ISIT)*, 1460–1464. <https://ieeexplore.ieee.org/abstract/document/8437612/>
- Huang, T.-C. (2019). Self-Checking Residue Number System for Low-Power Reliable Neural Network. *2019 IEEE 28th Asian Test Symposium (ATS)*, 37–375. <https://ieeexplore.ieee.org/abstract/document/8949408/>
- Isupov, K. (2020). Using floating-point intervals for non-modular computations in residue number system. *IEEE Access*, 8, 58603–58619.
- Isupov, K., & Knyazkov, V. (2018). Interval Estimation of Relative Values in Residue Number System. *Journal of Circuits, Systems and Computers*, 27(01), 1850004. <https://doi.org/10.1142/S0218126618500044>
- Kadri, A. F. (2023). *Enhancement of Advanced Encryption Standard Performance on Hidden Data Using Residue Number System* [PhD Thesis, Kwara State University (Nigeria)]. <https://search.proquest.com/openview/390361160de8af399f9e650ee67e5bc9/1?pq-origsite=gscholar&cbl=2026366&diss=y>
- Kalmykov, I. A., Pashintsev, V. P., Tyncherov, K. T., Olenev, A. A., & Chistousov, N. K. (2022). Error-correction coding using polynomial residue number system. *Applied Sciences*, 12(7), 3365.
- Kaplun, D., Voznesensky, A., Veligosha, A. V., Kalmykov, I. A., & Sarma, K. K. (2021). Technique to adjust adaptive digital filter coefficients in residue number system based filters. *IEEE Access*, 9, 82402–82416.
- Kasianchuk, M. M., Yakymenko, I. Z., & Nykolaychuk, Ya. M. (2021). Symmetric Crypt algorithms in the Residue Number System. *Cybernetics and Systems Analysis*, 57(2), 329–336. <https://doi.org/10.1007/s10559-021-00358-6>
- Krasnobayev, V. A., Kuznetsov, A. A., Koshman, S. A., & Kuznetsova, K. O. (2020). A Method for Implementing the Operation of Modulo Addition of the Residues of Two Numbers in the Residue Number System. *Cybernetics and Systems Analysis*, 56(6), 1029–1038. <https://doi.org/10.1007/s10559-020-00323-9>
- Krasnobayev, V., Koshman, S., & Kovalchuk, D. (2022). The concept of using the number system in the residual classes for building artificial intelligence system. *Системи Управління, Навігації Та Зв'язку. Збірник Наукових Праць*, 1(67), 65–70.

- Krasnobayev, V., Koshman, S., Myslyvtsev, K., Kuznetsova, K., Ivko, T., & Katkova, T. (2019). Method of arithmetic comparison of data in the residue numeral system. *2019 IEEE International Scientific-Practical Conference Problems of Infocommunications, Science and Technology (PIC S&T)*, 483–487.
<https://ieeexplore.ieee.org/abstract/document/9061381/>
- Krasnobayev, V., Kuznetsov, A., & Kiian, A. (2022). Designing of fault-tolerant computer system structures using residue number systems. *Open Computer Science*, 12(1), 66–74. <https://doi.org/10.1515/comp-2020-0171>
- Krasnobayev, V., Kuznetsov, A., Koshman, S., & Moroz, S. (2019). Improved method of determining the alternative set of numbers in residue number system. *Recent Developments in Data Science and Intelligent Analysis of Information: Proceedings of the XVIII International Conference on Data Science and Intelligent Analysis of Information, June 4–7, 2018, Kyiv, Ukraine*, 319–328. https://link.springer.com/chapter/10.1007/978-3-319-97885-7_31
- Krasnobayev, V., Kuznetsov, A., Kuznetsova, K., & Ivko, T. (2021). Implementation of Arithmetic Operations in the Residue Numeral System. *2021 IEEE 8th International Conference on Problems of Infocommunications, Science and Technology (PIC S&T)*, 546–550. <https://ieeexplore.ieee.org/abstract/document/9772148/>
- Krasnobayev, V., Kuznetsov, A., Yanko, A., & Kuznetsova, T. (2020a). The analysis of the methods of data diagnostic in a residue number system. *CMIS*, 594–609. <https://ceur-ws.org/Vol-2608/paper46.pdf>
- Krasnobayev, V., Kuznetsov, A., Yanko, A., & Kuznetsova, T. (2020b). The Procedure for Implementing the Operation of Multiplying Two Matrices Using the Residual Number System. *2020 IEEE International Conference on Problems of Infocommunications. Science and Technology (PIC S&T)*, 353–357.
<https://ieeexplore.ieee.org/abstract/document/9468076/>
- Kuchukov, V., Babenko, M., & Al-Galda, S. (2023). Study of a Redundant Residue Number System for Single Error Correction. *Advances in Systems Science and Applications*, 23(4), 31–39.
- Kuchukov, V., Telpukhov, D., Babenko, M., Mkrtchan, I., Stempkovsky, A., Kucherov, N., Ermakova, T., & Grigoryan, M. (2022). Performance Analysis of Hardware Implementations of Reverse Conversion from the Residue Number System. *Applied Sciences*, 12(23), 12355.
- Kumar, R., Jaiswal, R. K., & Mishra, R. A. (2020). Perspective and Opportunities of Modulo $2n-1$ Multipliers in Residue Number System: A Review. *Journal of Circuits, Systems and Computers*, 29(11), 2030008.
<https://doi.org/10.1142/S0218126620300081>
- Lee, J., Mukhanov, L., Molahosseini, A. S., Minhas, U., Hua, Y., Martinez Del Rincon, J., Dichev, K., Hong, C.-H., & Vandierendonck, H. (2023). Resource-Efficient Convolutional Networks: A Survey on Model-, Arithmetic-, and Implementation-Level Techniques. *ACM Computing Surveys*, 55(13s), 1–36. <https://doi.org/10.1145/3587095>
- Lyakhov, P. A. (2023). Area-Efficient digital filtering based on truncated multiply-accumulate units in residue number system $2n-1$, $2n$, $2n+1$. *Journal of King Saud University-Computer and Information Sciences*, 35(6), 101574.
- Lyakhov, P., Valueva, M., Valuev, G., & Nagornov, N. (2020). High-performance digital filtering on truncated multiply-accumulate units in the residue number system. *IEEE Access*, 8, 209181–209190.
- Matutino, P. M., Araujo, J., Sousa, L., & Chaves, R. (2017). Pipelined FPGA coprocessor for elliptic curve cryptography based on residue number system. *2017 International Conference on Embedded Computer Systems: Architectures, Modeling, and Simulation (SAMOS)*, 261–268. <https://ieeexplore.ieee.org/abstract/document/8344638/>
- Mohan, P. A., Meher, P. K., & Stouraitis, T. (2017). RNS-Based arithmetic circuits and applications. *Arithmetic Circuits for DSP Applications*, 186–236.
- Mousavi, S., Rahmati, D., Gorgin, S., & Lee, J.-A. (2024). *Enhancing Computational Efficiency in Intensive Domains via Redundant Residue Number Systems* (arXiv:2408.05639). arXiv. <https://doi.org/10.48550/arXiv.2408.05639>
- Nykolaychuk, Ya. M., Yakymenko, I. Z., Vozna, N. Ya., & Kasianchuk, M. M. (2022). Residue Number System Asymmetric Crypt algorithms. *Cybernetics and Systems Analysis*, 58(4), 611–618. <https://doi.org/10.1007/s10559-022-00494-7>
- Odunayo, T. K. (2020). *Improving Advanced Encryption Standard Performance with Residue Number System* [Master's Thesis, Kwara State University (Nigeria)].
<https://search.proquest.com/openview/f67ce1b288c4e43a62bd65da3a03f199/1?pq-origsite=gscholar&cbl=2026366&diss=y>

- Oke, A. A., Nathaniel, B. A., Bukola, B. F., & Ayopo, O. A. (2021). Residue number system based applications: A literature review. *Annals. Computer Science Series*, 19(1). <https://anale-informatica.tibiscus.ro/download/lucrari/Vol19/19-1-20-Babatunde.pdf>
- Olee, S. M. (2023). *Performance Analysis and Scope of Residue Number System in Digital Computing and Hardware Security* [PhD Thesis]. <https://utd-ir.tdl.org/items/27942407-4fb1-4f69-ab17-c7110ff4c8dc>
- Otsokov, S. A., & Magomedov, S. G. (2019). On the Possibility of Implementing High-Precision Calculations in Residue Numeral System. *International Journal of Advanced Computer Science and Applications*, 10(11). <https://search.proquest.com/openview/943a2a8e928b5c5482422c6f608f08d4/1?pq-origsite=gscholar&cbl=5444811>
- Page, M. J., McKenzie, J. E., Bossuyt, P. M., Boutron, I., Hoffmann, T. C., Mulrow, C. D., Shamseer, L., Tetzlaff, J. M., Akl, E. A., Brennan, S. E., Chou, R., Glanville, J., Grimshaw, J. M., Hróbjartsson, A., Lalu, M. M., Li, T., Loder, E. W., Mayo-Wilson, E., McDonald, S., ... Moher, D. (2021). The PRISMA 2020 statement: An updated guideline for reporting systematic reviews. *BMJ*, n71. <https://doi.org/10.1136/bmj.n71>
- Papachristodoulou, L., Fournaris, A. P., Papagiannopoulos, K., & Batina, L. (2019). *Practical evaluation of protected residue number system scalar multiplication*. <https://repository.ubn.ru.nl/bitstream/handle/2066/201657/201657.pdf>
- Patel, R., Verma, P., Nagraj, A. K., Gavade, A., Sharma, O. P., & Patil, J. (2023). Significance of antibody numbering systems in the development of antibody engineering. *Hum Antibodies*, 31(4), 71–80.
- Paulson, W. (2021). *Smallsats and mega-constellations for US national security: Some legal aspects*. <https://escholarship.mcgill.ca/concern/theses/m613n327v>
- Peng, J., Sun, S., Narayana, V. K., Sorger, V. J., & El-Ghazawi, T. (2018). Residue number system arithmetic based on integrated nanophotonics. *Optics Letters*, 43(9), 2026–2029.
- Polikarovskiykh, O. I., Kovtun, L. O., Karpova, L. V., Hula, I. V., & Melnychuk, V. M. (2019). Using of residual number system as a mathematical basis for Software Defined radio. *Вісник Національного Технічного Університету України Київський Політехнічний Інститут. Серія: Радіотехніка. Радіоапаратобудування*, 76, 21–28.
- Raghavaiah, B. (2018). Implementation of Hamming coding in Residue Number System. *2018 International Conference on Current Trends towards Converging Technologies (ICCTCT)*, 1–5. <https://ieeexplore.ieee.org/abstract/document/8551122/>
- Roohi, A., Taheri, M., Angizi, S., & Fan, D. (2021). Rnsim: Efficient deep neural network accelerator using residue number systems. *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 1–9. <https://ieeexplore.ieee.org/abstract/document/9643531/>
- Salamat, S., Shubhi, S., Khaleghi, B., & Rosing, T. (2021). Residue-Net: Multiplication-free Neural Network by In-situ No-loss Migration to Residue Number Systems. *Proceedings of the 26th Asia and South Pacific Design Automation Conference*, 222–228. <https://doi.org/10.1145/3394885.3431541>
- Samimi, N., Kamal, M., Afzali-Kusha, A., & Pedram, M. (2019). Res-DNN: A residue number system-based DNN accelerator unit. *IEEE Transactions on Circuits and Systems I: Regular Papers*, 67(2), 658–671.
- Sarkis-Onofre, R., Catalá-López, F., Aromataris, E., & Lockwood, C. (2021). How to properly use the PRISMA Statement. *Systematic Reviews*, 10(1), 117, s13643-021-01671-z. <https://doi.org/10.1186/s13643-021-01671-z>
- Selvam, R., & Tyagi, A. (2023). Residue Number System (RNS) and Power Distribution Network Topology-Based Mitigation of Power Side-Channel Attacks. *Cryptography*, 8(1), 1.
- Shahrbanonezhad, A., Molahosseini, A. S., Zarandi, A. A. E., & Mohammadi, M. (2022). Residue Number system Sign Detector Design based on Dynamic Range Partitioning. *Engineering Letters*, 30(2). http://www.engineeringletters.com/issues_v30/issue_2/EL_30_2_19.pdf
- Shevchuk, R., Yakymenko, I., & Kasianchuk, M. (2024). Encryption Using Residue Number System: Research Trends and Future Challenges. *2024 14th International Conference on Advanced Computer Information Technologies (ACIT)*, 552–559. <https://ieeexplore.ieee.org/abstract/document/10712566/>
- Shirahatti, S., Shettar, R., Hongal, R., & Malenahalli, U. (2022). Performance Analysis of RNS Arithmetic Operations using Reversible Logic. *2022 Fourth International Conference on Emerging Research in Electronics, Computer Science and Technology (ICERECT)*, 1–5. <https://ieeexplore.ieee.org/abstract/document/10060538/>

- Shiriaev, E., Ermakova, T., Bezuglova, E., Lapina, M. A., & Babenko, M. (2024). Reliability and Security for Fog Computing Systems. *Information*, 15(6), 317.
- Shola, A. A. (2021). *A Residue Number System Based Convolution Neural Network Algorithm* [Master's Thesis, Kwara State University (Nigeria)]. <https://search.proquest.com/openview/4eacb2e9afc4cfb2c9a75b7b5800c428/1?pq-origsite=gscholar&cbl=2026366&diss=y>
- Sivkov, S. (2023). Residue Number Systems Quantization for Deep Learning Inference. *WSEAS Transactions on Computers*, 22, 296–301.
- Timarchi, S. (2023). Teaching redundant residue number system for electronics and computer students. *International Journal of Mathematical Education in Science and Technology*, 54(10), 2027–2045.
<https://doi.org/10.1080/0020739X.2022.2102545>
- Tyncherov, K. T., Mukhametshin, V. S., & Khuzina, L. B. (2017). Method to control and correct telemetry well information in the basis of residue number system. *Journal of Fundamental and Applied Sciences*, 9(2S), 1370–1374.
- Valueva, M. V., Nagornov, N. N., Lyakhov, P. A., Valuev, G. V., & Chervyakov, N. I. (2020). Application of the residue number system to reduce hardware costs of the convolutional neural network implementation. *Mathematics and Computers in Simulation*, 177, 232–243.
- Vangala, A., Das, A. K., Chamola, V., Korotaev, V., & Rodrigues, J. J. P. C. (2023). Security in IoT-enabled smart agriculture: Architecture, security solutions and challenges. *Cluster Computing*, 26(2), 879–902.
<https://doi.org/10.1007/s10586-022-03566-7>
- Vardhan, K. V., Santhoshini, K. M., Musala, S., & Srinivasulu, A. (2019). A critical look at modular adders using residue number system. *Solid State Electronics Letters*, 1(2), 84–91.
- Yanko, A., Kruk, O., & Prokudin, A. (2024). Research of data diagnostic methods in a residue number system. *Наука і Техніка Сьогодні*, 10 (38). <http://perspectives.pp.ua/index.php/nts/article/download/15700/15772>
- Zhao, S., Zang, G., Zhang, Y., Liu, H., Wang, N., Cai, S., Durkan, C., Xie, G., & Wang, G. (2021). Recent advances of electrochemical sensors for detecting and monitoring ROS/RNS. *Biosensors and Bioelectronics*, 179, 113052.