The ISC Int'l Journal of Information Security
ISeCure

October 2025, Volume 17, Number 3 (pp. 1–10)
http://www.isecure-journal.org

Presented at the ISCISC'2025 in Tehran, Iran.

Defending CNNs Against Power Side-channel Attacks: A Residue Number System Solution *

Marzieh Morshedzadeh [1], Reyhaneh Hajimohammadali [1], and Ali Jahanian [1,??]
[1] *Faculty of Computer Science and Engineering, Shahid Beheshti University, Tehran, Iran*

ARTICLE INFO.

Keywords:
Neural Networks - Security - Side-Channel Attacks - Residue Number System - FPGA

Type:

doi:

Abstract

Neural networks are increasingly used in safety-critical domains such as autonomous driving and medical applications. Their dependence on sensitive parameters, however, makes them attractive targets for adversaries. Among the various existing threats, power side-channel attacks aimed at recovering model parameters are particularly concerning. This work presents a defence mechanism based on the Residue Number System (RNS) to strengthen neural network implementations against such attacks. Security can be improved by carefully selecting computational parameters without sacrificing accuracy. Our experimental results on FPGA hardware with LeNet-5 show up to 3× improvement in resistance to correlation power analysis (CPA) attacks before reverse conversion (RC) and 2× improvement after RC, while preserving classification accuracy. We believe this is the first systematic integration of RNS into convolutional neural network inference on FPGA hardware. While prior modular approaches protect only isolated operations, our method applies RNS arithmetic comprehensively, thereby providing stronger resilience against adversaries.

1 Introduction

Neural networks have revolutionised a wide array of domains, including computer vision, natural language processing, and business intelligence. In addition to these domains, deep neural networks (DNNs) play a crucial role in vital sectors like the military, healthcare, cybersecurity, real-time decision-making systems, and self-driving vehicles [1, 2].

While neural networks have an extensive range of applications, they face significant challenges related to security vulnerabilities and privacy threats. Their complexity, characterised by numerous parameters such as weights, biases, and the specifics of network architecture like the number of neurons and layers, makes them susceptible to attacks. These parameters complicate efforts to safeguard these networks from potential threats [3, 4].

Attackers may have different reasons for targeting neural networks, such as making money, accessing private information, invading people's privacy, or damaging a competitor's business. Accordingly, the use of neural networks in important applications has driven

* The ISCISC'2025 program committee effort is highly acknowledged for reviewing this paper.

Email addresses: m_morshedzadeh@sbu.ac.ir, r_hajimohammadali@sbu.ac.ir, jahanian@sbu.ac.ir

ISSN: 2008-2045

a significant increase in malicious activity. This shift has prompted researchers to focus more on strengthening the security of these systems [6, 8, 9].

Despite their computational power, neural networks are highly vulnerable to attacks, making their security a matter of utmost importance. Side-channel attacks, a common threat to these networks, are primarily used to extract parameters such as weights. One of the most serious side-channel attacks is based on power consumption. Specifically, neural networks are highly susceptible to these attacks due to power consumption patterns that exhibit significant leakage during operations. Such leakage results in the disclosure of critical network information targeted by adversaries. In particular, operations such as matrix multiplication, activation functions, and memory access are the primary contributors to this power leakage [10, 11].

Protecting neural networks is challenging due to their numerous parameters, and effective methods to defend against physical attacks, especially side-channel attacks, are still underdeveloped, leaving much room for improvement. To develop an efficient method to counter or mitigate side-channel attacks is of high importance [12].

For defending deep neural networks, common methods include introducing random noise into internal computations, masking (encrypting sensitive data using XOR), shuffling, and randomisation. Additionally, obfuscation methods such as layer skipping, layer branching, and layer deepening are employed to protect the architecture of DNNs against side-channel attacks [17–20].

Authors of [21] use the parametric noise injection (PNI) method to enhance the robustness of deep neural networks (DNNs) against adversarial attacks. This method involves injecting trainable Gaussian noise at various layers of the network, including weights and activations, all of which are optimised through an end-to-end training process. This technique enables the model to adaptively learn optimal noise levels, thereby improving its resilience against both white-box and black-box attacks.

In [18], the DNN-alias method is used to protect deep neural networks (DNNs) against side-channel attacks by employing various layer obfuscation techniques. Specifically, DNN-alias alters the runtime traces of DNN layers to ensure that all layers exhibit similar execution patterns, making it difficult for attackers to differentiate between different layers. The method utilises an evolutionary algorithm to optimise the combination of obfuscation operations, thereby maximising security while staying within a user-defined latency overhead budget.

Authors of [22] discuss various masking techniques, including Boolean masking, where sensitive values are split into randomised shares, and operations are performed independently on each share. Although this method has shown success in cryptographic circuits, extending it to DNNs presents considerable difficulties due to the high volume of multiply-accumulate (MAC) operations.

The author of [23] discusses the vulnerabilities of modern electronic systems, particularly in the context of side-channel attacks that exploit power consumption to extract sensitive information. The study emphasises the need for secure design methodologies that protect against both power leakage and cryptographic vulnerabilities. This work introduces a secure logic design based on the residue number system (RNS), which demonstrates improved resistance to side-channel attacks. The study also explores the impact of power distribution network topology on security, providing insights into how different configurations can mitigate information leakage.

The authors of [24] propose a secure design style that transforms binary inputs into multiple encrypted shares using RNS, providing both cryptographic and side-channel privacy. This study also highlights the effectiveness of RNS logic in masking power consumption, making it difficult for adversaries to distinguish between different input values.

The authors of [25] propose a novel approach to enhance the security of neural networks by using modular arithmetic to address the sign-bit leakage problem in masked BNNs. By replacing integer additions with modular counterparts, this method effectively eliminates the sign bit, thereby preventing information leakage and reducing susceptibility to side-channel attacks. Furthermore, the researchers develop a new layer architecture that incorporates modular arithmetic, ensuring that the operations remain efficient and secure. The study demonstrates that this method can achieve high accuracy with minimal information loss by carefully selecting the modulus k.

The authors of [26] explore differential power analysis (DPA) attacks on neural networks, specifically targeting the extraction of secret model parameters such as weights and biases during inference. The study demonstrates that neural networks, like cryptographic systems, are vulnerable to power side-channel attacks that can reveal sensitive information through power measurements. Furthermore, they conduct experiments on a binarised neural network (BNN) implemented on a SAKURA-X FPGA board, showing that first-order DPA attacks can successfully extract secret weights with as few as 200 traces. To address these vulnerabilities, the paper proposes a novel masking technique to protect neural network inferences from power-based side-channel attacks. The proposed countermeasure involves splitting inputs into randomised shares, which are processed independently to prevent correlation with secret values.

So far, many studies have been performed in the field of neural network security. The complexity of neural networks is much higher than that of encryption systems due to the numerous parameters that need protection (such as weights, activation functions, biases, number of neurons, and number of layers). The different and complex architectures of neural networks, along with their strong dependence on data (in the same network, the execution path for input data can vary significantly, which affects power consumption and execution time), make defending against attacks on these networks very challenging. Most studies have limitations and fundamental assumptions for a successful attack. While valuable for identifying weaknesses, these studies become invalid if conditions change or are removed. Consequently, presenting a method independent of these factors-capable of operating on any network-is essential. Serious research on the impact of mathematical parameters of numerical systems on the security of various network models has not been reported. Considering the importance of neural network security, this area is highly strategic and significant.

In this paper, we explore the potential of the unconventional residue number system (RNS) to enhance the security of neural networks against side-channel attacks. Implementing neural networks using this numerical system introduces a layer of complexity that makes it more difficult for attackers to breach. Conducting computations within the residue system significantly obscures key system parameters, thereby increasing ambiguity and enhancing security. The distinctive way in which multiplication and addition are performed in the residue system contributes to this increased complexity. The urgent need for innovative solutions to enhance the resilience of deep neural networks (DNNs) against sophisticated threats positions the RNS as a promising candidate. The inherent properties of the RNS effectively obscure power consumption patterns, thereby reducing the risk of information leakage and strengthening the network's defence mechanisms.

This paper is organised as follows. Section 2 describes the background information of the research. Section 3 illustrates the proposed protection method. In Section 4, experimental results are reported and discussed, and finally, Section 5 concludes the paper.

2 Basic Concepts

2.1 Side-Channel Attacks

Following the introduction of the first widely recognised side-channel attack on public-key cryptosystems by Paul Kocher [27], the security community developed a strong interest in physical attack methods, including timing analysis, power consumption, electromagnetic (EM) emissions, and even acoustic signals. Side-channel attacks exploit device physical emissions to extract sensitive information, such as cryptographic keys. Rather than targeting algorithms or protocols directly, these attacks exploit the physical attributes and behaviours of systems that unintentionally expose internal processes [27–29]. They leverage these unintentional leakage paths during computation to retrieve cryptographic keys or bypass security protections.

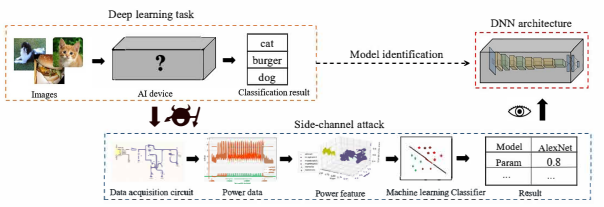

Figure 1. Side-channel attack on a DNN model [33]

Power side-channel attacks, a type of SCA, exploit physical signals emitted by devices to recover cryptographic keys. Fundamentally, device power consumption patterns correlate with active computations, thereby revealing confidential parameters. Specifically, an attacker may monitor the power usage of an FPGA during cryptographic operations and apply techniques such as Simple Power Analysis (SPA), Differential Power Analysis (DPA), or Correlation Power Analysis (CPA) to recover cryptographic keys [30, 31]. Figure 1 illustrates the flow of physical attacks on a neural network using the side-channel attack method, where voltages are collected by a sensor and analysed using machine learning (ML) techniques. Subsequently, model information, including the architecture, is extracted.

Simple Power Analysis (SPA) is a fundamental side-channel attack that exploits fluctuations in power consumption during cryptographic operations. Analysing these traces enables an attacker to identify the specific operations being executed [34].

Differential Power Analysis (DPA) is an advanced technique exploiting subtle fluctuations in power consumption during cryptographic operations. In this approach, numerous power traces are collected while the device processes different inputs; subsequently, statistical tools are applied to these traces to detect small variations that correlate with cryptographic keys. By generating hypothetical key values and building power models, the attacker can compare these predictions with actual traces to identify the correlations that reveal the correct key [34].

Correlation Power Analysis (CPA) is an advanced technique that builds on the principles of Differential Power Analysis (DPA) by focusing on statistical

relationships between measured power consumption and hypothetical intermediate values during cryptographic operations. During a CPA attack, researchers record power traces while the device is executing cryptographic operations and then use statistical methods to compare them with models built on assumed intermediate data, such as S-box outputs in algorithms like AES. By identifying which hypothetical scenario has the strongest correlation with the recorded power data, attackers can deduce the most likely values of cryptographic keys [34].
In this paper, we employ Correlation Power Analysis (CPA), which links power consumption measurements with hypothetical values derived from intermediate stages of cryptographic algorithms.

2.2 Vulnerabilities of Neural Networks

Artificial intelligence (AI) is widely used in numerous real-world applications, including weather forecasting, transportation systems, social media platforms, healthcare services, and advertising strategies [1, 32]. Within this context, machine learning (ML) is a specialised field that uses algorithms to process large-scale data, extract valuable knowledge, and support informed decision-making. Deep learning (DL), a subset of ML, enables machines to process raw data and automatically extract meaningful features. It has proven highly effective in analysing complex, high-dimensional data and addressing long-standing challenges in image, speech, text, video, and audio recognition.
Deep learning (DL) uses deep neural networks (DNNs) composed of input, hidden, and output layers. Among the various DL architectures, including fully connected feed-forward networks and multi-layer perceptrons, convolutional neural networks (CNNs) are the most widely used. An NN model's architecture and parameters define its structure and behaviour, requiring the model to be trained and fine-tuned using a training dataset. During this process, the parameters-specifically weights and biases-are adjusted using the backpropagation algorithm, enabling the trained network to perform classification or regression tasks [32].
In this paper, we focus on a specific type of NN called LeNet-5. Figure 2 shows the architecture of the LeNet-5 network. The network consists of three convolutional layers and two fully connected layers, with ReLU activations and average pooling [27].

Deep neural networks (DNNs) are highly vulnerable to various threats, including adversarial, fault injection, trojan, and side-channel attacks [4, 37]. Adversarial attacks introduce small, carefully crafted perturbations to input data, thereby causing the network to misclassify inputs. Fault injection attacks manipulate hardware or software components of

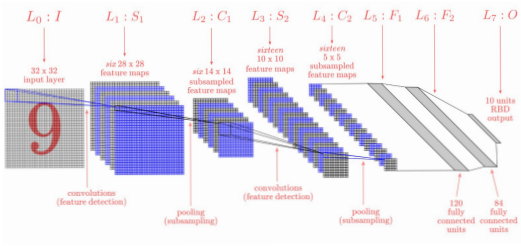

Figure 2. The LeNet-5 convolutional neural network model

the network, leading to incorrect computations and outputs.
Trojan attacks embed malicious functionality within the network, causing it to behave incorrectly under specific conditions. Additionally, side-channel attacks exploit physical implementation leakages-such as power consumption or timing-to infer sensitive data. The complex, multi-layered architecture of DNNs, combined with the high dimensionality of their processed data, makes them particularly vulnerable to these threats. Reliance on large amounts of training data and the non-linear transformations within the network make it challenging to detect and mitigate such attacks, thereby highlighting the need for robust defence mechanisms [4, 27, 38].

2.3 RNS-based Neural Networks

Long and unavoidable carry propagation chains often constrain the performance of binary computational systems. This limitation increases power consumption and degrades the speed of arithmetic operations within their data paths [38]. Unconventional number systems, such as the Residue Number System (RNS), provide an effective solution to mitigate these inefficiencies. The RNS is an unconventional, non-weighted system in which computations are carried out in parallel on the remainders of numbers divided by several coprime moduli. By decomposing long carry chains into several shorter parallel chains, RNS significantly accelerates addition, subtraction, and multiplication compared to weighted systems like two's complement. Figure 3 depicts the process of converting a decimal number to remainders, then performing mathematical operations on them, and finally reconstructing the original decimal number, as a block diagram.

In the RNS, binary numbers with long word-lengths are decomposed into shorter components, known as residues, based on a chosen set of moduli. Arithmetic operations-including addition, subtraction, and multiplication-are executed on these residues independently and in parallel, eliminating carry propagation across modulus channels. This characteristic significantly improves speed and reduces power consump-

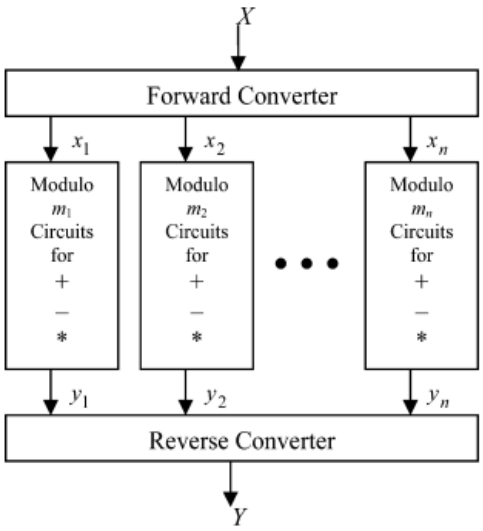

Figure 3. General structure of an RNS-based processor [34]

tion in RNS-based implementations, particularly for tasks involving high-volume arithmetic. Research indicates that the implementation of these applications using RNS provides numerous advantages. The parallel and modular nature of the RNS enhances the efficiency of data paths in processing structures dominated by arithmetic operations, thereby increasing speed and reducing power consumption without compromising latency.

Using the Residue Number System (RNS) to represent numbers and perform calculations, instead of the conventional binary system, is an effective approach that offers strong resistance against side-channel attacks. Employing the residue number system (RNS) in deep neural networks enhances resilience against side-channel attacks, particularly power side-channel attacks, for the following reasons: the residue number system (RNS) is a non-standard, non-weighted system in which computations are performed in parallel on the remainders obtained by dividing a number by several moduli that are pairwise coprime. The parallel nature of this system inherently facilitates the division of operations across multiple channels.

Within this system, arithmetic operations are distributed across multiple parallel channels. This parallelisation reduces the dependence of power consumption on the processed data, making power analysis more difficult for attackers. Additionally, the ability of RNS to split numbers into residues naturally introduces noise. This noise can obscure side-channel signals, making it difficult for attackers to rely on these signals to infer sensitive information.

Unlike traditional arithmetic, RNS operations do not depend on the carry bit, a major source of side-channel leakage. This can significantly reduce information leakage through physical channels. Since RNS

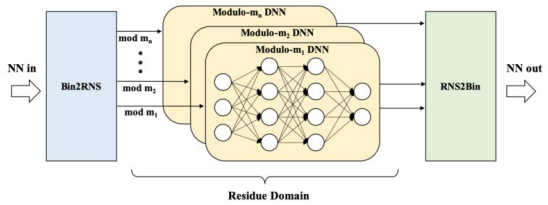

Figure 4. RNS-based neural network architecture [35]

operates with multiple moduli, reconstructing the original number from residues is non-trivial without knowledge of the entire set of moduli, which enhances security. Moreover, in the residue number system (RNS), calculations such as addition, subtraction, and multiplication are performed simultaneously across multiple independent channels. This distribution of power consumption across channels results in more consistent power usage and minimises sudden changes and power peaks. Consequently, it becomes difficult for an attacker to link power fluctuations with specific operations or operands, making such attacks harder to execute.

The numerical system chosen for representing numbers in deep neural networks significantly affects the accuracy of operations, speed, hardware requirements, energy consumption, and even network security. As neural networks advance and become deeper, the number of parameters and the volume of computations increase considerably, with most computations involving MAC (multiply-accumulate) operations. Due to the substantial computational demands of deep neural networks, employing RNS is effective for implementing these computations. Figure 4 illustrates a computing architecture for a DNN accelerator based on the residue number system.

There are two approaches to using this numerical system in deep neural networks: fully RNS-based and partially RNS-based implementations.

In the fully RNS-based approach, each input is initially converted to the chosen set of moduli, and all computations across the network layers are performed within the RNS domain. Finally, an inverse transformation converts the final results from the modular domain back to the conventional binary domain. In summary, this approach involves only one forward transformation and one inverse transformation.

However, in the partially RNS-based approach, only the convolutional layers, requiring a high number of multiplications and additions, are computed in the RNS domain. After the computations are completed and before the activation function is applied, the results are converted back to the binary number system using a reverse converter. Then, after applying the activation function (and any filtering), the output for the next layer is converted back to the RNS domain

using a forward converter. This method incurs overhead because forward and reverse converters must be repeatedly applied to the data.

3 The Proposed Protection Method

So far, the reason for using residue number systems in neural networks has been to increase computational speed and reduce power consumption. This is because, in these systems, operations are performed in parallel across multiple channels using fewer bits, thereby enabling faster computations and lower power consumption. However, the use of these number systems for protecting the architecture and important parameters of neural networks has not been seriously considered until now [37].

In this paper, we demonstrate that the use of the residue number system in neural network implementations, due to parallelisation across multiple channels and the independence of computational channels from one another (carry-free), increases ambiguity and helps hide important system parameters. Additionally, due to the specific nature of performing multiplication, addition, and comparison operations in this number system, system complexity increases, making attacks more difficult.

We experimentally performed addition and multiplication operations in the first convolutional layer of the LeNet-5 neural network using the residue number system (RNS) with three well-known moduli: $2^n+1, 2^n, 2^n-1$. Since these operations are executed in parallel across three channels-determined by the chosen number of moduli-we observed that power consumption during addition and multiplication is distributed across these channels, thereby reducing power leakage compared to the conventional binary case. Our implementation for testing the performance of the residue number system (RNS) uses a fully RNS approach, with both forward and reverse conversion modules utilised in the first layer. The primary goal is to prevent attackers from extracting the network weights, which are among the most critical parameters of a neural network. Without access to the weight values, an attacker cannot reverse-engineer the network or insert a Trojan into it.

We examined the protection of the neural network against side-channel attacks using the residue number system in two scenarios. In the first scenario, the attack targeted the outputs of the reverse conversion, while in the second scenario, it targeted the outputs of multiplication for each modulus separately. Figure 5 illustrates both scenarios.

To prevent side-channel attacks using this method, a critical issue is the selection of a moduli set in residue number systems (RNS). This selection plays a crucial role in enhancing the security of neural networks against such attacks. Selecting the right moduli, based on their number and form, ensures that all input values and intermediate results are adequately covered, thereby reducing vulnerabilities. The choice of moduli also influences the complexity of arithmetic operations; efficient computations can increase the difficulty for attackers who analyse power traces. A well-selected modular set can minimise the correlation between power consumption and processed data, thereby improving resistance to correlation attacks. Ultimately, the careful selection of moduli enhances both computational efficiency and neural-network security.

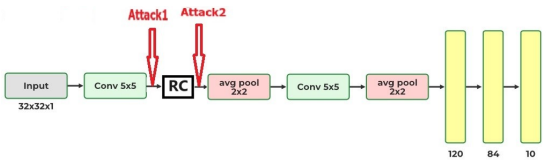

Figure 5. Attack scenarios on the RNS-based neural network before and after reverse conversion

In our implementation, only the first convolutional layer is executed entirely in the RNS domain, with both forward and reverse converters applied at its boundaries. Consequently, at the layer level, this part is described as a fully RNS implementation. However, since the subsequent layers of the network remain in the conventional binary domain, the design should be regarded as a partially RNS network at the overall system level.

4 Experimental Results

In this section, we evaluate the performance and security of our proposed residue number system architecture in comparison to traditional binary arithmetic networks.

4.1 Experimental Setup

In this article, we experimentally performed addition and multiplication operations in the first convolutional layer of the LeNet-5 neural network, using the residue number system (RNS) with the well-known pairwise coprime moduli set $\{2^n+1,\ 2^n,\ 2^n-1\}$.

In our design, we set $n = 10$; therefore, the chosen moduli set is $\{1023, 1024, 1025\}$, with bit-widths of 10, 10, and 11 bits. The configuration provides a dynamic range of approximately 2^{30}, which not only covers the 16-bit input/output space but also encompasses the maximum multiplication results in the MAC layer. We identified the maximum value generated during computation and accordingly determined the number of modulus bit-widths.

These operations are executed in parallel across

three channels, based on the chosen number of moduli. We observed that distributing power consumption during addition and multiplication across these channels results in reduced power leakage compared to conventional binary arithmetic. Following the computations, reverse conversion is performed to reconstruct 16-bit binary outputs, which are then passed to subsequent layers. This architecture enables parallel execution across the residue channels, effectively reducing the correlation between data and power consumption, thereby enhancing resistance to correlation power analysis (CPA) attacks.

4.2 Security Improvement

The binary implementation of LeNet-5 was highly vulnerable to CPA attacks, enabling successful key recovery using only 20,000 input samples. A noticeable correlation peak was observed at a weight of 12, 276, highlighting a strong link between power consumption and sensitive data.
In contrast, the RNS-enabled network required 40, 000 input samples for a successful CPA attack, compared with only 20, 000 for the binary representation. This is twice the number needed in the binary case. Although correlation peaks were still observed at the weight of 12, 276, their magnitude was significantly reduced. This reduction demonstrates improved resistance due to statistical obfuscation in the power consumption traces.

When examining the network before the reverse conversion, each residue channel was attacked individually. The findings revealed that:

- For the simplest modulus (2^n), a successful attack required 60, 000 input samples, indicating a threefold improvement over the binary configuration.
- For the two more complex moduli ($2^n + 1$ and $2^n - 1$), no successful attacks were observed within the tested sample range.

These findings confirm that using an RNS system significantly improves side-channel resistance, particularly before reverse conversion. In this stage, the independent processing of residue channels effectively disrupts power correlation patterns.

To further confirm the improvement in resistance, t-tests were conducted with the following results:

- In the binary network, the maximum and minimum power variations differed by 7.2 units.
- In the RNS-based network (after reverse conversion), this difference decreased to 6.6 units.

This reduction in power leakage shows that the RNS system effectively minimises information exposure through power traces, making CPA attacks less effective.

4.3 Visual Data Analysis

In this section, the correlation results are presented in the form of a chart. Figure 6 a shows the correlation plot for the binary LeNet-5 configuration using 20, 000 power traces. A distinct correlation peak is observed for weight index 12, 276, indicating a successful CPA attack.
Figure 6-b illustrates the CPA attack in partially-RNS LeNet-5 implementation before the reverse conversion for modulo 2^n. Among the three modulus configurations used, the modulo 2^n configuration, which is the simplest, exhibits a visible peak at index 12, 276 with 60, 000 traces. The modulo 2^{n+1} and 2^{n-1} configurations show no discernible peaks even at 150, 000 traces, indicating failed CPA attempts.

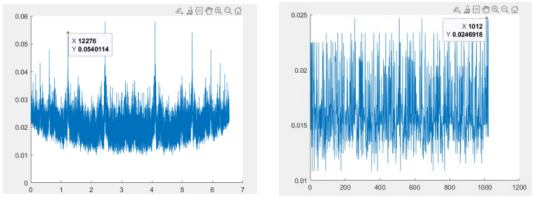

Figure 6. (a) CPA attack on the RNS-DNN after reverse conversion; (b) CPA attack on the RNS-DNN before reverse conversion for modulo 2^n

To complement the correlation power analysis (CPA), we employ Welch's t-test as a non-profiled leakage detection metric. This method determines whether statistically significant leakage exists by evaluating the differences between two groups of power traces separated by a secret-dependent condition (e.g., a specific bit of a weight value). If the absolute t-value exceeds the threshold of 4.5 at any sample point, it suggests potential leakage exploitable by an adversary.
We collected power traces for 20,000 input samples across both the standard binary and the full RNS-based LeNet-5 architectures. Each power trace consists of 20,000 sampled values. The traces were partitioned into two groups based on a chosen bit of the weight value. For each sample point, we computed the mean and standard deviation of the groups and derived the t-value using the Welch t-test formula:

$$T = \frac{\mu 1 - \mu 2}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{1}$$

The binary implementation exhibits multiple points exceeding the $|t|= 4.5$ threshold, confirming statistically significant leakage. In contrast, the RNS-based

implementation maintains t-values well below the threshold, indicating robust resistance to non-profiled side-channel analysis.

4.4 Overheads

The hardware cost of integrating RNS into a DNN is evaluated. The results are shown in Table 1.

Table 1. Hardware cost of implemented LeNet-5 in binary and RNS

Metric	Binary	Before RC	After RC
LUTs	12540	17890	19200
FFs	8230	10440	11050
DSPs	20	24	24

As shown in Table 1, the RNS implementations result in an area overhead of approximately 35%-50%. However, this trade-off is justified by a threefold improvement in resistance to side-channel attacks. For a fair comparison, both designs were synthesized under identical clock constraints. While the conversion process introduces slight latency, the operating frequency remains nearly unchanged. Furthermore, throughput is preserved by the parallel RNS channels, and energy consumption remains comparable to the baseline. Despite changes in arithmetic representation and data encoding, the MNIST classification accuracy is identical to the baseline.

5 Conclusion

In this paper, we implemented a defence mechanism against side-channel attacks using residue number systems (RNS) for LeNet-5 neural networks. Our implementation was evaluated in two configurations: fully RNS-based configurations and partially RNS-based configurations. In both cases, the attack results were assessed both before and after the RC (reverse conversion) process. The neural network evaluated was LeNet-5. The evaluation results obtained using the CPA (correlation power analysis) method showed that using the RNS made the attacks significantly harder without affecting the classification results of these networks. Furthermore, the results showed that attacking the network before the RC process was three times more difficult. The results also show that the choice of moduli significantly affects attack difficulty. Specifically, attacking the output of the $2^n + 1$ modulus was more challenging than attacking the outputs of the other two moduli. Although the implementation introduced some hardware overhead, the use of RNS remains advantageous due to the parallelisation and carry-free operations among the moduli.

References

[1] V. Sze, Y.-H. Chen, T.-J. Yang, and J. S. Emer, Efficient processing of deep neural networks: a tutorial and survey, Proc. IEEE, vol. 105, no. 12, pp. 2295–2329, 2017.

[2] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders, and K. R. Müller, Explaining deep neural networks and beyond: a review of methods and applications, Proc. IEEE, vol. 109, no. 3, pp. 247–278, 2021.

[3] M. Isakov, V. Gadepally, K. M. Gettings, and M. A. Kinsy, Survey of attacks and defenses on edge-deployed neural networks, in Proc. IEEE HPEC, 2019, pp. 1–8.

[4] A. Michel, S. K. Gha, and R. Ewetz, A survey on the vulnerability of deep neural networks against adversarial attacks, Prog. Artif. Intell., 2022, pp. 1–11.

[5] S. Almutairi and A. Barnawi, Securing DNN for smart vehicles: an overview of adversarial attacks, defenses, and frameworks, J. Eng. Appl. Sci., vol. 70, pp. 1–29, 2023.

[6] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks, in Proc. USENIX Security Symp., 2019.

[7] F. Tramer, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, Stealing machine learning models via prediction APIs, in Proc. USENIX Security Symp., 2016.

[8] G. Apruzzese, A. S. Anderson, S. Dambra, D. Freeman, F. Pierazzi, and K. Roundy, Real attackers don't compute gradients: bridging the gap between adversarial ML research and practice, in Proc. IEEE SATML, 2023, pp. 339–364.

[9] S. Mendez Real, Physical side-channel attacks on embedded neural networks: a survey, Appl. Sci., 2021, pp. 310–316.

[10] H. Chabanne, J.-L. Danger, L. Guiga, and U. Kuhne, Side-channel attacks for architecture extraction of neural networks, CAAI Trans. Intell. Technol., vol. 6, no. 1, pp. 3–16, 2021.

[11] A. Dubey, R. Cammarota, V. Suresh, and A. Aysu, Guarding machine learning hardware against physical side-channel attacks, CoRR, vol. abs/2109.00187, 2021.

[12] K. Ganesan, M. Fishkin, O. Lin, and N. E. Jerger, Blackjack: secure machine learning on IoT devices through hardware-based shuffling, arXiv preprint, arXiv:2310.17804, 2023.

[13] X. Liu, L. Xie, Y. Wang, J. Zou, J. Xiong, Z. Ying, and A. V. Vasilakos, Privacy and security issues in deep learning: a survey, IEEE Access, vol. 9, pp. 4566–4593, 2021.

[14] J. Krautter and M. B. Tahoori, Neural networks as a side-channel countermeasure: challenges and opportunities, in Proc. IEEE ISVLSI, 2021, pp. 272–277.
[15] M. Brosch, M. Probst, and G. Sigl, Counteract side-channel analysis of neural networks by shuffling, in Proc. DATE, 2022.
[16] T. D. Cnudde, M. Ender, and A. Moradi, Hardware masking revisited, TCHES, 2018.
[17] M. M. Ahmadi, L. Alrahis, O. Sinanoglu, and M. Shafique, DNN-alias: deep neural network protection against side-channel attacks via layer balancing, arXiv preprint, arXiv:2303.06746, 2023.
[18] J. Li, Z. He, A. S. Rakin, D. Fan, and C. Chakrabarti, Neurobfuscator: a full-stack obfuscation tool to mitigate neural architecture stealing, in Proc. IEEE HOST, 2021.
[19] R. P. Mcevoy, C. C. Murphy, et al., Isolated WDDL: a hiding countermeasure for differential power analysis on FPGAs, TRETS, 2009.
[20] Z. He, A. S. Rakin, and D. Fan, Parametric noise injection: trainable randomness to improve deep neural network robustness against adversarial attack, in Proc. CVPR, 2019, pp. 588–597.
[21] T. Schneider, A. Moradi, and T. Guneysu, Arithmetic addition over boolean masking towards first- and second-order resistance in hardware, Horst Gortz Institute for IT Security, Germany, 2015.
[22] R. Selvam and A. Tyagi, Residue number system (RNS) and power distribution network topology-based mitigation of power side-channel attacks, Cryptography, vol. 8, no. 1, p. 1, 2023.
[23] A. T.-S. R. Selvam, An evaluation of power side-channel resistance for RNS secure logic, Sensors, vol. 22, no. 1, 2022.
[24] A. Dubey, A. Ahmad, M. A. Pasha, R. Cammarota, and A. Aysu, Modulonet: neural networks meet modular arithmetic for efficient hardware masking, IACR TCHES, 2022, pp. 506–556.
[25] A. Dubey, R. Cammarota, and A. Aysu, MaskedNet: the first hardware inference engine aiming power side-channel protection, in Proc. IEEE HOST, 2020.
[26] P. C. Kocher, J. Jaffe, and B. Jun, Differential power analysis: leaking secrets, in Proc. Crypto, 1999, pp. 388–397.
[27] M. Randolph and W. Diehl, Power side-channel attack analysis: a review of 20 years of study for the layman, Cryptography, vol. 4, no. 2, p. 15, 2020.
[28] K. Gandolfi, C. Mourtel, and F. Olivier, Electromagnetic analysis: concrete results, in Proc. CHES, Springer, 2001, pp. 251–261.
[29] S. Mangard, E. Oswald, and T. Popp, Power analysis attacks: revealing the secrets of smart cards, Springer, 2008.
[30] J. Mishra and S. K. Sahay, Modern hardware security: a review of attacks and countermeasures, arXiv preprint, arXiv:2501.04394, 2025.
[31] S. Potluri and F. Koushanfar, SoK: model reverse engineering threats for neural network hardware, Cryptology ePrint Archive, Paper 2024/913.
[32] A. Yuan, G. Bai, L. Jiao, and Y. Liu, Offline handwritten English character recognition based on convolutional neural network, in Proc. IEEE DAS, 2012, pp. 125–129.
[33] Y. Xiang, Z. Chen, Z. Fang, H. Hao, J. Chen, Y. Liu, Z. Wu, Q. Xuan, and X. Yang, Open DNN box by power side-channel attack, IEEE Trans. Circuits Syst. II, 2020.
[34] A. A. Oke, B. A. Nathaniel, B. F. Bukola, and O. A. Ayopo, Residue number system based applications: a literature review, 2021.
[35] J. Peng, Y. Alkabani, S. Sun, V. J. Sorger, and T. El-Ghazawi, DNNARA: a deep neural network accelerator using residue arithmetic and integrated photonics, in Proc. ICPP, 2020.
[36] A. Babatunde, B. F. Balogun, A. O. Lawal, et al., Application of residue number system in information security: a systematic review, SLU J. Sci. Technol., vol. 11, pp. 195–219, 2025.
[37] Q. Xu, M. T. Arafin, and G. Qu, Security of neural networks from hardware perspective: a survey and beyond, in Proc. IEEE ASP-DAC, 2021, pp. 449–454.
[38] N. Singh, An overview of residue number system, in Proc. Nat. Seminar Devices, Circuits and Communication, Nov. 2008.

Marzieh Morshedzadeh Marzieh Morshedzadeh received her B.S. and M.S. degrees in Computer Engineering and Computer Architecture from Sharif University of Technology and Shahid Beheshti University in 2009 and 2011, respectively. She is currently a Ph.D. student at Shahid Beheshti University. Her research is focused on physical security aspects of deep neural networks.

Reyhaneh Hajimohammadali Reyhaneh Hajimohammadali received her B.S. and M.S. degrees in Computer Engineering and Computer Architecture from Azad University of Mashhad and Shahid Beheshti University in 2017 and 2024, respectively. Her current research interests include the hardware security of deep neural networks.

Ali Jahanian Ali Jahanian received his B.Sc. degree in Computer Engineering from the University of Tehran, Tehran, Iran, in 1996, and his M.Sc. and Ph.D. degrees in Computer Engineering from Amirkabir University of Technology, Tehran, Iran, in 1998 and 2008, respectively. He joined Shahid Beheshti University, Tehran, Iran, in 2008. His current research interests focus on hardware security and biochip design.