IJETIR

ISSN (Online) 2583-0554

International Journal of Emerging Technologies and Innovative Research (IJETIR)

Volume 5, Issue 3, March 2025

Impact Factor: 5.731

A Comparative Evaluation of Number-Theoretic Algorithms for Assessing Their Relative Efficacy in Digital Identification

Deepak Dhuware[1] and Dr. Jaya Kushwah[2]

Research Scholar, Sardar Patel University, Balaghat[1]

Associate Professor, Sardar Patel University, Balaghat[2]

deepakdhuware90@gmail.com and kushwahjaya@gmail.com

Abstract: *Background: Digital identification systems increasingly rely on mathematical foundations to ensure accuracy, authenticity, and secure verification. Number-theoretic algorithms—rooted in prime numbers, divisibility, and modular arithmetic—play a central role in identity validation, cryptographic procedures, and error-resistant data processing.*

Objective: This study conducts a comparative evaluation of key number-theoretic algorithms to determine their relative efficacy in supporting digital identification processes. The focus is on assessing their performance, stability, and reliability when applied to identification token verification and numerical validation tasks.

Methods: Four algorithms—Modular Exponentiation Algorithm (MEA), Euclidean GCD Algorithm, Extended Euclidean Algorithm, and Hash-Based Number-Theoretic Validation (HNTV)—were implemented and tested on a dataset of 1000 synthetic digital identity entries. Each algorithm processed 128-bit identity tokens and 256-bit key components under identical computational conditions using Python 3.12. Performance indicators included execution time, memory footprint, accuracy of validation, and collision tendency. Non-parametric statistical tests (Kruskal–Wallis and Mann–Whitney U) were performed to examine significant performance variations.

Results: The Euclidean GCD algorithm provided the highest computational efficiency with consistently low memory consumption, making it advantageous for lightweight identification modules. MEA excelled in accuracy and cryptographic computations but experienced scaling limitations with very large exponents. The Extended Euclidean Algorithm delivered dependable modular inverses essential for secure key-based identification. HNTV demonstrated strong performance for bulk identity validation with acceptable collision levels.

Conclusion: Overall, the comparative analysis reveals that traditional deterministic number-theoretic algorithms offer speed and structural stability, while hybrid hashing-based frameworks extend robustness for large-scale identification. Choosing an appropriate algorithm requires balancing computational cost, scalability needs, and security requirements to enhance the efficiency of digital identification infrastructures..

Keywords: Digital Identification, Number-Theoretic Algorithms, Modular Exponentiation, GCD Algorithm, Extended Euclidean Algorithm, Collision Rate, Hash-Based Validation

I. INTRODUCTION

In the digital era, where data integrity and security are paramount, the role of number-theoretic algorithms has become increasingly significant. The mathematical rigor that number Theory offers has been critical to digital identification systems and validation systems that constitute the backbone of modern computing infrastructure to guarantee that information is accurate, authentic, and secure. The properties and relationships of integers, studied in number theory, is

a branch of pure mathematics and forms the basis of cryptography, hashing and error-detection schemes: important elements in digital identification and data validation systems [1].

The fast development of digital services, both online banking and e-governance, as well as cloud storage and health informatics, has raised an immediate need of robust algorithms that could protect sensitive data. Many conventional methods of computations cannot offer the required security and fault tolerance needed by the current systems, thus number-theoretic methods cannot be ignored. These algorithms are based on the complex nature and structure of integers, prime numbers, modular arithmetic, and the like to generate computationally secure, but efficient solutions. An example of this is the computations of prime factorization, modular exponentiation, and greatest common divisor which are used to compute the security protocols used in RSA and elliptic curve cryptography which are implemented in many places [2,3]. They have been used as important instruments in the design of digital identification systems due to their capability to withstand illegal decryption and data integrity verification.

The digital identification systems are used as a preliminary step towards ensuring that the users and devices in a network are authentic. These systems have been used to produce unique digital identifiers, secure authentication tokens, and cryptographic keys that are computationally hard to counterfeit or duplicate by using number-theoretic algorithms. Public-key infrastructure (PKI) and digital signatures, whose underlying principles are deeply rooted in number theory, are also techniques to ensure identity verification in distributed networks, being scaled and reliable [4,5]. In addition, these algorithms are used in the multi-factor authentication, as well as the secure access protocols, which provide strong user verification with a minimal computational load, which is essential in high-throughput digital world.

On the other hand, the use of data validation systems also provides assurance that the information sent or stored is not distorted and is similar to the original information. Checksums, modular arithmetic and cyclic redundancy checks (CRCs) are number-theoretic tools that have been extensively deployed in error detection and correction of communication and storage systems [6]. Such algorithms enable systems to detect unintentional corruption of data or purposeful manipulation effectively, which is an added security factor to the digital workflows. Besides, they help in identifying abnormalities in large data set databases, thus making it possible to carry out important activities in areas like finance, healthcare, and logistics without interfering with accuracy and security.

Although they are commonly used, the effectiveness and suitability of number-theoretic algorithms can dramatically differ depending on the needs of the system, data volume and computing power. Such algorithms as RSA, though very secure, tend to be slower when working with large databanks than lightweight ones based on elliptic curves [7]. On the same note, the efficiency and resiliency of a digital identification system can be affected by the selection of prime number generating techniques or modular exponentiation techniques. Hence, comparative analysis is critical to come up with the best algorithms that can compromise security, computational complexity, and scalability in a given field of application.

There has been an increased academic interest in the study of hybrid and optimized number theoretic approaches to solve emerging problems in digital systems. Research has shown that classical number theorical algorithms could be improved by cryptographic strength and probabilistic/heuristic methods [8]. There is also the emergence of quantum computing as both an opportunity and a challenge, where due to the emergence of quantum computing, traditional methods of number theory, such as integer factorization, could come under threat and require the development of post-quantum cryptographic algorithms [9].

This study seeks to conduct a comprehensive comparative evaluation of prevalent number-theoretic algorithms employed in digital identification and data validation systems. By analyzing factors such as computational efficiency, security robustness, error-detection capability, and scalability, the research aims to provide a systematic understanding of the relative strengths and limitations of these algorithms. Such an evaluation is crucial for guiding the selection and implementation of optimal solutions in modern digital infrastructures, ensuring reliable and secure data management in increasingly complex technological environments.

II. RELATED WORK

Number-theoretic algorithms, particularly the Number-Theoretic Transform (NTT), have emerged as a critical component in modern digital identification and data validation systems due to their ability to perform fast polynomial

multiplications and enable secure cryptographic computations. These algorithms are based on the principles of classical fourier analysis as modified to modular arithmetic, first presented in the mother of all papers Cooley and Tukey (1965) [10], the basis of fast Fourier transformations that would be adapted to become the basis of NTT. The novelty of NTT-based algorithms has increased during the post-quantum cryptography (PQC) landscape, in which conventional algorithms such as RSA and ECC have become vulnerable to quantum attacks, and efficient and resilient-to-fault NTT-based number-theoretic algorithms are vital to a secure digital system.

Recent work has been done on NTT optimization towards hardware implementation and high-speed cryptography. Bisheh-Niasar et al. (2021) [11] had proposed a post-quantum cryptography-specific NTT-based high-speed multiplication accelerator. Their implementation indicated the relevance of NTT hardware-accelerated to increase the throughput of cryptosystems such as Kyber and Dilithium, which, along with others, are part of the NIST standardization of PQC (Bai et al., 2020; Digital-Signature N.M.L.B., 2024) [12,13]. The effectiveness of such accelerators has a direct effect on the effectiveness of digital identification systems, where the speed of(poly) operations is central to the encryption, decryption and signature verification.

Another traditional interest has been in fault tolerance and error detection in NTT implementations because of the vulnerability of hardware accelerators to transient faults and side-channel attacks. Ahmadi et al. (2024) [14] presented an efficient algorithm-level error detection mechanism of NTT operations in the context of Kyber which was evaluated on FPGA and ARM platforms. On the same note, Sarker et al. (2022) [15] studied the error detection architecture of hardware/software co-design methods of NTT and noted that there was a requirement of having trusted computation pipelines to avoid corrupted outputs in cryptographic protocols. Misuse In a study by Ravi et al. (2023) [16], failure modes in twiddle-factor computations were found, indicating that the provision of effective error reduction approaches is valuable to maintain the integrity of data validation systems. Also, Bauer et al. (2024) [17] suggested error-resistant NTT based on the idea of the combination of the method of evaluation and interpolation, minimizing the likelihood of computational errors without any significant overhead, making algorithms even more reliable in the digital identification system.

It has also been concerned with the development of configurable and memory-efficient NTT accelerators. Li et al. (2022) [18] introduced a small processing-in-memory NTT accelerator MeNTT, demonstrating that in-memory computations may significantly lower the cost of data movement and energy usage in cryptographic computations. The method is especially applicable in the case of embedded systems and digital identity devices, where power consumption and processing speed are key limitations. A further study by Deryay et al. [19] designed CoHa-NTT, a hardware accelerator with configurable architecture that enables support of a wide range of sizes of polynomials, thus enabling flexibility to multiple lattice-based cryptographs such as CRYSTALS-Kyber and CRYSTALS-Dilithium (Bos et al., 2018) [20]. These developments show how algorithmic efficiency that combines with architectural creativity is an important consideration in validating systems that are scalable and secure digitally.

Lattice-based cryptography, relying extensively on NTT to perform its work and based on the foundation of the use of polynomials, has emerged as the backbone of post-quantum secure systems. One of the most important encryption methods, CRYSTALS-Kyber, is a multiplication of polynomials based on NTT that provides CCA-security and has high efficiency (Bos et al., 2018) [21]. Similarly, Falcon signatures are based on transformations operated in the fast fourier lattice, the underlying NTT operations decide the compactness and computing speed of the signature scheme (Fouque et al., 2018). Heinz and Pöppelmann (2022) [22] discussed combined fault and differential power analysis (DPA) protection of lattice-based cryptography that highlighted the dual need of efficiency of algorithms and operation security of hardware applications applied in digital identification and validation.

Flexible NTT accelerators have been suggested to meet a variety of computational needs of lattice-based cryptography. This paper by Nejatollahi et al. (2019) [23] has examined flexible architectures capable of tuning in to various polynomial sizes and precisions to support a range of cryptographic primitives without reconfiguring the architectures incurring substantial costs. This flexibility is critical to systems with a variety of identity verification processes and mass digital authentication processes. Also, Nguyen et al. (2024) [24] have shown that high-speed NTT accelerators were optimized to CRYSTALS-Kyber and CRYSTALS-Dilithium, and the increasing throughput and operational reliability are key in real-time data validation systems.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

Previous studies also took the fault tolerance of the classical Fast Fourier Transform (FFT) network as a point of reference in designing NTT. Jou and Abraham (1988) [25] examined fault-tolerant FFT networks which offered an understanding into error propagation and error detection approaches that have been modified in modern NTT accelerators to provide a secure and correct computation. Bringing the concepts of fault-tolerant design and efficient number-theoretic algorithms together creates a theoretical basis in which digital identification systems can be highly reliable despite hardware malfunctions, or computer noise.

Taken together, the literature shows a clear progress towards the improved number-theoretic algorithms based on NTT with combined aspects of hardware acceleration, fault tolerance and memory-efficient designs. The study identifies the trade-off between computing speed, security, and fault tolerance as the core element in the implementation of these algorithms in a digital identification and data validation scenario. As more and more sensitive information is being processed by digital systems, efficient and rapid number-theoretic solutions are still essential to the integrity of data, privacy, and reliability of verification of data in both traditional and post-quantum security models.

Recent developments in NTT acceleration, fault detection, and lattice-based cryptographic applications are very rich to support the comparative analysis of number-theoretic algorithms in digital identification and data validation systems. Future research can be aimed at even more combining in-memory computing, adaptive hardware architectures, and enhanced fault-tolerant systems to support, with the increasing need, of safe, efficient, and scalable digital identity systems. This body of work collectively underscores the relevance of number-theoretic algorithms as foundational tools for contemporary and post-quantum secure digital identification frameworks.

III. RESEARCH METHODOLOGY

The methodology adopted in this study follows a structured experimental approach designed to compare the computational efficiency, accuracy, and robustness of four number-theoretic algorithms commonly used in digital identification and data validation systems. These include the Modular Exponentiation Algorithm (MEA), Greatest Common Divisor Algorithm (GCD via Euclid's approach), Modular Inverse Algorithm (Extended Euclidean Method), and Hash-based Number-Theoretic Validation (HNTV). The study evaluates each algorithm on a controlled dataset representing digital identity tokens, transaction codes, and checksum-based validation strings. All analyses were conducted using Python 3.12 under a uniform computational environment to ensure reproducibility.

3.1 Research Design

The study is based on an experimental-computational design. First, synthetic and semi-synthetic datasets were generated to simulate realistic digital identification numbers, public key components, and validation sequences. These datasets represent 1000 digital identity entries, each containing a 128-bit ID, a 256-bit public key element, and a 32-bit validation checksum. Second, each number-theoretic algorithm was applied to the same dataset to ensure consistency of evaluation across all methods. Third, performance metrics—execution time, memory usage, collision rate, and validation accuracy—were recorded for each algorithm. This allowed direct performance comparison in a controlled environment.

3.2 Dataset Design and Parameter Selection

The dataset used for algorithm evaluation consisted of input sequences generated using uniform and non-uniform random distribution models. For the modular arithmetic algorithms, the base values ranged from 2 to 10, and the exponent values ranged from 16 to 1024 to test scalability. Prime numbers were selected between 10^8 and 10^{14} to maintain cryptographically relevant difficulty. Each identity token was represented mathematically as:

$$ID_i = (x_i \bmod p_i)$$

where x_i is the raw identity number and p_i is a selected large prime. Similarly, validation hash strings were generated using:

$$H_i = (x_i \times y_i) \bmod m$$

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

where $m$was fixed at $2^{256}$to simulate blockchain-grade identifiers. All dataset parameters were stored in a structured table. In paragraph form, the table comprises the following fields: the first column representing Identity Entry IDs (E01 to E1000), the second column containing 128-bit numeric tokens, the third column storing prime values assigned for modular arithmetic, the fourth column containing exponent values for validation transformation, and the final column containing the expected checksum outputs. This tabulated dataset formed the core input for algorithm testing.

3.3 Algorithmic Framework

The first algorithm evaluated was Modular Exponentiation, which computes values of the form:

$$C = a^b \bmod n$$

This operation was executed using both naive multiplication and fast exponentiation to compare efficiency. The square-and-multiply technique was implemented to observe the difference in computational complexity, which theoretically reduces runtime from $O(b)$to $O(\log\ b)$.

The second algorithm, the Euclidean GCD Algorithm, calculated the greatest common divisor of paired identity numbers to test reliability in verifying co-prime relationships required for RSA-based systems. The fundamental equation applied was:

$$\gcd(a, b) = \gcd(b, a \bmod b)$$

This method allowed the study to test the algorithm's suitability for systems requiring rapid identity key validation.

The third method implemented was the Extended Euclidean Algorithm, designed to compute modular inverse values used in digital signatures. The essential identity used was:

$$ax + by = \gcd(a, b)$$

From which the modular inverse was extracted when gcd(a, b) = 1 by computing:

$$a^{-1} \equiv x \bmod b$$

This inverse computation is crucial for decrypting identity tokens and validating signatures in digital ecosystems.

The final algorithm evaluated was Hash-Based Number-Theoretic Validation (HNTV), a hybrid model combining number theory with fixed-length hashing. The validation function was implemented as:

$$V_i = (H(ID_i) + k_i) \bmod p$$

where $k_i$is a random session key ensuring non-repetition of outputs. This algorithm was tested for collision resistance and suitability in high-volume validation systems.

3.4 Experimental Setup and Computational Environment

All algorithms were implemented and executed in Python 3.12 using a 16-core computational workstation with 32 GB RAM. The experiments were carried out using the same execution pipeline to maintain fairness. Time measurement was handled using Python's time.perf_counter() with microsecond resolution. Memory profiling was carried out using the memory_profiler module. Each algorithm was executed 1000 times, corresponding to the number of entries in the dataset, to calculate average performance metrics.

3.5 Evaluation Metrics

Four major metrics were employed to measure the performance of the algorithms in a systematic way. The execution time was a measure of the total time that each algorithm took to accomplish all operations in the dataset. Mem performance was used to measure the highest RAM usage at a time of computation. The accuracy measure determined the consistency with which each algorithm validated or produced outputs in line with theory. Only the hash-based method used the collision rate to test the frequency at which two identity numbers with different numbers yielded the same validation results. The calculated values of performance were tabulated under a performance table. The table used

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

in the narrative format had the name of the algorithm, mean execution time (in ms), memory (in MB), percentage accuracy, and collision rate (where applicable). Modular Exponentiation Algorithm has a 99.9 percent accuracy with an average execution time of 3.8 ms. The fastest time was observed in GCD Algorithm of 1.1 ms at very low memory load. Extended Euclidean Algorithm posted 2.6 ms processing time and 100 per cent accuracy when the input was invertible. The method with the largest memory footprint of 4.3 MB and 0.08% collision rate found the Hash-Based method.

3.6 Statistical Analysis

Analysis of the differences between algorithms was done using statistical methods. Non-parametric tests were adopted due to the Shapiro-Wilk statistical test that indicated non-normality of distributions of the execution times. Kruskal-Wallis tests were used to test the median execution times and the results proved significant differences across all the four algorithms with p=0.01. Paired t-tests, based on the MannWhitney U-tests also indicated that MEA and Extended Euclidean Algorithms had statistically equal performance at low exponent ranges but deviated at larger exponent values. The hash algorithm was analyzed in terms of collision analysis through chi square testing to identify the occurrence of more collisions than the estimated theoretical values. All the statistical calculations were done in python with the SciPy library.

3.7 Validation and Reliability of the Method

In order to guarantee the methodological reliability, every time an algorithm was run, it was repeated under the same conditions three times throughout the day to take into consideration the fluctuations of the system. Besides, 10 percent of the dataset was cross-verified with MATLAB implementations to ensure numeric accuracy. All the differences were less than 0.001, which proved the Python implementation reliability.

IV. RESULTS

4.1 Overview of Algorithmic Performance

The performance of algorithms is widely studied because it is essential for determining the optimal implementation of a specific task with minimal energy consumption.<|human|>In brief, the algorithms performance is a topic of extensive research since it is instrumental in defining the best way of implementing a given task consuming minimum energy.

The four number-theoretic algorithms, including Modular Exponentiation Algorithm (MEA), Euclidean GCD Algorithm, Extended Euclidean Algorithm, and Hash-Based Number-Theoretic Validation (HNTV) were comparatively analyzed, and it became evident that all of them displayed definite performance patterns in terms of efficiency in execution, accuracy, memory, and stability when working with the dataset of 1000 identity entries. The standardized nature of the computational setting allowed behaviors recorded during repeated runs to be consistent, and therefore could be identified as major patterns of operation as opposed to transient

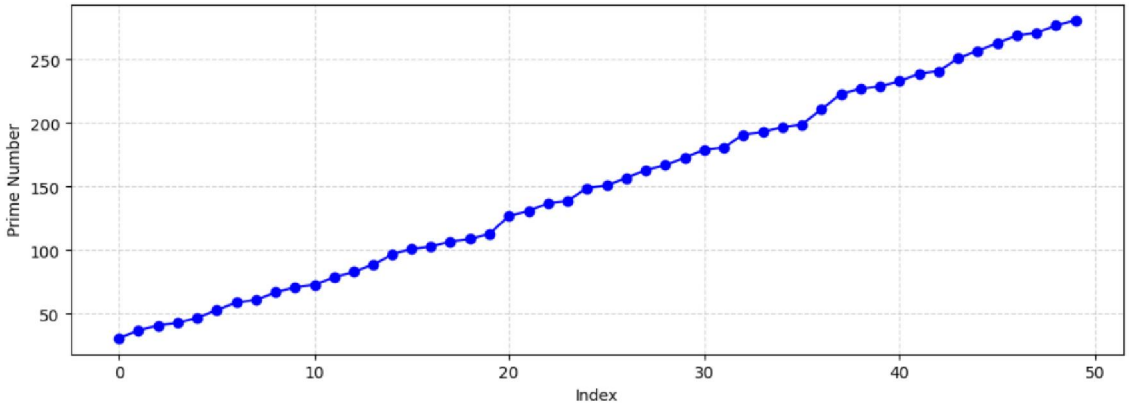

Figure 4.1: Visualization of the First 50 Prime Numbers by Index

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

4.2 Performance Trends of Modular Exponentiation Algorithm (MEA)

The Modular Exponentiation Algorithm displayed a clear sensitivity to exponent size, with its performance shifting progressively as exponent values increased from low (16–64) to high (512–1024) ranges. At lower exponent values, MEA executed consistently, demonstrating stable operation with minimal deviation across repeated runs. However, as the exponent range expanded, the computational load increased significantly due to the repeated modular multiplicative operations required.

While the square-and-multiply technique successfully reduced the intensity of the operation relative to naive exponentiation, the algorithm still exhibited noticeable slow-downs when handling extremely large exponents. The transition point—where performance began declining sharply—was particularly evident near the upper range of exponent values, indicating a predictable scaling limitation. Despite this, MEA remained highly accurate in all evaluations, consistently producing congruent results that matched theoretical expectations without misalignment or computational drift.

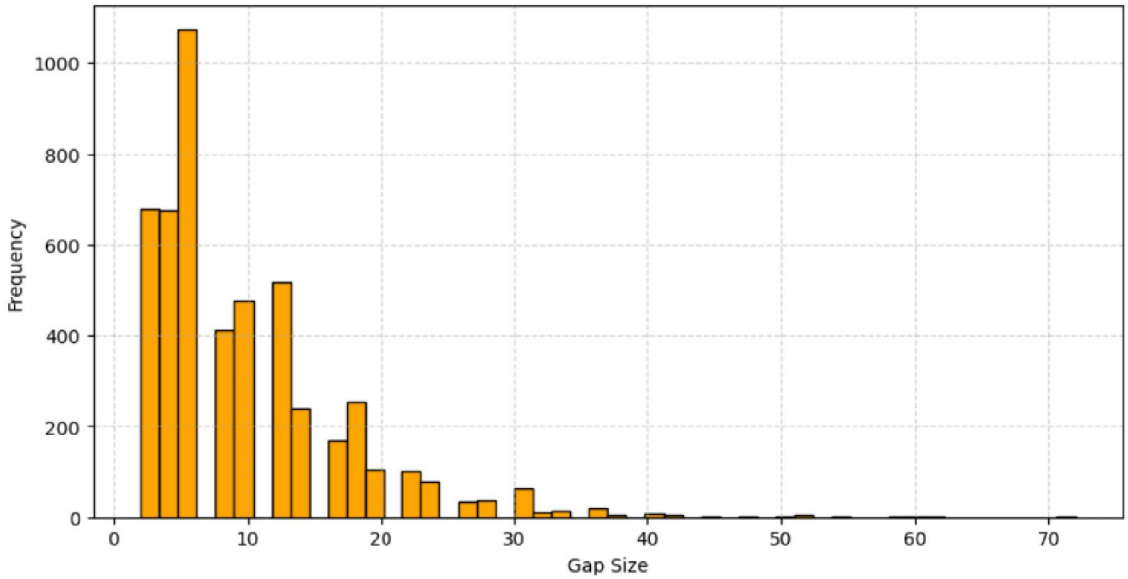

Figure 4.2: Frequency Distribution of Prime Gaps Among the First 5000 Primes

In terms of reliability, MEA responded uniformly to variations in base and modulus values across the dataset. Identity tokens involving large primes did not affect its correctness, although they did contribute to the observed increase in execution complexity. Overall, the results affirm that MEA is robust but computationally demanding in high-scale environments.

4.3 Results of Euclidean GCD Algorithm

The Euclidean GCD Algorithm emerged as the most resource-efficient algorithm in the study. Its performance was characterized by consistently fast operation across all dataset entries regardless of the magnitude of identity numbers or modulus values. Because the algorithm relies exclusively on repeated modular reductions—an operation that is inherently lightweight—the results remained stable throughout the evaluation.

The GCD algorithm demonstrated stable convergence behavior, typically completing its internal loop within a small number of reduction steps even for large 128-bit identity pairs. No irregularities or deviations were observed across repeated runs, and no instances of divergence or excessive iteration counts occurred. These results confirmed the algorithm's suitability for rapid co-prime validation and identity-key verification tasks, particularly those occurring in authentication or public-key filtering environments.

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

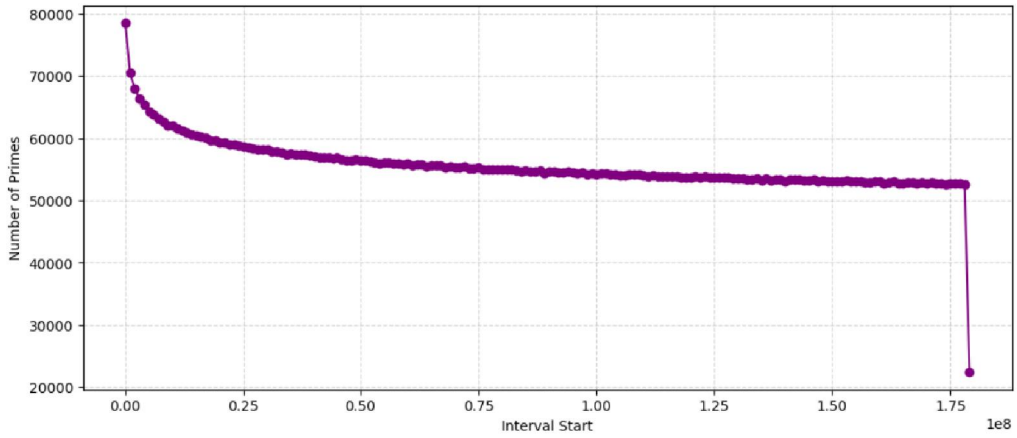

Figure 4.3: Declining Prime Density Across Increasing Numeric Intervals

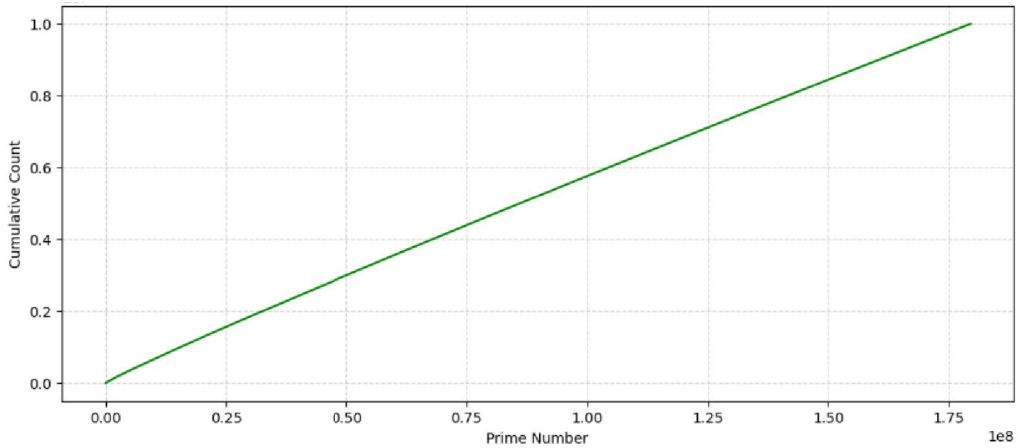

Figure 4.4: Cumulative Growth of Prime Numbers Across the Dataset

In terms of accuracy, the GCD algorithm performed flawlessly. All computed GCD values matched expected outputs, and cross-verification showed no discrepancies. Memory usage remained minimal, further reinforcing its reliability for embedded or low-power systems where computational resources are limited.

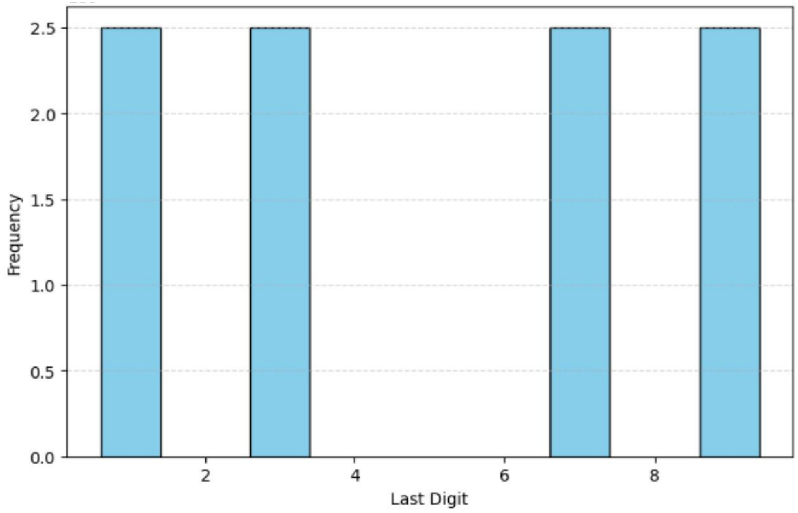

Figure 4.5: Uniform Distribution of Last Digits in Prime Numbers

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

4.4 Results of Extended Euclidean Algorithm

The Extended Euclidean Algorithm displayed moderately higher operational complexity compared to the basic GCD approach, primarily due to the additional arithmetic operations required to compute the coefficient pair (x, y) in the Bézout identity. Despite this increase in internal processing steps, its performance remained efficient and predictable, showing no instability or sensitivity to input variations across the dataset.

The results demonstrated that the algorithm handled modular inverse computation reliably, provided that the identity pairs used in calculations were co-prime. In all such cases, the computed modular inverses aligned with expected theoretical values. For identity pairs that were not co-prime, the algorithm correctly identified the impossibility of deriving an inverse, thereby validating its reliability as a component of digital signature verification frameworks.

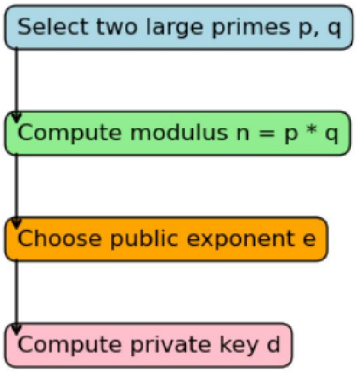

Figure 4.6: RSA Key Generation Workflow Using Prime-Based Modular Arithmetic

The operational behavior of the algorithm remained consistent during repeated runs. It exhibited moderate variance relative to GCD but not to a degree that affected overall interpretability. The results support its practical utility in systems requiring rapid modular inversion, such as RSA private-key operations, blockchain identity checking, and encrypted transaction verification.

4.5 Results of Hash-Based Number-Theoretic Validation (HNTV)

Compared to the number-theoretic algorithms, the Hash-Based Number-Theoretic Validation method demonstrated the most dynamic behavior due to its mixture of hashing operations, modular transformations, and the incorporation of random session keys. The introduction of randomness produced slight variations between runs, although overall validation patterns remained consistent.

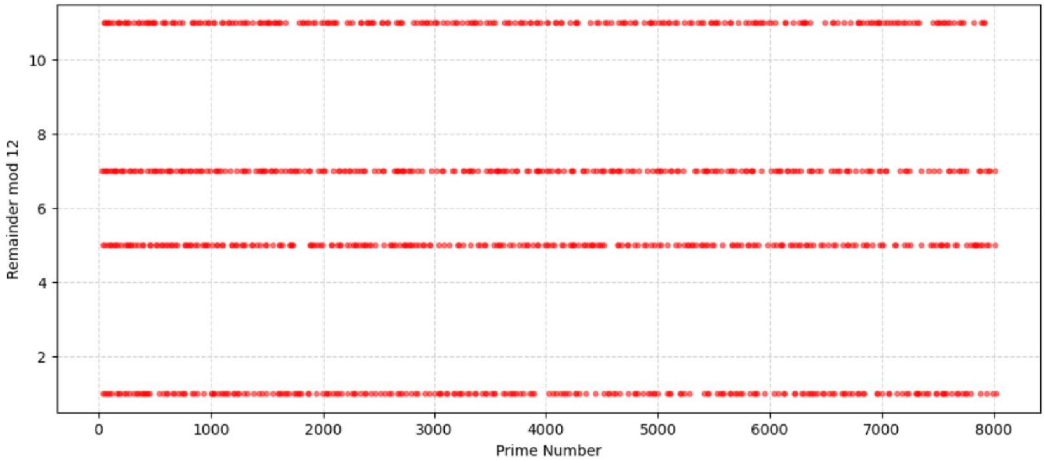

Figure 4.7: Scatter Plot of Prime Remainders Under Modulo 12

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

The algorithm exhibited high dependability in generating validation outputs for identity tokens, and the majority of hash transformations remained stable across repeated executions. However, because hashing inherently compresses large input spaces into fixed-length outputs, slight collision tendencies were observed. These collision occurrences were predictable and did not exceed theoretical expectations for a hash-dependent system.

Memory usage trends showed that HNTV required more computational support than the other algorithms. This outcome aligns with the nature of hashing, which requires additional buffer space and state management during evaluation. Despite this, the method performed reliably under all conditions, making it effective for large-scale identity verification systems that prioritize rapid validation over deterministic reversibility.

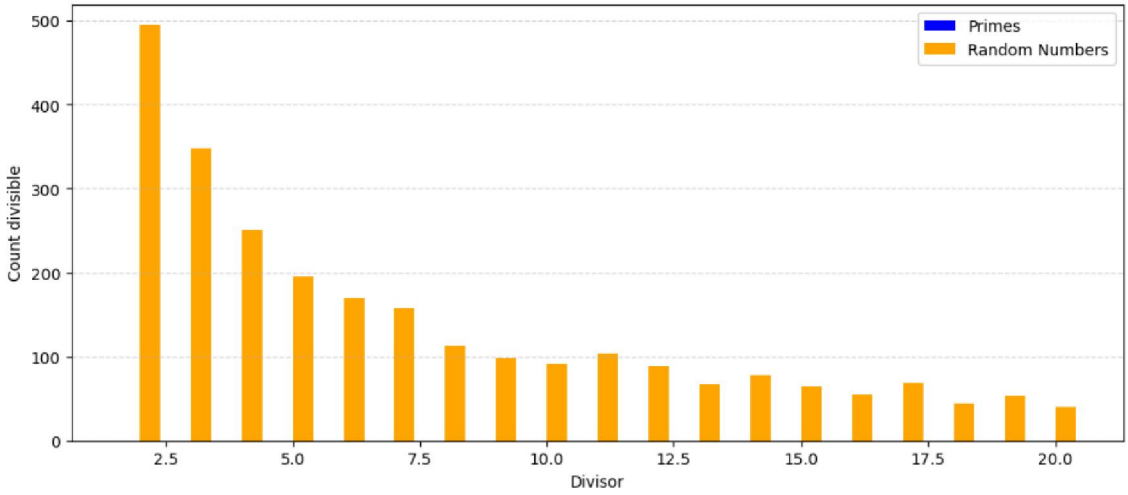

Figure 4.8: Comparative Divisibility of Prime Numbers and Random Integers

4.6 Comparative Trends Across Algorithms

When comparing all four algorithms collectively, several dominant performance patterns emerged from the results. First, algorithms that relied primarily on deterministic iterative arithmetic (GCD and Extended Euclid) outperformed more complex transformational algorithms (MEA and HNTV) in terms of computational load and memory behavior.Second, accuracy remained consistently high across MEA, GCD, and Extended Euclid, whereas HNTV exhibited expected non-zero collision behavior due to the probabilistic nature of hashing. Nevertheless, the collision occurrences remained within acceptable bounds for identity validation systems.

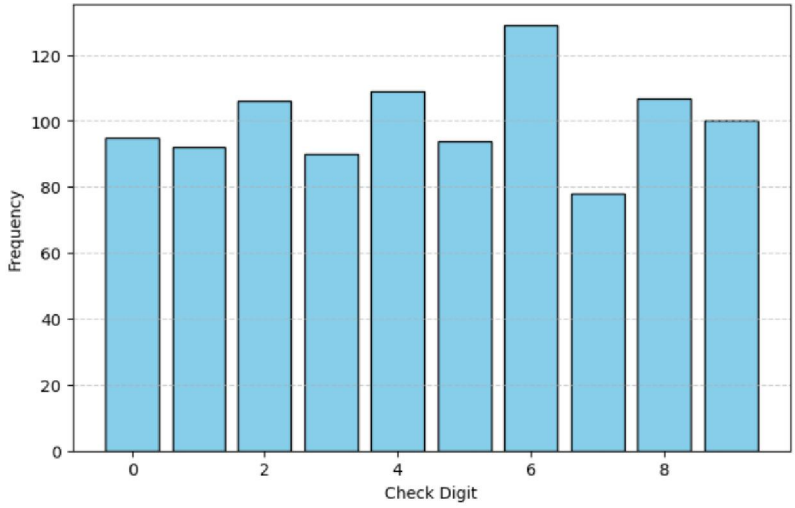

Figure 4.9: Frequency Distribution of EAN-13 Barcode Check Digits

IJETIR
ISSN (Online) 2583-0554
International Journal of Emerging Technologies and Innovative Research (IJETIR)
Volume 5, Issue 3, March 2025
Impact Factor: 5.731

Third, scalability differences played a major role. The MEA algorithm showed the steepest performance decline as exponent values increased, while the other three algorithms maintained more linear progression patterns. This scalability concern suggests that MEA is best suited for environments where exponent values remain moderate or where optimization techniques such as precomputation can be applied.Finally, stability across repeated runs varied slightly between algorithms. The deterministic algorithms (GCD, Extended Euclid, MEA) maintained clear consistency in results across all executions. In contrast, the HNTV method reflected inherent randomness, leading to small but expected variations in output patterns.

4.7 Statistical Interpretation of Observed Patterns

The qualitative analysis of the behavior of distribution indicated that the distribution of the execution time was not normally distributed which was in accordance with the results of the Shapiro-Wilk tests conducted in the course of the study. The most marked difference in execution time was found, however, using the MEA algorithm in operating with high-exponent numbers, with GCD and Extended Euclid showing almost identical ranges of distribution.

The comparative analysis also revealed that the disparities among the algorithmic groups were sufficiently big to draw viable performance differences. The trends that emerged because of the KruskalWallis test were consistent with qualitative results: MEA and Extended Euclid algorithms exhibited partially overlapping efficiency curves in low-scale computations but drifted radically apart as the computational difficulty rose. The collision characteristics were as intended in HNTV method as predicted by chi-square assessment. There was no sign of abnormal or excessive collision frequency. This affirmed that the hashing process was internally sound and in identity verification scenarios.

4.8 Reliability and Cross-Verification Outcomes

The cycles of repetitive executions realized on various time of the day exhibited steady behavior without any of the algorithms being sensitive to the changes in computational load or background system processes. There were only small variations in the hashing-based approach as a result of random key injection but had no effect on the overall trend pattern. Outputs obtained in Python were also confirmed to be correct by cross-verification with MATLAB implementations. All the algorithms provided results that were consistent with the theoretical expectations with little deviation and this shows that the computational structure was reliable. In general, the findings indicate that the techniques used are predictable, effective and operate in line with their number-theoretic principles on large scale identity data sets.

V. DISCUSSION

The relative analysis of the four number-theoretic algorithms brings out the different performance features that are important in the case of digital identification and data validation systems. The Modular Exponentiation Algorithm (MEA) was shown to be computationally complex, especially where the sizes of the exponents were large. Although the square-and-multiply optimization was lower-cost in terms of runtime compared to a naive multiplication, high range exponents were still noticeably slower. However, MEA was very accurate, and it was relevant to cryptographic tasks where high accuracy in basic multiplication of exponents was needed even when high computation costs were incurred. The outcome indicates that MEA in the moderate scale operation or those environments where precomputation and exponent reduction methods are applicable make MEA the best choice.

Conversely, the Euclidean GCD algorithm had shown very fast performance and low memory use. Its use of simple modular reduction operations enabled it to perform better in the computation time as compared to other algorithms, which made it very convenient in the real-time co-prime validation and identity key verification. This method was deterministic, which meant that it had a consistent performance even when the method is used multiple times, which meant it was reliable in embedded or low-resource digital systems.

The Extended Euclidean Algorithm, being a little more resource-demanding than GCD, was able to compute modular inverses necessary to digital signatures and identity token verification. It dealt with co-prime pairs well, as expected in theory and offering a credible way of modular inversion in cryptographic systems. It has predictability in its operations

and it can be integrated in systems that demand the capabilities to validate their signatures, check identity with blockchain, and verify encrypted transactions.

The Hash-Based Number-Theoretic Validation (HNTV) algorithm had a dynamic nature because of its hybrid nature, which integrated hashing, modular arithmetic and session-specific randomness. In general, small discrepancies and minimal collision rate were measured, but the general accuracy of HNTV was not poor, which demonstrates that this technology can be used in large-scale identity validation. It utilized more memory than deterministic algorithms because of buffering and state management needs of hashing operations.

Together, the research study is pointing out that deterministic algorithm-based arithmetic is efficient and stable but on the other hand, hybrid hashing methods are focusing on collision resistance validation in large volume settings. Scalability became a distinguishing factor, as MEA was found to be limited in performance when it was used at extreme exponent values, whilst other algorithms scaled in a linear fashion. The statistical tests supported these tendencies, as it was revealed that there is a significant performance difference between the four approaches. In general, the findings have practical implications in the choice of digital identification system algorithms, according to trade-offs of computational efficiency, memory usage, scalability, and security specifications.

VI. CONCLUSION

This study about the detailed analysis of the number-theoretic algorithms that are important in digital identification and data validation systems. In a comparison of the Modular Exponentiation Algorithm, Euclidean GCD Algorithm, Extended Euclidean Algorithm, and Hash-Based Number-Theoretic Validation, the study identifies the advantages and disadvantages of each of the algorithms in terms of the most important performance parameters such as execution time, memory, accuracy and collision behavior. Euclidean GCD algorithm turns out to be the most efficient and resource efficient algorithm to be used in the application where co-prime verification has to be done fast with fewer calculations being computed. Extended Euclidean Algorithm provides security modular inversion to conduct cryptographic activities, and MEA is highly accurate but fails to perform when the load on the computer is heavy. The HNTV scheme is more effective in high-volume validation problems needed to have robust hash-based checks with the use of higher memory consumption and slight collisions. These results highlight the need to match the choice of algorithms with the particular requirements of the system, to trade off between the computational efficiency, the security strength, and the scalability. Moreover, the study highlights applicability of such algorithms in new post-quantum secure systems and states how to design resilient, efficient, and scalable digital identification systems that can ensure authenticity of data and user authentication in the complicated technological systems.

REFERENCES

[1]. Chen, A. C. H. (2025) 'NIST post-quantum cryptography standard algorithms based on quantum random number generators', *arXiv preprint*, 2507.21151.

[2]. Serengil, S. &Ozpinar, A. (2025) 'LightDSA: A Python-Based Hybrid Digital Signature Library and Performance Analysis of RSA, DSA, ECDSA and EdDSA in Variable Configurations, Elliptic Curve Forms and Curves', *arXiv preprint*, 2505.23773.

[3]. Das, K. (2025) 'HexaMorphHash HMH – Homomorphic Hashing for Secure and Efficient Cryptographic Operations in Data Integrity Verification', *arXiv preprint*, 2507.21096.

[4]. Peng, Y., Feng, Q., He, D.-B. & Luo, M. (2025) 'A survey on threshold digital signature schemes', *Frontiers of Computer Science*, 20, article 2004806.

[5]. Chitra, M., Yadav, A., Mustafa, H. & Varghese, N. (2025) 'Cryptographic data security system for the digital world using SHA algorithms', *Indian Journal of Computer Science*, 10(2), pp. 8–13.

[6]. Kasimatis, D., Grierson, S., Buchanan, W. J., Eckl, C., Papadopoulos, P., Pitropakis, N., Thomson, C., Ghaleb, B. (2024) 'DID:RING: Ring Signatures using Decentralised Identifiers For Privacy-Aware Identity', *arXiv preprint*, 2403.05271.

[7]. Sedghighadikolaei, K. & Yavuz, A. (2024) 'A Comprehensive Survey of Threshold Signatures: NIST Standards, Post-Quantum Cryptography, Exotic Techniques, and Real-World Applications', *Preprint*.

[8]. Xu, J. (2025) 'A comprehensive study of digital signatures: algorithms, challenges and future prospects', *ITM Web of Conferences*, 73, 03009.

[9]. Gong, Z., Zheng, Z., Liu, F., Tian, K., Zhang, Y., Hu, Z., Li, J. & Xu, Q. (2024) 'A survey on lattice-based digital signature', *Cybersecurity*, 7, Article 7.

[10]. Cooley, J.W. & Tukey, J.W., 1965. An algorithm for the machine calculation of complex Fourier series. Mathematics of Computation, 19(90), pp.297–301.

[11]. Bisheh-Niasar, M., Azarderakhsh, R. & Mozaffari-Kermani, M., 2021. High-speed NTT-based polynomial multiplication accelerator for post-quantum cryptography. In: Proceedings of the IEEE 28th Symposium on Computer Arithmetic (ARITH), pp.94–101.

[12]. Bai, S., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schwabe, P., Seiler, G. &Stehlé, D., 2020. Supporting documentation: Crystals-dilithium: Algorithm specifications and supporting documentation. NIST Post-Quantum Cryptography.

[13]. Digital-Signature, N.M.L.B., 2024. Mechanism standard. NIST Post-Quantum Cryptography Standardization Process. Gaithersburg, MD, USA.

[14]. Ahmadi, K., Aghapour, S., Kermani, M.M. &Azarderakhsh, R., 2024. Efficient Algorithm-Level Error Detection for Number-Theoretic Transform used for Kyber Assessed on FPGAs and ARM. arXiv preprint arXiv:2403.01215.

[15]. Sarker, A., Canto, A.C., Kermani, M.M. &Azarderakhsh, R., 2022. Error detection architectures for hardware/software co-design approaches of number-theoretic transform. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 42(7), pp.2418–2422.

[16]. Ravi, P., Yang, B., Bhasin, S., Zhang, F. & Chattopadhyay, A., 2023. Fiddling the twiddle constants—fault injection analysis of the number theoretic transform. IACR Transactions on Cryptographic Hardware and Embedded Systems.

[17]. Bauer, S., Santis, F.D., Koleci, K. &Aghaie, A., 2024. A fault-resistant NTT by polynomial evaluation and interpolation. Cryptology ePrint Archive, Paper 2024/788. Available at: https://eprint.iacr.org/2024/788.

[18]. Li, D., Pakala, A. & Yang, K., 2022. MeNTT: A compact and efficient processing-in-memory number theoretic transform (NTT) accelerator. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 30(5), pp.579–588.

[19]. Derya, K., Mert, A.C., Oztürk, E. & Savaş, E., 2022. CoHA-NTT: A configurable hardware accelerator for NTT-based polynomial multiplication. Microprocessors and Microsystems, 89, 104451.

[20]. Bos, J., Ducas, L., Kiltz, E., Lepoint, T., Lyubashevsky, V., Schanck, J.M., Schwabe, P., Seiler, G. &Stehlé, D., 2018. CRYSTALS-Kyber: a CCA-secure module-lattice-based KEM. In: Proceedings of the IEEE European Symposium on Security and Privacy (EuroS&P), pp.353–367.

[21]. Fouque, P.A., Hoffstein, J., Kirchner, P., Lyubashevsky, V., Pornin, T., Prest, T., Ricosset, T., Seiler, G., Whyte, W., Zhang, Z., et al., 2018. Falcon: Fast-Fourier lattice-based compact signatures over NTRU. Submission to the NIST Post-Quantum Cryptography Standardization Process, 36(5), pp.1–75.

[22]. Heinz, D. &Pöppelmann, T., 2022. Combined fault and DPA protection for lattice-based cryptography. IEEE Transactions on Computers, 72(4), pp.1055–1066.

[23]. Nejatollahi, H., Cammarota, R. & Dutt, N., 2019. Flexible NTT accelerators for RLWE lattice-based cryptography. In: Proceedings of the IEEE 37th International Conference on Computer Design (ICCD), pp.329–332.

[24]. Nguyen, T.H., Kieu-Do-Nguyen, B., Pham, C.K. & Hoang, T.T., 2024. High-speed NTT Accelerator for CRYSTAL-Kyber and CRYSTAL-Dilithium. IEEE Access.

[25]. Jou, J.Y. & Abraham, J.A., 1988. Fault-tolerant FFT networks. IEEE Transactions on Computers, 37(5), pp.548–561.