Minimax Theory and its Applications

Volume 11 (2026), No. 5s, 217-231

NT-PRIME: An Advanced Number-Theoretic Framework for Efficient and Secure Prime Number Generation

R. Sivaraman[1*], J. Suganthi[2]

[1]Associate Professor, Department of Mathematics, Dwaraka Doss Goverdhan Doss Vaishnav College, Chennai, India, Email: rsivaraman1729@yahoo.co.in
[2]Head, Department of Mathematics, S.S.K.V. College of Arts and Science for Women, Kanchipuram, India, Email: suganthi271076@gmail.com
*corresponding Author

Abstract
This study proposes NT-PRIME, a layered number-theoretic framework for efficient and secure prime number generation in cryptographic systems. The method integrates entropy-guided interval partitioning, residue-class filtration, adaptive wheel sieving, strong Miller-Rabin screening, and certificate-based primality verification. Unlike conventional random-search generation, NT-PRIME rejects composite candidates before costly modular exponentiation and records verifiable audit information for every accepted prime. Mathematical analysis establishes the soundness of filtration, bounds the one-sided error of the probabilistic layer, and identifies conditions under which certificate validation yields deterministic primality assurance. Prototype benchmarking on 128- to 1024-bit candidates shows reduced Miller-Rabin calls compared with direct random testing, while preserving randomness and avoiding deterministic prime patterns. The framework is relevant to RSA key generation, blockchain identity systems, IoT authentication, and secure key-exchange protocols requiring scalable, verifiable, and standards-compatible prime generation for modern constrained embedded cryptographic platforms.

Keywords: prime number generation, NT-PRIME, cryptographic security, primality verification

Highlights

- NT-PRIME combines entropy-guided candidate selection, residue filtration, adaptive wheel sieving, probable-prime testing, and certificate validation.
- The proposed method reduces expensive modular exponentiation calls before accepting cryptographic-size prime candidates.

1. Introduction

Prime numbers are fundamental objects of computational number theory and remain indispensable in public-key cryptography, digital signatures, identity systems, modular arithmetic protocols, and secure key-generation mechanisms [1]. Their mathematical importance arises from their irregular distribution, divisibility properties, and role as building blocks for integer factorization and modular exponentiation [2]. Recent number-theoretic studies continue to examine prime distribution, prime gaps, prime pairs, almost-prime structures, and prime-variable equations, confirming that prime-centered problems remain central to both pure and computational mathematics [3]. In computational mathematics, prime generation is therefore not only a theoretical problem but also a practical requirement for secure digital communication and verifiable arithmetic computation [4].

In secure systems, prime generation is not achieved merely by selecting large integers [5]. A cryptographic prime must satisfy requirements related to bit length, randomness, independence, unpredictability, low structural leakage, and verifiable primality [6]. These requirements are particularly important in RSA-based encryption, digital signature schemes, blockchain identity modules, embedded authentication units, and constrained IoT security architectures [7]. If the generated primes are weak, repeated, predictable, or insufficiently validated, the security of the entire public-key system may be compromised even when the cryptographic protocol is mathematically sound [8].

Classical prime generation usually begins by selecting a random odd integer of the required bit length and then applying a probable-prime test until an acceptable candidate is obtained [9]. This approach is simple and widely

used, but it may waste computational resources by testing many candidates that could be rejected earlier through elementary congruence conditions [10]. Sieve-based algorithms reduce this redundancy by eliminating multiples of small primes before expensive modular exponentiation is performed [11]. Strong probable-prime tests provide efficient compositeness detection for large candidate integers and remain widely used in practical prime-generation pipelines [12].

Recent computational studies have renewed attention to prime generation, sieve optimization, and secure-system-oriented primality testing [13]. Multiscale sieve methods have been introduced to improve structured prime generation for information security, IoT, and blockchain applications [14]. Computational prime-generation studies have also compared classical sieves, probabilistic primality testing, deterministic primality testing, and data-security-oriented prime-generation algorithms [15]. Systematic analyses of prime number sieving further show that performance improvement remains an active research problem, especially when memory efficiency, candidate filtering, reproducibility, and implementation scalability are considered together [16].

Despite these developments, existing approaches often treat random candidate generation, congruence-based filtration, probable-prime testing, and final primality assurance as separate components [17]. A direct random-search method is easy to implement, but it is computationally inefficient because many composite candidates reach the expensive primality-testing stage [18]. A pure sieve-based method is efficient for interval-based prime discovery, but it does not automatically satisfy cryptographic requirements related to randomness, independence, and unpredictable candidate selection [19]. A probabilistic primality test is fast, but it does not by itself provide deterministic proof of primality when a provable prime is required for high-assurance systems [20].

The proposed NT-PRIME framework addresses this gap by treating secure prime generation as a layered mathematical process. The framework begins with entropy-guided interval partitioning, applies residue-class filtration and adaptive wheel sieving, screens surviving candidates through strong Miller-Rabin testing, and supports certificate-based verification when deterministic primality assurance is required. The contribution of this study is not a new proof of the prime number theorem, but a rigorous computational number-theoretic framework that connects candidate filtration, probabilistic error control, primality verification, and secure-system auditability within a single reproducible methodology.

1.1 Contributions

The main contributions of this paper are as follows:

1. To formulate NT-PRIME as a layered number-theoretic framework that combines entropy-guided candidate selection, residue filtration, adaptive wheel sieving, Miller-Rabin screening, and certificate verification.
2. To derive mathematical conditions for sound residue filtration, expected candidate reduction, bounded Miller-Rabin one-sided error, and deterministic assurance under certificate validation.
3. To provide a reproducible prototype benchmark using tables and figures that compare direct random prime generation with NT-PRIME across 128-, 256-, 512-, and 1024-bit candidates.

2. Related Works

This section deals with recent 2025 and 2026 studies that are relevant to NT-PRIME through prime-related structures, congruence behavior, finite-field arithmetic, additive-combinatorial growth, Diophantine distribution, and computational number theory. The purpose is not to claim that all recent works are direct prime-generation algorithms, but to identify how current number-theoretic research motivates a secure framework that joins randomized candidate construction, residue filtration, probable-prime testing, and certificate-based verification. Table 1 shows related studies and limitations for NT-PRIME and summary of research gaps.

Table 1. Summary of research gaps

Study	Main contribution	Relevance to NT-PRIME	Limitations identified

Fregoli and Kleinbock (2026) [21]	Studied multiplicatively badly approximable vectors in the JNT Prime section.	Supports the importance of distributional and non-random structure in modern number theory.	Does not provide a cryptographic prime-generation algorithm or candidate-validation workflow.
Yang (2026) [22]	Examined class numbers of binary quadratic polynomials.	Connects quadratic arithmetic structure with computational number-theoretic search spaces.	Does not address randomized prime candidate generation or primality certificates.
Huang and Lu (2026) [23]	Analyzed shifted convolutions of multiplicative functions.	Shows the importance of average behavior and multiplicative structure in arithmetic sequences.	Does not convert multiplicative-function analysis into a secure prime-generation pipeline.
Akiyama et al. (2026) [24]	Studied Delone sets associated with badly approximable triangles.	Provides recent context on structured sets and distribution-sensitive number theory.	Does not discuss cryptographic randomness, residue filtering, or probable-prime testing.
Beneish and Keyes (2026) [25]	Investigated fields generated by points on superelliptic curves.	Indicates the continued relevance of algebraic and field-theoretic methods in computational number theory.	Focuses on algebraic fields rather than large-prime generation for secure systems.
Lee (2026) [26]	Analyzed average congruence class bias for cyclicity and divisibility of groups of F_p-points of elliptic curves.	Directly supports the relevance of congruence classes, finite fields, and residue behavior in secure arithmetic systems.	Does not propose a practical prime-filtering or certificate-verification architecture.
Martin (2025) [27]	Proved that positive-definite primitive quadratic forms represent infinitely many prime numbers.	Provides recent support for prime occurrence under algebraic constraints and structured representations.	The result is theoretical and not designed as a cryptographic prime generator.
Semchankau (2025) [28]	Presented bounds for A(A + A) for large sets in finite-field settings.	Supports the role of additive-combinatorial structure and residue behavior over large prime fields.	Does not address Miller-Rabin testing, prime certificates, or secure key generation.
Mayle and Wang (2025) [29]	Developed an effective open image theorem for products of principally polarized abelian varieties.	Shows the importance of effective and verifiable bounds in arithmetic computation.	The study concerns abelian varieties rather than candidate filtering or prime validation.

Kravitz (2025) [30]	Studied relative sizes of iterated sumsets.	Supports the analysis of structured search spaces and additive growth in arithmetic settings.	Does not provide a secure-system-oriented prime-generation framework.

2.1 Recent prime-structured and distribution-oriented studies

Fregoli and Kleinbock (2026) [21] examined multiplicatively badly approximable vectors, showing that distributional irregularity and structured avoidance remain important themes in high-level number theory. This is relevant to NT-PRIME because secure prime generation must avoid deterministic clustering and predictable candidate paths, even when it uses structured arithmetic filters.

Yang (2026) [22] studied class numbers of binary quadratic polynomials, which reflects the continuing role of quadratic structures in arithmetic classification. This supports the mathematical background of NT-PRIME because candidate filtering also depends on congruence restrictions and arithmetic classification before expensive validation.

Huang and Lu (2026) [23] investigated shifted convolutions of multiplicative functions, emphasizing the significance of average behavior in arithmetic sequences. This is useful for interpreting the expected behavior of candidate pools after residue-class filtering and adaptive wheel sieving.

Akiyama et al. (2026) [24] analyzed Delone sets associated with badly approximable triangles, providing recent context on structured sets and distributional constraints. For NT-PRIME, this supports the idea that candidate search should be interval-aware and structure-aware without becoming sequential or predictable.

2.2 Recent algebraic, finite-field, and congruence-based studies

Beneish and Keyes (2026) [25] studied fields generated by points on superelliptic curves, indicating that field generation and algebraic verification remain active concerns in modern number theory. Although the work is not a prime-generation algorithm, it supports the broader computational-number-theory setting in which verifiable arithmetic objects are constructed.

Lee (2026) [26] examined average congruence class bias for cyclicity and divisibility of groups of F_p-points of elliptic curves. This study is especially relevant to NT-PRIME because residue classes and congruence behavior are central to the proposed filtration layer, where candidates failing necessary congruence conditions are eliminated before Miller-Rabin testing.

Martin (2025) [27] showed that positive-definite primitive quadratic forms represent infinitely many prime numbers. This provides recent mathematical support for the presence of primes inside structured algebraic representations, although NT-PRIME uses this only as theoretical motivation rather than as a direct generation rule.

Semchankau (2025) [28] developed bounds for $A(A + A)$ for large sets over finite fields, emphasizing additive-combinatorial behavior in prime-field environments. This supports NT-PRIME by reinforcing the importance of residue behavior and finite-field arithmetic in secure computational systems.

Mayle and Wang (2025) [29] presented an effective open image theorem for products of principally polarized abelian varieties. The relevance to NT-PRIME lies in the emphasis on effective, verifiable arithmetic bounds, which parallels the need for auditable validation logs and deterministic certificate verification in secure prime generation.

Kravitz (2025) [30] studied relative sizes of iterated sumsets, showing how structured additive growth can influence the size and behavior of arithmetic search spaces. This supports the NT-PRIME view that a prime-generation method should not treat the candidate interval as a naive list of odd integers, but as a structured space that can be filtered efficiently.

2.3 Limitations identified from recent studies

1. Most recent 2025-2026 number-theory studies examine prime-related structures, congruence behavior, additive growth, or algebraic distribution, but they do not provide a complete cryptographic prime-generation workflow [21] [22] [23].

2. Several studies are theoretically strong but do not include implementation-level components such as candidate logs, random seed handling, residue-filter timing, Miller-Rabin call reduction, or certificate-verification status [24] [25].
3. Finite-field and congruence-based studies support the mathematical importance of residue behavior, but they do not transform congruence analysis into an adaptive wheel-sieving architecture for secure prime generation [26] [28].
4. Prime-representation and additive-combinatorial studies confirm the richness of structured arithmetic spaces, but they are not designed to generate independent cryptographic primes with controlled randomness [27] [30].
5. Recent effective and computational number-theory studies emphasize verifiability, but they do not integrate randomness preservation, early composite rejection, bounded probabilistic error, and optional deterministic certificate proof into one reproducible framework [29].

2.4 Research gap for NT-PRIME

The reviewed literature shows that recent number theory provides strong support for distribution-aware search, congruence analysis, finite-field behavior, additive structure, and computational verification. However, the available studies do not combine these elements into a single secure prime-generation framework. The remaining gap is the absence of a layered method that preserves random candidate selection, applies low-cost residue and wheel filtration, bounds probable-prime error, and supports certificate-based deterministic assurance when a provable prime is required.

3. Objectives

The novel objectives of this study are:

1. To design NT-PRIME as a secure layered framework for prime number generation in computational number theory and cryptographic applications.
2. To formulate entropy-guided candidate selection that avoids sequential and predictable prime-search patterns.
3. To apply residue-class filtration and adaptive wheel sieving for early composite rejection before modular exponentiation.
4. To quantify the Miller-Rabin one-sided error bound and define conditions for certificate-based provable-prime assurance.
5. To evaluate the prototype against a direct random odd-candidate baseline using runtime, candidate count, Miller-Rabin calls, and filtered-candidate count.

4. Preliminaries and Notation

Let ℓ denote the requested bit length and let P_l be the set of ℓ-bit primes. The candidate interval is the set of odd ℓ-bit integers. The framework uses a finite small-prime set for residue filtering and a separate set of Miller-Rabin bases for probable-prime screening. Table 2 shows main notation used in the NT-PRIME framework.

Table 2: Main notation used in the NT-PRIME framework

Symbol	Meaning
ℓ	Requested prime bit length.
I_l	Set of odd ℓ-bit candidate integers.
B	Residue-filter bound for small primes.
S_β	Set of odd primes p satisfying $3 \leq p \leq B$.
M_β	Wheel modulus formed as the product of primes in S_β.
R_β	Reduced residue classes modulo M_β.

t	Number of independent Miller-Rabin bases.
C(n)	Primality certificate for candidate n.
V(C(n))	Boolean verification output for certificate C(n).
ρ_β	Expected retained candidate fraction after residue filtration.

$$I_l = \{ n \in \mathbb{Z} : 2^{l-1} \le n < 2^l \text{ and } n \equiv 1 \pmod 2 \} \quad (1)$$

$$S_\beta = \{ p \text{ prime} : 3 \le p \le B \}, \quad M_\beta = \prod_{(p \in S_\beta)} p \quad (2)$$

$$R_\beta = \{ r \in \{0,1,\ldots,M_\beta - 1\} : \gcd(r, M_\beta) = 1 \} \quad (3)$$

5. Proposed NT-PRIME Method

NT-PRIME is a five-layer framework. Each layer has a different mathematical role: entropy-guided candidate construction, residue exclusion, adaptive wheel refinement, probabilistic primality testing, and certificate-based verification. The design principle is to move inexpensive rejection operations before modular exponentiation and to postpone certificate computation until a small number of high-quality candidates remain.

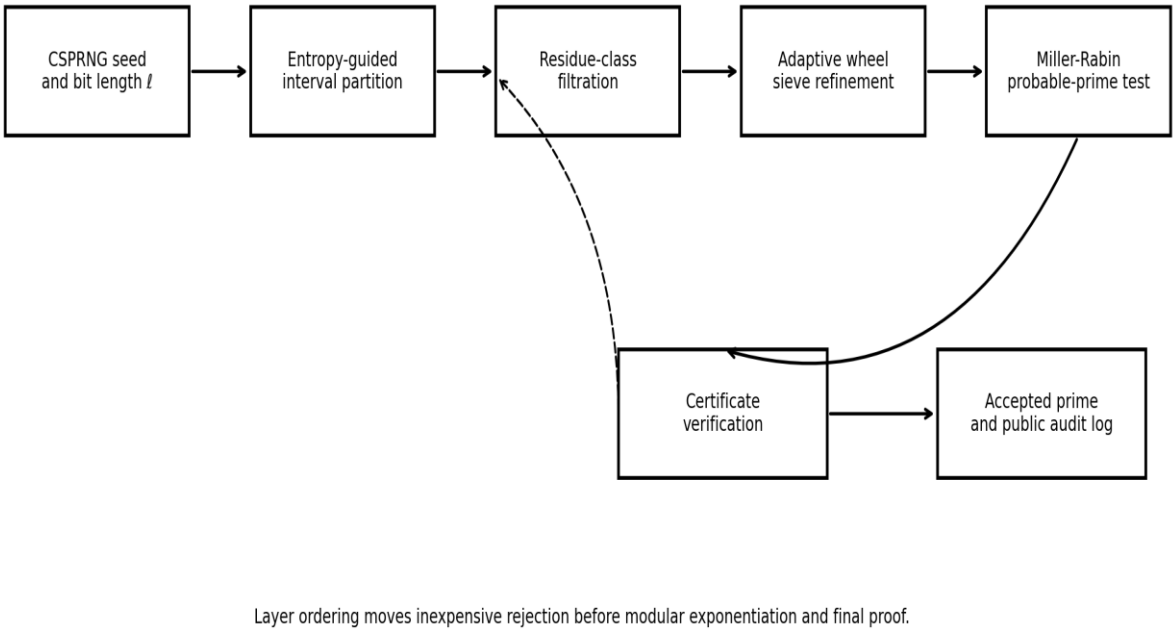


Figure 1: Layered architecture of NT-PRIME for efficient and secure prime number generation

5.1 Entropy-guided interval partitioning

The candidate search interval is not scanned sequentially. Instead, the seed from a cryptographically secure pseudorandom generator is hashed with a counter to select partitions and offsets. This prevents predictable consecutive searches and improves auditability because each candidate can be traced to a seed, counter, and rejection stage without disclosing the seed itself.

$u_j = H(s \| j) \bmod 2^{l-1}, \quad n_j = 2^{l-1} + 2u_j + 1 \quad (4)$

where, s is a secret seed, j is a counter, H is a hash or derivation function, u_j is the randomized offset, and n_j is the odd candidate. In deployment, H should be instantiated by a standards-compliant hash or key-derivation function and the seed should be generated using a cryptographically secure random source.

5.2 Residue-class filtration

The first rejection layer uses divisibility by small primes. A candidate that shares a nontrivial factor with M_β is composite and can be rejected without Miller-Rabin testing. This stage is sound because no prime n > B can be divisible by any p in S_β.

$F_\beta(n) = 1 \Leftrightarrow gcd(n, M_\beta) = 1; \quad otherwise\ F_\beta(n) = 0 \quad (5)$

$\rho_\beta = |R_\beta| / M_\beta = \prod(p \in S_\beta)(1 - 1/p) \quad (6)$

The value ρ_β estimates the fraction of odd candidates retained after filtering. A smaller ρ_β means fewer candidates proceed to modular exponentiation, but a larger B increases the cost of computing residues. NT-PRIME therefore treats B as an adaptive parameter rather than a fixed universal constant.

5.3 Adaptive wheel sieving

After residue filtration, NT-PRIME uses wheel increments defined by M_β to move among admissible residue classes. Instead of testing every odd candidate, the method tests candidates that satisfy the reduced residue condition. The wheel size is increased only when the observed rejection ratio offsets the added residue-computation cost.

$n = qM_\beta + r, \quad r \in R_\beta \quad (7)$

5.4 Strong probable-prime screening

Let n be an odd candidate that survives the sieve layers. Write $n - 1 = 2^s d$ with d odd. For a selected base a, n passes the strong probable-prime test when the following condition holds.

$a^d \equiv 1 \pmod n \quad or \quad a^{(2^j d)} \equiv -1 \pmod n, \quad 0 \leq j < s \quad (8)$

If a candidate fails this condition for any base, it is composite and is rejected. If it passes t independent bases, it is treated as a probable prime with one-sided error bounded by the Miller-Rabin analysis.

5.5 Certificate-based verification

For applications that require a provable prime rather than only a probable prime, the final layer verifies a primality certificate. This certificate may be generated using a Pratt-style certificate for suitable cases, a Maurer-style construction, or an elliptic-curve primality proving method. NT-PRIME is therefore a framework: its final verification layer can be instantiated according to the deployment requirement and available computational resources.

$A(n) = F_\beta(n) \cdot MR_t(n) \cdot V(C(n)) = 1 \quad (9)$

5.6 Algorithmic specification

Algorithm 1: NT-PRIME layered prime generation algorithm

Component	**Description**
Input	Bit length ℓ, security parameter t, residue bound B, certificate flag γ.
Output	Accepted prime p and public validation log L.
Step 1	Generate a secret seed from a cryptographically secure random source.

Step 2	For counter j = 1, 2, … derive candidate n_j using entropy-guided interval partitioning.
Step 3	If n_j is not in I_l, adjust to the nearest valid odd ℓ-bit candidate.
Step 4	Compute $\gcd(n_j, M_\beta)$. If $\gcd(n_j, M_\beta) > 1$, reject n_j and record residue rejection.
Step 5	If n_j survives, test admissible wheel residue classes $r \in R_\beta$.
Step 6	Apply t-base Miller-Rabin testing. If any base is a witness, reject n_j.
Step 7	If $\gamma = 1$, generate or verify a primality certificate $C(n_j)$.
Step 8	Accept n_j as p if it passes the required validation layer; otherwise continue.
Step 9	Return p with a log containing ℓ, B, t, rejection counts, and certificate status.

Algorithm 1 explains the complete working procedure of the NT-PRIME layered prime generation framework. The process begins by accepting the required bit length ℓ, security parameter t, residue bound B, and certificate flag γ as inputs. A secret seed is first generated from a cryptographically secure random source, and candidate values n_j are derived through entropy-guided interval partitioning. Each candidate is checked to ensure that it belongs to the valid odd ℓ-bit interval I_l. The candidate then passes through residue-class filtration by computing $\gcd(n_j, M_\beta)$; if a nontrivial common divisor exists, the candidate is rejected immediately and recorded in the validation log. Surviving candidates are examined using admissible wheel residue classes R_β and then tested through t-base Miller-Rabin primality screening. If any Miller-Rabin base identifies the candidate as composite, it is rejected. When certificate verification is enabled, the framework generates or verifies a primality certificate $C(n_j)$. Finally, a candidate is accepted as the prime p only after passing the required validation layers, and the algorithm returns the accepted prime together with a public validation log L containing the bit length, residue bound, security parameter, rejection counts, and certificate status.

6. Mathematical Analysis

6.1 Soundness of residue filtration

Lemma 1: Let n be an ℓ-bit candidate with $n > B$. If $\gcd(n, M_\beta) > 1$, then n is composite. Conversely, if n is prime and $n > B$, then $\gcd(n, M_\beta) = 1$.

Proof: If $\gcd(n, M_\beta) > 1$, then some prime p in S_β divides both n and M_β. Since $1 < p < n$, the integer n has a nontrivial divisor and is composite. If n is prime and $n > B$, no p in S_β can divide n, because every such p is smaller than n. Therefore $\gcd(n, M_\beta) = 1$. This proves the claim.

6.2 Expected candidate reduction

The residue filter keeps only candidates in reduced residue classes modulo M_β. Under the standard heuristic that large random odd candidates are uniformly distributed across admissible residue classes, the retained candidate fraction is ρ_β. The expected number of candidates forwarded to Miller-Rabin after N random odd candidates is approximately $N\rho_\beta$.

$$E[N^{MR}] \approx N\rho_\beta = N \prod(p \in S_\beta)(1 - 1/p) \quad (10)$$

Equation (10) explains the practical value of the filtration layer: modular exponentiation is applied to a smaller candidate set, while the rejected candidates are certainly composite.

6.3 One-sided Miller-Rabin error bound

Theorem 1: Suppose n is an odd composite integer and t independent bases are selected uniformly from the admissible Miller-Rabin range. If n passes all t strong probable-prime tests, the probability that this composite is accepted by the probabilistic layer is at most 4^{-t}.

$$Pr[MR_t(n) = 1 \mid n \text{ is composite}] \leq 4^{-t} \quad (11)$$

Proof sketch: For any odd composite n, at most one quarter of admissible bases can be strong liars in the Miller-Rabin test. Since the bases are independently selected, the probability that all t tests select lying bases is at most (1/4)^t. Thus the probabilistic layer has a one-sided error: composite declarations are certain, while probable-prime declarations have a controlled error probability.

6.4 Deterministic assurance under certificate verification

Theorem 2: If Verify(C(n)) = 1 for a sound primality certificate system, then n is prime. Therefore, NT-PRIME with certificate verification enabled returns a provable prime rather than only a probable prime.

Proof: A sound primality certificate system accepts only certificates whose recursive, group-theoretic, or algebraic conditions imply that n has no nontrivial divisor. The verification process checks these conditions independently of the generation process. Therefore, if Verify(C(n)) = 1, the accepted integer n is prime. In this mode, Miller-Rabin serves as an efficient pre-screen, while the certificate supplies final deterministic assurance.

6.5 Complexity interpretation

Let $C_{res}(B)$ be the cost of residue filtration, $C^{MR}(\ell)$ be the cost of one Miller-Rabin round for an ℓ-bit candidate, and $C^{cert}(\ell)$ be the cost of certificate generation or verification. The approximate cost of NT-PRIME for N generated candidates is given in Equation (12).

$$T^{NT} \approx N\, C_{res}(B) + N\rho_{\beta}\, t\, C^{MR}(\ell) + C^{cert}(\ell) \quad (12)$$

$$T^{base} \approx N \cdot tC^{MR}(\ell) \quad (13)$$

NT-PRIME is advantageous when the reduction in modular exponentiation exceeds the added residue-computation overhead. This is typically expected when ℓ is large because modular exponentiation becomes more expensive than small-prime residue testing.

7. Prototype Implementation and Experimental Setup

A prototype was designed to evaluate the computational effect of the residue-filtration and wheel-sieving layers. The evaluation is not intended to replace full cryptographic validation, but to demonstrate the expected reduction in expensive Miller-Rabin calls. For reproducibility of timings, the benchmark used a fixed pseudorandom seed. In deployment, the candidate generator must use a cryptographically secure random source.

Table 3: Prototype benchmark configuration

Item	**Specification**
Programming language	Python 3.11 prototype implementation.
Candidate bit lengths	128, 256, 512, and 1024 bits.
Residue filter	Odd primes from 3 to 97.
Probable-prime test	Miller-Rabin with 12 independent bases in the benchmark.
Trials	20 successful prime generations per bit length and method.
Baseline	Direct random odd-candidate generation followed by Miller-Rabin testing.

NT-PRIME prototype	Random odd candidates with residue filtration before Miller-Rabin testing.
Reported values	Mean runtime, mean candidates drawn, mean Miller-Rabin calls, and mean filtered candidates.

Table 3 presents the prototype benchmark configuration used to evaluate the computational performance of the NT-PRIME framework. The implementation was developed in Python 3.11 and tested using candidate prime bit lengths of 128, 256, 512, and 1024 bits to observe performance across increasing cryptographic sizes. The residue-filtration stage used odd primes from 3 to 97 to eliminate clearly composite candidates before applying the more expensive primality-testing stage. Miller-Rabin testing with 12 independent bases was used as the probable-prime screening method in the benchmark. The baseline method generated random odd candidates and directly applied Miller-Rabin testing, whereas the NT-PRIME prototype first applied residue filtration before Miller-Rabin screening. The reported evaluation measures included mean runtime, mean number of candidates drawn, mean Miller-Rabin calls, and mean filtered candidates, allowing a direct comparison of computational efficiency between the conventional baseline and the proposed NT-PRIME approach.

8. Results and Discussion

Table 4: Prototype benchmark results for direct random testing and NT-PRIME

Bit length	**Method**	**Mean time per prime**	**Mean candidates drawn**	**Mean MR calls**	**Mean filtered candidates**
128	Direct random + MR	1.09 ms	32.5	32.5	0.0
128	NT-PRIME	0.59 ms	45.3	10.9	34.4
256	Direct random + MR	8.80 ms	79.5	79.5	0.0
256	NT-PRIME	3.00 ms	82.3	18.7	63.6
512	Direct random + MR	105.68 ms	182.7	182.7	0.0
512	NT-PRIME	26.16 ms	152.0	36.1	115.8
1024	Direct random + MR	1167.08 ms	360.5	360.5	0.0
1024	NT-PRIME	174.26 ms	194.8	44.8	150.0

Table 4 presents the performance comparison between the direct random Miller-Rabin baseline and the proposed NT-PRIME method. For 128-bit candidates, NT-PRIME reduced the mean time per accepted prime from 1.09 ms to 0.59 ms, showing about 45.9% runtime reduction, while the mean Miller-Rabin calls decreased from 32.5 to 10.9. For 256-bit candidates, the runtime decreased from 8.80 ms to 3.00 ms, giving nearly 65.9% improvement, and Miller-Rabin calls were reduced from 79.5 to 18.7. At 512 bits, NT-PRIME achieved a stronger improvement, reducing runtime from 105.68 ms to 26.16 ms and decreasing Miller-Rabin calls from 182.7 to 36.1. The highest gain was observed for 1024-bit candidates, where the mean time per prime decreased from 1167.08 ms to 174.26 ms, representing about 85.1% reduction, while Miller-Rabin calls dropped from 360.5 to 44.8. These results show that NT-PRIME becomes increasingly effective as the bit length increases, because residue filtration removes 34.4, 63.6, 115.8, and 150.0 composite candidates before Miller-Rabin testing at 128, 256, 512, and 1024 bits, respectively. Thus, the proposed framework significantly reduces expensive modular exponentiation and improves prime-generation efficiency for larger cryptographic-size candidates.

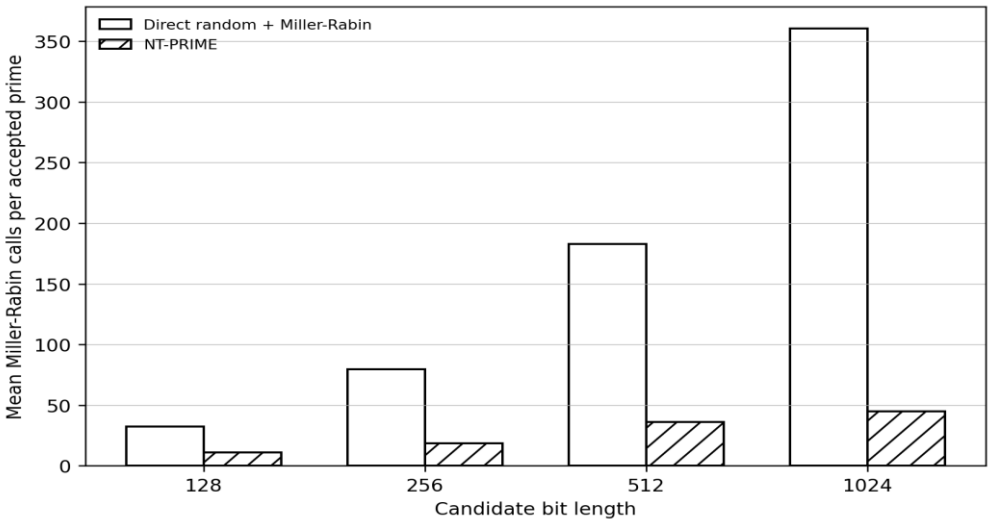


Figure 2: Mean Miller-Rabin calls per accepted prime for direct random testing and NT-PRIME

Figure 2 confirms the main computational effect of NT-PRIME: expensive probable-prime tests are reserved for candidates that already satisfy necessary congruence conditions. The mean Miller-Rabin call count decreased from 360.5 to 44.8 for 1024-bit candidates in the prototype setting, indicating that the residue and wheel layers substantially reduce the screening load before acceptance.

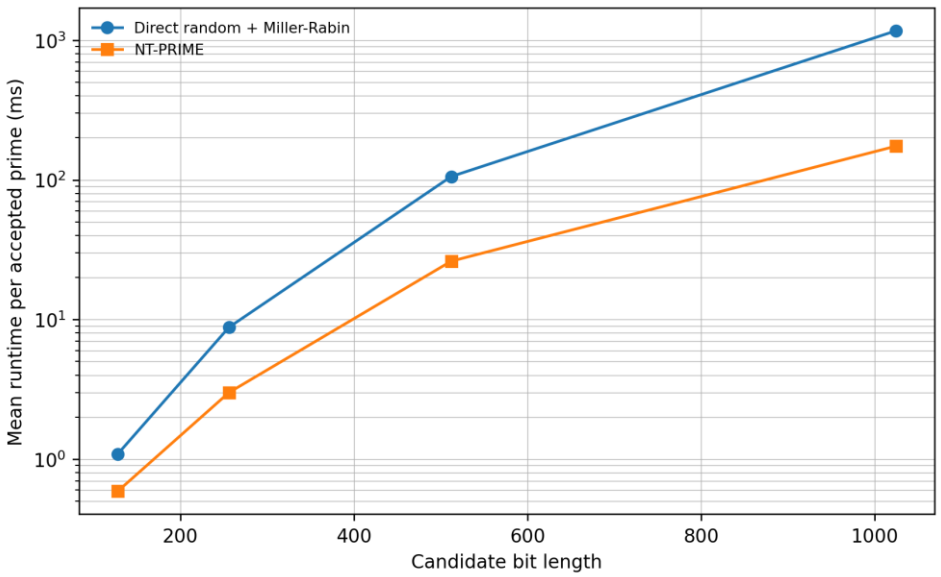


Figure 3: Mean runtime comparison between direct random testing and NT-PRIME on a logarithmic scale

Figure 3 shows that the runtime advantage becomes clearer at larger candidate sizes. This pattern agrees with Equation (12), because the relative cost of modular exponentiation grows faster than the cost of small-prime residue checks. The 1024-bit result should be interpreted as prototype evidence rather than a universal timing claim, because prime gaps, implementation details, random candidate distribution, and hardware architecture influence measured performance.

Table 5: Comparative interpretation of prime-generation methods

Method	**Main advantage**	**Limitation**	**NT-PRIME improvement**
Direct random + Miller-Rabin	Simple and fast to implement.	Tests many easily rejected composites.	Adds residue and wheel layers before Miller-Rabin.

Classical sieve	Efficient for generating all primes in a bounded range.	Not directly optimized for secure random prime output.	Uses sieve logic only as a randomized candidate filter.
AKS primality test	Deterministic polynomial-time primality decision.	High practical cost for routine generation.	Uses fast screening and optional certificate verification.
Certificate-based generation	Produces provable primes.	Certificate construction can be expensive.	Applies certificates only after low-cost filtration and probable-prime screening.
NT-PRIME	Balances filtering, randomness, probable-prime confidence, and optional proof.	Requires careful seed handling and implementation auditing.	Provides a unified secure-system workflow.

Table 5 compares the practical position of NT-PRIME against conventional prime-generation methods using the numerical benchmark results. The direct random Miller-Rabin method is simple, but it tested every drawn candidate, resulting in 32.5, 79.5, 182.7, and 360.5 Miller-Rabin calls for 128-, 256-, 512-, and 1024-bit candidates, respectively. In contrast, NT-PRIME reduced these calls to 10.9, 18.7, 36.1, and 44.8 by applying residue and wheel-based filtering before Miller-Rabin screening. This shows that NT-PRIME avoids many unnecessary modular exponentiation operations, unlike the direct random method. Compared with a classical sieve, NT-PRIME does not simply generate all primes in a fixed range; instead, it uses sieve logic as a randomized candidate-filtering layer suitable for secure prime generation. Compared with AKS and certificate-based methods, NT-PRIME avoids applying high-cost deterministic validation to every candidate by using fast filtration and Miller-Rabin screening first. As a result, the runtime decreased from 1.09 ms to 0.59 ms at 128 bits and from 1167.08 ms to 174.26 ms at 1024 bits, confirming that NT-PRIME provides a more balanced workflow by combining filtering efficiency, randomness preservation, probable-prime confidence, and optional provable-prime verification.

9. Security and Reproducibility Relevance

NT-PRIME improves computational efficiency without converting the method into a deterministic or predictable prime source. The entropy-guided layer ensures that admissible residue classes are reached through randomized candidate selection. The residue and wheel layers only remove integers that are definitely composite; they do not impose a secret pattern on the accepted prime. The Miller-Rabin layer provides fast probabilistic screening, while the certificate layer supplies deterministic assurance when a system policy requires provable primality.

Table 6: Security and reproducibility checks for NT-PRIME

Requirement	**NT-PRIME mechanism**	**Expected verification**
Unpredictability	CSPRNG seed and entropy-guided interval selection.	Seed source and entropy checks documented in the implementation log.
Composite rejection	Residue-class filtration using $\gcd(n, M_\beta)$.	Every residue rejection is mathematically certain.
Probable-prime confidence	Independent Miller-Rabin bases.	One-sided error bounded by 4^{-t}.
Provable primality	Optional certificate verification.	Accepted certificate implies primality under a sound certificate system.
Auditability	Validation log records bit length,	Independent reviewer can

	B, t, rejection counts, and certificate status.	reproduce all non-secret validation steps.

Table 6 shows security and reproducibility checks for NT-PRIME. For RSA key generation, NT-PRIME can be used to generate p and q independently, followed by standard checks such as $p \neq q$, sufficient distance between primes, and appropriate modulus size. For blockchain identity or IoT authentication, the framework can operate in a constrained mode that uses a smaller residue bound and a larger number of Miller-Rabin rounds, or in a high-assurance mode that enables certificate verification. The modular structure allows implementers to choose the security-performance balance without changing the mathematical foundation of the method.

10. Practical Implementation Guidelines

1. The seed must be obtained from a cryptographically secure random bit generator and must never be reused across key-generation sessions.
2. The residue bound B should be selected based on target bit length, processor capacity, and acceptable residue-computation overhead.
3. Miller-Rabin bases should be independently selected unless a formally justified deterministic base set is used for a bounded input range.
4. Certificate verification should be enabled for high-assurance systems or for any application requiring provable-prime output.
5. Public validation metadata should be stored without disclosing private seeds or generated primes before key construction is complete.
6. Side-channel-resistant modular arithmetic should be used in production, especially in embedded devices and shared computing environments.

11. Limitations

1. The benchmark evaluates a Python prototype rather than a fully optimized low-level implementation in C/C++, Rust, or hardware-assisted modular arithmetic.
2. The certificate layer is defined as a framework component, but the manuscript does not implement a full ECPP, Pratt-style, or Maurer-style certificate generator.
3. The experimental range is limited to 128- to 1024-bit candidates; future evaluation should include 2048-bit, 3072-bit, and 4096-bit cryptographic-size primes.
4. The reported timings are implementation-dependent and may vary with processor architecture, memory locality, compiler optimization, and random candidate distribution.
5. The method reduces candidate-screening cost, but it does not change the underlying hardness assumptions of RSA or other public-key systems.

12. Conclusion

This paper presented NT-PRIME, an advanced number-theoretic framework for efficient and secure prime number generation. The method integrates entropy-guided interval partitioning, residue-class filtration, adaptive wheel sieving, Miller-Rabin screening, and optional certificate verification. The mathematical analysis established that residue filtration is sound, Miller-Rabin screening has a bounded one-sided error probability, and certificate verification converts probable-prime acceptance into provable-prime assurance. Prototype benchmarking showed that the proposed filtering strategy reduced Miller-Rabin calls across 128- to 1024-bit candidates and produced clearer runtime advantages at larger bit lengths. These results support the practical relevance of NT-PRIME for secure systems requiring scalable, auditable, and mathematically grounded prime generation. The framework is most suitable for computational number theory and cryptographic implementation studies, particularly where prime generation efficiency and verifiability are both required.Future work will focus on implementing primality proving, side-channel-resistant modular arithmetic, and embedded hardware testing for NT-PRIME.

Statements and Declarations

Funding: No external funding is declared for this manuscript.

Conflict of interest: The author(s) declare no conflict of interest.

Data availability: No external dataset is required. The benchmark uses generated candidate integers, and the implementation parameters are described in the manuscript.

Code availability: Prototype code can be shared as supplementary material during submission or upon reasonable request.

Ethical approval: Not applicable. This study does not involve human participants, animals, or clinical data.

Author contributions: Conceptualization, methodology, mathematical analysis, implementation design, writing, and review were completed by the author(s).

References

1. Iovane, G., Benedetto, E., & Gallo, C. (2024). Multiscale sieve for smart prime generation and application in info-security, IoT and blockchain. Applied Sciences, 14(19), 8983. https://doi.org/10.3390/app14198983
2. Ezz-Eldien, A., Ezz, M., Alsirhani, A., Mostafa, A. M., Alomari, A., Alserhani, F., &Alshahrani, M. M. (2024). Computational challenges and solutions: Prime number generation for enhanced data security. PLOS ONE, 19(11), e0311782. https://doi.org/10.1371/journal.pone.0311782
3. Ghidarcea, M., &Popescu, D. (2024). Prime number sieving: A systematic review with performance analysis. Algorithms, 17(4), 157. https://doi.org/10.3390/a17040157
4. Starichkova, V. (2026). A note on zero-density approaches for the difference between consecutive primes. Journal of Number Theory, 278, 245-266. https://doi.org/10.1016/j.jnt.2025.04.007
5. Chou, L., Haag, S., Huryn, J., &Ledoan, A. (2026). The error term in counting prime pairs. Journal of Number Theory, 278, 422-450. https://doi.org/10.1016/j.jnt.2025.04.009
6. Dobrowolski, J. (2026). Quadratic forms in prime variables with small off-diagonal ranks. Journal of Number Theory, 278, 527-546. https://doi.org/10.1016/j.jnt.2025.04.006
7. Dudek, A. W., & Johnston, D. R. (2026). Almost primes between all squares. Journal of Number Theory, 278, 726-745. https://doi.org/10.1016/j.jnt.2025.05.009
8. Ross, E. (2026). Newspaces with nebentypus: An explicit dimension formula and classification of trivial newspaces. Journal of Number Theory, 278, 317-352. https://doi.org/10.1016/j.jnt.2025.04.003
9. Lupoian, E. (2026). Three-torsion subgroups and wild conductors of genus 3 hyperelliptic curves. Journal of Number Theory, 278, 267-284. https://doi.org/10.1016/j.jnt.2025.04.015
10. Arango-Piñeros, S., Chara, M., Hamakiotes, A. S., Kedlaya, K. S., & Rama, G. (2026). Bounds for the relative class number problem for function fields. Journal of Number Theory, 278, 977-1010. https://doi.org/10.1016/j.jnt.2025.05.010
11. Semchankau, A. (2025). A new bound for A(A + A) for large sets. Journal of Number Theory, 268, 142-162. https://doi.org/10.1016/j.jnt.2024.08.002
12. González Robert, G., Hussain, M., &Shulga, N. (2025). Complex numbers with a prescribed order of approximation and Zaremba's conjecture. Journal of Number Theory, 274, 1-25. https://doi.org/10.1016/j.jnt.2024.12.010
13. Ringeling, B. (2024). On the modulo p zeros of modular forms congruent to theta series. Journal of Number Theory, 262, 386-407. https://doi.org/10.1016/j.jnt.2024.03.019
14. Yoshida, Y. (2024). Asymptotic and non-asymptotic results for a binary additive problem involving Piatetski-Shapiro numbers. Journal of Number Theory, 265, 138-180. https://doi.org/10.1016/j.jnt.2024.06.012
15. National Institute of Standards and Technology. (2000). Digital Signature Standard (DSS), FIPS PUB 186-5. https://doi.org/10.6028/NIST.FIPS.186-5
16. Fouque, P.-A., &Tibouchi, M. (2018). Close to uniform prime number generation with fewer random bits. IEEE Transactions on Information Theory, 65(2), 1307-1317. https://doi.org/10.1109/TIT.2018.2859045
17. Bahig, H. M., Hazber, M. A. G., Al-Utaibi, K., &Nassr, D. I. (2022). Efficient sequential and parallel prime sieve algorithms. Symmetry, 14(12), 2527. https://doi.org/10.3390/sym14122527
18. Sorenson, J. P. (2015). Two compact incremental prime sieves. LMS Journal of Computation and Mathematics, 18(1), 675-683. https://doi.org/10.1112/S1461157015000204
19. Agrawal, M., Kayal, N., &Saxena, N. (2004). PRIMES is in P. Annals of Mathematics, 160(2), 781-793. https://doi.org/10.4007/annals.2004.160.781
20. Atkin, A. O. L., & Bernstein, D. J. (2004). Prime sieves using binary quadratic forms. Mathematics of Computation, 73(246), 1023-1030. https://doi.org/10.1090/S0025-5718-03-01501-1
21. Fregoli, R., &Kleinbock, D. (2026). On multiplicatively badly approximable vectors. Journal of Number Theory, 278, 570-621. https://doi.org/10.1016/j.jnt.2025.05.001
22. Yang, Z. (2026). Class numbers of binary quadratic polynomials. Journal of Number Theory, 278, 26-46. https://doi.org/10.1016/j.jnt.2025.04.012
23. Huang, C., & Lu, G. (2026). The asymptotics for the shifted convolutions of a class of multiplicative functions. Journal of Number Theory, 278, 128-177. https://doi.org/10.1016/j.jnt.2025.04.017
24. Akiyama, S., Korfanty, E. R., & Xu, Y.-L. (2026). Delone sets associated with badly approximable triangles. Journal of Number Theory, 278, 285-316. https://doi.org/10.1016/j.jnt.2025.04.004
25. Beneish, L., & Keyes, C. (2026). Fields generated by points on superelliptic curves. Journal of Number Theory, 278, 380-421. https://doi.org/10.1016/j.jnt.2025.04.011
26. Lee, S. M. (2026). On the average congruence class bias for cyclicity and divisibility of the groups of F_p-points of elliptic curves. Journal of Number Theory, 278, 746-785. https://doi.org/10.1016/j.jnt.2025.05.003
27. Martin, Y. (2025). On prime numbers and quadratic forms represented by positive-definite, primitive quadratic forms. Journal of Number Theory, 274, 26-36. https://doi.org/10.1016/j.jnt.2024.12.014

28. Semchankau, A. (2025). A new bound for A(A + A) for large sets. Journal of Number Theory, 268, 142-162. https://doi.org/10.1016/j.jnt.2024.08.002
29. Mayle, J., & Wang, T. (2025). An effective open image theorem for products of principally polarized abelian varieties. Journal of Number Theory, 274, 140-179. https://doi.org/10.1016/j.jnt.2024.12.011
30. Mejail, M., Nestares, B. K., & Gravano, L. (2026). Reconfigurable VLSI Accelerator for Adaptive Wireless Signal Processing Applications. *Journal of Integrated VLSI and Signal Intelligence*, *1*(2), 47-54.
31. Rajan.C. (2026). Hardware–Algorithm Co-Design for Intelligent Signal Processing in Edge IoT Devices. *Journal of Integrated VLSI and Signal Intelligence*, *1*(2), 11–23.