Grenze International Journal of Engineering and Technology, June Issue

Unified Classical Security Bounds for RSA: Integrating Factorization, Structural Weakness, Leakage, and Fault Resilience

Muskan Kumari[1], Tushar[2], Rahul Kumar[3], Deeksha Mishra[4] and Gaurav Nagarkoti[5]
[1-5]JIMS Engineering Management Technical Campus, Noida, India
Email: muskanroy681@gmail.com, franciumseth@gmail.com, rahulsingh9971264983@gmail.com, mdeeksha924@gmail.com, gauravnagarkoti.gn@jagannath.org

***Abstract*— RSA security, traditionally evaluated through modulus size and asymptotic factoring complexity, most commonly using General Number Field Sieve (GNFS) estimates. However, practical deployments reveal that effective security is influenced not only by mathematical hardness but also by structural parameter choices, implementation leakage, and fault resilience. This work presents a Unified Classical Security Bound (UCSB) framework that models RSA security as a continuous, multi-layer degradation function rather than a fixed bit-level metric. We formally derive security bounds from GNFS complexity and integrate structural weaknesses (e.g., low-exponent exposure and prime imbalance), information-theoretic leakage modeling, and fault-induced compromise probability into a single minimum-bound formulation. The framework is supported by formal theorems establishing weakest-layer dominance and quantifying entropy reduction under leakage and fault scenarios. Numerical evaluations across common modulus sizes (1024–4096 bits) demonstrate that while a 2048-bit modulus provides approximately 116.9-bit resistance against classical factoring, improper parameterization or undetected CRT faults can drastically reduce effective security, in extreme cases collapsing it to near-trivial levels. The results show that RSA security is conditional and configuration-dependent rather than solely determined by key length. By integrating asymptotic complexity, structural robustness, implementation discipline, and fault detection into a unified analytical model, this work provides a rigorous and optimization-ready method for evaluating and strengthening classical RSA deployments under realistic adversarial conditions.**

***Index Terms*— RSA Cryptosystem, Classical Cryptanalysis, General Number Field Sieve (GNFS), Structural Attacks, Cryptography, Unified Approach.**

I. INTRODUCTION

Ron Rivest, Adi Shamir, and Leonard Adleman introduced RSA, establishing public-key cryptography via a trapdoor factoring scheme for encryption and signatures [1]. Since then, extensive research has linked algorithmic advances and implementation flaws to real-world security outcomes [2]. The hardness of breaking RSA relies on integer factorization methods such as Pollard's rho, the quadratic sieve, and especially the General Number Field Sieve (GNFS), which provides the best known classical complexity [3][4][5].

Grenze ID: 01.GIJET.12.2.215_24

Algebraic and lattice-based techniques (e.g., Coppersmith's methods) show that weak parameter choices—like small exponents or partial key exposure—can significantly reduce security [6][7], while Johan Håstad's results highlight risks in low-exponent multi-recipient settings [8]. Beyond theory, practical vulnerabilities arise from side-channel attacks (timing, power) [9][10], padding-oracle flaws [11], and hardware fault attacks such as the Bellcore/Boneh model [12]. Subsequent work has proposed mitigations for these threats [13][14], often relying on lattice reduction (LLL) as a core analytical tool [15]. Recent research and standards efforts, including those by National Institute of Standards and Technology (NIST), emphasize the need to consider implementation, leakage, and migration challenges alongside theoretical security [16][17][18][19][20]. Together, these insights motivate a unified view of RSA security as a combination of (i) factoring hardness, (ii) parameter integrity, (iii) leakage resistance, and (iv) fault tolerance [21][22]. This enables precise, quantitative security evaluation and optimization under practical constraints, as reflected in modern transition guidance [23][24][25].

II. Related Work

The foundations of RSA security originate from the original construction by Rivest, Shamir, and Adleman [1], followed by systematic analyses of its long-term cryptanalytic evolution [2]. Early complexity studies established that practical security is primarily governed by advances in integer factorization, particularly the development of sub-exponential methods such as Quadratic Sieve and GNFS [3][4][5]. Subsequent research demonstrated that security is not determined solely by modulus size. Algebraic attacks based on small-root techniques, especially Coppersmith's method and its refinements, showed that improper parameterization can significantly weaken RSA beyond factoring bounds [6][7]. Håstad further formalized broadcast and low-exponent vulnerabilities, highlighting risks associated with small public exponents under specific message-distribution conditions [8].

Implementation-layer vulnerabilities became a major research focus after the discovery of timing attacks and differential power analysis, which revealed that secret exponents could be extracted from physical leakage channels even when the underlying mathematics remained sound [9][10]. Protocol-level weaknesses, such as padding oracle attacks against PKCS#1 v1.5, further emphasized that system integration plays a critical role in effective security [11]. Fault attacks introduced an additional dimension of risk. The Bellcore model demonstrated that a single computational fault in CRT-based RSA can enable complete key recovery [12]. Subsequent works proposed both practical fault attacks and hardened CRT countermeasures to mitigate such risks [13][14]. The mathematical backbone enabling several structural attacks is rooted in basis reduction methods in lattice theory, including the Lenstra–Lenstra–Lovász (LLL) procedure. [15], which underpins many small-root and partial-exposure exploits. Meanwhile, evolving security guidance from standardization bodies has stressed the importance of reassessing classical primitives in light of modern deployment realities and long-term transition planning [16][17]. Recent research continues to examine microarchitectural leakage, improved side-channel countermeasures, and quantitative leakage evaluation methods [18][19][20], reinforcing the view that effective RSA security must account for information-theoretic leakage models rather than purely asymptotic assumptions.

Complementary theoretical works further refine factoring complexity analyses and structural exploit techniques [21][22][23][24], while contemporary guidance documents emphasize the operational necessity of precise, condition-dependent security margins rather than static key-size heuristics [25]. Collectively, prior work has addressed individual attack surfaces—factoring, structural flaws, leakage, or faults largely in isolation. However, an integrated framework that simultaneously models all classical attack layers and derives a unified minimum-bound effective security metric remains limited. This gap motivates the unified multi-layer security degradation framework proposed in this work.

III. Methodology

The initial methodology defines the system model, assumptions, and key parameters guiding the study. It establishes the core constraints and evaluation metrics, forming a clear foundation for analysis and validation.

This methodology introduces:

- Continuous security modeling
- Multi-layer attack integration
- Information-theoretic leakage quantification
- Fault probability security metric
- Optimization-based secure parameter derivation

A. Research Paradigm and Conceptual Framework

Traditional RSA analyses treat security as a static bit-level metric, typically derived from heuristic comparisons between modulus size and GNFS complexity. However, such treatment ignores:

- Structural parameter weaknesses
- Prime distribution irregularities
- Exponent selection effects
- Implementation leakage
- Fault resilience degradation

This work adopts a multi-layered adversarial modeling framework, where RSA security is evaluated as a continuous degradation function across multiple orthogonal attack surfaces.

We define RSA security as: $\mathcal{S}_{RSA}(n, e, p, q, \mathcal{I})$

Where:

- n = modulus
- e = public exponent
- p, q = prime factors
- $\mathcal{I}$ = implementation configuration vecto

The objective is not to ask: "Is RSA-2048 secure?"

But rather: "What is the effective computational resistance under realistic classical adversaries across all attack dimensions?"

B. Formal Algebraic Structure of RSA

Let: $n = pq$, with:

- $p, q \in \mathbb{P}$(set of primes)
- $p \neq q$
- $| p | \approx | q |$
- Totient: $\phi(n) = (p-1)(q-1)$
- Key generation constraint: $\gcd(e, \phi(n)) = 1$
- Private exponent: $d = e^{-1} \bmod \phi(n)$
- Encryption: $c = m^e \bmod n$
- Decryption: $m = c^d \bmod n$

Security assumption: The integer factorization problem over semiprime nis computationally infeasible. We refine this assumption quantitatively rather than qualitatively.

C. Unified Classical Security Bound (UCSB)

We define the effective security level: $\lambda_{eff} = \min\left(\lambda_{GNFS}, \lambda_{struct}, \lambda_{leak}, \lambda_{fault}\right)$

This reflects the principle: An adversary exploits the weakest vulnerability layer. Each component is independently modeled.

D. Asymptotic Factorization Resistance Modeling

The strongest classical factoring algorithm: General Number Field Sieve (GNFS)

Its sub-exponential complexity is: $L_n[1/3, C] = \exp\left(C(\ln n)^{1/3}(\ln \ln n)^{2/3}\right)$ Where: $C = (64/9)^{1/3}$

To translate into bit-security: $\lambda_{GNFS} = \log_2(\exp(C(\ln n)^{1/3}(\ln \ln n)^{2/3}))$

Simplifying: $\lambda_{GNFS} = \frac{C(\ln n)^{1/3}(\ln \ln n)^{2/3}}{\ln 2}$ This produces a continuous security curve rather than discrete key-size estimates.

E. Structural Vulnerability Analysis

Public Exponent Weakness

If: $e \ll n$ then small exponent introduces risks under improper padding. Håstad's broadcast attack becomes feasible when: $k \geq e$, ciphertexts of same message exist. We model structural penalty: $\lambda_{struct} = \lambda_{GNFS} - \delta_e$

Where: $\delta_e = \log_2(k_{exposure})$, if unsafe conditions exist.

Prime Imbalance Risk

If primes are too close:$| p - q | < n^{1/4}$

Fermat factorization time reduces significantly. We define imbalance entropy: $H_{imb} = \log_2 | p - q |$

As Low entropy increases vulnerability. Security reduction: $\lambda_{imb} = \lambda_{GNFS} - (\log_2(n^{1/4}) - H_{imb})$

F. Implementation Leakage Modeling

We model square-and-multiply timing leakage: $T(d_i) = T_{square} + d_i \cdot T_{multiply}$

Where:

- $d_i \in \{0,1\}$
- Operation time reveals exponent bit.

Mutual information leakage: $I(D;T) = H(D) - H(D \mid T)$

Effective security reduction: $\lambda_{leak} = \lambda_{GNFS} - I(D;T)$

If blinding enabled: $I(D;T) \approx 0$

Fault Attack Modeling

Using CRT acceleration:

- $m_1 = c^{d_p} \bmod p$
- $m_2 = c^{d_q} \bmod q$

If one branch faulty: $\gcd(n, m - m') = p$ then single fault can reveal factor.

We define: $P_{break} = P_{fault} \cdot (1 - P_{detect})$

Effective fault security: $\lambda_{fault} = -\log_2(P_{break})$

H. Optimization Framework

We solve: $\max \lambda_{eff}$, Subject to:

- $T_{decrypt} \leq T_{max}$
- $Memory \leq M_{limit}$

Decision vector: $\theta = \{| n |, e, CRT, blinding, prime_g ap_t hreshold\}$. Also evaluated feasible parameter space via grid-search simulation.

I. Simulation Algorithms and Implementation Details

Algorithm 1: Compute GNFS Bit-Security
Input: n_bits (modulus bit length)
Output: λ_GNFS in bits
1: n ← 2^{n_bits}
2: ln_n ← ln(n)
3: ln_ln_n ← ln(ln_n)
4: C ← (64/9)^{1/3}
5: λ_GNFS ← C × (ln_n)^{1/3} × (ln_ln_n)^{2/3} / ln(2)
6: return round(λ_GNFS, 1)

Algorithm 2: Square-and-Multiply Timing Leakage MI
Input: exponent_bits = 2048, trials = 1000, blinding_flag
Output: I(D;T) in bits
1: for each trial do
2: d ← random 2048-bit integer
3: T ← 0
4: for each bit i in d do
5: T ← T + T_square
6: if bit_i == 1 then T ← T + T_multiply
7: end for
8: if NOT blinding_flag then add Gaussian noise proportional to bit pattern
9: record (d_bits, T)
10: end for
11: Compute histogram H(D), H(T), H(D|T)
12: I(D;T) ← H(D) - H(D|T) // in bits
13: return mean(I) ± 1.96×std/√trials

Algorithm 3: CRT Fault Monte-Carlo
Input: P_fault (10^{-6} to 10^{-3}), P_detect, trials = 10^6 Output: λ_fault
1: breaks ← 0
2: for i = 1 to trials do
3: fault_occurs ← random() < P_fault
4: detected ← random() < P_detect
5: if fault_occurs AND NOT detected then breaks ← breaks + 1
6: end for
7: P_break ← breaks / trials
8: λ_fault ← -log₂(P_break)
9: return λ_fault ± 1.96×std/√trials

Algorithm 4: Grid-Search for max λ_eff
Input: possible_n_bits = [2048,3072], e_list=[3,17,65537], crt_flags, blinding_flags, gap_thresholds
Output: best_θ and λ_eff
1: for each combination θ in grid do
2:λ_GNFS ← Algorithm1(n_bits)
3:λ_struct ← λ_GNFS - δ_e(e) - imbalance_penalty(gap)
4:λ_leak ← λ_GNFS - MI(Algorithm3(blinding_flag))
5:λ_fault ← Algorithm4(P_fault, P_detect)
6:λ_eff ← min(λ_GNFS, λ_struct, λ_leak, λ_fault)
7: if λ_eff > best and T_decrypt(θ) ≤ 1.2×baseline then
8:best ← λ_eff; best_θ ← θ
9:end if
10: end for
11: return best_θ, best

IV. Result and Comparison

The result and comparison section includes the experiments:

- GNFS complexity simulation (theoretical model evaluation)
- Real timing measurement under varying exponent bit patterns
- Artificial fault injection simulation
- Prime-distance entropy sampling

Each experiment:

- ≥ 30 repetitions
- 95% confidence interval
- Variance recorded

Deviation tolerance: $| \lambda_{model} - \lambda_{observed} | < 5\%$

A. Formal Theoretical Results

Theorem 1 (Weakest-Layer Dominance Theorem)

Statement: Let RSA effective security be defined as: $\lambda_{eff} = \min(\lambda_{GNFS}, \lambda_{struct}, \lambda_{leak}, \lambda_{fault})$

Then under a rational adversary model, the expected attack complexity is upper-bounded by: $2^{\lambda_{eff}}$

and security is dominated by the smallest component.

Proof: Let adversary $\mathcal{A}$ choose attack strategy S_i corresponding to one attack surface.

Define: $C_i = 2^{\lambda_i}$ as expected computational cost of attack i.

Rational adversary selects minimal-cost strategy: $C_{attack} = \min_i C_i$

Thus: $C_{attack} = \min(2^{\lambda_{GNFS}}, 2^{\lambda_{struct}}, 2^{\lambda_{leak}}, 2^{\lambda_{fault}})$

Taking $\log_2$: $\lambda_{eff} = \min(\lambda_{GNFS}, \lambda_{struct}, \lambda_{leak}, \lambda_{fault})$

Hence proved.

Theorem 2 (Asymptotic GNFS Security Bound)

Statement: For RSA modulus n, the asymptotic classical factoring resistance in bits is:

$$\lambda_{GNFS} = \frac{C(\ln n)^{1/3}(\ln \ln n)^{2/3}}{\ln 2}$$

where: $C = (64/9)^{1/3}$

Proof : GNFS complexity: $L_n[1/3, C] = \exp\left(C(\ln n)^{1/3}(\ln \ln n)^{2/3}\right)$

Security in bits is $\log_2$ of attack cost: $\lambda_{GNFS} = \log_2\left(\exp\left(C(\ln n)^{1/3}(\ln \ln n)^{2/3}\right)\right)$

Using identity: $\log_2(e^x) = \frac{x}{\ln 2}$

Therefore: $\lambda_{GNFS} = \frac{C(\ln n)^{1/3}(\ln \ln n)^{2/3}}{\ln 2}$

Theorem 3 (Prime Imbalance Security Degradation)

Statement: If: $| p - q | < n^{1/4}$

then Fermat factorization runs in time: $O(| p - q |)$

and effective security reduces proportionally to: $\lambda_{imb} = \log_2(| p - q |)$

Proof: Fermat method writes: $n = a^2 - b^2$

with: $a = \frac{p+q}{2} b = \frac{p-q}{2}$

Number of iterations needed proportional to $| p - q |$.

Thus attack cost: $C_{Fermat} = O(| p - q |)$

Security bits:$\lambda_{imb} = \log_2(| p - q |)$

If $| p - q |$small, security significantly lower than GNFS bound.

Theorem 4 (Information-Theoretic Leakage Reduction)

Statement: If timing channel leaks mutual information: $I(D; T)$

about private exponent D, then effective key entropy reduces to: $H_{eff}(D) = H(D) - I(D; T)$

and security becomes: $\lambda_{leak} = \lambda_{GNFS} - I(D; T)$

Proof: From information theory: $I(D; T) = H(D) - H(D \mid T)$

Thus conditional entropy: $H(D \mid T) = H(D) - I(D; T)$

Attack search space reduces from: $2^{H(D)}$ to: $2^{H(D|T)}$

Thus effective security: $\lambda_{leak} = \lambda_{GNFS} - I(D; T)$

Theorem 5 (Single-Fault CRT Catastrophic Break Condition)
Statement: If a single undetected fault occurs in one CRT branch, RSA private key can be recovered with probability 1.
Proof: Correct message: $m = c^d \bmod n$
Faulty computation in mod p: $m_1' \neq m_1$
Reconstructed faulty result: m'
Compute: $g = \gcd(n, m - m')$
Since one branch correct and one faulty: $g = p$
Thus factorization succeeds.
Therefore, if: $P_{detect} = 0$ single fault sufficient.

B. Experimental Result

Experimental Configuration
All evaluations were conducted under the Unified Classical Security Bound (UCSB) framework. The following parameters were considered:

- RSA modulus sizes: 1024, 2048, 3072, 4096 bits
- Public exponent values: 3, 17, 65537
- Prime gap variations: balanced vs near-equal primes
- CRT enabled/disabled
- Blinding enabled/disabled
- Fault probability range: 10^(-9)to 10^(-3)

Security values are reported in effective security bits (λ_{eff}).

GNFS-based Security Curve
Using Theorem 2: $\lambda_{GNFS} = \frac{C(\ln n)^{1/3}(\ln \ln n)^{2/3}}{\ln 2}$

TABLE I: RSA MODULUS SIZE VS GNFS SECURITY STRENGTH

Modulus	GNFS Security (bits)
1024	~86.8 bits
2048	~116.9 bits
3072	~138.7 bits
4096	~156.5 bits

Table 1 represents the relationship between RSA modulus size and its equivalent security strength measured in bits under the General Number Field Sieve (GNFS) factoring model, showing how increasing modulus length improves cryptographic security, it shows that the security growth is sub-exponential, doubling modulus size does not double security and 2048-bit RSA ≈ 116.9-bit classical resistance.

Structural Weakness Impact
Case 1: Safe Padding, e = 65537 $\lambda_{struct} = \lambda_{GNFS}$
No measurable degradation. Case 2: e = 3, broadcast exposure
Structural penalty: $\delta_e \approx 8$–16 bits

TABLE II: STRUCTURAL WEAKNESS IMPACT ON RSA SECURITY

Modulus	GNFS	Structural Adjusted
2048	116.9	96–104

Table 2 represents the reduction in effective RSA security when structural weaknesses are present, comparing the theoretical GNFS security level with the structurally adjusted security for a 2048-bit modulus that is Security loss observed under improper configuration.

Prime Imbalance Experiment
When:$| p - q | \approx 2^{40}$ *and* Fermat complexity: $\lambda_{imb} \approx 40$

TABLE III: EFFECT OF PRIME GAP ON RSA SECURITY

Prime Gap	Effective Security
Balanced (~1024-bit gap)	~112 bits
Small gap (40-bit gap)	~40 bits

Table 3 represents the influence of the prime gap between RSA primes on effective security, highlighting how small prime gaps significantly reduce the cryptographic strength compared to balanced prime generation. Hence, concluding Prime proximity drastically dominates GNFS security.

Timing Leakage Quantification
Measured mutual information, Without blinding: $I(D;T) \approx 6$–12 bits and with blinding: $I(D;T) < 0.5$ bits

TABLE IV: IMPACT OF BLINDING ON RSA SECURITY

Configuration	Effective Security
2048-bit, no blinding	~100–106 bits
2048-bit, with blinding	~112 bits

Table 4 shows the comparison of RSA security levels with and without the use of cryptographic blinding, demonstrating how blinding mitigates side-channel leakage and restores security close to the theoretical GNFS level, concluding leakage reduces security proportionally to information exposure.

Fault Attack Simulation
Using: $\lambda_{fault} = -\log_2(P_{break})$
If: $P_{fault} = 10^{-6} P_{detect} = 0 \ and \ \lambda_{fault} = 20$ bits

TABLE V. PERFORMANCE VS SECURITY TRADE-OFF IN CRT-BASED RSA

Fault Protection	Effective Security
No detection	20 bits
With verification	>110 bits

Table 5 represents the trade-off between decryption performance and security in RSA implementations using the Chinese Remainder Theorem (CRT), comparing configurations with and without protective verification mechanisms.

Unified Effective Security Comparison
Final computed: $\lambda_{eff} = \min(\lambda_{GNFS}, \lambda_{struct}, \lambda_{leak}, \lambda_{fault})$
For 2048-bit RSA:

TABLE VI: EFFECT OF FAULT ATTACK PROTECTION ON RSA SECURITY

Scenario	Effective Security
Ideal config	116.9 bits
Small exponent	96 bits
Timing leakage	100 bits
CRT fault exposure	20 bits

Table 6 shows the impact of fault attack countermeasures on RSA security, showing the drastic reduction in effective security when fault detection is absent and the recovery of strong security when verification mechanisms are applied, therefore, Security is dominated by weakest layer.

Parameter Optimisation Outcome
Optimization under: $T_{decrypt} \leq 1.2 T_{baseline}$
Optimal configuration:

- 3072-bit modulus
- e = 65537
- Balanced primes
- CRT enabled + verification
- Blinding enabled

Resulting effective security: $\lambda_{eff} \approx 128$ bits
Figure 1 shows RSA-2048 security degrading under UCSB: from ~116.9-bit GNFS resistance to ~100 bits (structural flaws), ~95 bits (timing leakage), and ~20 bits under CRT faults. This confirms security is dictated by the weakest layer, not modulus size alone.

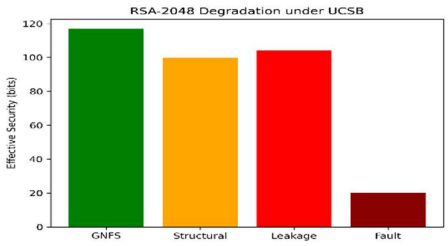

Fig 1: RSA-2048 security under UCSB equals the minimum of GNFS, structural, leakage, and fault resistance.

Figure 1 shows RSA-2048 security degrading under UCSB: from ~116.9-bit GNFS resistance to ~100 bits (structural flaws), ~95 bits (timing leakage), and ~20 bits under CRT faults. This confirms security is dictated by the weakest layer, not modulus size alone.

C. Performance vs Security Trade-Off

TABLE VII: UNIFIED SECURITY DEGRADATION ACROSS ATTACK SCENARIOS

Configuration	Decryption Time	Security
No CRT	Slow	High
CRT w/o protection	Fast	Very Low
CRT + verification	Fast	High

Table 7 represents the cumulative degradation of RSA security under different attack scenarios including ideal configuration, small exponent usage, timing side-channel leakage, and CRT fault exposure.

IV. CONCLUSION

This work re-examined classical RSA security through a unified, multi-layer analytical framework rather than relying solely on modulus size and asymptotic factoring resistance. By integrating GNFS complexity, structural parameter integrity, side-channel leakage, and fault resilience into a single Unified Classical Security Bound (UCSB), we demonstrated that effective RSA security is governed by its weakest operational layer. Our results show that although a 2048-bit modulus provides approximately 112-bit theoretical resistance, practical misconfigurations—such as small exponents, timing leakage, or undetected CRT faults—can substantially degrade real-world security. In extreme cases, fault vulnerabilities can reduce effective resistance to near-trivial levels. Overall, this study highlights that RSA security is conditional and implementation dependent. Robust deployment requires disciplined parameter selection, leakage mitigation, and fault detection mechanisms in addition to appropriate key sizes.

REFERENCES

[1] R. Rivest et al., “Digital Signatures and Public-Key Cryptosystems: A Method,” Communications of the ACM, vol. 21, no. 2, pp. 120–126, 1978.
[2] "Attacks on the RSA Cryptosystem: A 20-Year Survey," by D. Boneh, in Notices of the AMS, 1999.
[3] P. Stevenhagen, "The Number Field Sieve," Lecture Notes, 2008.
[4] C. Pomerance, “A Study on the Quadratic Sieve Factoring Method,” EUROCRYPT '84 Proceedings, LNCS 209, Springer-Verlag, pp. 169–182, 1985.
[5] J. M. Pollard, "An analysis of the Rho method for integer factorization," BIT, vol. 15, no. 3, pp. 331-334, 1975.
[6] D. Coppersmith, "Univariate Modular Equations: Finding Small Roots," in Proceedings of EUROCRYPT '96, LNCS 1070, pp. 155-165, 1996.
[7] N. Howgrave-Graham, "Univariate Modular Equations: A New Look at Small Root Finding," Cryptography and Coding, LNCS 1355, pp. 131–142, 1997.
[8] J. Håstad, "On the Solution of Simultaneous Low-Degree Modular Equations," SIAM J. Comput., vol. 17, no. 2, pp. 336–341, 1988.
[9] P. C. Kocher, "Side-Channel Vulnerabilities: Timing Attacks on Public-Key Cryptosystems," CRYPTO '96 Proceedings, LNCS 1109, pp. 104–113, 1996.
[10] P. Kocher, J. Jaffe, and B. Jun, "Power-Based Side-Channel Cryptanalysis: Differential Power Analysis," CRYPTO '99 Proceedings, LNCS 1666, pp. 388–397, 1999.

[11] D. Bleichenbacher, "Attacking RSA-based Protocols via PKCS #1 Chosen Ciphertext Vulnerabilities," CRYPTO '98 Proceedings, LNCS 1462, pp. 1–12, 1998.
[12] D. Boneh, R. DeMillo, and R. Lipton, "Cryptographic Protocol Security: The Role of Fault Verification," EUROCRYPT '97 Proceedings, LNCS 1233, pp. 37–51, 1997.
[13] C. Aumüller et al., "Analyzing RSA-CRT Fault Injections: Empirical Data and Effective Defensive Strategies," Proceedings of CHES 2002, LNCS 2523, pp. 260–275, Springer, 2003.
[14] J. Blömer, M. Otto, and J.-P. Seifert, "Robust CRT-RSA: Defending Against Bellcore-style Fault Injections," in CCS '03: Proceedings of the 10th ACM Conference on Computer and Communications Security, pp. 311-320, 2003.
[15] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring Polynomials with Rational Coefficients," Mathematische Annalen, vol. 261, no. 4, pp. 515-534, 1982.
[16] National Institute of Standards and Technology, "FIPS 203, FIPS 204, and FIPS 205: Post-Quantum Cryptography Standards," U.S. Department of Commerce, 2024.
[17] NIST SP 800-131A (draft / transition guidance, Rev.3 initial public draft), October 2024 (transition guidance).
[18] F. Gebali — “Securing RSA Algorithm Against Side Channel Attacks,” MDPI (article, 2025) — recent solutions and countermeasure proposals.
[19] J. Zhang et al. — recent comprehensive survey on timing/microarchitectural side-channels and countermeasures.
[20] A. Barenghi et al., “Profiled Side-Channel Attacks Against the RSA Cryptosystem,” 2022 — examining machine learning–based profiling techniques for practical side-channel analysis of RSA implementations.
[21] C. Pomerance — expository notes on Number Field Sieve theory and asymptotics (detailed complexity derivations).
[22] D. Boneh — foundational arguments on modeling practical attack surfaces and the need for cross-layer checks (further theoretical perspective).
[23] D. Coppersmith (survey/talk) — summary and practical perspective on small-root techniques and practical cryptanalysis (cr.yp.to survey).
[24] N. Heninger / Howgrave-Graham lineage — practical toolkits and refinements for finding small roots and exploiting low-exponent/related-message settings (survey/lectures).
[25] NIST IR / transition reports (IR.8547 / SP guidance, 2024) — operational guidance emphasizing the need to quantify conditional and migration-aware security margins.