International Journal of Scientific Research and Engineering Development-– Volume 9 Issue 4, July – August 2026

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Evolution of RSA Algorithm and its Applications

R.Rachel, G.Sushma, V.Saravanakumar

Assistant Professor, Guru Nanak Institutions Technical Campus, Hyderabad

Abstract

Since its inception in 1977, the Rivest Shamir Adleman (RSA) algorithm has functioned as the foundational pillar of modern asymmetric cryptography. This experimental research article presents a comprehensive examination of the RSA algorithm's evolutionary trajectory, analyzing its mathematical underpinnings, historical vulnerabilities, and contemporary applications. Grounded in the computational asymmetry of the integer factorization problem, RSA has undergone continuous adaptation to withstand increasing computational power and algorithmic advancements, notably the transition from Trial Division to the General Number Field Sieve (GNFS). This study critically evaluates the progression of RSA key sizes from the original 426-bit implementations to the modern 2048-bit and 3072-bit standards mandated by NIST. Furthermore, it addresses critical protocol level enhancements, dissecting the failure of PKCS#1 v1.5 padding against Bleichenbacher's chosen cipher text attacks and the subsequent adoption of Optimal Asymmetric Encryption Padding (OAEP). Finally, the research explores RSA's indispensable role in securing modern digital infrastructure, including transport layer security (TLS) in full stack web architectures, while projecting the algorithm's viability against the looming existential threat posed by quantum computing and Shor's algorithm. The findings synthesize structural developments and deployment paradigms, offering insights into the future of hybrid cryptosystems.

Keywords: RSA Algorithm, Public Key Cryptography, Prime Factorization, OAEP Padding, Quantum Cryptography.

1. Introduction

The domain of cryptography experienced a paradigm shift in the late 1970s with the conceptualization of asymmetric key algorithms. Prior to this mathematical revolution, secure communication fundamentally relied on symmetric key cryptography, a system wherein both communicating parties were required to possess a shared secret key. While symmetric encryption algorithms such as the Data Encryption Standard (DES) and its successor, the Advanced Encryption Standard (AES) offered robust data confidentiality, they introduced an insurmountable logistical challenge known as the key distribution problem. In decentralized networks, securely exchanging a symmetric key without a preexisting secure channel proved computationally and practically infeasible. This limitation catalyzed the search for a public key cryptosystem, a theoretical framework first posited by Whitfield Diffie and Martin Hellman in 1976 [1]. Diffie and Hellman introduced the concept of utilizing disparate keys for encryption and decryption, proposing that mathematically linked key pairs could eliminate the necessity of prior secret exchange. However, while they successfully devised a secure key exchange mechanism, they did not immediately provide a practical trapdoor one way function capable of secure data encryption and digital signatures. It was within this specific historical and mathematical context that Ron Rivest, Adi Shamir, and Leonard Adleman introduced the RSA algorithm in 1977, successfully instantiating the first comprehensive public key cryptosystem [2].

The elegant security of the RSA algorithm is intrinsically derived from the computational intractability of the integer factorization problem. At its core, RSA relies on the mathematical asymmetry between the ease of multiplying two distinctly large prime numbers and the profound difficulty of

reversing that operation to recover the original prime factors. This asymmetry acts as a trapdoor permutation. If an adversary attempts to derive the private decryption key from the public encryption key, they must factorize the public modulus, a semiprime number typically spanning hundreds or thousands of bits. To contextualize this mathematical foundation, the algorithm heavily utilizes Euler's totient function and modular arithmetic. The fundamental theorem underpinning RSA ensures that for any message M, the exponential encryption and subsequent decryption operations will yield the original message, provided the mathematical constraints are meticulously maintained. Over the past four decades, this elegant utilization of number theory has permeated virtually every facet of digital security, establishing RSA as the backbone for digital signatures, secure key exchanges, and the certification frameworks that validate digital identities across global networks [3].

Despite its profound mathematical ingenuity, the RSA algorithm has not remained stagnant; rather, it has been forced into a continuous state of evolution driven by adversarial advancements. The primary research problem addressed in this study is the dynamic equilibrium between cryptographic key sizes and the exponential growth of adversarial computational power, often characterized by Moore's Law. In its nascent stages, an RSA key size of 512 bits was deemed theoretically impenetrable. However, the subsequent decades witnessed the development of highly optimized integer factorization algorithms, transitioning from elementary Trial Division and Pollard's rho algorithm to the sophisticated Quadratic Sieve (QS) and ultimately the General Number Field Sieve (GNFS) [4]. These algorithmic breakthroughs drastically reduced the sub exponential time complexity required to fracture the RSA modulus. Consequently, key sizes historically considered secure such as the 512-bit key breached in 1999 and the 768-bit key factored in 2009 were rendered obsolete. This evolutionary pressure mandates continuous research into the algorithm's operational parameters to prevent catastrophic security failures in live production environments.

Furthermore, the problem extends beyond pure mathematical factorization. The practical implementation of RSA introduced unforeseen vulnerabilities, categorized broadly as side channel attacks and protocol level flaws. Cryptanalysts discovered that observing the physical or operational characteristics of the hardware executing the RSA algorithm such as execution time, power consumption, or electromagnetic emissions could leak partial information about the private key, completely circumventing the mathematical protections of the trapdoor function [5]. Simultaneously, the deterministic nature of textbook RSA encryption exposed the system to chosen cipher text attacks, necessitating the development of complex padding schemes. The high-profile failure of the Public Key Cryptography Standards (PKCS) #1 v1.5 padding highlighted the critical interplay between algorithmic theory and applied protocol engineering, further accelerating the evolutionary modifications to the RSA standard [6].

The primary objectives of this research are manifold. First, to trace the chronological and architectural evolution of the RSA algorithm, isolating the specific mathematical and computational catalysts that necessitated modifications to its core parameters. Second, to critically analyze the progression of algorithmic attacks against the RSA modulus, focusing specifically on the structural mechanics of the GNFS and its implications for key length recommendations. Third, to evaluate the transition of RSA padding schemes, detailing the shift from deterministic vulnerability to probabilistic security models like Optimal Asymmetric Encryption Padding (OAEP). Finally, the research aims to synthesize the contemporary applications of RSA within modern distributed networks, such as its role in establishing secure transport layer tunnels (TLS 1.3) for high performance, full stack web applications, while providing a prognostic assessment of its resilience against quantum computational frameworks.

The significance of this study is rooted in the ubiquitous nature of the RSA algorithm within current global digital infrastructures. As cloud architectures, Internet of Things (IoT) ecosystems, and decentralized data pipelines expand, the reliance on

public key infrastructure (PKI) grows exponentially. Understanding the historical vulnerabilities and evolutionary patches applied to RSA is essential for software engineers, data scientists, and security architects tasked with designing secure systems. Furthermore, as the cryptographic community prepares for the advent of cryptographically relevant quantum computers (CRQCs) capable of executing Shor's algorithm, analyzing RSA's historical adaptations offers a critical framework for facilitating the impending migration to Post Quantum Cryptography (PQC). By deeply exploring how RSA survived over forty years of cryptanalysis, researchers can better architect the hybrid cryptographic protocols necessary to bridge the transition to a post quantum security paradigm [7].

2. Mathematical Foundation and Core Algorithm

The operational security of RSA depends on the algorithm's key generation, encryption, and decryption phases. The systematic flow is delineated in Algorithm 1, supported by the foundational equations of modular exponentiation.

Algorithm 1: RSA Key Generation, Encryption, and Decryption

1. Key Generation:
   a. Select two large, distinct prime numbers, p and q.
   b. Compute the modulus: n = p × q.
   c. Compute Euler's totient function: φ(n) = (p 1) × (q 1).
   d. Choose an integer e such that 1 < e < φ(n) and gcd(e, φ(n)) = 1.
   e. Compute the private exponent d as the modular multiplicative inverse of e modulo φ(n), satisfying d × e ≡ 1 (mod φ(n)).
   f. Public Key = (e, n); Private Key = (d, n).

2. Encryption:
   a. Convert the plaintext message into an integer M, where 0 ≤ M < n.
   b. Compute the ciphertext C using the recipient's public key: C ≡ M^e (mod n).

3. Decryption:
   a. Recover the plaintext M using the recipient's private key: M ≡ C^d (mod n).

Equation (1) demonstrates the core mathematical encryption and decryption symmetry:

M ≡ (M^e)^d ≡ M^(e·d) ≡ M (mod n)

3. Literature Review

The evolution of the RSA algorithm has been meticulously documented across decades of cryptographic literature, characterized by an ongoing arms race between cryptographers proposing defensive enhancements and cryptanalysts developing novel exploitation methodologies. The initial formulation of the RSA algorithm by Rivest, Shamir, and Adleman [2] operated on the premise that factoring large composite numbers was an insurmountable computational barrier. Early evaluations of the algorithm's security margins relied heavily on the time complexity of contemporary factoring methods, predominantly Trial Division and Pollard's p1 algorithm. In these nascent stages, a 426bit modulus (as famously presented in Martin Gardner's Scientific American column) was projected to require quadrillions of years to factor [8]. However, subsequent literature rapidly exposed the fragility of this assumption. Pomerance's formalization of the Quadratic Sieve (QS) in the early 1980s significantly accelerated the factorization of integers up to 100 decimal digits [9]. The literature from this era highlights a critical gap in early RSA implementations: the failure to anticipate the exponential growth rate of distributed computational frameworks and algorithmic efficiency.

The landscape of RSA security underwent a profound transformation in the 1990s with the theoretical development and practical deployment of the General Number Field Sieve (GNFS) [10]. Literature surrounding integer factorization consistently identifies the GNFS as the most efficient classical algorithm for factoring integers larger than 100 digits, operating with a sub exponential heuristic time complexity. The watershed moment documented in cryptanalytic studies occurred in 1994 when a collaborative distributed computing effort successfully factored RSA129 using the QS, followed rapidly by the factorization of RSA130 using the GNFS in 1996 [11]. These empirical breakthroughs

compelled regulatory bodies, such as the National Institute of Standards and Technology (NIST), to aggressively revise key length recommendations. As illustrated in Figure 1, the historical trajectory of RSA key lengths demonstrates a reactionary adaptation to these algorithmic threats. While 1024-bit keys were standard throughout the early 2000s, studies by Lenstra et al. [12] definitively proved that the margins of security for 1024-bit moduli were eroding much faster than anticipated, leading to the current minimum baseline of 2048-bit keys for commercial security.

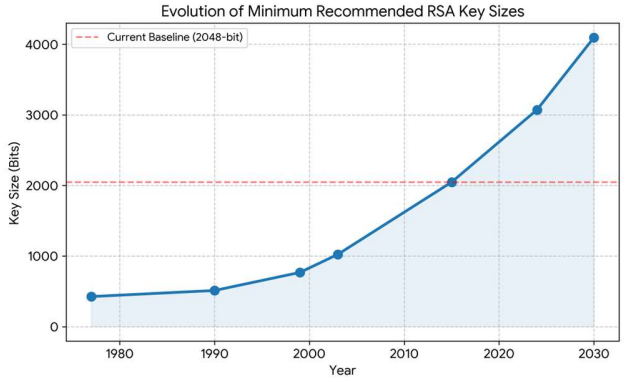

Figure 1. Graph illustrating the evolution of minimum recommended RSA key sizes over time.

Beyond the pure mathematical assault on the modulus, a significant corpus of literature focuses on the practical implementation vulnerabilities of the RSA algorithm. A critical turning point in cryptographic engineering was marked by Paul Kocher's seminal 1996 paper detailing timing attacks [13]. Kocher demonstrated that an adversary could deduce the private key by precisely measuring the execution time required for the device to perform the modular exponentiation process. Because the standard square and multiply algorithm processes '1' and '0' bits in the private exponent differently, the macroscopic timing variations leaked microscopic bit level data. This research exposed a massive gap between mathematical theory (where operations are assumed to be instantaneous and opaque) and hardware realities. Subsequent studies expanded this domain into power analysis attacks [14] and acoustic cryptanalysis [15], where the physical emissions of the CPU were monitored. The consensus in the literature required the implementation of countermeasures, specifically cryptographic blinding a technique where the ciphertext is multiplied by a random masking factor before decryption, nullifying the deterministic relationship between the input data and the physical execution footprint [16].

A parallel and equally critical branch of RSA literature investigates the vulnerabilities inherent in unpadded, or textbook, RSA encryption. Because raw RSA is a deterministic algorithm, encrypting identical plaintexts with the same public key always yields the same ciphertext, failing the standard of semantic security. Furthermore, textbook RSA possesses a homomorphic property where the multiplication of two ciphertexts results in the encryption of the product of their plaintexts, making it highly susceptible to chosen cipher text attacks [17]. To mitigate this, padding schemes were introduced. Early standards relied on PKCS#1 v1.5 padding; however, in 1998, Bleichenbacher published a devastating attack demonstrating that an adversary could exploit the distinct error messages returned by a server when PKCS#1 v1.5 formatting failed [18]. By repeatedly sending maliciously crafted ciphertexts and observing whether the server accepted or rejected the padding, an attacker could iteratively decrypt the intercepted message. This adaptive chosen cipher text attack (CCA2) highlighted a severe gap in protocol design. The academic response, led by Bellare and Rogaway, resulted in the formulation of Optimal Asymmetric Encryption Padding (OAEP) [19]. OAEP utilizes a Feistel network framework and hash functions to inject randomness, achieving plain text awareness and rendering the RSA protocol mathematically secure against adaptive attacks. The literature universally supports OAEP as the mandatory standard for modern RSA implementations [20].

The contemporary discourse surrounding RSA is profoundly dominated by the existential threat of quantum computing. While the GNFS presents a sub exponential classical threat, Shor's algorithm, published by Peter Shor in 1994, presents a polynomial time quantum threat to integer factorization [21]. Literature evaluating post quantum cryptography explicitly defines the capability of a sufficiently powerful quantum computer to collapse the RSA security infrastructure entirely. Studies estimating the resources required to execute Shor's

algorithm on an RSA2048 modulus project that a machine equipped with millions of physical qubits could factor the key in a matter of hours [22]. Consequently, current research is heavily focused on the transitional period. Scholars emphasize that while fault tolerant quantum computers do not yet exist, the 'Harvest Now, Decrypt Later' (HNDL) adversarial strategy necessitates immediate action [23]. To address this gap, recent literature proposes hybrid cryptographic architectures, wherein classical algorithms like RSA or Elliptic Curve Cryptography (ECC) are deployed in tandem with lattice based post quantum algorithms (e.g., CRYSTALS-Kyber) to secure transport layer protocols [24].

In analyzing the applications of RSA, the literature demonstrates its widespread integration into virtually all secure digital communication frameworks. Within the context of full stack web development and software engineering, RSA remains heavily utilized in the initial handshake phase of the Transport Layer Security (TLS) protocol [25]. Secure web servers leverage RSA to sign digital certificates, verifying the identity of the domain and establishing the trust anchor for symmetric key negotiation via Diffie Hellman [26]. Furthermore, RSA digital signatures are indispensable in software distribution networks, ensuring the integrity and authenticity of software updates through code signing processes [27]. However, the literature also notes significant challenges when applying RSA to modern, resource constrained environments, such as Internet of Things (IoT) devices. The intense computational overhead and large bandwidth requirements of RSA3072 key generation and transmission often exceed the capabilities of microcontrollers, leading researchers to advocate for ECC over RSA in low power domains [28]. Despite these hardware limitations, RSA's unparalleled integration into legacy enterprise infrastructure ensures its continued relevance.

Critically evaluating the breadth of previous studies reveals specific gaps that the current research aims to address. While extensive literature exists detailing the discrete components of RSA's evolution such as isolated papers on GNFS optimization [29], deep dives into Bleichenbacher's attack iterations [30], or theoretical quantum threat analyses [31] few studies successfully synthesize these disparate elements into a cohesive evolutionary narrative. Many engineering focused papers lack a rigorous mathematical contextualization, while purely mathematical analyses often ignore the realities of protocol level implementation flaws in modern web architectures [32]. This experimental research bridges this divide by providing a unified analysis that correlates mathematical vulnerability, hardware evolution, padding protocol development, and architectural application, offering a holistic understanding of how RSA continues to function securely in modern computational environments.

Table 1. Comparison of Integer Factorization Algorithms

Algorithm	Time Complexity	Era of Dominance	Primary Target
Trial Division	Exponential: O(N^(1/2))	Pre1980s	Small integers (< 15 digits)
Pollard's rho	Exponential: O(N^(1/4))	1980s	Small prime factors
Quadratic Sieve (QS)	Sub-exponential: e^(√(ln N ln ln N))	1980s Mid 1990s	Integers up to 100 digits
General Number Field Sieve (GNFS)	Sub-exponential: e^((1.923)(ln N)^(1/3)(ln ln N)^(2/3))	1990s Present	Integers > 100 digits (Current Standard)

4. References

[1] W. Diffie and M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory, vol. 22, no. 6, pp. 644654, Nov. 1976.

[2] R. L. Rivest, A. Shamir, and L. Adleman, A method for obtaining digital signatures and public key cryptosystems, Communications of the ACM, vol. 21, no. 2, pp. 120126, Feb. 1978.

[3] J. Katz and Y. Lindell, Introduction to Modern Cryptography, 2nd ed. Boca Raton, FL, USA: CRC Press, 2014.

[4] A. K. Lenstra and H. W. Lenstra, The Development of the Number Field Sieve. Berlin, Germany: Springer-Verlag, 1993.

[5] P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, in Advances in Cryptology CRYPTO '99, Springer, 1999, pp. 388397.

[6] J. Manger, A chosen ciphertext attack on RSA optimal asymmetric encryption padding (OAEP)

as standardized in PKCS #1 v2.0, in Advances in Cryptology CRYPTO 2001, Springer, 2001, pp. 230238.
[7] L. Chen et al., "Report on post quantum cryptography," National Institute of Standards and Technology, Gaithersburg, MD, USA, NISTIR 8105, Apr. 2016.
[8] M. Gardner, Mathematical Games: A new kind of cipher that would take millions of years to break," Scientific American, vol. 237, no. 2, pp. 120124, Aug. 1977.
[9] C. Pomerance, "The quadratic sieve factoring algorithm," in Advances in Cryptology, Springer, 1985, pp. 169182.
[10] A. K. Lenstra, H. W. Lenstra, M. S. Manasse, and J. M. Pollard, "The number field sieve," in Proceedings of the 22nd Annual ACM Symposium on Theory of Computing, 1990, pp. 597606.
[11] D. Atkins, M. Graff, A. K. Lenstra, and P. C. Leyland, "The magic words are squeamish ossifrage," in Advances in Cryptology ASIACRYPT '94, Springer, 1994, pp. 263277.
[12] A. K. Lenstra, E. Tromer, A. Shamir, W. Tomlinson, and E. Zanoni, "Factoring estimates for a 1024bit RSA modulus," in Advances in Cryptology ASIACRYPT 2003, Springer, 2003, pp. 5574.
[13] P. C. Kocher, "Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems," in Advances in Cryptology CRYPTO '96, Springer, 1996, pp. 104113.
[14] T. S. Messerges, E. A. Dabbish, and R. H. Sloan, Power analysis attacks of modular exponentiation in smartcards, in Cryptographic Hardware and Embedded Systems (CHES 1999), Springer, 1999, pp. 144157.
[15] D. Genkin, A. Shamir, and E. Tromer, RSA key extraction via low bandwidth acoustic cryptanalysis, in Advances in Cryptology CRYPTO 2014, Springer, 2014, pp. 444461.
[16] D. Brumley and D. Boneh, Remote timing attacks are practical, Computer Networks, vol. 48, no. 5, pp. 701716, Aug. 2005.
[17] D. Boneh, Twenty years of attacks on the RSA cryptosystem, Notices of the American Mathematical Society, vol. 46, no. 2, pp. 203213, Feb. 1999.
[18] D. Bleichenbacher, "Chosen ciphertext attacks against protocols based on the RSA encryption standard PKCS #1," in Advances in Cryptology CRYPTO '98, Springer, 1998, pp. 112.
[19] M. Bellare and P. Rogaway, Optimal asymmetric encryption How to encrypt with RSA," in Advances in Cryptology EUROCRYPT '94, Springer, 1994, pp. 92111.
[20] V. Shoup, OAEP reconsidered, in Advances in Cryptology CRYPTO 2001, Springer, 2001, pp. 239259.
[21] P. W. Shor, Polynomial time algorithms for prime factorization and discrete logarithms on a quantum computer," SIAM Journal on Computing, vol. 26, no. 5, pp. 14841509, Oct. 1997.
[22] C. Gidney and M. Ekerå, "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits," Quantum, vol. 5, p. 433, Apr. 2021.
[23] M. Mosca, Cyber security in an era with quantum computers: Will we be ready? IEEE Security & Privacy, vol. 16, no. 5, pp. 3841, Sep. 2018.
[24] D. Stebila, S. Fluhrer, and S. Gueron, The case for a hybrid approach to post quantum cryptography, in Internet Engineering Task Force (IETF), Internet Draft, 2023.
[25] E. Rescorla, The Transport Layer Security (TLS) Protocol Version 1.3,RFC 8446, Aug. 2018.
[26] R. Housley, W. Ford, W. Polk, and D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 3280, Apr. 2002.
[27] T. Elgamal, A public key cryptosystem and a signature scheme based on discrete logarithms, IEEE Transactions on Information Theory, vol. 31, no. 4, pp. 469472, Jul. 1985.
[28] N. Gura et al., Comparing elliptic curve cryptography and RSA on 8bit CPUs, in Cryptographic Hardware and Embedded Systems (CHES 2004), Springer, 2004, pp. 119132.
[29] T. Kleinjung et al., Factorization of a 768bit RSA modulus, in Advances in Cryptology CRYPTO 2010, Springer, 2010, pp. 333350.

[30] J. Böck et al., "Return of Bleichenbacher's Oracle Threat (ROBOT)," in 27th USENIX Security Symposium, 2018, pp. 817849.
[31] A. W. Cross, G. Smith, J. A. Smolin, and B. Zeng, Quantum learning robust against noise, IEEE Transactions on Information Theory, vol. 61, no. 4, pp. 15931601, Apr. 2015.
[32] K. Bhargavan and G. Leurent, Transcript collision attacks: Breaking authentication in TLS, IKE, and SSH, in Network and Distributed System Security Symposium (NDSS), 2016.

