Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/13/20240864

Evaluating RSA encryption: Primality testing, pollard's algorithms, and security challenges

Ziqian Liu

The Faculty of Science and Engineering, University of Nottingham Ningbo China, Ningbo, 315100, China

scyzl14@nottingham.edu.cn

Abstract. In the dynamic realm of cryptography, Rivest, Shamir, and Adleman (RSA) encryption stands as a pivotal element in ensuring secure communications. This analysis offers a detailed exploration of RSA encryption, emphasizing critical aspects such as primality testing, the intricacies of Pollard's factorization algorithms, and the overarching security challenges intrinsic to this prevalent encryption paradigm. Delving deep into the heart of RSA, the research underscores the paramount role of primality testing in the RSA key generation process and critically evaluates the efficiency and reliability of diverse primality test methodologies. Moreover, it navigates the potential pitfalls introduced by Pollard's algorithms and ponders their consequential implications for RSA's security matrix. Beyond these technicalities, the analysis brings to the forefront a spectrum of security challenges besieging RSA. This encompasses nuances like vulnerabilities arising from diminutive private keys, pitfalls linked to common modulus attacks, and susceptibilities stemming from cache timing discrepancies. By illuminating both the robust facets and inherent vulnerabilities of RSA encryption, this scholarly work elevates the current narrative on cryptographic security. It accentuates the perpetual necessity for meticulous scrutiny and agile adaptability in the quest to shield sensitive digital information in our progressively interconnected world.

Keywords: Fermat's Little Theory, Rabin-Miller Test, Pollard's Rho, Pollard's p-1.

1. Introduction

Rivest, Shamir, and Adleman (RSA) encryption employs a dual-key system, with public and private components. The public key, characterized by a modulus and an encryption exponent, facilitates message encryption [1]. In contrast, the private key, possessing the same modulus but paired with a decryption exponent, remains concealed for the decryption process. RSA's robust security foundation hinges on the immense computational difficulty inherent in deducing the product of two substantial prime numbers [2, 3]. Such intricacies make decryption without the secret key virtually unfeasible, reinforcing RSA's pivotal role in the domain of secure digital communication and cryptography. Its legacy underscores its lasting relevance in the cryptographic realm.

The realm of modern cryptography has brought to the fore a fascinating method known as RSA encryption. At the heart of this system lie two integral sets of keys: the public key, often represented by the notation (n, e), and its counterpart, the fiercely guarded private key, which is denoted by (n, d). These sets of keys are the linchpins that hold the entire encryption and decryption process together. Delving into the mechanics, when one opts to send an encrypted message using RSA, the original content is not simply cloaked in an unreadable format. Instead, it undergoes a metamorphosis where it is transformed into a numerical avatar, symbolized by M. This conversion ensures that the text is prepped for encryption. Upon reaching its destination, the recipient, armed with their exclusive private key (n, d), unravels the message. They achieve this by calculating M = C^d % n, a process that can appear esoteric to the uninitiated but is an epitome of elegance in the world of cryptography. Section 1 of our exploration goes under the hood of RSA, diving deep into its foundational process: the primality testing methods. It's of paramount importance that the numbers chosen as the bedrock of this encryption method are unambiguously prime. The distinction between a number that's genuinely prime and one that's mistakenly considered so could be the difference between an impregnable message and a cryptographic catastrophe. As cited in [4], various methods ensure the veracity of these prime numbers, enhancing the robustness of the RSA system. In Section 2, the narrative transitions from the construction of RSA to dissecting numbers. The spotlight here is on Pollard's factorization algorithms, a suite of techniques engineered to deconstruct numbers down to their prime elements. Understanding factorization is not merely an academic exercise; it provides insights into the potential chinks in the cryptographic armor, giving us a holistic view of the system's strengths and vulnerabilities. However, like all cryptographic techniques, RSA is not without its potential pitfalls. Section 3 embarks on a journey through the labyrinth of security challenges and intrinsic limitations associated with RSA. While RSA remains a stalwart in the encryption domain, being cognizant of its vulnerabilities ensures that users can fortify their defenses and make informed decisions [5, 6]. Drawing the curtains in Section 4 not only summarizes the journey but also gazes into the horizon, identifying promising areas for future research. The dynamic field of cryptography is ever-evolving, and with every challenge overcome, new horizons beckon. The world of RSA blends mathematical artistry with computational prowess. Through its intricacies, there is a reminder of the ingenuity of human intellect and the endless pursuit of security in the interconnected digital age [7, 8].

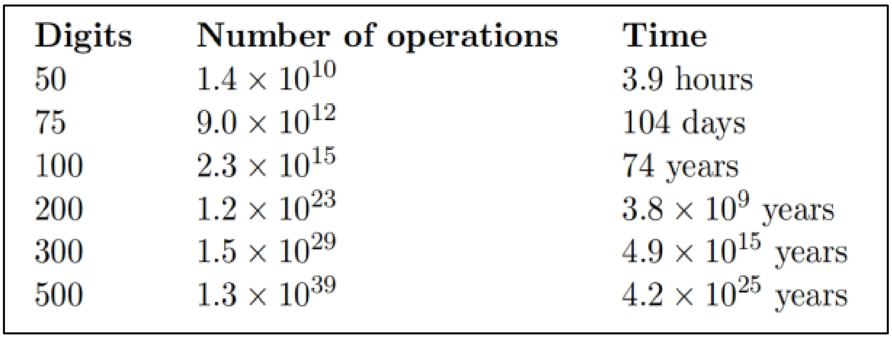
2. Examination of Primality Testing Methods

2.1. Brute Force

Brute force is a general cryptanalysis method that seeks to dismantle a cryptosystem by attempting every possible key [9]. To determine p and q through brute force, the following code could be utilized:

```
function primeFactors = find_prime_factors(n)
    primeFactors = [];
    for i = 2:n
        while mod(n, i) == 0
            primeFactors = [primeFactors, i];
            n = n / i;
        end
    end
end
```

However, as the number of digits in n increases, the execution time of this method also rises. Presented below is Table1, originally introduced in 1978 by the creators of RSA. Assuming that each operation in the Schroeppel factoring technique takes one microsecond to execute, we provide the data below for varying lengths of n [10].



Digits	Number of operations	Time
50	1.4×10^{10}	3.9 hours
75	9.0×10^{12}	104 days
100	2.3×10^{15}	74 years
200	1.2×10^{23}	3.8×10^{9} years
300	1.5×10^{29}	4.9×10^{15} years
500	1.3×10^{39}	4.2×10^{25} years

Figure 1. Effects of Key Size on RSA Encryption Efficiency: A Computational Analysis (Photo/Picture credit: Original).

In that case, the larger key will slow down the encryption and decryption process because it requires more computations to create the key and to encrypt and decrypt data. As shown in Figure 1.

2.2. Fermat's Little Theory

Fermat's Little Theorem provides a strong mathematical basis for the RSA algorithm, demonstrating its security properties. Here's how it's applied within the RSA framework:

Euler's Totient Theorem asserts that for every pair of co-prime integers "a" and "n," a raised to the power of φ(n) is congruent to 1 modulo n. In RSA, "n" represents the modulus. The selection of various parameters in the algorithm is guided by the Euler's Totient function, φ(n). For key generation in RSA, it's common practice to select a fixed prime number, like 65537, due to its efficiency in encryption operations. Using Fermat's Little Theorem ensures that this number remains co-prime with the modulus 'n' and its corresponding φ(n). This co-primality is a core requirement for ensuring encryption is secure. As a part of this process, Fermat's Little Theorem also aids in determining the private exponent 'd' by computing the modular multiplicative inverse of the public exponent. This mathematical process provides a solid foundation, ensuring that the keys used in RSA encryption and decryption are both legitimate and secure.

The act of encryption and decryption in RSA involves modular exponentiation [11]. The integrity of these processes leans heavily on the Euler's Totient Theorem. It's essential for the chosen public exponent to be co-prime to φ(n), which is often achieved by choosing a small prime number. Consequently, any message encrypted using the public exponent can be decrypted successfully using its corresponding private exponent. To illustrate the RSA's correctness using Fermat's Little Theorem, consider the prime numbers p=17 and q=19. Their product gives n=323, and φ(n) is 288. Choosing an integer e, like 13, which is co-prime to 288, one can determine 'd' such that the product of e and d is congruent to 1 modulo φ(n). Through iterative processes, it's found that when the constant is 32, d equals 709, making 709 a suitable integer. This process underscores the mathematical validity and reliability of the RSA encryption and decryption mechanism.

2.3. Rabin-Miller Test

The Rabin-Miller test's main principle is to effectively check a number's primality via randomness. The Rabin-Miller test, in contrast to deterministic primality tests like the Sieve of Eratosthenes, yields a result that is probably accurate rather than a result that is guaranteed to be true. It means that even though it occasionally produces false positive results, the likelihood of such errors can be drastically reduced by running the test numerous times using different random selections [12].

In applications like cryptography, where quickly finding prime numbers is essential, the Rabin-Miller test excels due to its speed and effectiveness for big numbers. However, because of its probabilistic character, it occasionally yields false-positive findings; for this reason, it is frequently used in conjunction with other primality tests or in algorithms that can accept some false-positive results.

3. Exploration of Pollard's Factorization Algorithms

If either algorithm finds either, it can be used iteratively to find the factors of an or b [13].

3.1. Pollard's Rho

The Birthday Paradox algorithm serves as a foundational building block in the vast domain of computational mathematics, particularly where it intersects with number theory. This algorithm offers an insightful look into probabilistic events and the seemingly counterintuitive occurrence of collisions. Such an understanding is vital as one transitions to the intricate issue of decomposing composite numbers into prime components. Yet, there are limitations to the Birthday Paradox algorithm. Its considerable space and time demands, particularly during the replication process of comparisons, coupled with its nonlinear surge in collision probability, can often be constricting. Shifting focus, another computational marvel emerges: the Pollard's Rho algorithm. Much like the Birthday Paradox, it provides foundational knowledge. However, its true brilliance is revealed in the sophisticated realm of integer factorization. Celebrated for its ingenious deployment of randomness combined with cycle detection, the Pollard's Rho algorithm stands out in isolating small prime factors of expansive integers. Such capabilities have profound implications, especially in fields demanding rigorous encryption and paramount security. Further diving into this topic introduces us to the Pollard's p-1 method. Renowned for its prowess in integer factorization and cryptanalysis, this algorithm's primary mission is to pinpoint non-trivial prime factors in composite structures. Such a function is invaluable, especially when decoding or assessing the fortitude of cryptographic constructs, such as the RSA system. Using a principle rooted in Fermat's Little Theorem, the Pollard's p-1 technique not only helps decrypt but also facilitates a deeper understanding of number theory subjects. It also emerges as a reliable instrument for primality verification, presenting profound insights into the innate properties of numbers. To sum it up, the Pollard's p-1 method's capacity to uncover prime factors of composite structures holds immense value both in theoretical number theory studies and the practical arena of cryptography.

3.2. Pollard's p-1

The Pollard's p-1 algorithm is a sophisticated mathematical approach that plays a pivotal role in integer factorization and cryptanalysis. The primary objective of this algorithm is to identify non-trivial prime factors of composite numbers. This capability is especially significant when considering the decryption processes and evaluating the robustness of cryptographic systems like RSA.

The strength of Pollard's p-1 method lies in its ability to detect smooth prime factors efficiently. By harnessing the power of Fermat's Little Theorem, the algorithm can zero in on prime factors with remarkable precision. This theorem, rooted deeply in number theory, allows for the identification of numbers that have particular relationships with their prime factors. Furthermore, the algorithm isn't just restricted to factorization [14]. It plays a broader role in the realm of number theory. For instance, Pollard's p-1 serves as an essential tool for primality testing, offering deep insights into the properties and behaviors of numbers. The versatility of the algorithm bridges the gap between theoretical research and practical applications. In essence, the significance of the Pollard's p-1 method is manifold. Its knack for uncovering prime factors of composite numbers has tremendous implications. From an academic standpoint, it enriches number theory research by offering a new lens through which to view number properties. On the practical front, its applications in cryptography underscore its importance in maintaining the security and integrity of digital communications in our increasingly connected world.

4. Security Challenges and Limitations

RSA encryption, while being a foundational pillar of modern cryptography, is persistently threatened by an array of sophisticated attacks. The Small Private Key Attack, for instance, jeopardizes data confidentiality by potentially allowing adversaries to guess private keys systematically. This vulnerability can grant unauthorized access, putting sensitive information at peril. The Common Modulus Attack weakens the very trust on which digital systems operate. Any exposure of private keys due to shared moduli can compromise the security of numerous entities.

Furthermore, Timing Attacks, by compromising the integrity of RSA operations, can reveal vital components through meticulous timing observations, leading to significant breaches. Fault-Based Attacks create avenues for malevolent actors to interfere with cryptographic processes, resulting in private key exposure. Weaknesses in key generation, tapped by Low Private Exponent Attacks, might enable intruders to deduce private keys, thereby undermining data encryption [15]. More subtly, Padding Oracle Attacks can slowly extract private key details, imperiling data confidentiality. Meanwhile, Coppersmith's Attacks harness intricate mathematical methodologies to reconstruct private key data, emphasizing the need for rigorous countermeasures. With Partial Key Exposure Attacks, even a slight disclosure can be catastrophic, as attackers might deduce the entire key. Cache Timing Attacks, capitalizing on cache access variations, can reveal private keys, threatening the RSA encryption's foundational integrity. Beyond these vulnerabilities, RSA grapples with inherent limitations. The considerable computational demands of RSA operations can stretch computing capabilities, inducing unwanted lags, especially in time-sensitive scenarios. Additionally, as security paradigms evolve, there's an increasing push for larger RSA key sizes. This push, while enhancing security, imposes practical challenges, particularly for devices with limited resources. Given these complexities, there's an evident need to seek alternative cryptographic methodologies or refine RSA's implementation. This pursuit is crucial to maintain a harmony between high-end security and computational agility, ensuring RSA remains an influential player in the ever-evolving digital realm.

5. Conclusion

In essence, understanding the intricacies of RSA encryption necessitates a profound exploration of its foundational elements—primality testing and identifiable vulnerabilities, notably those highlighted by Pollard's algorithms. The unshakeable foundation of RSA's security lies in the formidable generation of prime numbers, fortified by rigorous mathematical principles. However, as the landscape of cybersecurity continuously morphs and adapts, RSA confronts a spectrum of challenges. These range from the rudimentary, such as vulnerabilities arising from diminutive private keys, to the more complex and nuanced, like the perils of cache timing attacks. Addressing these looming threats mandates a multifaceted strategy. It begins with an infallible approach to key generation, then dovetails into the judicious selection of algorithms and culminates in perpetual vigilance. The advent of the digital age has accelerated the urgency of addressing data security concerns. Within this framework, the role of RSA encryption becomes paramount. As digital transactions, communications, and records become commonplace, ensuring the sanctity of these interactions is not just a technical necessity but a societal imperative. While RSA stands as a formidable bulwark in the digital defense lineup, no system is entirely impervious. The world witnessed this when vulnerabilities were discovered and later addressed. Yet, these instances serve as stark reminders that the quest for digital security is an ongoing one.

References

[1] Milanov, E. (2009). The RSA algorithm. RSA laboratories, 1-11.

[2] Barnes, C. (2004). Integer factorization algorithms. Oregon State University.

[3] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[4] Boneh, D., & Shacham, H. (2004). Fast Variants of RSA. Cryptology ePrint Archive, Report 2004/086.

[5] Schindler, W., & Wagner, D. (2005). Cryptanalysis of the RSA Implementations of Crypto – Coprocessors. Cryptographic Hardware and Embedded Systems, 1-12.

[6] Boneh, D., & Venkatesan, R. (1998). Breaking RSA may not be equivalent to factoring. In Advances in Cryptology—EUROCRYPT'98: International Conference on the Theory and Application of Cryptographic Techniques Espoo, Finland, May 31 – June 4, 1998 Proceedings 17 (pp. 59-71). Springer Berlin Heidelberg.

[7] Takagi, T. (1998). Fast RSA-type cryptosystem modulo pkq. In Advances in Cryptology—CRYPTO'98: 18th Annual International Cryptology Conference Santa Barbara, California, USA August 23–27, 1998 Proceedings 18 (pp. 318-326). Springer Berlin Heidelberg.

[8] Rabin, M. O. (1980). Probabilistic algorithms in finite fields. SIAM Journal on computing, 9(2), 273-280.

[9] Muhammad, S. J., Chiroma, H., & Mahmud, M. (2014). Cryptanalytic attacks on Rivest, Shamir, and Adleman (RSA) cryptosystem: issues and challenges. J Theor Appl Inf Technol, 61(1), 2349.

[10] Samandari, N., Nazari, N. M., Olfat, J. A., Rafi, R., Azizi, Z., & Ulfat, W. I. (2023). Applications of Fermat's Little Theorem. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 14(03), 209-215.

[11] Boneh, D. (1999). Twenty years of attacks on the RSA cryptosystem. Notices of the AMS, 46(2), 203-213.

[12] Sarnaik, S., Bhakkad, R., & Desai, C. (2015, March). Comparative study on Integer Factorization algorithm -Pollard's RHO and Pollard's P-1. In 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom) (pp. 677-679). IEEE.

[13] Kim, J. H., Montenegro, R., Peres, Y., & Tetali, P. (2010). A birthday paradox for Markov chains with an optimal bound for collision in the Pollard rho algorithm for discrete logarithm.

[14] Galbraith, Steven D. (2012), "14.2.5 Towards a rigorous analysis of Pollard rho", Mathematics of Public Key Cryptography, Cambridge University Press, pp. 272-273, ISBN 9781107013926.

[15] Hegde, N., & Deepthi, P. (2015). Pollard RHO algorithm for integer factorization and discrete logarithm problem. International Journal of Computer Applications, 121(18), 14-17.