Vasudeo Patil, 2026, 14:4
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

A Dual-Key Based RSA Cryptosystem

Vasudeo Patil[1], Arpit Vijay Raikwar[2]

[1]Professor, Department of Mathematics, Arts, Science Commerce College, Chikhaldara Dist Amravati India
[2]Research Scholar, Department of Mathematics, Arts, Science Commerce College, Chikhaldara Dist Amravati India

Abstract- Rivest–Shamir–Adleman (RSA) is a widely used public-key cryptosystem based on a single public-private key pair. This paper proposes a dual-key RSA cryptosystem that employs two public keys with their corresponding private keys, introducing additional key dependency into the encryption and decryption processes. The correctness of the proposed scheme is established through a formal mathematical proof and illustrated with a numerical example. The proposed scheme and conventional RSA are implemented in Java using Apache NetBeans IDE 25, and their computational performance is evaluated under identical parameters. The experimental results indicate that the proposed scheme achieves performance comparable to conventional RSA. The security analysis using ProVerif verifies message and private-key secrecy under a symbolic adversarial model. The proposed dual-key RSA cryptosystem represents a structured and efficient extension of the classical RSA scheme suitable for academic and experimental cryptographic studies.

Keywords: RSA Cryptosystem, Public Key Cryptography, Dual Key Encryption, and Cryptographic Security.

I. INTRODUCTION

Cryptography is the study of techniques for securing information by transforming plaintext into ciphertext such that only authorized parties can recover the original message [1]. Its primary objective is to protect sensitive digital data from unauthorized access. Current research in this field focuses on developing cryptographic algorithms that not only enhance security but also remain efficient in terms of computational complexity [2, 3, 4, 5]. In symmetric key cryptography [6], both the sender and the receiver share a single secret key for encryption and decryption. Although this approach is computationally efficient, it suffers from the key distribution problem: if the secret key is intercepted, the confidentiality of all communications is compromised.

To overcome this limitation, Diffie and Hellman introduced their key exchange protocol in 1976 [7], which laid the foundation for public key cryptography. In this paradigm, each user possesses a pair of mathematically related keys, namely a public key for encryption and a private key for decryption [8]. Among the various public key algorithms, the RSA cryptosystem, introduced by Rivest, Shamir, and Adleman in 1977 [9], remains one of the most widely adopted schemes. Classical RSA employs a single pair of public and private keys, and its security is primarily based on the computational difficulty of integer factorization. However, the reliance on a single private key may pose limitations in scenarios involving key exposure or misuse. Motivated by this observation, the present work proposes a dual-key-based RSA cryptosystem that employs two public keys and two corresponding private keys. The proposed structure introduces additional key dependency during encryption and decryption while preserving the mathematical foundation of the classical RSA scheme.

II. LITERATURE REVIEW

The RSA cryptosystem has been a foundational public-key cryptographic scheme for decades, but its classical form faces limitations in computational efficiency and vulnerability to factorization-based attacks. Over the years, numerous variants have been proposed to address these issues, focusing on either enhancing security, improving computational efficiency, or both. This section reviews key contributions, organized into three thematic categories: security-oriented modifications, efficiency-oriented modifications, and hybrid approaches that aim to balance security and performance.

Security-Oriented Modifications

Several studies have focused primarily on strengthening RSA's resistance to factorization or key recovery attacks. Dhakar et al. [10] introduced the Modified RSA Encryption Algorithm (MREA), which uses four prime numbers instead of the standard two to increase the difficulty of modulus factorization. Their scheme also incorporated modified encryption and decryption procedures, improving theoretical security, although the additional computational overhead was significant.

Thangavel et al. [11] proposed a key generation mechanism using four distinct prime numbers, which strengthened the system against factorization attacks. While effective from a security perspective, the higher number of primes resulted in increased computation time, illustrating the recurring trade-off between robustness and efficiency in RSA variants. Minni et al. [12] presented a novel approach that replaced the traditional modulus N = p × q with a new parameter X, defining public and private keys as (k_1,X) and (k_2,X), respectively, constrained by (k_1 · k_2) mod X = 1. Decryption was performed via

$$PT = \sqrt{CT^{k_2} \bmod X}$$

However, this construction inadvertently exposed the private key, as k_2 could be easily derived from k_1 and X, making the system less secure than conventional RSA. Al Barazanchi et al. [13] introduced a three-key RSA variant to increase structural complexity. Although this added algorithmic depth, further assessment is required to evaluate its resilience against advanced cryptanalytic attacks and scalability to large key sizes.

Efficiency-Oriented Modifications

A second body of work emphasizes reducing computational overhead while retaining the basic RSA structure. Das et al. [14] proposed a novel encryption and decryption algorithm that streamlines the modular arithmetic operations that dominate RSA computation. Their method is particularly relevant for resource-constrained environments, such as mobile devices and embedded systems, where efficiency is critical. Hermawan et al. [15] extended this direction by integrating eight prime numbers and applying the Chinese Remainder Theorem (CRT) to handle multiple message components simultaneously.

Encryption used eight public keys, while decryption relied on two private keys. This parallelization accelerated the encryption-decryption process but increased key management complexity, highlighting a practical limitation of such efficiency-oriented approaches. Ghadi [16] introduced per-symbol key variation, converting plaintext messages to hexadecimal and encrypting each symbol using a sequence of public exponents {e_i}, with corresponding private exponents {d_i} for decryption. This method increases encryption diversity, though efficiency for large datasets remains a concern.

Hybrid Approaches

Balancing Security and Efficiency Some studies aim to achieve both security enhancement and efficiency improvement. Kamardan et al. [17] combined multiple prime factors with CRT, resulting in faster decryption while maintaining increased theoretical resistance to factorization. Nevertheless, brute-force attacks on the private exponent d remained possible, underscoring that efficiency gains alone do not guarantee comprehensive security. Pal et al. [18] presented a modified and optimized RSA scheme employing four prime numbers for key generation. A notable feature of their approach is the redefinition of the public exponent as f = (e · a) + b, with decryption executed via its modular inverse d = f^(-1) .

This modification seeks to achieve a better balance between security and computational feasibility. While promising, the long-term resilience of this scheme to advanced factorization algorithms or side-channel attacks has yet to be rigorously tested. V. Patil et al. [19] introduced the generalized discrete logarithm problem (GDLP), which incorporates (m-1) distinct exponents and can be analyzed via the index calculus algorithm. However, not strictly an RSA variant, GDLP reflects broader trends in exploring new cryptographic hardness assumptions for both security and efficiency.

II. THE RSA CRYPTOSYSTEM

The RSA cryptosystem, introduced in 1977 by Rivest, Shamir, and Adleman, remains one of the most widely deployed public-key algorithms for ensuring data confidentiality and authenticity [20, 21]. RSA employs a pair of mathematically related keys: a public key for encryption, which may be openly distributed, and a private key for decryption, which must remain secret. The security of RSA is grounded in the computational intractability of factoring large composite integers, i.e., products of two prime numbers [9]. Since no efficient polynomial-time algorithm exists for integer factorization, RSA is considered secure when sufficiently large key sizes are employed. For validation and experimentation, a Java-based prototype was developed using the Apache NetBeans IDE (version 25). The implementation was designed to evaluate the fundamental security properties of confidentiality, integrity, and authenticity.

RSA Key Generation

The RSA key generation process is defined as follows:

1. Select two large random prime numbers x and y of equal bit length.
2. Compute the modulus $n = xy$ and Euler's totient $\phi(n) = (x-1)(y-1)$.
3. Choose a public exponent g such that $gcd(g, \phi(n)) = 1$ and $1 < g < \phi(n)$.
4. Compute the private exponent h as the modular inverse of g, satisfying $gh \equiv 1 \ (mod\ \phi(n))$.
5. The public key is defined as (g, n) and the private key as (h, n).

RSA Encryption

To encrypt a plaintext message M (where M is a positive integer less than n), the sender computes:

$$E \equiv M^g \ (mod\ n) \qquad (1)$$

where E denotes the ciphertext. The value E is then transmitted to the receiver.

RSA Decryption

Upon receiving the ciphertext E, the receiver recovers the plaintext using the private key (h, n):

$$N \equiv E^h \ (mod\ n) \qquad (2)$$

where N corresponds to the original message M. $gh \equiv 1 \ (mod\ \phi(n))$,, decryption correctly yields M.

Proposed Modified RSA Cryptosystem

The proposed Modified RSA Cryptosystem extends the classical RSA scheme by introducing two distinct public keys and two corresponding private keys. Unlike traditional RSA, which relies on a single key pair, the modified approach enhances robustness against cryptanalytic attacks by requiring an adversary to compromise multiple key pairs simultaneously. This multikey structure significantly increases computational complexity for potential attackers, thereby strengthening the overall security of communication. For validation, a Java-based implementation was developed using Apache NetBeans IDE (version 25). The prototype was tested against fundamental security properties, namely confidentiality, integrity, and authenticity.

Key Generation

The key generation process for the proposed scheme is defined as follows:

1. Select two large prime numbers x and y of equal bit length.
2. Compute the modulus $n = xy$ and Euler's totient $\phi(n) = (x-1)(y-1)$.
3. Choose two public exponents g and i such that $gcd(g, \phi(n)) = 1, gcd(i, \phi(n)) = 1$, and $1 < g, i < \phi(n)$.
4. Compute the corresponding private exponents h and j as modular inverses of g and i, respectively, satisfying:
   a. $gh \equiv 1 \ (mod\ \phi(n)), ij \equiv 1 \ (mod\ \phi(n))$ (3)
5. Impose additional constraints to ensure non-trivial key differentiation:
   a. $g \neq h,\ i \neq j.$ (4)
6. The public keys are (g, n) and (i, n), while the private keys are (h, n) and (j, n).

The explicit conditions g ≠ h and i ≠ j are enforced to prevent degenerate cases where encryption and decryption exponents coincide. Without these constraints, the scheme could collapse into a trivial equivalence between public and private components, thereby weakening its cryptographic strength.

Encryption

To encrypt a plaintext message M (where M < n), the sender computes:

$$E \equiv M^{g.i} \ (mod\ n) \qquad (5)$$

where E denotes the ciphertext. The value E is then transmitted securely to the receiver.

Decryption

Upon receiving the ciphertext E, the receiver recovers the plaintext using the private keys (h,n) and (j,n)

$$N \equiv E^{h.j} \ (mod\ n) \qquad (6)$$

$(g \cdot i)(h \cdot j) \equiv 1 \ (mod\ \phi(n))$, the decrypted value N correctly yields the original message M. The use of two independent key pairs ensures that both confidentiality and resilience against single-key exposure are reinforced.

III. MATHEMATICAL PROOF OF THE MODIFIED RSA CRYPTOSYSTEM

Correctness of Encryption and Decryption

Let x and y be two distinct large prime numbers. Define $n = xy$ and $\varphi(n) = (x-1)(y-1)$. Select integers g, i such that $gcd(g, \varphi(n)) = 1$ and $gcd(i, \varphi(n)) = 1$, with $1 < g, i < \varphi(n)$. The public keys are (g, n) and (i, n), while the corresponding private keys are (h, n) and (j, n).

Let $M \in Z_n^*$ be the plaintext message. The encryption and decryption steps are defined as: $E \equiv M^{g.i} \ (mod\ n)$, $N \equiv E^{h.j} \ (mod\ n)$.

Substituting the encryption equation into the decryption equation gives

$$N \equiv M^{(g \cdot i)(h \cdot j)} \ (mod\ n)$$

Since $(g \cdot i)(h \cdot j) \equiv (gh)(ij) \equiv 1 \ (mod\ \varphi(n))$

Therefore, by Euler's theorem, $M^{\varphi(n)} \equiv 1 \ (mod\ n)$ and hence $N \equiv M \ (mod\ n)$.

Thus, the original plaintext is successfully recovered after decryption. Hence, the proposed Modified RSA Cryptosystem satisfies the correctness property of public-key encryption while employing two independent public-key/private-key exponent pairs.

Experimental Results

The proposed modified RSA cryptosystem introduces structural changes that enhance the strength of the cryptosystem. This approach utilizes two distinct pairs of public and private keys, thereby increasing the cryptographic strength and improving resilience against unauthorized access and potential cryptanalytic attacks [22]. While the generation and utilization of multiple key pairs introduces a slight increase in processing time compared to standard RSA, this trade-off results in a substantially improved security posture. Given that ensuring the confidentiality and integrity of data is typically a higher priority than minimizing computational overhead, the additional time required is considered acceptable within the context of secure communications.

To rigorously assess the performance and efficiency of both the traditional and modified RSA cryptosystems, a series of experiments were conducted using randomly generated prime numbers with key sizes of 512, 1024, and 2048 bits.The experiments were executed on a PC equipped with a 12th Gen Intel® Core™ i5-1235U CPU (1.30 GHz) and 8 GB RAM running a 64-bit Windows operating system using Apache NetBeans IDE 25. A custom Javabased implementation was developed and executed within the Apache NetBeans IDE to measure and compare key generation time, encryption time, decryption time, and total computation time for each algorithm. It is important to note that due to the use of randomly generated primes in each trial, some variability in the observed results was expected and recorded.

Despite these variations, consistent trends were observed across multiple runs, reinforcing the reliability of the comparative analysis. The experimental results are summarized in Table 1, Table 2, Table 3, and Table 4, corresponding to different key lengths. Additionally, Figures 1 to 4 provide visual representations of the timing comparisons between the traditional RSA and the modified RSA implementations. The results demonstrate that although the modified RSA incurs a marginal increase in key generation time, the overall performance overhead remains minimal, particularly for larger key sizes. These findings suggest that the proposed RSA variant achieves a balance between computational efficiency and

enhanced security, making it a viable candidate for future secure communication systems.

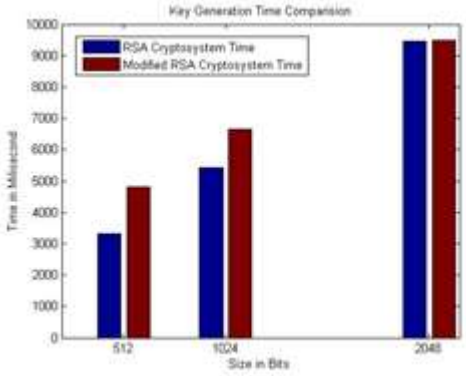

Figure 1: Key Generation Time Comparison

Size (Bits)	RSA (ms)	Modified (ms)
512	3317	4801
1024	5413	6663
2048	9457	9496

Table 1: Key Generation Time Comparison

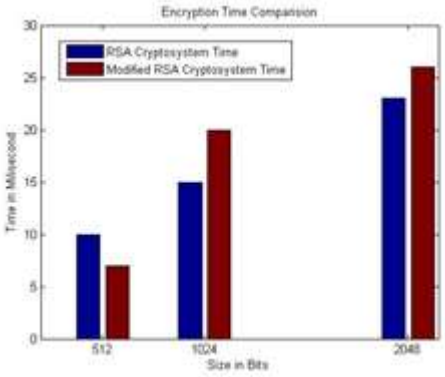

Figure 2: Encryption Time Comparison

Size (Bits)	RSA (ms)	Modified RSA (ms)
512	10	7
1024	15	20
2048	23	26

Table 2: Encryption Time Comparison

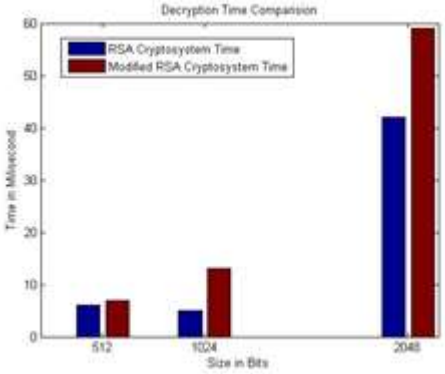

Figure 3: Decryption Time Comparison

Size (Bits)	RSA (ms)	Modified RSA (ms)
512	6	7
1024	5	13
2048	42	59

Table 3: Decryption Time Comparison

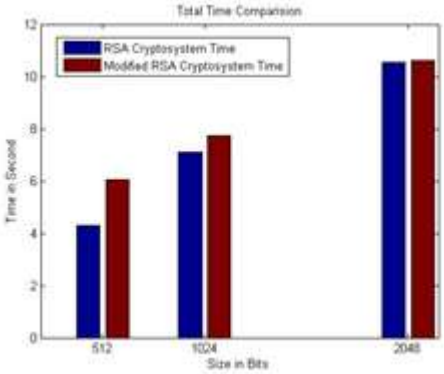

Figure 4: Total Time Comparison

Size (Bits)	RSA (s)	Modified RSA (s)
512	4.317	6.042
1024	7.090	7.720
2048	10.540	10.620

Table 4: Total Time Comparison

```
Fixed Prime p: 27647
Fixed Prime q: 65537
Modulus n: 1811901439
Phi(n): 1811808256
Public Key (e, n): (1253, 1811901439)
Private Key (d, n): (1043994861, 1811901439)
Original Message: 123
Encrypted Message: 726701541
Decrypted Message: 123
--------------------------------------------------------------------------------
BUILD SUCCESS
--------------------------------------------------------------------------------
Total time:  0.992 s
Finished at: 2025-10-01T08:42:33+05:30
```

Figure 5. Execution Time for RSA

```
Prime p: 27647
Prime q: 65537
Modulus n = p*q: 1811901439
Euler Totient ?(n): 1811808256
Public Exponent e: 1253
Public Exponent o: 1255
Private Exponent d: 1043994861
Private Exponent g: 1550503639
Original Message: 123
Encrypted Message: 1004117695
Decrypted Message: 123
--------------------------------------------------------------------------------
BUILD SUCCESS
--------------------------------------------------------------------------------
Total time:  0.927 s
Finished at: 2025-10-01T08:44:59+05:30
```

Figure 6. Execution Time for Modified RSA

In our experimental evaluation, we conducted a comparative analysis of the standard RSA cryptosystem and a modified RSA variant using fixed prime numbers and exponent values, implemented in Java Apache NetBeans IDE (version 25). Execution time measurements were collected for both encryption and decryption processes under identical computational conditions. The results, illustrated in Figures 5 and 6, show that the standard RSA algorithm required approximately 0.992 seconds, whereas the modified RSA completed the same operations in approximately 0.927 seconds. This indicates a measurable reduction in computational time for the modified RSA.

The observed performance behavior can be attributed to the structural modifications introduced through the use of dual exponents and corresponding modular inverse operations. While these additions introduce some overhead during key generation, the overall computational performance remains comparable to that of classical RSA for fixed parameters. Furthermore, the proposed scheme preserves the fundamental mathematical structure of RSA while introducing additional key dependency through multiple key parameters.

This added structural complexity may increase the difficulty of certain cryptanalytic analyses without altering the underlying hardness assumptions of RSA. Overall, the results indicate that the modified RSA scheme provides a balanced trade-off between computational efficiency and structural enhancement, making it suitable for academic and experimental evaluation of multi-key public-key cryptosystems.

IV. FORMAL VERIFICATION USING PROVERIF

To validate the correctness and secrecy properties of the proposed dualkey RSA cryptosystem, the protocol was modeled using the ProVerif tool [23, 24]. ProVerif is a widely used framework for the symbolic analysis of cryptographic protocols under the Dolev–Yao adversary model. The complete ProVerif specification is provided in Listing 1, where the dual-key encryption–decryption relation is defined, and secrecy queries are formulated to analyze whether an adversary can compromise the protocol. Specifically, the following properties were examined:

1. Secrecy of the plaintext message (m) — ensuring that the encrypted message remains confidential and cannot be learned by an adversary.
2. Secrecy of the first private key (sk1) — verifying that the adversary cannot derive the first private decryption key.
3. Secrecy of the second private key (sk2) — verifying that the adversary cannot derive the second private decryption key.

```
type s1key.
type p1key.
type s2key.
type p2key.

free c: channel.
free sk1: s1key [private].
free sk2: s2key [private].

fun pk1(s1key): p1key.
fun pk2(s2key): p2key.
fun encrypt(bitstring, p1key, p2key): bitstring.

reduc forall x: bitstring, y: s1key, z: s2key;
decrypt(encrypt(x, pk1(y), pk2(z)), y, z) = x.

(* Query if attacker can learn m *)
free m: bitstring [private].
query attacker(m).
query attacker(sk1).
query attacker(sk2).

process
out(c, encrypt(m, pk1(sk1), pk2(sk2)))
```

Listing 1: ProVerif model of the dual-key RSA protocol

The execution results of the ProVerif analysis are summarized in Listing 2. The tool confirms that all secrecy queries hold, i.e., not attacker(m) is true, not attacker(sk1) is true, and not attacker(sk2) is true. These results indicate that, even in the presence of an active adversary with full control over the communication channel, the protocol preserves the confidentiality of the plaintext message and ensures that both private keys remain undisclosed.

```
-- Query not attacker(m[]) in process 0
RESULT not attacker(m[]) is true.
-- Query not attacker(sk1[]) in process 0
RESULT not attacker(sk1[]) is true.
-- Query not attacker(sk2[]) in process 0
RESULT not attacker(sk2[]) is true.
```

Listing 2: Verification output of the ProVerif analysis

This formal verification supports the following security properties of the proposed dual-key RSA scheme:

- **Message confidentiality —** ensuring that encrypted data cannot be revealed to an adversary.

- **Private key secrecy —** guaranteeing that neither private key can be derived through protocol execution.
- **Resistance to channel-based attacks —** preventing an adversary from recovering sensitive information even under full network control.

By combining formal verification with the mathematical correctness analysis presented earlier, the study provides a structured validation of the proposed dual-key RSA cryptosystem. While ProVerif operates in a symbolic model, the results demonstrate that the protocol satisfies essential secrecy properties and maintains consistency with the theoretical foundations of RSA-based encryption.

Experimental Demonstration of Avalanche Effect

One of the essential properties of a secure cryptographic system is its sensitivity to small changes in the input, commonly referred to as the avalanche effect. Ideally, if a single bit in the plaintext is altered, approximately half of the ciphertext bits should change. This ensures strong diffusion, making the system resistant to differential and chosen-plaintext attacks. To evaluate this property, an experimental test was carried out using the proposed Modified RSA scheme. A random plaintext M1 was selected and its ciphertext E1 was generated. A second plaintext M2 was obtained by flipping a single random bit of M1, and the corresponding ciphertext E2 was then computed.

The Hamming distance between E1 and E2 was measured to calculate the avalanche effect. The results of one such trial are shown in Table 5. The results show that flipping just one bit in the plaintext resulted in 52.44 (%) of the ciphertext bits being different. This value is very close to the ideal 50 (%), confirming that the proposed Modified RSA algorithm demonstrates strong diffusion characteristics. Such behavior indicates robustness against cryptanalytic techniques and makes the scheme highly suitable for secure applications such as digital payments, authentication systems, and confidential communications.

Table 5: Avalanche Effect Demonstration for Modified RSA

Plaintext	Ciphertext	Avalanche Effect (%)
$M_1 = 46234681013677$ $67906367946029493957786064935$	$E_1 = 3555199631195$ $51702087597725744866867585588$	52.44%
$M_2 = 46234681013$ $67762954607788887972858189568039$	$E_2 = 67692360235$ $23534891842468515045841478366790$	

Numerical Example of the Proposed Dual-Key RSA Cryptosystem

To illustrate the working of the proposed dual-key based RSA cryptosystem, a simple numerical example is presented using small prime numbers for clarity.

1. Select two prime numbers $x = 43$ and $y = 67$.
2. Compute the modulus: $n = xy = 43 \times 67 = 2881$. 3. Compute Euler's totient function: $\phi(n) = (x-1)(y-1) = 42 \times 66 = 2772$.
4. Select two public exponents $g = 353$ and $i = 1753$ such that $gcd(g, \phi(n)) = 1, gcd(i, \phi(n)) = 1$, with $1 < g, i < \phi(n)$.
5. Compute the corresponding private exponents $h = 1013$ and $j = 1741$ such that $gh \equiv 1\ (mod\ \phi(n)), ij \equiv 1\ (mod\ \phi(n))$.
6. The public keys are $(353, 2881)$ and $(1753, 2881)$, and the private keys are $(1013, 2881)$ and $(1741, 2881)$.
7. Let the plaintext message be $M = 55$.
8. Encryption is performed using both public keys: $E = M^{g.i}\ mod\ n = 55353 \cdot 1753\ mod\ 2881 = 55618809\ mod\ 2881$.
9. The resulting ciphertext is: $E = 2565$.
10. Decryption is performed using both private keys: $N = E^{h.j}\ mod\ n = 25651013 \cdot 1741\ mod\ 2881 = 25651763633\ mod\ 2881$.
11. The decrypted value is: $N = 55$.

which matches the original plaintext message.

This numerical example confirms the correctness of the proposed dual-key RSA cryptosystem. The encryption and decryption processes successfully recover the original message, demonstrating that the modified scheme preserves the fundamental properties of the classical RSA algorithm while introducing additional key dependency.

V. CONCLUSION

In this paper, a modified RSA cryptosystem based on a dual-key structure has been presented. The

proposed scheme extends the classical RSA framework by employing two public keys and two corresponding private keys, thereby introducing additional structural complexity into the encryption and decryption processes while preserving the fundamental mathematical principles of RSA. The correctness of the proposed cryptosystem has been established through formal mathematical analysis, and a numerical example has been provided to clearly demonstrate the validity of the encryption–decryption mechanism. To evaluate practical feasibility, both the classical RSA and the proposed dual-key RSA schemes were implemented in Java using Apache NetBeans IDE 25.

A comparative performance analysis under fixed parameters shows that the proposed scheme achieves computational performance comparable to standard RSA, with no significant increase in execution time. At the same time, the use of dual keys increases key dependency and reduces the risks associated with single-key compromise. The security properties of the proposed cryptosystem were examined using the ProVerif tool. The formal verification results confirm the secrecy of the plaintext message and the private keys under a symbolic adversarial model.

An avalanche effect analysis was also conducted, demonstrating adequate diffusion and sensitivity to small changes in the plaintext. Overall, the proposed dual-key-based RSA cryptosystem provides a practical and structured enhancement of the classical RSA scheme. This work offers a solid foundation for further academic research on multi-key public-key cryptosystems and may be extended in future work to explore optimization strategies and applicability in resource-constrained environments.

Disclosure statement

The authors declare that there are no conflicts of interest regarding the publication of this paper. No financial, personal, or professional relationships have inappropriately influenced the work reported in this manuscript.

Biographical Note

Dr. V. R. Patil is an Associate Professor and Head of the Department of Mathematics at Arts, Science and Commerce College, Chikhaldara. He brings over 19 years of extensive research experience and has authored more than 52 research papers published in reputed academic journals. Dr. V. R. Patil has also presented his work at numerous national and international conferences, demonstrating his strong expertise and significant contributions to the domains of Relativity and Cryptography.

Mr. Arpit Vijay Raikwar is a Research Scholar in the Department of Mathematics, pursuing his Ph.D. under the guidance of Dr. V. R. Patil at Arts, Science and Commerce College, Chikhaldara. His research interests focus on Public Key Cryptography and Data Security.

Data Availability Statement (DAS)

All data supporting the findings of this study are openly available within this published article. The complete Java source code implementations of standard RSA, Modified RSA, and the ProVerif model used for validation are also openly accessible as supplementary materials.

Declaration of Funding

No funding or sponsorship was received for conducting this study.

REFERENCES

1. D. Davies, A brief history of cryptography, Information Security Technical Report 2 (2) (1997) 14–17. doi:https://doi.org/10.1016/S1363-4127(97)81323-4. URL https://www.sciencedirect.com/science/article/pii/S1363412797813234
2. R. S. Jamgekar, G. S. Joshi, File encryption and decryption using secure rsa, International Journal of Emerging Science and Engineering (IJESE) 1 (4) (2013) 11–14.
3. T. S. Obaid, Study a public key in rsa algorithm, European Journal of Engineering and Technology Research 5 (4) (2020) 395–398.

4. S. Nisha, M. Farik, Rsa public key cryptography algorithm, A Review. International Journal of Scientific and Technological Research 6 (2017) 187–191.
5. A. I. Khyoon, Modification on the algorithm of rsa cryptography system (2005).
6. W. Diffie, M. Hellman, Special feature exhaustive cryptanalysis of the nbs data encryption standard, Computer 10 (6) (1977) 74–84. doi:10.1109/C-M.1977.217750.
7. W. Diffie, M. Hellman, New directions in cryptography, IEEE Transactions on Information Theory 22 (6) (1976) 644–654. doi:10.1109/TIT.1976.1055638.
8. W. Stallings, Cryptography and network security: Principles and practices (01 2003).
9. R. L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems 21 (2) (1977) 120–126. doi:10.1145/359340.359342. URL https://doi.org/10.1145/359340.359342
10. R. S. Dhakar, A. K. Gupta, P. Sharma, Modified rsa encryption algorithm (mrea), in: 2012 Second International Conference on Advanced Computing Communication Technologies, 2012, pp. 426–429. doi:10.1109/ACCT.2012.74.
11. M. Thangavel, K. Kuppusamy, R. Udayakumar, K. Subashini, An enhanced and secure rsa cryptosystem using four prime numbers, Procedia Computer Science cio on "International Conference on Communications and Network Security 2014" (2014) —.
12. R. Minni, K. Sultania, S. Mishra, D. R. Vincent, An algorithm to enhance security in rsa, in: 2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), 2013, pp. 1–4. doi:10.1109/ICCCNT.2013.6726517.
13. I. Al_Barazanchi, S. A. Shawkat, M. H. Hameed, K. S. L. Al-badri, Modified rsa-based algorithm: a double secure approach, TELKOMNIKA (Telecommunication Computing Electronics and Control) 17 (6) (2023). doi:10.12928/telkomnika.v17i6.13201. URL https://telkomnika.uad.ac.id/index.php/TELKOMNIKA/article/view/13201
14. S. B. Das, S. K. Mishra, A. K. Sahu, A new modified version of standard rsa cryptography algorithm, in: Smart Computing Paradigms: New Progresses and Challenges, Vol. 767 of Advances in Intelligent Systems and Computing, Springer Singapore, 2020, pp. 281–287. doi:10.1007/978-981-13-9680-924.
15. N. T. E. Hermawan, E. Winarko, A. Ashari, Eight prime numbers of modified rsa algorithm method for more secure single board computer implementation, International Journal of Advanced Science, Engineering and Information Technology 11 (6) (2021) 2375–2384. doi:10.18517/ijaseit.11.6.14106.
16. D. M. Ghadi, Improving the robustness of rsa encryption through inputbased key generation, Modelling, Measurement & Control B 11 (1) (2024) 217–223. doi:10.18280/mmep.110124. URL https://www.iieta.org/journals/mmep/paper/10.18280/mmep.110124
17. M. G. Kamardan, N. Aminudin, N. Che-Him, S. Sufahani, K. Khalid, R. Roslan, Modified multi prime rsa cryptosystem, in: Journal of Physics: Conference Series, 995, Vol. 995, IOP Publishing, 2018, p. 012030. doi:10.1088/1742-6596/995/1/012030.
18. P. Pal, B. C. Sahana, J. Poray, Morsa: An innovative modified and optimized rsa framework using parallel computing environment, Modelling, Measurement & Control B 12 (6) (2025) 2085–2096. doi:10.18280/mmep.120624. URL https://www.iieta.org/journals/mmep/paper/10.18280/mmep.120624
19. V. Patil, A. V. Raikwar, An efficient index calculus algorithm for evaluating the generalized discrete logarithm problem, in: Proceedings of the National Conference on Recent Trend in Mathematics & Space Science (NCRTMASS).
20. Y. Kumar, R. Munjal, H. Sharma, Comparison of symmetric and asymmetric cryptography with existing vulnerabilities and countermeasures, International Journal of Computer Science and Management Studies 11 (03) (2011) 60–63.
21. L. P. Saikia, Simulation and analysis of modified rsa cryptographic algorithm using five prime numbers, International Journal on Recent and Innovation Trends in Computing and Communication 5 (6) (2017) 224–228.

22. R. C. Merkle, Secure communications over insecure channels, Commun. ACM 21 (4) (1978) 294–299. doi:10.1145/359460.359473. URL https://doi.org/10.1145/359460.359473
23. B. Blanchet, An efficient cryptographic protocol verifier based on prolog rules, in: Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW), 2001, pp. 82–96. doi:10.1109/CSFW.2001.930138.
24. B. Blanchet, Modeling and verifying security protocols with the applied pi calculus and proverif, Foundations and Trends in Privacy and Security 1 (1-2) (2016) 1–135. doi:10.1561/3300000004.