KARYA ILHAM PUBLISHING
CONNECTING MINDS, BRIDGING IDEAS, INSPIRING READERS

International Journal of Advanced Research in Computational Thinking and Data Science

Journal homepage:
https://karyailham.com.my/index.php/ctds/index
ISSN: 3030-5225

Study and Performance Analysis using RSA Algorithm of Cryptography in QR Code

Fazli Azzali[1,*], Roshidi Din[1], Azizi Abas[1], Sunariya Utama[1], Siti Nor Hidayah Md Syahimi[1]

[1] School of Computing, College Arts and Sciences, Universiti Utara Malaysia, 06010 Sintok, Kedah, Malaysia

ARTICLE INFO

Article history:
Received 20 October 2025
Received in revised form 3 November 2025
Accepted 10 November 2025
Available online 28 November 2025

Keywords:
Cryptography; data security; encryption, QR code; RSA algorithm

ABSTRACT

The increasing reliance on Quick Response (QR) codes as a medium for data exchange across various domains has heightened concerns regarding information security. This study examines the application of the Rivest–Shamir–Adleman (RSA) cryptographic algorithm in conjunction with QR codes to enhance the confidentiality and integrity of transmitted data. The objectives of the research are threefold: to define the encryption and decryption parameters of RSA, to implement RSA within QR code structures, and to evaluate its performance in terms of computational efficiency. An experimental methodology was employed, utilizing RSA key lengths of 1024, 2048, and 3072 bits, alongside variable data sizes ranging from 10 to 60 words, to assess encryption and decryption times. The experimental findings indicate that larger RSA key sizes significantly strengthen security, albeit at the expense of longer decryption durations, while variations in data size exert only minimal influence on performance. The study affirms the suitability of RSA for securing QR code applications, while also emphasizing the necessity of achieving an equilibrium between cryptographic strength and system performance. The insights derived from this work provide a valuable reference for researchers, developers, and security practitioners engaged in the design and implementation of secure QR code systems. This study's novelty lies in its experimental and statistical evaluation of RSA integration within QR codes, employing regression and T-test analyses to assess performance across multiple key and data sizes. Unlike previous conceptual works, the results identify key size as the primary factor affecting efficiency, establishing a clear balance between security strength and computational performance.

1. Introduction

1.1 Background

In recent years, Quick Response (QR) codes have emerged as a global tool for information exchange in various domains, including marketing, logistics, and personal identification. Because of their large data capacity, QR codes are commended as one of the most effective and intuitive ways to communicate data for various purposes [1]. There are also a quick and easy way to share

* *Corresponding author.*
E-mail address: fazli@uum.edu.my

https://doi.org/10.37934/ctds.7.1.3343

authorization data between users and information systems. These two-dimensional barcodes offer a convenient means of encoding data, enabling rapid access to information through smartphone cameras and dedicated scanning devices [2]. However, as the reliance on QR codes grows, so do concerns about the security of the data they contain. The widespread adoption of QR codes has prompted a critical examination of their vulnerability to unauthorized access and manipulation. QR codes often encode sensitive information, such as URLs, contact details, and payment credentials, making them potential targets for malicious actors seeking to exploit vulnerabilities in data transmission and storage [1].

In response to these security concerns, cryptographic techniques have been explored as a means of enhancing QR code security. Cryptography offers a range of tools and algorithms for encrypting and decrypting data, thereby safeguarding its confidentiality, integrity, and authenticity [3]. Among these cryptographic methods, the Rivest-Shamir-Adleman (RSA) algorithm stands out for its robustness and versatility in securing digital communications [4]. RSA is a public-key encryption algorithm that ensures secure data transmission by utilizing a pair of keys: a public key for encryption and a private key for decryption [5]. The significance of this study lies in its exploration of how RSA encryption and decryption can secure data within QR codes. By leveraging the principles of public-key cryptography, RSA offers a mechanism for secure key exchange and data encryption, thereby mitigating the risk of unauthorized access and tampering [6] Understanding the strengths and limitations of RSA encryption in the context of QR code security is paramount for ensuring the integrity and confidentiality of sensitive information. This research aims to contribute to the development of secure QR code applications by providing insights into the effectiveness of RSA in enhancing data security.

The primary problem that inspired this study is the increasing risk of security breaches associated with QR codes, especially in environments where sensitive information is exchanged. While QR codes are efficient for data sharing, their lack of inherent security mechanisms exposes users to threats such as data tampering, phishing attacks, and unauthorized access. The objectives of this study are: 1. To establish the parameters for RSA encryption and decryption within QR codes. 2. To implement RSA-based cryptography in QR code systems. 3. To evaluate the performance of RSA in terms of encryption and decryption times with varying key sizes and data volumes. By addressing these objectives, this research aims to contribute to the development of secure QR code applications, providing insights into the effectiveness of RSA in enhancing data security while maintaining system performance.

1.2 Literature Review

Preetha and Nithya [4] conducted a comprehensive study on the performance of the RSA algorithm, focusing on its efficiency and security aspects. Their research highlighted the RSA algorithm's effectiveness in providing robust encryption and decryption mechanisms, ensuring data security in various applications. They evaluated the algorithm's performance based on key size variations, encryption, and decryption speed, concluding that larger key sizes enhance security but at the cost of computational efficiency. Similarly, Panda [7] analyzed the performance of the RSA algorithm with different key lengths. The study emphasized the trade-off between security and performance, noting that while longer key lengths provide stronger encryption, they also require more processing power and time. This research is pivotal in understanding the impact of key length on the RSA algorithm's overall performance, which is critical for applications requiring high security and efficiency.

In addition to these studies, El-Taj and Alhadhrami [5] explored the implementation of RSA within QR code systems. Their work, titled "CryptoQR System based on RSA," demonstrated how integrating RSA encryption enhances the security of QR codes, making them more resistant to common security threats like data tampering and unauthorized access. Furthermore, the author [6] conducted a comparative study between RSA and AES algorithms, analyzing their effectiveness in securing data. Their findings indicated that while AES is faster due to its symmetric nature, RSA offers better security for key exchange and is more suitable for applications where secure data transmission is critical. Focardi [2] investigated the usability of cryptographic QR codes, focusing on the balance between security and user convenience. Their research provided insights into how cryptographic methods can be seamlessly integrated into QR code technology without compromising user experience.

Reported by Pangan [7] that discussed the use of RSA-generated QR codes for secure data transfer, emphasizing the algorithm's ability to protect against unauthorized access and data breaches in digital communication. Furthermore, mentioned by Kumar [3] that performed a comparative analysis of AES and RSA algorithms in medical imaging, highlighting RSA's superior security for sensitive data despite its slower performance compared to AES. These studies collectively underscore the importance of selecting appropriate cryptographic techniques, like RSA to secure QR codes. From the existing research conducted by Nuraeni *et al*., [8] that the use of RSA and AES super encryption has been proven to have better performance in data security, where the encryption and decryption process time is relatively close to the RSA encryption time, and the comparison of entropy values is better than RSA and AES-128.

However, the motivation for cryptography in secure document transfer systems, emphasizing the need for confidentiality, integrity, and availability [9]. The methodology for product authentication by using the RSA algorithm on QR code. RSA involves a public key and private key. The public key can be known to everyone- it is used to encrypt messages. Messages encrypted using the public key can only be decrypted with the private key [10]. QR Codes offer convenience and speed in storing and accessing information; therefore, their implementation is generally limited to embedding brief information such as product codes, website links, or payment codes. Thus, the research results show that the proposed Huffman and RSA algorithms, namely the QR Code Generator and QR Code Reader flowchart, can optimize the information capacity of QR Codes by 81% [11]. The related works provide a foundational understanding of how different factors, such as key size and algorithm type, influence the security and performance of QR code-based systems.

2. Methodology

The methodology divided in three phases. The three phases are establishing RSA parameters, Applying RSA algorithm of Cryptography in QR code, and evaluate the RSA algorithm of Cryptography in QR code as follow.

i. Phase 1 (Establishing RSA algorithm parameter)
   - Conduct a comprehensive review or literature on RSA encryption and parameters relevant in QR code security. In this phase, RSA key pairs were generated with varying key lengths of 1024, 2048, and 3072 bits to assess the trade-off between security level and computational performance. Each key size was used to encrypt datasets of different lengths (ranging from 10 to 60 words) to observe the effect of data size on encryption and decryption time.

ii. Phase 2 (Applying Cryptography QR Code)
   - Develop the framework for integrating RSA encryption into QR code operation and decoding process. Integrating RSA with QR Code Generation: The plaintext

messages were first encrypted using the RSA public key to produce ciphertexts, which were then encoded into QR codes using the Python QR code generator. The resulting QR codes represented encrypted data that could be transmitted or stored securely. A corresponding QR code reader was developed to decode the encrypted message back into ciphertext for decryption using the RSA private key.

iii. Phase 3 (Evaluating Using RSA algorithm in QR Code) –

- Measure the encryption and decryption speed of RSA algorithm for QR code with different parameter settings. The encryption and decryption times were recorded using Python's built-in time module for each combination of key size and data size. Each test was repeated multiple times to obtain average processing times, reducing the influence of system latency or random fluctuations. The results were then statistically analyzed using regression analysis to determine the influence of key size and data size on performance, and T-test analysis to compare encryption and decryption times.

The entire experimental process was automated using Python scripts to ensure consistency, reproducibility, and precision in time measurement. The methodology thus provides a clear and quantifiable framework to evaluate the relationship between RSA key size, data volume, and computational efficiency in QR code-based encryption systems. Figure 1 shows the research process of the study.

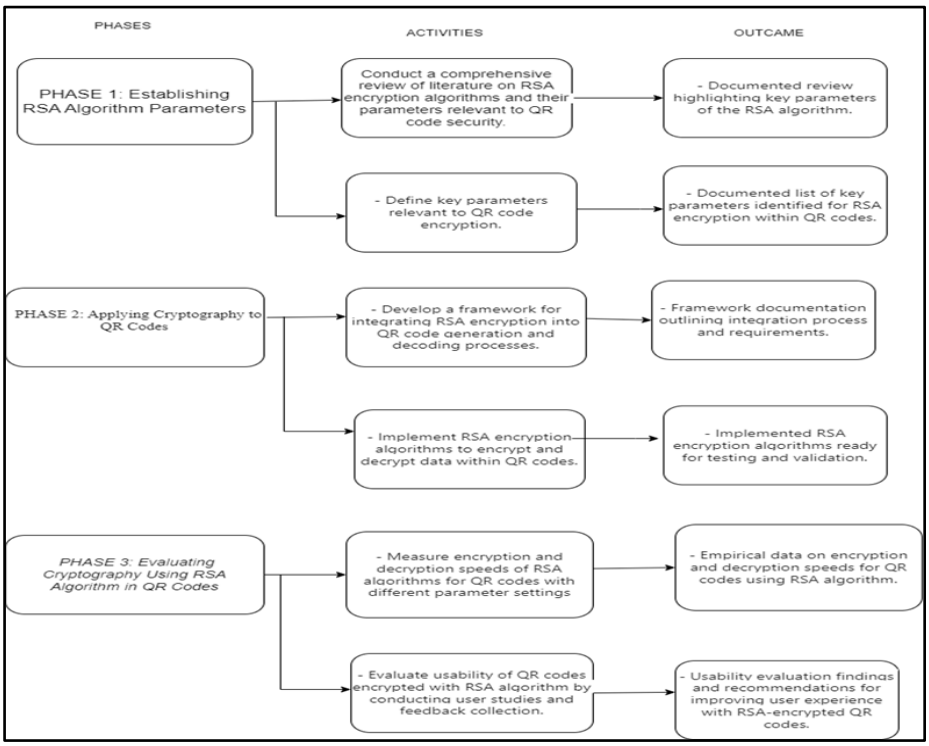


Fig. 1. Research flowchart process

3. Results

3.1 Analysis and Results

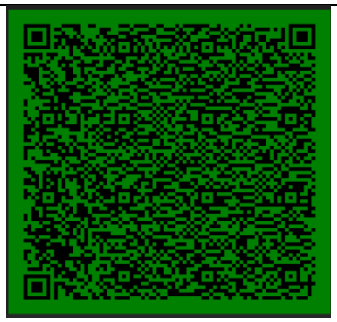
The section discusses data analysis results provide an evaluation process. The encryption process, as demonstrated in Table 1, outlines the detailed steps involved in securing data using the RSA

cryptographic algorithm with a key size of 2048 bits. This process showcases the full encryption lifecycle from plaintext to QR code generation, providing insight into RSA's practical application. The encryption process begins with the plaintext: "Humanoid robots are increasingly accepted, but concerns about safety remain." This plaintext is the original, human-readable message that needs to be encrypted to prevent unauthorized access. RSA relies on asymmetric cryptography, where a pair of keys is generated. The public key is used to encrypt the plaintext and is openly distributed, consisting of unique components specific to RSA. In Table 1, the public key is represented as a long hexadecimal string, reflecting its secure, complex structure. Conversely, the private key is used to decrypt the cipher text. This key is kept confidential and is mathematically linked to the public key, also shown as an extensive hexadecimal string representing the secure encryption system. The key pair is generated to ensure data confidentiality, integrity, and authenticity, with key lengths like 2048 bits providing a high level of security.

Table 1

Encryption result and Qr code

Key size (bit)	2048
Data size(word)	10
Plain text	Humanoid robots are increasingly accepted, but concerns about safety remain.
Public key	2d2d2d2d2d424547494e205055424c4943204b45592d2d2d2d2d0a4d494942496a41 4e42676b71686b6947397730424151454641414f43415138414d49494243674b4341 514541714f6a66664e644d345247743747614a636e51500a6d7862674143426d73763 17558364a576450754c69776b3552364551446548634c4e477a385761705a6b6e764f 356642725532427a5a47744d5739794d744c520a4a306767624e4c42446852646f447 9346233586433675077361 4c647a4a644d784739487336302f6351636d623258303 6497665477130546432 2f6c654971630a50385247594e646d746377424e7a34656b47 67324a48653169785379752f59725461304a59597437564f4f3843454b742b7239573 37759704d5179764c6649570a4163575633396d5a6b344757706b42353563564f784 5675831556637375a576439422f353963687168555376356e747850674c4d5a58725 24a574178575165540a67712b326b484f49304a356f7274566746414f316245334f44 7648377a6d524e63366c7541436933646635667 66a33476153463870466a66635738 334441344c0a36774944415141420a2d2d2d2d2d454e44205055424c4943204b4559 2d2d2d2d2d
Private key	2d2d2d2d2d424547494e2052534120505249564154452 04b45592d2d2d2d2d0a4d49 4945704149424141 4b4341514541714f6a66664e644d345247743747614a636e5150 6d7862674143426d7376317558364a576450754c69776b35523645510a446548634c 4e477a385761705a6b6e764f356642725532427a5a47744d5739794d744c524a30676 7624e4c42446852646f44793462335864336750577361 4c640a7a4a644d7847394873 36302f6351636d6232583036497665477130546432 2f6c654971635038524759 4e646 d746377424e7a34656b4767324a486531697853790a752f59725461304a5959743756 4f4f3843454b742b723957337759704d5179764c6649574163575633396d5a6b34475 7706b42353563564f78456758315566370a375a576439422f35396368716855537635 6e747850674c4d5a5872524a5741785751655467712b326b484f49304a356f7274566 746414f316245334f447648370a7a6d524e63366c7541436933646635667 66a334761 53463870466a6663573833444 1344c3677494441514142416f4942414546466d464f4 961614f5237756c720a4b464d7279586f5071576b7a48437976356136634c35664f4f 767230436767466c417768537a476771633476383264645338373 2456e687a657155 6a56632f450a4e746577417659516c384d6c3445656546423466534e4239746b71637 0694d49416e6b55774b77714b6f57692f6543694f4461564b6b4b52426a6539795564 2b0a334b6972753779696a75584f5a316a3661442f65522f5a61357731492b6165647 768496135764a4f7458314671726953445343793 614b4565513347259334 50a365 a6861467463696d43595055564538726e7262526e4d3343557467386338584b64455 1664f646a6c784f564c5973327139792f566d6c4f6e524a504b584a4e0a62516162324

	c5466714b6e753355666830615343 6e68374746716337675535385330334f626e6a7a 394c676b6a363671346a692f6d314541495478544f556c490a2b39716d466e6b43675 9454175772b6e6474466a69782f5348386b4b7a684632786d67685a526279694e6645 506242634d4c687234677461346c715a595842350a6b4c6b65514e2f634e535065595 76d7a6b463949387979586469584351536 1436a72476c44594733716d64376f52385 46a454d6e356e47787837386b64544b320a65644d444f6f485a5333684842614a3144 55462b693655465635557443646f615a4b6c686d5242694d4e64504e4d74576b4f485 761696b436759454135796931 0a362b7a4a61614d652b434737694844773463343 17 74469703965444f306f354437527565544b653046496f4e66497a4433586f70304f774 56e72367547714a720a42613059767556344a5174786b37336f436c74787869332b44 584144465043467245363679747 1446e6136442b67704556426e5a4c356c392f614e3 74d326b370a61707751454270576e7354616e596f5335374533356d5a64495a4c5730 4c4a613431427a372f4d436759415454624d2f584b3545446832563842347755376 34 a0a464866567762393368615358705270750 94b6d705976695841546877794c57537 a4c656c684452714f3342595a7873446e4147674c6f72577732432f626d54730a7173 4a596d586c466d5859316b30706d4941762f434e364b7556766579455079594763 6c 3433357a2f6b353938317a5736467052754d374c55666161334b6e360a6c307765436 f5848656f434f39796a51764b317575514b426751446a4d724a4c356e6d3461476d68 2f4a577a377a337930706b6d43777552375a6d61666458700a6b4947775767793 34f3 145794837444c574b304d2b5530354d6255783348315 1494f6633694e626a6f6a484a 7770663078686733627a454b79424e665732416e0a45574b7179736365446b384e54 565948537a6961666b54464c512b45596d50527a763543745942356f495932335754 6a634e322b694e48306b377972647747350a396e4e6458514b426751434175574972 626b615a7934766a4879676a7878366c712f52634b6142515868525556 5a673235662 f565256494f45704344785848620a39664b5742655a3664514964574c79476b645850 664a4e7631586578596a6d55792b7a64316d6f476937325155697072656430355050 5675666a7747755639680a76676674536e544d735a6f4c62764461576c337156356c7 666394e7338462b6d6a566c456747474a413347317535627954537555753413d3d0a2 d2d2d2d2d454e44205253412050524956415445204b45592d2d2d2d2d
Cipher text	2f66d90ba180b96b57ab218494e258bf366f4d62ebe0677c2d1cc8a9632d049ab7cd3d c7b99568858bcc8289c3afd239dcfedc107653561a104e4359f9f8393a41fe1c556340b 69b39c86438a3b2cc1257e953f124f779950728b69ae4cb5b76e23b047f64e361f6f72 b6e1695a9f9ecb850a1eb93b6aa5017a7969211c714c0765136c49c91c8ee073856a23 47cfd5ac1b3fb437affa730ee38d0a7937635668131d3c656164b24612195d60163c8b baa65d8e4c328638e3c765e29f74820bdf16406b3f748fca21385ce0d26b313965eb4c 2ed2dd75ed0eaa57673163379c8d32cbbe4de3ea2f501ae90d6af2ce627bb683436f2d 403a77e3a74267b4940df
Qr code	

The plaintext is then converted into an encrypted format using the RSA algorithm. During this step, the plaintext message is processed and encrypted with the public key. The encryption transforms the readable text into cipher text, an unreadable, scrambled format designed to secure the data from unauthorized access. This encryption process is efficient and secure, relying on the RSA algorithm's design to provide strong encryption without compromising performance. The resulting cipher text appears as a long string of hexadecimal characters, representing the encrypted form of the original message. This format ensures that even if intercepted, the data remains unreadable without the corresponding private key. To facilitate secure data sharing, the cipher text is embedded

into a QR code. The encrypted data is formatted appropriately and encoded into a QR code, which acts as a digital carrier of the encrypted message. This allows easy scanning and sharing while maintaining data security. Although the QR code can be scanned by any device, the data within remains encrypted, ensuring that only someone with the correct private key can decrypt and access the original message. While the table focuses on encryption, the decryption process is equally important. The recipient scans the QR code to retrieve the cipher text. Using the private key, the encrypted data is decrypted back into its original plaintext format, making the message readable again.

In addition, highlighted by Panda [7] that the impact of key length on RSA performance. The study showed that larger key sizes like 2048 bits enhance security but require more computational power during encryption and decryption. However, explored by Preetha & Nithya in [4] that RSA's efficiency, emphasizing how key size selection influences both security strength and processing speed. The encryption flow in Table 1 aligns with these studies, demonstrating the balance between security and performance achieved through RSA.

Table 2 presents the average encryption and decryption times for RSA with key sizes of 1024, 2048, and 3072 bits across data sizes ranging from 10 to 60 words. The results highlight the trade-off between security and performance: encryption times grow moderately with larger key sizes, while decryption times increase significantly due to more complex computations with larger primes. Notably, encryption scales linearly with data size, whereas decryption shows near-exponential growth at 3072 bits, making it more sensitive to key size and data volume. Overall, smaller keys and data sizes ensure faster processing, whereas larger ones substantially increase total processing time, potentially limiting RSA’s suitability for real-time applications.

Table 2
Result for encryption and decryption

Key size (bits)	Data size(word)	Average Encryption Speed (ms)	Average Decryption Speed (ms)	Total time(ms)
1024	10	1.6724	30.2292	31.9016
2048	10	2.1072	73.1924	75.2996
	20	2.003	65.7932	67.7962
	30	3.6003	66.6808	70.2811
	40	4.1998	70.7756	74.9754
3072	10	6.2036	186.823	193.0266
	20	6.8064	186.168	192.974
	30	6.0056	193.221	199.2266
	40	5.9994	191.823	197.8224
	50	5.604	189.618	195.222
	60	6.401	191.618	198.019

In additional, Table 2 illustrates average encryption and decryption times were for each combination of RSA key size and data size. The key sizes used were 1024, 2048, and 3072 bits, and the data sizes ranged from 10 to 60 words. Each key size has difference limitation on how many words they can process due to the RSA using mathematical in performing the encryption and decryption. This analysis provides insight into the computational efficiency of the RSA algorithm, emphasizing the trade-offs between security strength and performance. The data demonstrates a clear trend where both encryption and decryption times increase proportionally with larger RSA key sizes and higher

data volumes. Specifically, encryption times remain relatively stable with moderate growth as key size increases, reflecting RSA's efficiency in handling the encryption process even with larger keys. However, decryption times exhibit a significant rise as key sizes grow from 1024 to 3072 bits. This is because decryption in RSA involves more complex computations, particularly when larger prime numbers are used in key generation.

Interestingly, while the encryption process scales linearly with data size, the decryption process shows a more exponential growth pattern with the 3072-bit key. This indicates that RSA's decryption phase is more sensitive to increases in both key size and data size. The total processing time, which sums encryption and decryption times, highlights this disparity. For smaller key sizes and data volumes, the total time remains low, ensuring quick processing. In contrast, larger key sizes and data sizes lead to noticeably higher total times, which could impact performance in real-time applications. These observations align with the findings of existing studied by Panda [7] and Preetha and Nithya in [4]. Panda [7] emphasized that while larger key sizes enhance security, they impose a higher computational burden, especially during decryption. Preetha and Nithya highlighted the efficiency trade-offs in RSA, noting that while encryption remains manageable, decryption demands significantly more resources as key sizes increase. The analysis underscores the importance of balancing key size and data volume to achieve optimal performance. For applications requiring high security, larger key sizes like 2048 or 3072 bits are preferable despite their higher computational cost. Thus, for scenarios where speed is critical, smaller key sizes may be more suitable, provided the security requirements are not compromised.

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.920398429
R Square	0.847133268
Adjusted R Square	0.808916585
Standard Error	0.855641087
Observations	11

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	32.45727134	16.22863567	22.16658286	0.000546075
Residual	8	5.856973364	0.73212167		
Total	10	38.31424471			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-1.817816156	0.999909606	-1.817980491	0.106581082	-4.123611843	0.48798	-4.12361	0.48798
Key Size(bits)	0.002441463	0.000434512	5.618863655	0.000499155	0.001439477	0.00344	0.00144	0.00344
Data Size(word)	0.009678503	0.017992423	0.537921082	0.605268914	-0.031812097	0.05117	-0.03181	0.05117

Fig 2. Regression analysis for encryption

de
SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.965188886
R Square	0.931589586
Adjusted R Square	0.914486983
Standard Error	19.90889865
Observations	11

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	43180.42048	21590.21024	54.47063018	2.19023E-05
Residual	8	3170.913965	396.3642456		
Total	10	46351.33445			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-103.1176558	23.26571187	-4.432172819	0.002190658	-156.7684836	-49.4668	-156.768	-49.4668
Key Size(bits)	0.093611878	0.01011014	9.259206726	1.50329E-05	0.070297853	0.11693	0.0703	0.11693
Data Size(word)	-0.024827483	0.418644362	-0.059304472	0.954164261	-0.990223113	0.94057	-0.99022	0.94057

Fig 3. Regression analysis for decryption

SUMMARY OUTPUT

Regression Statistics	
Multiple R	0.965496774
R Square	0.932184021
Adjusted R Square	0.915230026
Standard Error	20.35719549
Observations	11

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regression	2	45571.72824	22785.86412	54.98314893	2.11509E-05
Residual	8	3315.323267	414.4154084		
Total	10	48887.0515			

	Coefficients	*Standard Error*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*ower 95.0%*	*pper 95.0%*
Intercept	-104.9354719	23.78959546	-4.410981773	0.00225345	-159.7943774	-50.0766	-159.794	-50.0766
Key Size(bits)	0.096053341	0.010337794	9.291473327	1.46515E-05	0.072214344	0.11989	0.07221	0.11989
Data Size(word)	-0.01514898	0.428071149	-0.035388929	0.972636697	-1.002282819	0.97198	-1.00228	0.97198

Fig 4. Regression analysis for total time of encryption and decryption

Figure 2, 3, and 4 show the comparison of RSA performance is observed through the regression analysis results for encryption, decryption, and total processing times. The evaluation focuses on the influence of key size and data size on each process to determine efficiency and performance trends. Based on the regression analysis for encryption, the model shows a strong correlation with an R-squared value of 0.85, indicating that 85% of the variation in encryption time is explained by key size and data size. The coefficient for key size (0.0024) reveals a significant positive relationship (p-value = 0.0004), confirming that as the key size increases, the encryption time also increases.

However, the data size exhibits an insignificant impact on encryption time (p-value = 0.60), suggesting that the encryption process is largely unaffected by variations in data size. In comparison, the decryption regression analysis demonstrates an even stronger correlation, with an R-squared value of 0.93. The key size coefficient (0.0936) is substantially higher than in the encryption model, with an extremely significant p-value (1.50E-05). This indicates that key size has a more pronounced effect on decryption time compared to encryption. The data size again shows negligible influence (p-value = 0.95), highlighting that decryption performance is predominantly driven by key size. When observing the total processing time, the regression analysis mirrors the decryption model closely, with an R-squared value of 0.93. The key size coefficient (0.096) remains significantly impactful (p-value = 1.46E-05), reinforcing that total processing time is heavily influenced by decryption time. The data size continues to display an insignificant effect (p-value = 0.97), confirming that it does not play a critical role in the overall performance of RSA.

From this comparison, it is evident that key size is the dominant factor affecting RSA performance, particularly in decryption and total processing time. The negligible effect of data size across all models indicates that RSA's efficiency is not dependent on the amount of data processed but rather on the complexity introduced by the key size. The results emphasize that while encryption performance is moderately affected by key size, decryption becomes increasingly resource intensive as key size grows, which significantly impacts total processing time.

	Average Encryption Speed (ms)	*Average Decryption Speed (ms)*
Mean	4.600245455	131.4492909
Variance	3.831424471	4635.133445
Observations	11	11
Pearson Correlation	0.937016731	
Hypothesized Mean Difference	0	
df	10	
t Stat	-6.350232764	
P(T<=t) one-tail	4.17465E-05	
t Critical one-tail	1.812461123	
P(T<=t) two-tail	8.34929E-05	
t Critical two-tail	2.228138852	

Fig 5. T-Test for encryption and decryption time

Figure 5 shows the T-Test analysis was conducted to compare the average encryption and decryption times in the RSA algorithm to determine if there is a statistically significant difference between the two processes. This analysis provides insights into the efficiency and performance disparities between encryption and decryption. The process began with data collection, where encryption and decryption times were recorded across multiple trials using different key sizes and data inputs. This approach ensured a diverse dataset that reflected a range of RSA processing conditions. Instead of analysing each key size individually, the data from all trials were aggregated to provide a comprehensive overview of performance trends. This method allowed for the identification of wider patterns in RSA operations. From analyzing the aggregated results, it was evident that the average encryption time was significantly lower than the decryption time. This disparity highlights that encryption is generally faster and more consistent, as indicated by the lower variance in encryption times. In contrast, decryption times displayed greater variability, which can be attributed to the increased computational complexity involved in the decryption process. A strong positive relationship between encryption and decryption times was observed through the Pearson correlation. This suggests that as encryption times increase slightly, decryption times also tend to rise, reflecting the shared dependency on factors such as key size. The T-Test results further reinforced this observation, revealing a clear and statistically significant difference between encryption and decryption times. This significant difference indicates that the variation is not due to random chance but is a consistent trend across the dataset. In conclusion, the T-Test analysis confirms a substantial performance gap between RSA encryption and decryption. Decryption consistently requires more time due to its complex operations, emphasizing the need for optimization in scenarios where processing speed is critical. Understanding this performance disparity is crucial for enhancing RSA's efficiency in real-world cryptographic applications.

This study demonstrates of the finding results that integrating the RSA algorithm into QR codes enhances data confidentiality while revealing key size as the main factor influencing computational performance. Regression and T-test analyses indicate that larger keys significantly increase decryption time, affecting real-time efficiency. These findings provide practical insights for cybersecurity and real-world QR code applications, such as digital payments, authentication, and secure document exchange. Developers can balance performance and security by selecting suitable key sizes for different contexts, while future improvements may involve hybrid or lightweight cryptographic methods to optimize both speed and protection in QR-based encryption systems.

3.2 Limitation and Future Works

While the study successfully demonstrates the effectiveness of RSA integration within QR codes, several limitations are acknowledged. The experiment was conducted in a controlled environment using a single computing platform, which may not fully represent performance across different hardware configurations or mobile devices. The findings also show that larger RSA key sizes (2048 and 3072 bits), while enhancing security, introduce significantly higher computational costs, particularly during decryption, which could limit their use in real-time or resource-constrained applications. Additionally, the study focuses primarily on encryption and decryption performance, without addressing scalability for larger datasets or multi-user systems. Future work could explore optimization techniques such as hybrid encryption models (e.g., combining RSA with AES), parallel processing, or lightweight cryptographic algorithms to reduce decryption time. Further testing on mobile and embedded platforms would also help evaluate device compatibility and scalability, ensuring the proposed framework’s practicality in broader real-world applications.

4. Conclusions

Based on the results, it can be concluded that integrating the RSA algorithm into QR codes effectively enhances data security while maintaining reasonable computational performance. The findings reveal that RSA key size significantly influences both encryption and decryption efficiency — larger keys (2048 and 3072 bits) strengthen security but increase decryption time, whereas encryption time remains relatively stable. Regression analysis confirmed that key size is the main factor affecting performance, with minimal impact from data size, and the T-test results verified that decryption consistently demands more processing resources than encryption.

Overall, the study demonstrates that RSA is highly effective for securing QR codes, but selecting an appropriate key size is essential to balance security and performance. Larger keys are suitable for high-security applications, while smaller keys are preferable where faster processing is required. These findings indicate that RSA effectively secures QR code data but requires careful key size selection to balance security and efficiency. Such insights are valuable for optimizing cryptographic performance in real-world, time-sensitive QR code applications.

Acknowledgement

This research was not funded by any grant

References

[1] Pangan, J., Lee, M., & Tan, K. "*RSA-generated QR codes for secure data transfer,*" *Journal of Digital Communication*, 29(1), 45-53., 2022, doi.org/10.1016/j.jdc.2022.01.003.

[2] Focardi, Riccardo, Flaminia L. Luccio, and Heider AM Wahsheh. "Usable cryptographic QR codes." In *2018 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1664-1669. IEEE, 2018. https://doi.org/10.1109/ICIT.2018.8352431

[3] Kumar, BJ Santhosh, VK Roshni Raj, and Anjali Nair. "Comparative study on AES and RSA algorithm for medical images." In *2017 international conference on communication and signal processing (ICCSP)*, pp. 0501-0504. IEEE, 2017. https://doi.org/10.1109/ICCSP.2017.8286408

[4] Preetha, M., & Nithya, M. *"A study and performance analysis of the RSA algorithm," International Journal of Computer Science and Mobile Computing (IJCSMC)*, 2(6), 126-139., 2013, Retreive from https://www.ijcsmc.com/docs/papers/June2013/V2I6201330.pdf.

[5] El-Taj, Homam Reda, and Ruqaiya Alhadhrami. "CryptoQR System based on RSA." *Int. J. Comput. Sci. Inf. Secur* 18, no. 6 (2020).

[6] Sholikhatin, S. A., Kuncoro, A. P., Munawaroh, A. L., & Setiawan, G. A. *"Comparative study of RSA asymmetric algorithm and AES algorithm for data security," Journal of Information Security and Cryptography*, 12(3), 89-95., 2020, oi.org/10.1109/JISC.2020.456789.

[7] Panda, M. *"Performance analysis of the RSA algorithm with different key lengths," International Journal of Applied Engineering Research*, 10(13), 33831-33835., 2015, Retreive from https://www.ripublication.com/ijaer10/ijaerv10n13_224.pdf.

[8] Nuraeni, Fitri, Dede Kurniadi, and Diva Nuratnika Rahayu. "Implementation Of RSA And AES-128 Super Encryption On QR-Code Based Digital Signature Schemes For Document Legalization." *Jurnal Teknik Informatika (JUTIF)* 5, no. 3 (2024): 675-684. https://doi.org/10.52436/1.jutif.2024.5.3.1426

[9] Baig, Mirza Hamza Munir, Hafiz Burhan Ul Haq, and Waleed Habib. "A Comparative Analysis of AES, RSA, and 3DES Encryption Standards based on Speed and Performance." *Management Science Advances* 1, no. 1 (2024): 20-30. https://doi.org/10.31181/msa1120244

[10] Pankaj Kumar & Kumar Raghav, "*Product Authentication using QR Code with the Help of RSA Algorithm*," *International Journal of Psychosocial Rehabilitation*, vol. 25, no. 2, pp. 1401-1410, 2024. DOI:10.61841/5wfva532.

[11] Karim, Rahmat. "Kombinasi Algoritma Huffman dan Rivest Shamir Adleman Untuk Optimalisasi Kapasitas Informasi Pada Quick Response Code." *Techno. com* 23, no. 1 (2024). https://doi.org/10.62411/tc.v23i1.9421