International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Quantum Resilient RSA Variants for cloud security

Prince Asare Aning
Department of Computer Science
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

Narain Kobby Gaisie
Department of Computer Science
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

Dr. Dominic Asamoah
Department of Computer Science
Kwame Nkrumah University of Science and Technology
Kumasi, Ghana

***Abstract* - Rivest Shamir Adleman (RSA) has proved to be less secured due to the advent of Quantum Computing, as it is broadly relied on in protecting cloud systems and applications. This study proposes a quantum-resilient cryptographic framework that combines RSA with quantum Kyber512 algorithm and advanced encryption standard (AES) to enhance cloud communications defense against security risks from traditional and quantum attacks. We use Kyber512 to safely encapsulate keys that then become AES-128 session keys for encryption in encrypt-authenticate-exchange (EAX) mode. After generating these AES keys, they are twice protected by using RSA-2048 encryption. We tested this method over a range of message sizes from 64 to 16,384 bytes. Results from our experiments showed a shorter average encryption time of 0.00206 seconds with a minimum of 0.0009 seconds and a maximum of 0.0057. seconds. Decryption was fast, with an average of 0.0036 seconds ranging between 0.0033 and 0.0039 seconds. Nothing changed in the ciphertext expansion, as 768 bytes were always used for Kyber and 256 bytes for the AES key that is RSA encrypted. We processed Shor's algorithm by simulating it with small RSA keys, revealing its capability to read public and private keys. Our method is shown to have speed and security benefits and emerged at the top compared to other pioneering encryption modules, making it practical for flexible use in major cloud environments. By using this method, cloud security receives a practical encryption solution compatible with upcoming technologies that defend against both modern and future quantum threats.**

Keywords – Quantum resilience, hybrid encryption, RSA variants, lattice cryptography, cloud security

I. INTRODUCTION

The growth in using cloud systems for managing and transferring confidential information means that effective and secured cybersecurity solutions are essential and imminent. RSA, which is broadly relied on for ensuring confidentiality and integrity of information in the cloud as thought to be indestructible encryption is now threatened by the advancement of quantum computing. If we apply Shor's algorithm to RSA encryption, it could quickly factor large prime numbers which is the computational foundation of RSA encryption [1]. Traditional and classical computers on the other hand would take centuries to execute this task. Since scalable quantum systems may be developed soon, researchers now see the limitations of classical cryptography in that, it cannot be trusted to defend against quantum attacks [2]. As a result of recent work in Post Quantum Cryptography, NIST has recommended Kyber512 as a candidate for future quantum-safe key encapsulation exchange mechanisms and it mainly relies on an algorithm lattice-based hardness assumptions as the computational foundation [3]. While post-quantum alternatives like Kyber768 exhibit high security, we still need to integrate these algorithms with systems and cloud solutions that use classic RSA infrastructure [4]. There have been proposed pioneering models that constitute the combination of RSA and AES, although they do not offer sufficient protection from quantum adversaries, mainly in cloud situations where maintaining confidentiality for an extended period is vital [5]. In addition, current cryptographic methods face issues when trying to balance performance, security, and scaling their solutions [6]. Many common cryptographic structures do well in traditional threat scenarios, yet their use with post-quantum components becomes clumsy or hard on systems thereby introducing overheads in processing by the existing computing system [7]. Most cloud systems rely on RSA encryption today, the safety and confidentiality against the emerging threats posed by quantum computing technologies are threatened [8]. Due to the development of algorithms such as Shor, the standard security of RSA is weakening as quantum computers are thought to factor such integers exponentially faster than current computers [9]. These threats could severely damage the long-term confidentiality and trust that cloud platforms rely on. Most strategies for dealing with these risks are either classical cryptographic methods that cannot resist quantum attacks or unique post-quantum primitives with sophisticated infrastructure that potentially can disrupt the existing systems, increasing overheads in processing, with limited ways to ease adoption [10]. Moreover, many hybrids cryptographic encryption systems do not provide a balance between security and performance, with no clear integration policies, which hinders their implementation into cloud systems. Also, most cloud security frameworks based on RSA won't be able to adapt to post-quantum standards, and they lack the protection

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

against quantum attacks that true layered encryption can give [11]. Consequently, these providers have insecure or unfit solutions that cannot meet the twin needs for interoperability and future resilience. This challenge clearly depicts the major gap in the traditional cryptographic environment, thus the lack of an encryption system that combines the efficiency of RSA-based technologies with keen attention to interoperability with current systems, quantum and post-quantum resilience, and protection modern systems demand. To address this global problem, this research proposes and evaluates a hybrid encryption framework that combines RSA, AES, and Kyber512 to ensure secure, scalable, and future-proof data communication in cloud environments. This paper introduces a novel hybrid encryption framework that combines RSA, AES, and the post-quantum Kyber512 algorithm to secure cloud communication systems. Unlike previous models that either lack quantum resistance or fail to support legacy infrastructures, the proposed model offers forward secrecy, post-quantum key encapsulation, and classical interoperability. The system has remarkably low encryption/decryption latency and predictable ciphertext expansion as the message size changes, which allows it to be used in real-time cloud settings that are quantum-era forward defensible. In summary, the significance of this work is as follows:

- The study proposed a hybrid system that utilizes RSA, AES, and Kyber512 to encrypt cloud data safely, scalable, and resilient against quantum attacks.
- A modular system that uses Kyber512 for quantum secure key encapsulation and AES for fast symmetric encryption, with RSA used to wrap the symmetric key for classical compatibility.
- Accesses how well the framework performs in encrypting and decrypting messages of different sizes about speed, the size of the encoded text and computational efficiency.

The subsequent sections of the paper are organized with the background and related works in Section 2. The methodology used for the study is highlighted in Section 3. Experimental results and findings from the study are presented and discussed in Section 4. Eventually, we present the study's conclusion and possible areas for future work.

II. RELATED WORKS

A. Cloud Security in the Digital Era

Cloud computing has transformed information and data management procedures of firms and persons by handling data at scale, flexible, and providing cost-efficient alternatives, in storing, processing, and deploying applications [12]. But along with the advantages of cloud computing, there are security issues that are implicit [13]. Distributed cloud environments, multi-tenancy, and third-party data manipulation expand the possible attack surface and generate vulnerabilities that can be used by adversaries [14]. Data confidentiality is one of the essential issues in cloud computing [15]. Organizations trust cloud service providers (CSPs) to keep sensitive data including financial details, personal data, intellectual property, and business proprietary information. Poor protection may result in data leaks and unauthorized access as well as insider threat [16]. When data is sent over shared infrastructure and open networks, it becomes vulnerable to Eavesdropping, man-in-the-middle attacks, and malicious interception [17]. Data integrity is yet another important security aspect of the cloud [18]. It is necessary that stored and transmitted data cannot be altered, to keep the trust and accuracy of operations. Alteration of information either at rest or in transit can lead to disastrous consequences, particularly in healthcare, finance, and defense industries [19]. To prevent such risks cryptographic hashing, digital signatures, and authentication protocols are often employed. The cloud also presents a challenge in terms of data availability, and service continuity. The unavailability of cloud services may be due to Distributed Denial of Service (DDoS) attacks, system failure, and configuration errors, which influence the conduct of business and lead to potential loss of revenues [20]. The cloud providers mitigate these threats through redundancy, load balancing, and incident response [21]. However, the most important aspect of cloud security strategy is encryption [22]. The cloud security paradigm is complicated by the emerging quantum threat. Classical cryptographic algorithms such as RSA and ECC become vulnerable as quantum computers remain more viable. Historical data may be decrypted by quantum adversaries and quantum pseudonyms may enable attackers to weaken long-term confidentiality. To recap it all, cloud security is a multifaceted issue, which embraces confidentiality, integrity, and availability as well as regulatory compliance. In this ecosystem, encryption is one of the foundation technologies. Nevertheless, the emergence of quantum computers and more advanced cyber threats require transition to more secure, scalable and quantum-resistant encryption schemes. This paves the way to the combination of classical encryption with quantum schemes such as Kyber512 to secure sensitive data in the cloud, both in the current and future environment.

B. Classical Cryptography and Its Limitations

Classical encryptions rely on RSA to distribute the keys asymmetrically and AES data encryption to provide high-performance symmetric encryption. The merits of RSA include the hardness of factoring large integers, whereas AES block ciphers perform well and are efficient [23]. Hybrid solutions in which RSA keys are used to protect AES session keys and AES to protect cloud payloads achieve high security and efficiency levels in cloud environments [24]. However, quantum computing is a threat to RSA and AES. The Shor algorithm makes ECC and RSA vulnerable, as it allows efficient computation of the factorization and discrete logarithms [25]. At the same time, Grover's algorithm makes the brute force key search $\sqrt{N}$ times faster, practically decreasing symmetric algorithms such as AES in half [26]. Under Grover, quantum calculations indicate that AES-256 is exposed to an effective 128-bit of security, which is acceptable only as keys are elongated [27]. In the meantime, RSA and ECC are completely vulnerable to breakage when practical fault-tolerant quantum computers become available. As a result, classical hybrid RSA-AES systems, even those that are performant to the current cloud infrastructure, cannot offer long-term forward secrecy or quantum resistance,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

requiring the deployment of quantum cryptographic primitives [28].

C. AES-ECC Hybrid Cryptographic Model

The combination of traditional symmetric and asymmetric cryptography algorithms to increase the performance and security of data protection systems has been the direction of research in hybrid encryption and cloud security. According to Muhammed et al., [29], which assessed the ability of three hybrid systems, AES-RSA, AES-Elliptic Curve Cryptography (ECC), and AES-ElGamal, across various file formats and, thus, ascertain their relevance to cloud data storage as was developed. They reported that the AES-RSA had a good trade-off between the speed of encryption and high computational cost, taking on average 0.1667 seconds of encryption time and 0.1953 seconds of decryption time. AES-ECC, with a reputation for a small key size and enhanced security, also had higher encryption times, 0.7902 seconds, which means that it can only be useful on non-real-time applications. AES-ElGamal, however, turned out to be the poorest performer, with the time used to encrypt and decrypt averaging 0.9648 seconds and 0.3393 seconds, respectively. None of the three models involved the use of quantum-resistant cryptography, as they all offered added security compared to the standalone use of AES or RSA. All combined results from encryption schemes are shown in table I. It is indeed a major shortcoming of this approach since all asymmetric ciphers (RSA, ECC, ElGamal) fall prey to the Shor algorithm when very large-scale quantum computers are available. This is a major research gap that is filled by this current research by adopting Kyber512 in the hybrid model. Besides offering far superior encryption and decryption times of 0.0021 seconds and 0.0036 seconds, the QR-RSA paradigm also offers post-quantum security, thus establishing a new standard in a hybrid encryption paradigm. This makes it an applicable solution to future secure applications and not its classical counterpart.

TABLE I. Average Performance Metrics from AES-Based Hybrid Cryptosystems (Muhammed et al., 2024).

Encryption Scheme	Avg. Encryption Time (s)	Avg. Decryption Time (s)
AES-RSA	0.1667	0.1953
AES-ECC	0.7902	0.2232
AES-ElGamal	0.9648	0.3393

D. Comparative Review of AES-ECC and Post-Quantum Hybrid Encryption Models

Rehman et al., [30] recommended a hybrid encryption algorithm using AES and Elliptic Curve Cryptography (ECC) to enhance the security of cloud data in relation to data leakage, unauthorized access, and computational efficiency. The results from their model are presented in table II. They employed ECC to generate keys and AES to encrypt the data, which made their system use less encryption overhead when compared with the conventional encryption protocols, and more confidentiality was achieved, especially in limited-resource systems like mobile platforms. The AES ECC model, however, is not quantum resistant since ECC, similarly to RSA, is vulnerable to the Shor algorithm. Alternatively, the Quantum-Resilient RSA (QR-RSA) system that is described in this study incorporates Kyber512, a post-quantum cryptographic algorithm based on lattices, that can guarantee protection against advanced threats that will occur in the quantum era. Using classical efficient architecture and implementing quantum-safe key encapsulation, the model in question offers performance benefits with a long-term security profile that would be applied to modern cloud networks.

TABLE II. Results from AES-ECC Model by Rehman et al. (2021).

Author	Encryption Time (s)	Decryption Time (s)
Rehman et al., (2021)	0.0025	0.0019

III. METHODOLOGY

A. Research Design

Fig.1. shows the proposed design of the hybrid quantum-resilient encryption framework designed for secure and efficient data transmission in cloud environments. The proposed scheme integrates a classical RSA algorithm with the post-quantum lattice-based Kyber512 Key Encapsulation Mechanism (KEM) from the CRYSTALS suite, forming a composite cryptosystem that leverages the security of RSA and the quantum resistance of Kyber. Additionally, AES is employed for high-speed symmetric encryption of the actual data payload, making the overall framework suitable for real-time cloud applications. Besides data collection, the framework has four key steps which are: key generation, key establishment, data encryption and data decryption.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

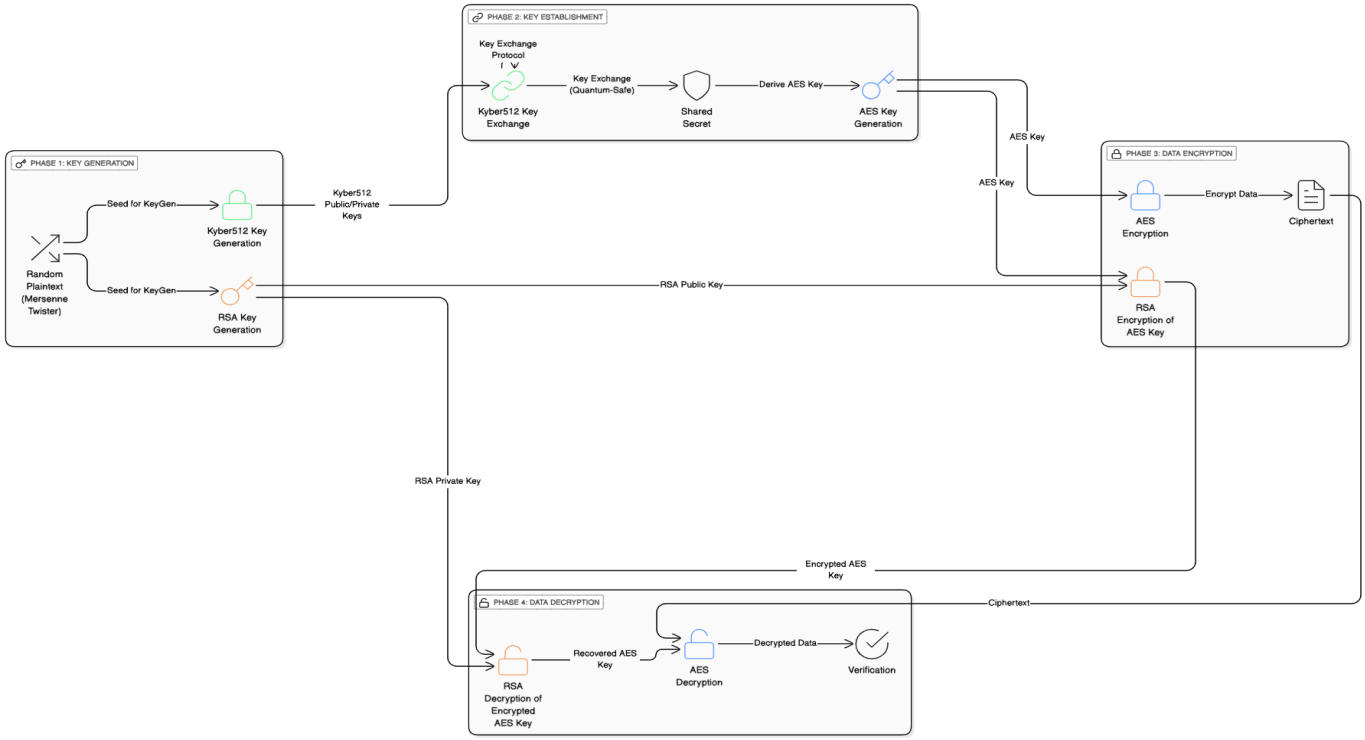

Fig. 1. Proposed design of QR-RSA encryption framework

B. *Data Collection*

The data utilized in the experiment was a simulated dataset to assess the computational efficiency and reliability of the RSA Kyber hybrid encryption to process different-sized data as may be experienced in cloud-based communication. As the goal is neither classification nor learning with data, but evaluating the encryption performance over realistic payload sizes, plaintext samples were dynamically generated in the application with a secure random number generator. All plaintext inputs consisted of randomly generated characters, expressed in the UTF-8 encoding, similar to the nature of the data that would be sent in a real-world cloud setup (authentication tokens, user metadata, log entries, documents, and excerpts). Five input sizes were chosen to experiment with as given below:

• 64 bytes, the typical size of small messages, for example, API keys or tokenized credentials.
• 256 bytes - a good size for encrypting JSON or parameter files.
•1024 bytes (1 KB) -representing small textual documents or logs.
•4096 bytes (4 KB) -standard block-sized cloud data transfer simulation. 16384 bytes (16 KB) - simulating huge text payloads or file partial uploads.

In each size category, Python random and string libraries were used to create random alphanumeric strings. The strings thus took the encryption pipeline, consisting of three sequential steps:

• Encapsulation of a Kyber512 public key to derive a shared AES key,
• AES encryption of the resulting plaintext with the derived symmetric key in EAX mode.
• RSA encryption of the AES key with a 2048-bit RSA public key.

This programmatically generated and in-memory controlled dataset guaranteed consistent, reproducible, and random Dataset between test runs. This approach removed all external factors like Input/output (I/O) overhead or file format discrepancies and permitted benchmarking of the encryption and decryption times to be done reliably under the same conditions.

C. *Mersenne Twister Algorithm for Random Plaintext Generation*

The system leverages the random module of the Python programming language to generate realistic data payloads in cloud environment by utilizing the Mersenne Twister (MT19937) pseudo-random number generator. This is started by initializing the generator and character set that is made up of upper cases, lower case letters, and digits. A desired plaintext size k is selected from a predefined set (64, 256, 1024, 4096, or 16,384 bytes), after which k characters are randomly sampled with replacement using random.choices(). These characters are joined into a single string and encoded in UTF-8 format to form the final plaintext, which is then forwarded to the AES encryption module. This process ensures uniformity, reproducibility, and randomness suitable for encryption benchmarking. A plaintext message m is defined as n random bytes. Each byte b_i is independently drawn from a uniform distribution over the values 0 to 255 and belongs to the modular space Z_{256} corresponding to 8-bit

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

binary representation. Given a message length $n \in \{64,256,1024,4096,16384\}$, a random message $m \in \{0,1\}^{8n}$ is generated that:

$$m = \{b_1, b_2, b_n\},\ b_i \sim uniform(0,255) \quad b_i \in Z_{256} \qquad (1)$$

D. *Kyber 512 key generation with Module-Learning with Rounding (MLWR) Algorithm*

Kyber512 key generation process starts with initializing system-level entropy or pseudorandom generator like Mersenne Twister with entropy of secure random seed source. This seed is used to initiate the Kyber512 lattice-based cryptographic engine, which outputs a public/private key pair rooted in the Module Learning with Rounding (MLWR) problem. In mathematical computation, A represents matrix of uniformly random polynomials, S is a small secret vector, e is a small error vector drawn from a noise distribution, q is a prime modulus and c becomes the public key. As depicted in equation 2, the public key is later used in the Kyber512 Key Exchange to securely encapsulate a symmetric AES key, while the private key remains with the recipient for subsequent decryption or decapsulation. Making the generated key pair resilient to attacks by threat actors ensures quantum resistance affirming its suitability for secure communication. The mathematical formulation of the MLWR-based key generation is the following:

$$c = A \cdot s + e \mod q \qquad (2)$$

E. *RSA key generation phase*

As shown in Fig. 2. the RSA key generation phase starts with the input of a secure random seed, which initializes the key generation mechanism. This seed derived from system entropy make sure that generated keys are both secure and unpredictable as well. Using this entropy source, the system generates an RSA key pair, which is made up of a public key and a private key. The RSA public key is directed toward Phase 3 as previously shown in figure 1 where it is used to encrypt the AES session key during the hybrid encryption process. The RSA private key is reserved for Phase 4 as previously shown in Fig. 1, where it is used to securely decrypt the AES key and enable access to the protected message content. This structured key output ensures that asymmetric encryption operations can be securely and efficiently integrated within the overall QR-RSA hybrid model, providing backward compatibility and layered security alongside the quantum Kyber512 scheme.

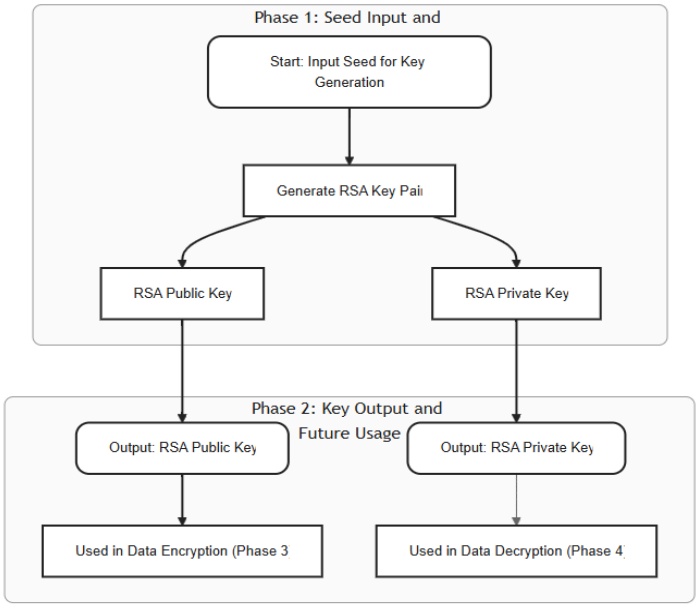

Fig. 2. RSA key generation flow in the QR-RSA framework

F. *Key establishment phase*

A quantum-safe key exchange is done using a prior-generated Kyber512 public key. A shared secret is securely established between communicating parties using the Kyber512 encapsulation protocol. From this shared secret, an AES symmetric key is then obtained, which forms the session key for data encryption. This AES key is prepared for dual usage. It is passed directly to the AES encryption module for payload encryption and simultaneously encrypted using the RSA public key to make sure secure key wrapping happens as seen in Fig. 3. This post-quantum-classically-compatible system combines the security of Kyber512 with classical interoperability of RSA.

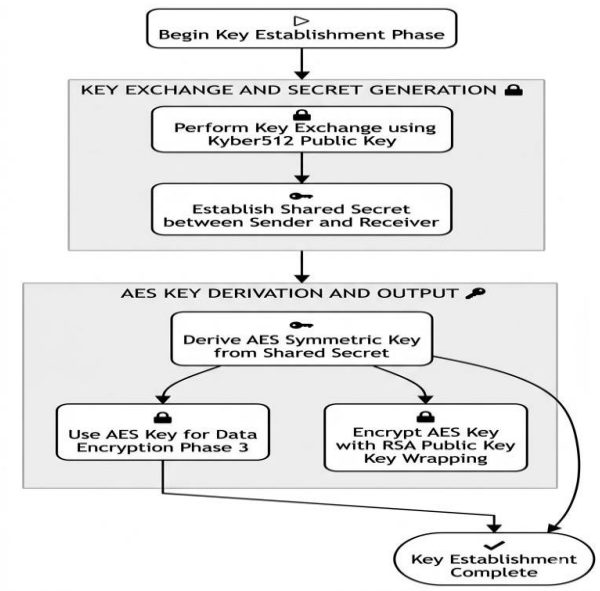

Fig. 3. Key establishment flow in the QR-RSA model.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

Algorithm: Kyber512 key establishment

INPUT: Security Parameter k

OUTPUT: Shared key k

1. $(pk, sk) \leftarrow KeyGen(k)$

2. $(ct, k_s) \leftarrow Encapsulate(pk)$

3. $k_r \leftarrow Decapsulate(ct, sk)$

4. if $k_s = k_r$ then

5. $k \leftarrow k_s$

6. else

7. $k \leftarrow \perp$

8. end if

9. Return k

G. RSA-Kyber-AES Hybrid Encryption

Fig. 4. shows the runtime communication sequence between the client, encryption modules (AES and RSA), and the receiver during secure data transmission. This diagram provides a dynamic view of Phase 3 and Phase 4 as described in the system architecture in Fig. 1. This begins with the client sending plaintext data to the AES encryption module. This module also obtains a pre-selected AES key that was acquired through a Kyber512 post-quantum key exchange in the key establishment stage. This key is then used by the AES module to encrypt the data resulting in the ciphertext. For protection of AES key during transmission, the client forwards it to the RSA encryption module. This module encrypts the AES key using the receiver's RSA public key making sure that only the intended receiver who has the corresponding RSA private key can decrypt it. Once the encryption process has been done, then the system proceeds to the transmission part where the ciphertext and the RSA-encrypted AES key is transmitted to the receiver. The decryption module called RSA would generate the original AES key at the receiver end upon input of the private key. This recovered key is fed into the AES decryption module, and the cipher text is decrypted, yielding original plain text data. A verification check shows the authenticity of the encrypted message. This modular and layered process achieves data confidentiality, integrity, and secure key exchange. Furthermore, the use of Kyber512 (in the broader architecture) adds quantum-resilient key establishment, while AES ensures high-speed encryption, and RSA provides interoperability with existing security systems.

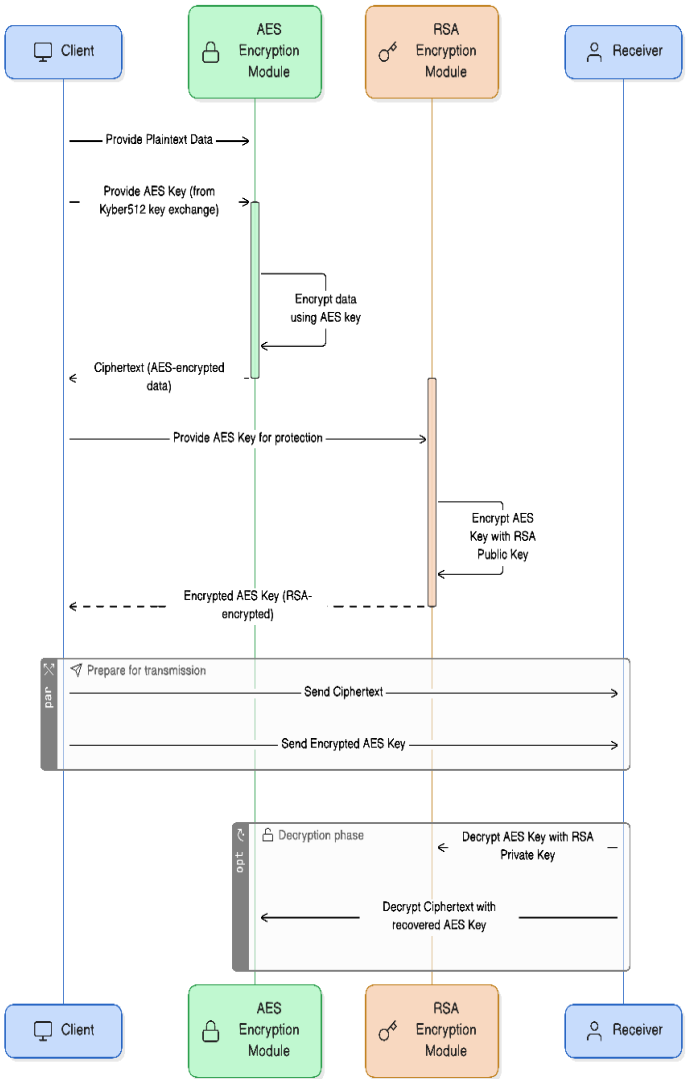

Fig. 4. Hybrid encryption showing secure communication workflow

H. Data Decryption and Verification

In the final phase of the RSA-Kyber-AES hybrid architecture, the receiver performs the decryption process in a structured sequence to securely recover the original plaintext message.

This begins when the receiver acquires two key components from the sender, that is, the AES-encrypted ciphertext and the AES key itself, which has been wrapped using the sender's RSA public key. First, the RSA private key of the receiver is used to decrypt the AES key, which gains the symmetric key to decrypt data. Having successfully retrieved the AES key, the receiver then employs it to decrypt the ciphertext through the AES decryption algorithm in safe mode. This decryption process provides the initial plaintext data, which the sender encrypted originally. After decryption, the system provides an integrity check procedure, which establishes whether the

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

decrypted message has not been altered or corrupted. Data decryption and verification sequence are illustrated in Fig. 5.

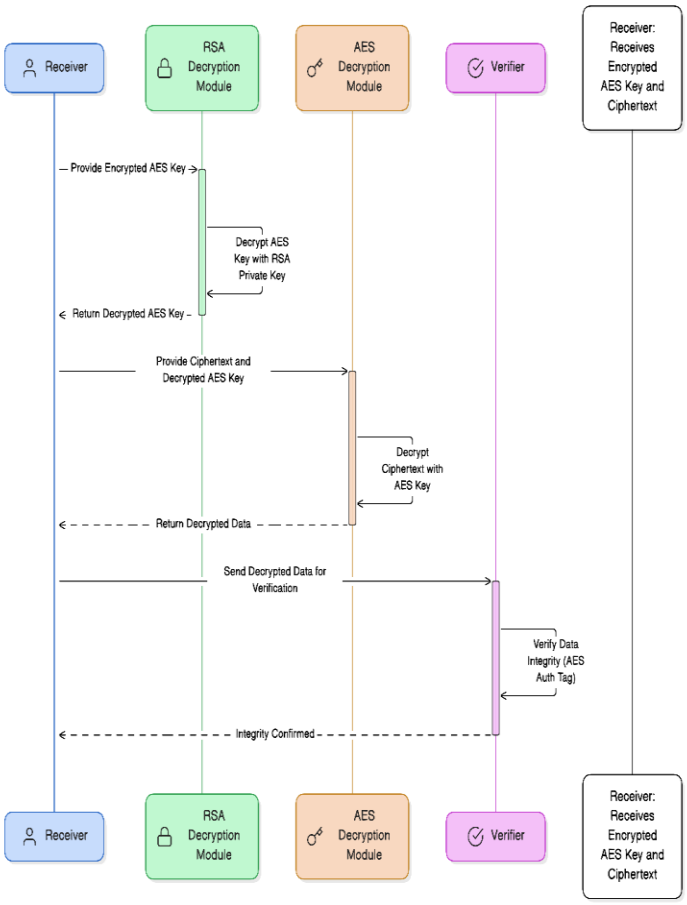

Fig. 5. RSA-Kyber-AES hybrid decryption phase

IV. RESULTS AND DISCUSSION

A. Experimental Results

The hybrid architecture (QR-RSA) was simulated in a hypothetical cloud communication environment and results is shown in table III. Encryption and decryption time averages 0.0021 seconds and 0.0036 seconds respectively on message size in the range of 64bytes to 16384bytes and indicates the framework's capability of sustaining performance near real-time. The key encapsulation overhead received in ciphertext analysis was as follows:

TABLE III. Results from proposed QR-RSA hybrid encryption model under varying message sizes.

Message Size (Bytes)	Avg. Encryption Time (s)	Avg. Decryption Time (s)	Kyber Ciphertext Size (Bytes)	RSA-wrapped AES Key Size (Bytes)
64	0.0057	0.0039	768	256
256	0.0017	0.0033	768	256
1024	0.0010	0.0037	768	256
4096	0.0009	0.0036	768	256
16384	0.0010	0.0035	768	256

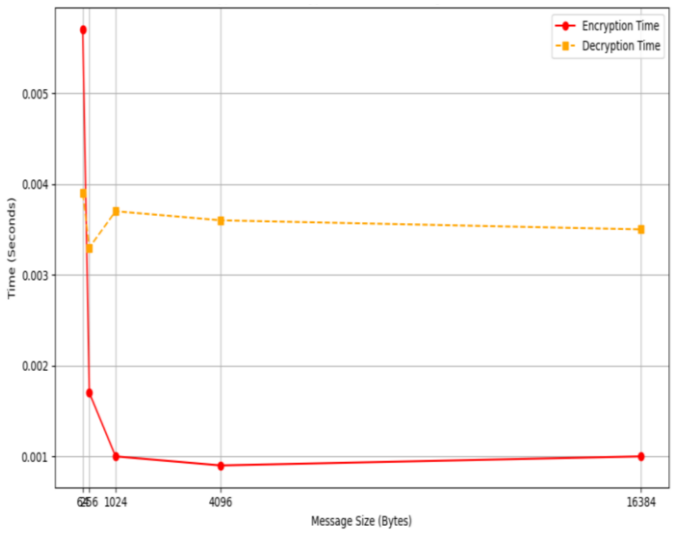

Fig. 6. Encryption and decryption time vs message size

B. File Size Expansion in QR-RSA Encryption

QR-RSA encryption results in the plaintext data increasing in size by a measurable amount so that, in most cases, the encrypted information is larger than the original text, often called file size expansion or impact. Fig. 7. shows this phenomenon where plaintext sizes are compared to the corresponding ciphertext sizes with change in the message length. The chart contains two important factors: 1. Bar Charts (Left Y-axis) which depicts the lengths of the plaintext (light blue) and resulting ciphertext (dark blue) of messages of different lengths (64 byte to 16,384 byte). 2. Red Line Plot (right Y-axis) that represents the percentage of encryption overhead calculated as:

$$Overhead\,(\%) = \left(\frac{Ciphertext\ Size - Plaintext\ Size}{Plaintext\ Size}\right) \times 100 \qquad (3)$$

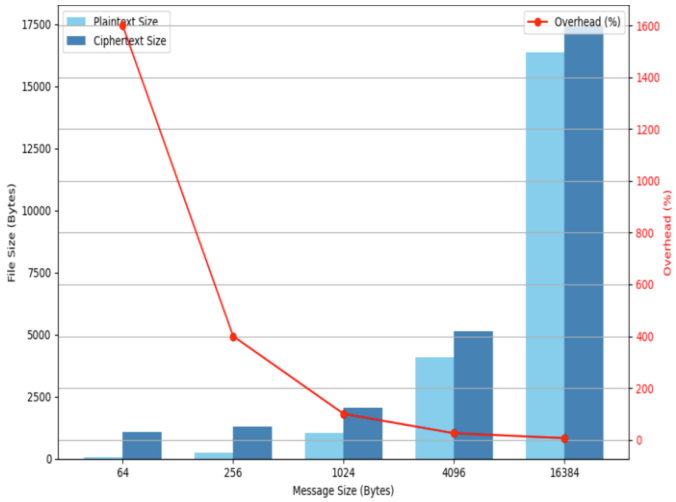

Fig. 7. File size expansion in QR-RSA encryption

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

TABLE IV. Comparison of the proposed QR-RSA model with previous studies.

Author	Framework	Encryption Time(s)	Decryption Time(s)	Post Quantum Secure
Rehman et al. (2021)	AES–ECC	0.0025	0.0019	No
Muhammed et al. (2024)	AES–RSA	0.1667	0.1953	No
Muhammed et al. (2024)	AES-ECC	0.7902	0.2232	No
Muhammed et al. (2024)	AES-EI-GAMAL	0.9648	0.3393	No
Proposed model	**QR-RSA**	0.0021	0.0036	YES

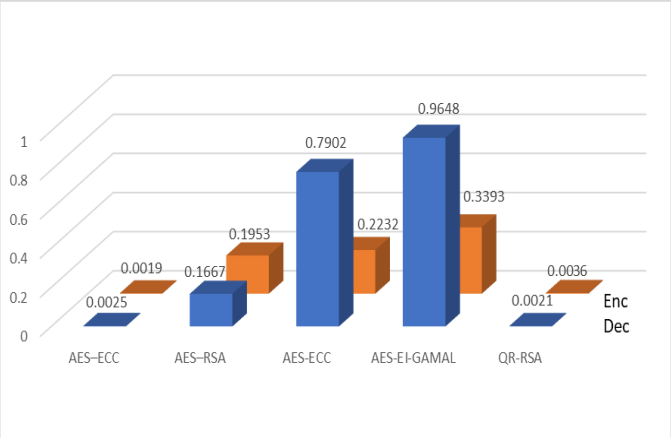

Fig. 8. Comparative encryption and decryption time with QR-RSA model.

Fig. 9. Quantum security indicator (green = secure)

C. *Discussion*

An analytical comparison of QR-RSA concerning current models such as AES, RSA, AES ECC, and AES ElGamal would be pertinent to enjoy the superior performance and security advantage offered by the proposed hybrid architecture.

The strongest argument of QR-RSA is its extremely low encryption and decryption times for all payloads as seen in Fig. 8. To demonstrate that, the QR-RSA model has average encryption and decryption time between 0.0009 s and 0.0057 s with 64-byte to 16,384-byte message sizes and 0.0033 s to 0.0039 s with 64-byte to 16,384-byte message sizes, respectively, as shown in table III. Remarkably, encryption time decreases with an increase in message size, to as low as 0.0009 s on 4096-byte messages, which is an indication of the scalability of the model.

In contrast, table IV. shows that QR-RSA emerges superior as compared to other models. Even the AES to ECC model suggested by Rehman et al. [30], whose purpose was to offer low-latency cloud security, had encryption and decryption times of 0.0025 and 0.0019 s, respectively, which is a bit faster in decryption and slower in encryption than QR-RSA. Even though QR-RSA does outperform all these models and shows an even profile over all message sizes, as shown in Fig. 8. The figure shows consistent low and steady execution time irrespective of payload, as QR-RSA is efficient and consistent, which is significant in real-time cloud communications.

One disadvantage widely found in hybrid encryption schemes is ciphertext expansion. But QR-RSA graciously meets this challenge. The size of the increase in the ciphertext can be seen in table III and Fig. 7, which it is predictable and manageable.

This consistent growth eases the process of planning resources in the system. A clear comparison of the plaintext to resultant ciphertext, as a bar chart, and a red line that shows the overhead of this encryption in percent, has been displayed in Fig. 7. The overhead is small, even on the biggest file (16,384 bytes), relative to the security gain. This is an encapsulation constant (Kyber + RSA key wrapping) that is constant at scale compared with legacy schemes that tend to display exponentially growing encapsulation overhead or unpredictable memory consumption. In addition to bare performance, quantum resistance of QR-RSA offers a strategic advantage. Fig. 9 shows clearly that the QR-RSA model (when highlighted in green) can withstand throughout quantum attack as compared to the other schemes that fail to pass the test, such as AES-RSA, AES-ECC, and AES-ElGamal.

The catch is in the fact that QR-RSA uses the Kyber512 lattice-based algorithm to perform key encapsulation, which is standardized by NIST as a post-quantum one. This provides resistance to the Shor algorithm that makes RSA, ECC, and ElGamal obsolete in a quantum environment. Quantum resistance is not addressed by any model in table IV, and therefore, QR-RSA is the only suitable model which assures forward secrecy and future-proof encryption.

This has been supported by the Quantum Security Indicator in Fig. 9, where QR-RSA is the only one with a security mark, thus, it is not only better in the current measurements, but also indispensable in survival. Conversely, when applied to older cryptographic schemes such as AES-ECC or AES-RSA, performance gains become irrelevant in the era of quantum cryptography because of cryptographic flaws.

V. CONCLUSION

The main goal of this study was to design, implement, and evaluate a hybrid encryption model (QR-RSA) that ensures data confidentiality, efficient performance, and quantum-resilient security in cloud communication systems. This goal has been effectively addressed by combining the RSA, AES,

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

and Kyber512 cryptography algorithms into a framework. RSA algorithm provided secure key encapsulation, AES provided fast and lightweight data encryption, and Kyber512 provided a powerful post-quantum protection in key exchange. All these features formed a model that had resistance to traditional and emergent quantum risks. During the research process, a prototype of the QR-RSA framework was created and tested through a set of experimental runs with different message sizes. The system was described to have robust encryption and decryption functionalities with consistent processing performance rates, thus confirming its competence in supporting real-time and scalable data protection requirements. The architecture was designed to be modular an attribute that offers flexibility and ease of fit into existing infrastructure especially when considering cloud-based applications, where performance and security is of utmost importance.

The QR-RSA model satisfies the core research objectives by offering a post-quantum-ready hybrid cryptographic scheme, achieving low-latency encryption and decryption performance, and ensuring scalability across diverse file sizes. The effective achievement of these goals makes the model a forward-compatible option that ensures data protection in a scenario where both classical and quantum computation risks co-exist. Nevertheless, there are also some limitations to the study. Although, the constant overheads that Kyber512 and RSA introduced is reasonable in large message contexts, it can be inefficient with small payloads or devices with limited resources. In addition, this implementation lacks digital signature functionality and efficient deployment to mobile and embedded devices. Future work must attempt to include lightweight authentication schemes, minimize overhead during transmission of small-sized data, as well as implement the model within practical cloud or IoT platforms that can be more fully validated. Further examination in diverse network environments would also increase familiarity with its resilience in operation.

Finally, the study provides a practical, safe, and future-oriented hybrid encryption model that will match the rising need to protect data using quantum-resistant methods. It is true to say that the QR-RSA model is among the contributions that can advance cryptographic solutions toward the post-quantum age.

ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g.” Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

REFERENCES

[1] M. Kumar and B. Mondal, “Study on implementation of Shor’s factorization algorithm on quantum computer,” *SN Computer Science*, vol. 5, no. 4, p. 413, 2024, doi: 10.1007/s42979-024-02771-y.

[2] K. Cherkaoui Dekkaki, I. Tasic, and M.-D. Cano, “Exploring post-quantum cryptography: Review and directions for the transition process,” *Technologies*, vol. 12, no. 12, p. 241, 2024, doi: 10.3390/technologies12120241.

[3] J. Bos *et al*., “CRYSTALS - Kyber: A CCA-secure module-lattice-based KEM,” in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, 2018, pp. 353–367, doi: 10.1109/EuroSP.2018.00032.

[4] J. I. Escribano Pablos and M. I. González Vasco, “Secure post-quantum group key exchange: Implementing a solution based on Kyber,” *IET Communications*, vol. 17, no. 6, pp. 758–773, 2023, doi: 10.1049/cmu2.12561.

[5] D. Das, “A hybrid algorithm for secure cloud computing,” *International Journal of Wireless and Mobile Computing*, vol. 18, no. 2, p. 116, 2020, doi: 10.1504/IJWMC.2020.105693.

[6] P. Jindal and B. Singh, “Performance evaluation of security-throughput tradeoff with channel adaptive encryption,” *International Journal of Computer Network and Information Security*, vol. 5, no. 1, pp. 49–55, 2013, doi: 10.5815/ijcnis.2013.01.06.

[7] R. Renner and R. Wolf, “Quantum advantage in cryptography,” *AIAA Journal*, vol. 61, no. 5, pp. 1895–1910, 2023, doi: 10.2514/1.J062267.

[8] M. Sharma *et al*., “Leveraging the power of quantum computing for breaking RSA encryption,” *Cyber-Physical Systems*, vol. 7, no. 2, pp. 73–92, 2021, doi: 10.1080/23335777.2020.1811384.

[9] A. Ekert and R. Jozsa, “Quantum computation and Shor’s factoring algorithm,” *Reviews of Modern Physics*, vol. 68, no. 3, pp. 733–753, 1996, doi: 10.1103/RevModPhys.68.733.

[10] D. Sikeridis, P. Kampanakis, and M. Devetsikiotis, “Assessing the overhead of post-quantum cryptography in TLS 1.3 and SSH,” in *Proceedings of the 16th International Conference on Emerging Networking EXperiments and Technologies*, 2020, pp. 149–156, doi: 10.1145/3386367.3431305.

[11] D. Ott, K. Paterson, and D. Moreau, “Where is the research on cryptographic transition and agility?” *Communications of the ACM*, vol. 66, no. 4, pp. 29–32, 2023, doi: 10.1145/3567825.

[12] E. Roloff *et al*., “High performance computing in the cloud: Deployment, performance and cost efficiency,” in *2012 IEEE 4th International Conference on Cloud Computing Technology and Science*, 2012, pp. 371–378, doi: 10.1109/CloudCom.2012.6427549.

[13] S. Z. Al-Otaibi, “Data security challenges and solutions in cloud computing: Critical review,” *Communications in Mathematics and Applications*, vol. 13, no.2, pp. 795–806, 2022, doi: 10.26713/cma.v13i2.2032.

[14] W. Hashim and N. A.-H. K. Hussein, “Securing cloud computing environments: An analysis of multi-tenancy vulnerabilities and countermeasures,”*SHIFRA*,2024,p.816,doi:10.70470/SHIFRA/2024/002.

[15] S. S. Kausalye and S. Kumar Sharma, “Data confidentiality in cloud storage: A survey,” 2021, doi: 10.3233/APC210257.

[16] T. Ntloedibe, T. Foko, and M. A. Segooa, “Cloud leakage in higher education in South Africa: A case of University of Technology,” *South African Journal of Information Management*, vol. 26, no. 1, 2024, doi: 10.4102/sajim.v26i1.1783.

[17] N. Tissir, N. Aboutabit, and S. El Kafhali, “Detection and prevention of Man-in-The-Middle attack in cloud computing using Openstack,” *Bulletin of Electrical Engineering and Informatics*, vol. 14, no. 1, pp. 377–387, 2025, doi: 10.11591/eei.v14i1.8103.

[18] N. Thakur, A. Singh, and A. L. Sangal, “Data integrity authentication techniques in cloud computing: A survey,” pp. 1255–1267, 2020, doi: 10.1007/978-981-15-0751-9_115.

[19] P. Goswami *et al*., “Investigation on storage level data integrity strategies in cloud computing: Classification, security obstructions, challenges and vulnerability,” *Journal of Cloud Computing*, vol. 13, no. 1, p. 45, 2024, doi: 10.1186/s13677-024-00605-.

[20] M. K. Parmar and M. L. Rahevar, “Compromising cloud security and privacy by DoS, DDoS, and botnet and their countermeasures,” pp. 159–169, 2019, doi: 10.1007/978-3-030-00665-5_17.

[21] R. Saxena and S. Dey, “DDoS attack prevention using collaborative approach for cloud computing,” *Cluster Computing*, vol. 23, no. 2, pp. 1329–1344, 2020, doi: 10.1007/s10586-019-02994-2.

[22] P. Ramos Brandão, “The importance of authentication and encryption in cloud computing framework security,” *International Journal on Data Science and Technology*,vol. 4,no. 1, p. 1, 2018, doi: 10.11648/j.ijdst.20180401.11.

[23] M. H. M. Baig, H. B. Ul Haq, and W. Habib, “A comparative analysis of AES, RSA, and 3DES encryption standards based on speed and

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 04 , April - 2026

performance," *Management Science Advances*, vol. 1, no. 1, pp. 20–30, 2024, doi: 10.31181/msa1120244.

[24] B. A. Buhari *et al*., "Performance evaluation of symmetric data encryption algorithms: AES and Blowfish," *Saudi Journal of Engineering and Technology*, vol. 4, no. 10, pp. 407–414, 2019, doi: 10.36348/SJEAT.2019.v04i10.002.

[25] S. H. Murad and K. H. Rahouma, "Implementation and performance analysis of hybrid cryptographic schemes applied in cloud computing environment," *Procedia Computer Science*, vol. 194, pp. 165–172, 2021, doi: 10.1016/j.procs.2021.10.070.

[26] V. Mavroeidis *et al*., "The impact of quantum computing on present cryptography," *International Journal of Advanced Computer Science and Applications*, vol. 9, no. 3, 2018, doi: 10.14569/IJACSA.2018.090354.

[27] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying Grover's algorithm to AES: Quantum resource estimates," pp. 29–43, 2016, doi: 10.1007/978-3-319-29360-8_3.

[28] S. Jaques, M. Naehrig, M. Roetteler, and F. Virdia, "Implementing Grover oracles for quantum key search on AES and LowMC,"pp. 280–310, 2020, doi: 10.1007/978-3-030-45724-2_10.

[29] J. Buchmann, J. Braun, D. Demirel, and M. Geihs, "Quantum cryptography: A view from classical cryptography," *Quantum Science and Technology*, vol. 2, no. 2, p. 020502, 2017,doi: 10.1088/2058-9565/aa69cd.

[30] S. Rehman, N. T. Bajwa, M. A. Shah, A. O. Aseeri, and A. Anjum, "Hybrid AES-ECC model for the security of data over cloud storage," *Electronics*,vol. 10,no.21,p.2673,2021,doi:10.3390/electronics10212673.