Next-Gen Data Security Using Hybrid Cryptography and Email-Based Two-Factor Authentication

Nadakuditi Vijay Vamsi[1], Radhika Rani Chintala[2]
[1,2]*department of CSE, Koneru Lakshmaiah Educational Foundation, Vaddeswaram, Guntur, A.P, India*

***Abstract*—Now-a-days it must be mandatory for the organization to properly secure data using cryptographic techniques-the advancing digital communication and cloud storage. Although the traditional model hybrid encryption mechanisms work well, they cannot be easily adopted in high throughput environments due to the performance bottleneck and vulnerability aspects. This paper presents ARB3, an innovative algorithm in hybrid cryptography. This next-generation algorithm combines the merits of Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA), and the BLAKE3 hashing algorithm. The three components brought together form a fast, secure, and efficient data protection mechanism: high speed symmetric encryption from AES, secure key establishing capabilities from RSA, and fast and secure hashing for data integrity verification from BLAKE3. This model also adds an electronic email 2FA to give an extra layer to access control and user verification: time-sensitive one-time passwords issued to their registered email account. Through rigorous test cases, ARB3 has been found to perform better as regards both encryption and decryption speed and throughput improvements over common cyber threats against traditional models. Hence, this hybrid model is scalable and robustly geared up to provide a state-of-the-art solution for contemporary applications where the demand for high-performance adaptable crypto systems is becoming more urgent.**

***Index Terms*—Hybrid Cryptography, AES, RSA, Blake3, Symmetric, Asymmetric, Hash function, two-factor authentication (2FA)**

I. INTRODUCTION

Information security greatly depends on cryptography. Cryptography is a key aspect of information security measures. According to the Economic, cryptography is the technique of turning regular plain communication into incomprehensible writing or vice reverse such that intended recipient will able to read and process it. The word ‘cryptograph’ was formed from the Greek words praphia (graphia), which means writing, and kryptos (krnptos), which means hidden or secret [1]. algorithms for cryptography: The methods required to ensure the confidentiality of information are the subject of this study. By using cryptography, the real text is formed as an unintelligible writing.[2]

It is divided into two classes: symmetric and asymmetric cryptography. Three are there: "Cryptography can be divided into the following three categories depending upon the types of keys used: secret key (symmetric) cryptography, public key (asymmetric) cryptography, and hash functions."

A. Symmetric Cryptography

Symmetric encoding uses an encoding algorithm and a key to transform data to ciphertext. The input text can be extracted from the cipher text using the same key and decryption technique. Symmetric key encoding has old history, according to author [3]. AES, DES, and Blow fish are few of different symmetric key cryptography methods.

B. Asymmetric Cryptography

Public key cryptography, known as asymmetric key encoding, encode and decode data using two distinct keys [4]. Every communication participant in this approach employs two keys: a public key that is distributed with every participant and a secret key that is only known by intended recipient. There is a link between the public and secret keys, despite their apparent differences. This method can offer nonrepudiation, integrity, and authentication [5].

C. Hash Function

A hashing function is a method that results in a fixed length string of characters, also called a hashing value,

from an input, such as a file, text, or message. One of hash functions properties is determinism, which is the notion that the same input always produces the same output. Because the output length is fixed and all outputs are of the same length, hash functions are also made to be efficient. They produce results quickly enough, regardless of the length of the input.[6]

In cryptographic applications, hash functions are of extreme importance because they can help in ensuring the integrity of data by creating a digital fingerprint for that input data as verification. Some popular hashing algorithms include MD5, which has become obsolete for security, and the quite more secure version SHA-256 from the widely-adopted SHA-2 family. A new contemporary hashing algorithm, BLAKE3, is comparatively much more recognized for its flexibility and higher performance and security, especially in high-performance hashing applications. The pre-image and collision resistance help ensure integrity and security of data in many systems.[7]

D. Hybrid Cryptography

Hybrid cryptography is deeply instituted in contemporary digital communications and data protection security infrastructure. Its implantation in online transactions, secure messaging, cloud storage, and electronic voting prove its versatility and indispensable role in maintaining input text security is increasingly interconnected widely. Cyber threats are growing and evolution always means that hybrid cryptography will find an indispensable place in maintaining secure security protocols.

E. Uses of Hybrid Cryptography: -

1. Hybrid cryptography is essential for the protection of web communications. It is this that HTTPS relies on, SSL/TLS, which realizes an encrypted process of data transmission based on hybrid cryptography. This uses asymmetric encryption to protect the secret session key exchange, after which the session key will encrypt the actual data to achieve real faster response times.

2. Email encryption (PGP/SMIME): Secure email transmission is guaranteed via hybrid cryptography. Asymmetric encryption is used by the protocols S/MIME and PGP (Pretty Good Privacy) to first encrypt the symmetric keys that are used to safeguard the emails' actual content.

3. Data transmission and storage in cloud are handled using hybrid cryptography. As a result, data can be safely maintained while encoding and decoding times are decreased by the usage of such cryptography. While asymmetric encoding is used to secure the key itself, symmetric encoding will encrypt the input that is kept on the cloud.[8]

4. Virtual Private Networks (VPNs): To protect the security of input flowing over the internet, VPNs employ hybrid cryptography. Asymmetric encryption guarantees a secure key exchange between the VPN client and server, while symmetric encryption protects the data.

5. Hybrid cryptography protects credit card type information because it hides sensitive financial information, enciphers with symmetric key the transaction details while securing the exchange of this key using asymmetric encryption. File encryption: Applications use hybrid cryptography where large files need to be exchanged securely. In this particular case, asymmetric encryption secures the key used for its transmission, and symmetric key encrypts the file content.[9]

F. Why Prefer Hybrid Cryptography: -

Hybrid cryptography is greater than symmetric or asymmetric cryptography alone because it combines the privacy of asymmetric encoding for key transformed with the performance and efficiency of symmetric encoding.[10] The problem of safely sharing the secret key via an unsecured channel severely hinders pure symmetric cryptography, even though it is fast. While pure asymmetric cryptography is safe for key exchange, it is highly inefficient for huge datasets since it must be applied to each individual piece of data. By giving an intermediate level of power and security balancing, hybrid cryptography gets beyond these restrictions and provides both extremely high performance and strong security. Because of this, it is a good option for many contemporary applications that require both encryption strength and efficiency in comparable protocols.

G. Two-Factor Authentication (2FA): -

Among these, strong encryption algorithms like AES, RSA, and BLAKE3 form the basis for secure data transmission and storage; however, additional requirements are placed on user authentication from modern security challenges. Password access alone has shown weaknesses in being broken through phishing, credential stuffing, and brute-force attacks. [11] Hence, to mitigate this significant security gap, we integrate email-based Two-Factor Authentication (2FA) into our ARB3-based encryption system for an extra layer of protection.

In this model, authentication is not based solely on something the user knows (i.e., password) but also on something the user possesses-access to a registered email account. When logging in or gaining access to the system, the user is prompted to enter a unique One-Time Password (OTP), which expires within a specified time frame and is sent to their email address. Once the right OTP is input, the user is granted access; hence, even if someone else knows the password, they cannot enter the system without email verification.[12]

This kind of 2FA is lightweight, simple, and extremely effective. Nothing is needed besides access to email along with which usability is enhanced; most users have it. When combined with vigorous cryptography, 2FA enhances the overall security posture of the system even further by denying unauthorized access and mass ensuring that only verified users may carry out encryption or decryption processes.

The integration of email-based 2FA along with ARB3 thus provides a complete solution securing data via hybrid cryptography while equally strengthening access control to the system, especially in environments where data sensitivity and user authentication are prime concerns.

II. LITERATURE REVIEW

In order to improve a file's security features, this study combines file deduplication with Advanced Encryption Standard encryption. Deduplication was accomplished using the Blake3 hashing technique, which allows duplicate data blocks to be removed in order to maximize storage capacity. AES encryption ensures strong confidentiality of unique files. In this instance, it was striking a balance between robust encryption and data optimization. This strategy is better suited to the problems with cloud computing, where security and efficiency are the main requirements. [13] The study examines current developments in these fields, emphasizing the importance of integrating encryption and deduplication to create a safe yet effective cloud architecture.

The study focuses on a sophisticated encryption processing system that uses a modified DES algorithm to process accounting data. [14] By optimizing the system architecture from hardware acceleration points and integrating the more secure AES algorithm than the DES algorithm, the authors show significant time savings for encryption activities. This demonstrates that in addition to improved security training, appropriate key management, backup, and recovery procedures are required. System enhancements are intended to provide businesses with dependable and efficient solutions for protecting the integrity and confidentiality of accounting data.

To secure input from unauthorized access and destruction, this paper primarily focuses on data security. It has been said that hybrid cryptography, it combines the Pros of public-key and symmetric-key encoding, an effective way to secure digital data. To ensure that only authorized persons only can access file, it first encrypts it using its symmetric key and encrypts it again using the recipient's public key. The requirement to securely store data has been expanding along with the volume of data, and the rise in malware and phishing attempts has made this issue more important. It also tackles the issue of their local servers' inability to manage high data volumes, encouraging the adoption of cloud computing in spite of its security drawbacks. Researchers have suggested a novel technique to protect important data files after criticizing well-known algorithms like AES, DES, and RSA. [15]

Sherief H. Murad and Kamel H. Rahouma's book "Hybrid Cryptography for Cloud Security: Methodologies and Designs" [16] aims to investigate how various cryptographic algorithms might be hybridized to improve cloud security. It examines a variety of hybrid cryptological models from 2013 to 2020 and offers information on their applicability, constraints, design, and implementation. It makes the point that symmetric and asymmetric systems work better together and that their security and performance

may be enhanced. Although hybrid cryptography has limits in terms of key distribution and computational expense, the research ends with a robust solution for cloud hybrids that provides protection for sensitive data. I would support studies that try to close the gaps and improve Protected measures.

Higher security now requires in all areas are guarantee the confidentiality of the data. A good solution is Robust hybrid encoded, which mix both symmetric and asymmetric cryptography algorithms to encode messages. Symmetric encoded algorithms such as AES and asymmetric algorithms such as RSA, which offer secure key exchange methods, have gained attention due to their effectiveness in encrypting massive datasets. Double encoded technique can be employs the guarantee file security in cloud, according to recent research reports. For example, data security achieves the use of Protect key management with RSA algorithms and high-speed encryption and decryption techniques like AES. Jaspin et al. (2021) claim that using double encryption with AES and RSA significantly improves data confidentiality and integrity [17]. By distributing encrypted data among cloud servers, Swarnalatha et al. (2023) improved security by further developing methods for file splitting and encryption2. Key management and overhead computation issues are currently being worked on as open issues. Achieving a favorable performance-security trade-off in optimized hybrid systems will be the main focus of future research, particularly for cutting-edge technologies like blockchain and the Internet of Things.

"Secure File Storage on Cloud using Hybrid Cryptography" [18] is one of them; it examines hybrid cryptography methods to improve cloud storage input text security. 1. The suggested method encrypts and decrypts data using 3DES in conjunction with Blowfish encryption 3. A hybrid approach is used to assure improved security because the technique's recognized weakness is the weakness of a single cryptographic approach. For total protection against unwanted access, data encryption using this suggested paradigm is split into three sections that will be encrypted using various algorithms. 2. Along with reviewing relevant work in the field, this study discusses different encryption methods and their efficiency. There is also a thorough discussion of the specifics of the suggested system's implementation, including all of its advantages in terms of data integrity, very high security, authentication, and confidentiality.

High data security demands that researchers investigate hybrid cryptographic systems. Although it can be difficult, creating hybrid cryptographic systems that combine symmetric and asymmetric encoding methods is necessary for input text security. Asymmetric algorithms like ElGamal and RSA can be used to provide secure key exchange mechanisms, whereas symmetric algorithms like Twofish and AES are well-known for their speed and effectiveness in encrypting big datasets. Research on integrating these capabilities is underway. For example, a hybrid implementation of RSA and Twofish has been shown to reduce the size of the ciphertext while also producing improved computing performance. AES and RSA work together to provide overall increased security. Indeed, by guaranteeing high encryption levels and effective key management, Kumar et al.'s recent research from 2020 has demonstrated that hybrid cryptosystems offer the potential to significantly improve data security in cloud computing scenarios. Additionally, it has been investigated how to integrate ElGamal with symmetric algorithms to offer security that is resistant to quantum errors. All of these advances have been made, in fact, but the areas that still require research are those that deal with computing overhead and key management complexity. Thus, future advancements would focus on these hybrid systems' scalability and optimal performance. [19]

Cloud computing revolutionized IT via services such as IAAS, PAAS and SAAS; however, they have introduced security issues, particularly authentication. Conventional techniques present, e.g., password-based and biometric authentication, cannot perform well as they are either prone to brute force attacks or require a lot of computation, respectively. To solve these issues, the paper suggests a secure two-factor authentication (2FA) mechanism that uses one-way hash functions, nonce-based schemes, and Elliptic Curve Cryptography (ECC). The proposed solution offers enhanced security against MITM, replay and session hijacking attacks, greater user applicability, and lower computational cost compared to RSA since it utilizes ECC for secure and efficient use of encryption, especially for memory-constrained devices. The solution introduced uses user IDs, salted

passwords, OTPs and cloud certificates to efficiently meet stringent authentication needs without compromising user-friendly operations. The model prefers sensitive information in clouds as being safe from unauthorized access and cyberattacks, the sign of an evolution in cloud security practices.[20]

The paper discusses two-factor authentication (2FA) systems and their implementation at CMU. Password breaches are now a major safety risk due to phishing attacks and poor password hygiene; as such, 2FA becomes a necessary safety measure. User behaviours and perception surrounding the mandatory adoption of Duo 2FA at CMU were studied. Users initially found 2FA an inconvenience, but they acknowledged its usefulness and the security it afforded. Positive Duo experiences turned many of them to begin using Duo for other accounts. In the research, misconceptions, insecure behaviours, and design problems limiting the adoption are also identified. Recommendations include strategic messaging, a careful design plan for implementation, and mandates to boost adoption rate. This research will enhance the understanding: It describes how a compromise between usability and security could be struck to support the large rollout of 2FA, and thus offers lessons for boosting user acceptance and system efficacy.[21]

III. THE COMPREHENSIVE THEORETICAL BASIS

The fundamental workings of AES, Blake3, and the improved RSA algorithm will be covered in this part.

A. Advanced Encryption Standard:

The Advanced Encryption Standard is referred to as AES.[22] The symmetric block cipher algorithm family includes this one. To encode and decode, it uses a key with 128, 192, or 256 bits. As seen in Fig 1. this technique makes use of an SP network with various rounds, which again rely on the size of the key being used.[23] In every round, there are four stages: replacing the bytes, shifting the rows, mixing the columns, and finally adding the round key.

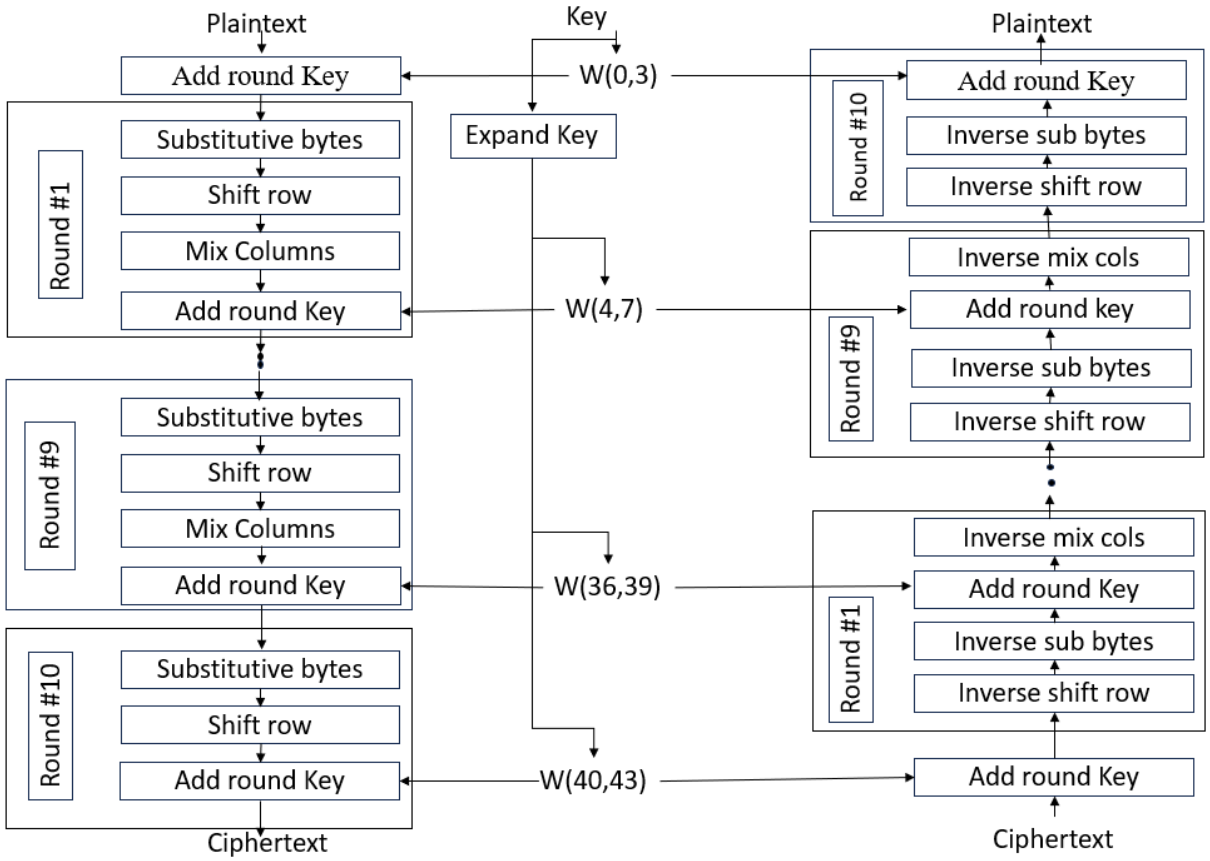

Fig 1: AES Encryption and decryption system

B. Rivest-Shamir-Adleman (RSA)

RSA is asymmetric, i.e., it employs two different keys: a private key and a public key. The private key remains secret, whereas the public key is shared with all. Rivest, Shamir, and Adleman are known to have developed the RSA algorithm. The most popular aspect of RSA is its secure encryption technique. The exponential application of prime integers of the positive integers to encrypt and decrypt is the basis of RSA.[24] RSA algorithm works as shown in Fig 2.

Two exponent variables are employed in this approach: e for public and d for private. M and C represent plaintext and ciphertext, respectively, to denote the encryption and decryption process as follows: the encryption and decryption process [25].

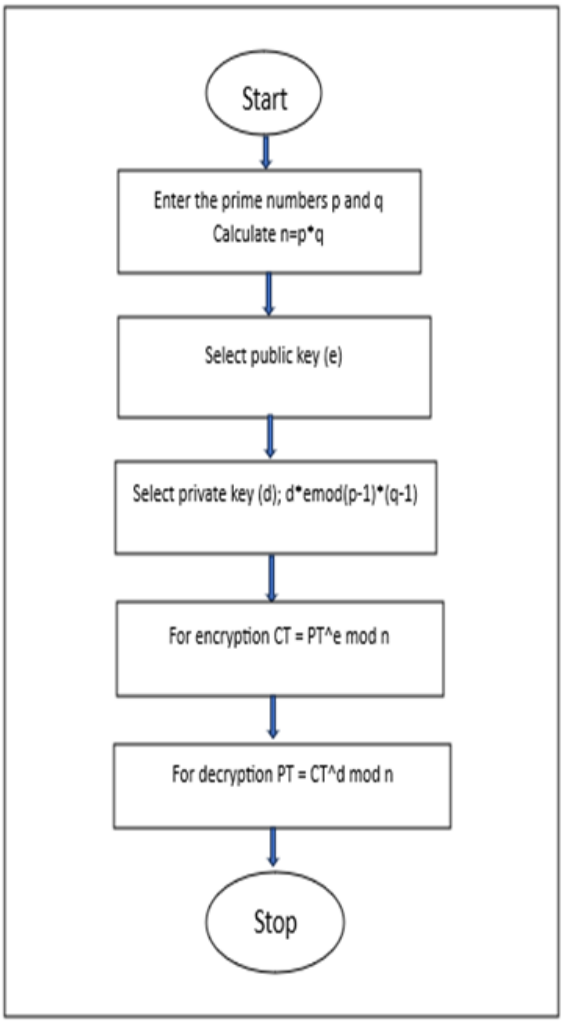

Fig 2: RSA Algorithm

C. Blake 3

BLAKE3: BLAKE3 is a cryptographic hash function developed to offer integrity and authentication of data while having higher efficiency and security with a fixed-size hash output. The primary functionality of obtaining one-time hash from the input data in BLAKE3 is to map integrity in data. In this way, Blake3 carries much importance in holding accuracy along with trustworthiness of transmitted information. The performance. Software comparing the two SHA-3 finalists' performance shows that Blake is almost three times faster than Keccak on a contemporary CPU for a 512-bit hash [11], but only 1.26 times better for a 256-bit hash. Performance is expressed in cycles per byte.

The primary steps involved in BLAKE3's operation are as follows:

Chunking: The input text is separated into discrete pieces, usually with a size of 1 KB. Parallel execution is made possible by the independent processing of these portions. Padding is applied to finish the chunk if the data is less than 1 KB. Every chunk has attributes like chunk length and whether it is the last chunk, along with an identifier (such as its position in the data).[26]

Compression Function: Compression Function: A compression function that carries out cryptographic mixing receives each chunk. This function applies operations like addition, XOR, and bit rotation on a set of 32-bit words. It is based on BLAKE2's permutation function. Simple patterns won't show up in the output since the mixing stage makes Sure the input data is evenly distributed throughout the chunk.

Merkle Tree Structure: A binary Merkle tree is built using the hash values from every chunk. The process proceeds up the tree until a single root node is created by combining chunks in pairs and compressing their hashes to create parent nodes. Because of this structure, BLAKE3 may effectively enable parallel computation by processing chunks separately before joining them.

Finalization: Using particular domain settings and keys, the root node—which stands for the sum of the hashes of all the chunk goes through a last compression stage. This process guarantees the final hash's security and uniqueness. To reach the required length, usually 256 bits, which is the average output size for BLAKE3, the output is either truncated or expanded.

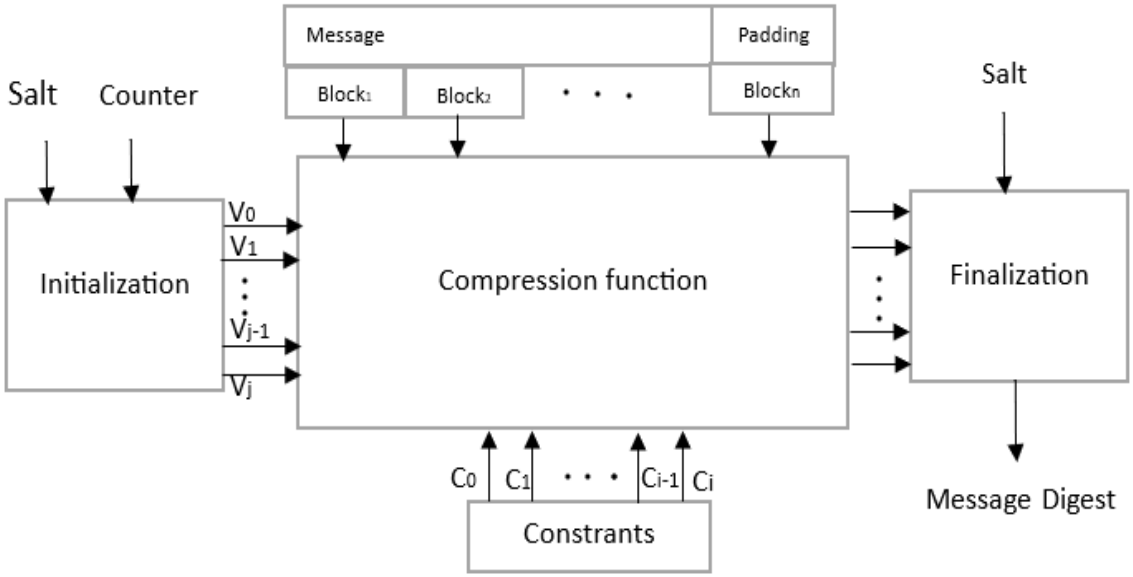

Fig 3: Architecture of the Blake3 hash function

Output: The final hash is displayed in hexadecimal format as a 256-bit digest. This digest can be used as a fingerprint for data comparison or to confirm the integrity of the data, among other uses.[27]

As demonstrated in the architectural diagram of BLAKE3 Fig 3. a hash function accepts the input, and it gives the fixed-size hash value that uniquely represents the data. Therefore, it is very efficient and secure; hence it can be applied in hybrid systems for data integrity verification. Hybrid cryptography checks whether there is any change in the data during transmission or not. It does this by taking the hash values of the plaintext data before and after transmission and comparing them.

D. Two-Factor Authentication (2FA): -

The Two-Factor Authentication process through email begins when an end user logs onto the login page using a client-side application, such as a web browser. The first part involves the entry of the credentials, typically a password and a username, in the login form. Upon submitting this form, the data are passed to the server, which will verify the credentials against those stored in the user database. If a wrong password is provided, it denies access immediately [28].

If the proper password is provided, however, the second authentication level is triggered. The system requests the user for a second authentication passkey: an OTP (One-Time Password), which is automatically generated by the system, temporarily stored in the OTP keys database, and sent out via email to the user's registered account. The user inputs this OTP themselves in a form on the client side.

Then the user clicks the submit button, and the OTP is sent back to the server for verification. The server checks the OTP entered against that which is stored in the OTP database, as well as checking if it has expired or not. Where there is an invalid or expired OTP, access is denied. Where the OTP is valid, the server verifies the user and grants access to the system.

This two-step authentication significantly increases security in the sense that even if the password is compromised, an attacker will be unable to do anything without possessing additional access to the respective user's email account. The procedure also represents an effective means of demonstrating identity for systems that handle sensitive or secure transactions as shown in Fig 4.

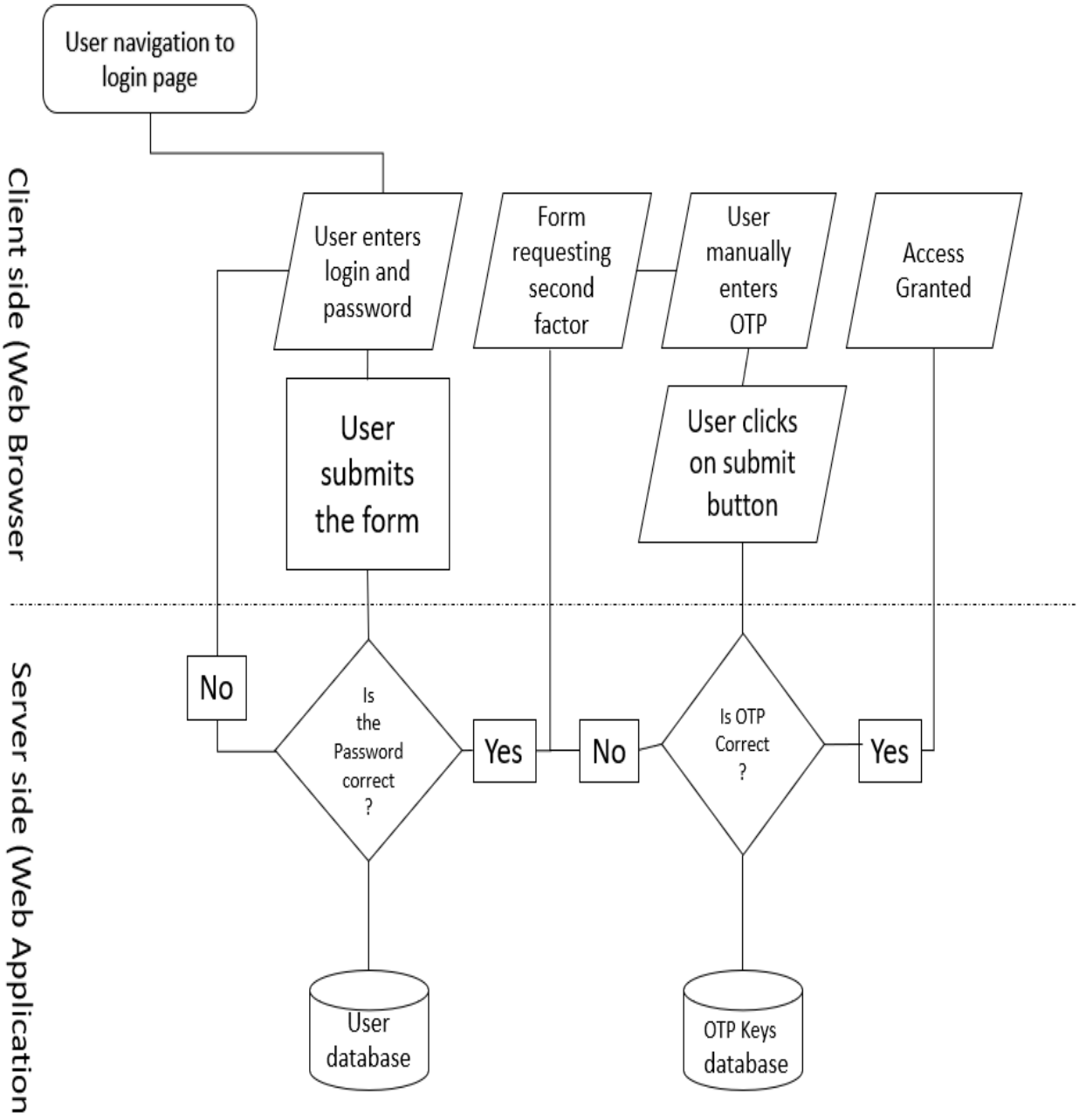

Fig 4: Architecture of Email-Based Two-Factor Authentication

IV. PROPOSED METHOD

The hybrid cryptography function of the ARB3EncryptorApp program, which combines many cryptographic techniques to securely encrypt and decrypt files, is concealed by a graphical user interface. File data is encrypted using the Advanced Encryption Standard (AES); the AES key is securely encrypted by RSA; file data is hashed using the Blake3 hash function to confirm data integrity; and two-factor authentication via email with a one-time password (OTP) ensures that only authorized users perform encryption and decryption. Confidentiality, strong authentication, and integrity verification are all represented by this system as shown in the Table 1.

Table1: - Security Parameters in ARB3 Hybrid Cryptography Model

Component	Parameter	Purpose
AES (Symmetric Encryption)	Key Size: 128 / 192 / 256 bits	Used for high-speed encryption and decryption of data.
	SP Network Rounds	Number of rounds varies based on key size (10, 12, or 14).
	Byte Substitution, Row Shifting, Column Mixing, Round Key Addition	Core AES operations ensuring confusion and diffusion.
RSA (Asymmetric Encryption)	Public and Private Key Pair (e.g., 2048 bits)	Secures the AES key during transmission (key exchange).
BLAKE3 (Hash Function)	Hash Output Length (256 bits default)	Ensures data integrity and provides a digital fingerprint.
	Collision and Pre-image Resistance	Security properties preventing tampering or spoofing.
Two-Factor Authentication (2FA)	Email-based One-Time Password (OTP) Time-sensitive Code	Adds an extra verification step for access control. OTPs expire after a short period for added security.

3.1 Main Components and Functionalities

The application begins with file selection, allowing users to search and choose any file on their system. Once a file is selected, it proceeds to RSA key generation, which produces a 2048-bit RSA key pair: the public key to encrypt the AES key during the encryption process and the private key to decrypt the AES key during the decryption process. In addition to the RSA key generation process, a secure digest of the original file is created during encryption using the Blake3 hashing algorithm. This digest will be retained and compared later when decrypting for file integrity. For key security, the application uses PBKDF2 (Password-Based Key Derivation Function to derive a secure AES key from a password provided by the user, and random salt is used to resist brute-force attacks.

3.2 Encryption Process (Hybrid Cryptography)

The password, email address, and file path are provided to the user when it is being encrypted.The user will need to accept a secure one-time password (OTP) that is being sent to the email address they have provided.Once successful entry verification of the OTP, the code proceeds to generate the AES key by using PBKDF2 and then subsequently encrypts the file using AES in CBC mode. It then encrypts the AES key with RSA public key cryptography. The process outputs three files: a “. mail" file with the email address for future access, an encrypted file with “. enc" attached to its name, and a file named "file. Hash" that has the Blake3 hash. User notification is also sent when the encryption process is complete as shown in Fig 5.

3.3 Decryption Process with OTP Verification

The user selects the .enc file to be decrypted. The program automatically searches for the related. mail and .hash files. It sends a fresh one-time password (OTP) to the email address after reading the email ID from the .mail file. OTP authentication must be completed again in order to proceed with decryption. Once authentication is successful, the RSA private key is used to decrypt the AES key, and the AES key is then used to decode the encrypted file. After removing the padding, the result is saved as a .dec file, indicating the decrypted form of the original file as shown in Fig 5.

3.4 OTP-Based Two-Factor Authentication (2FA)

OTP is a random one-time password, which is created during both encryption and decryption. The OTP is sent via SMTP to the user's email through Python's smtplib. Therefore, authentication credentials of the sender's email need to be configured in the application. The authentication layer here forms a second-factor authentication, which means that even if the file encrypted somehow was decrypted without having any authorized access in Fig 5, no one would be able to decrypt it without the user's email access.

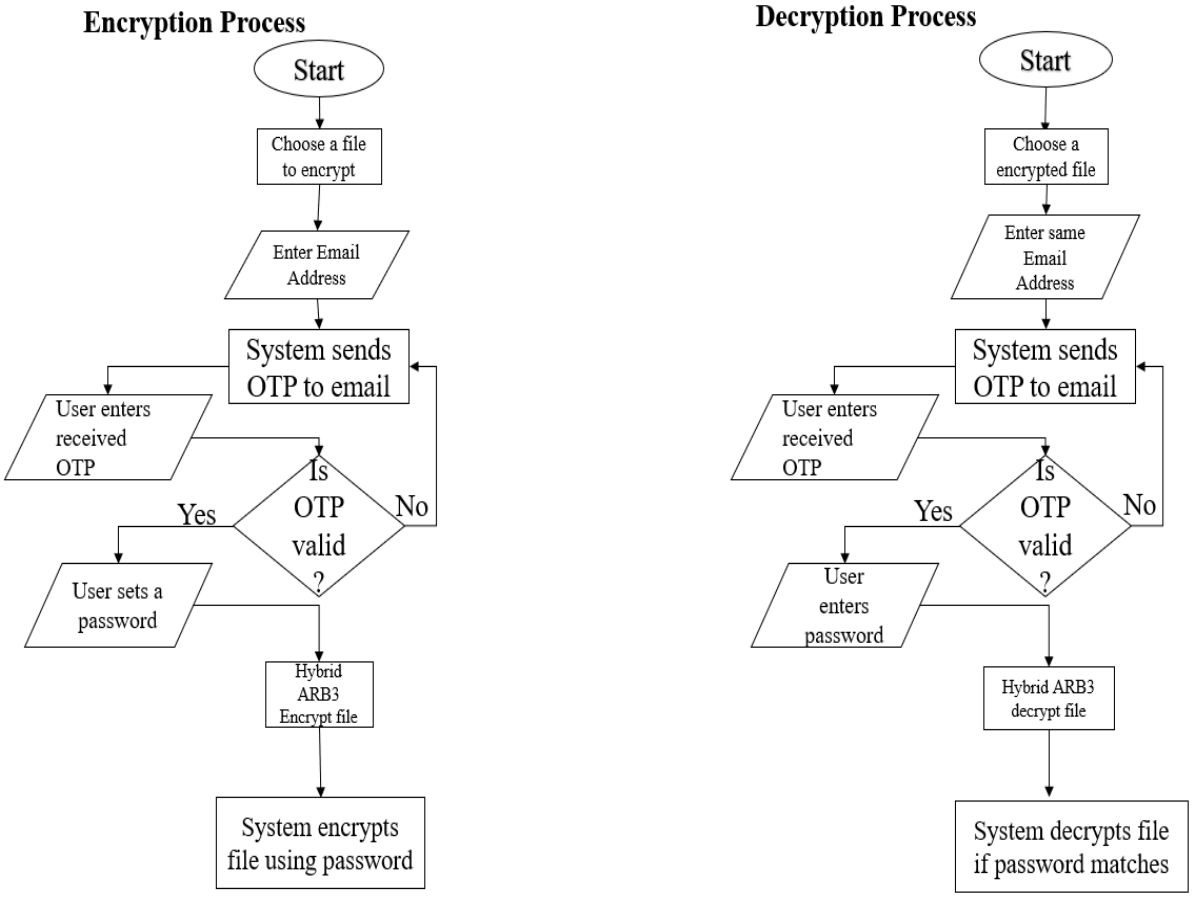

Fig 5: Proposed ARB3 and Email-Based Two-Factor Authentication Flowchart

3.5 Security Highlights

The hybrid encryption model is operated through this application by synergizing the superior speed and efficiency of symmetric AES encryption with the secure key exchange process of the RSA encryption scheme. User authentication is enforced with two-factor authentication (one-time password plus password). The integrity of the file is ensured by Blake3 hashing, which allows verification against tampering. PBKDF2 also denies brute-force attacks, adding another layer to the entire encryption process.

V. RESULTS AND DISCUSSION

A. Encryption and Decryption Time in Seconds

The encoding and decoding of times of algorithms depict a vast variation between performance levels. Of all these, it is the combination ARB3 (AES + RSA + Blake3) that triumphs over the rest by achieving the fastest speed in both encryption and decryption. The remaining algorithms vary from mild to extreme degrees of efficiency from one another. Some combined algorithms, such as RSA + HMAC and RSA +MD5, perform significantly worse in terms of time taken. There are also several hybrid variants which show discrepancies in the speed of both encryption

and decryption-with either process consuming much more time than the other. Compared with earlier hybrid algorithms, the new ARB3 algorithm shows a marked improvement in terms of time efficiency as shown in Fig 6. For overall speed, ARB3 seems to be the best bet-the most obvious superiority over all other algorithmic combinations in the comparison.

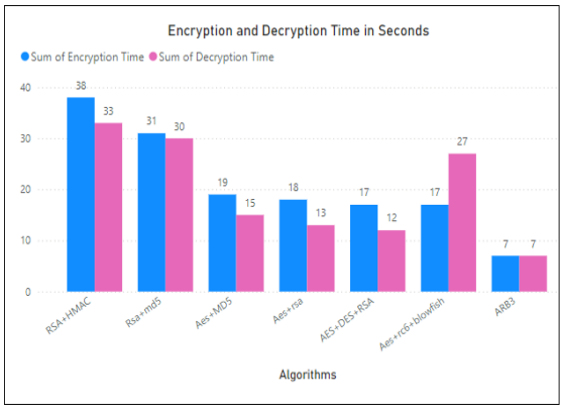

Fig 6: Encryption and decryption time in seconds

B Throughput:

Throughput is determined by the encryption and decryption times, which show how quickly an encryption can communicate. scheme [29].

The standard for it is: As $(T_P\ (E_t + D_t)) * 100$, throughput

Were,

- T_p Enter plain text (Bytes)
- E_t and D_t Encryption-Decryption Time (seconds)

Throughput metrics are used to calculate the efficiency of algorithms. If we consider the sole speed of processing among all data-cryptographic algorithms, then ARB3 has the highest throughput. During throughput testing, it was noted to achieve an astonishing figure of 28,445,023.15 bytes per second by stopping all other processes as shown in Fig 7. Automated encoding and decoding of large message blocks operate at such high throughputs because it is likely that shows the importance of rapid surrounding velocity in work. For several other algorithms, throughput is at best neutral to stagnant, showing that most of the time is spent managing data. This observation confirms the dramatic throughput results noted above, validating the high efficiency of ARB3 noted in previous studies. Due to its relative efficiency against other cryptographic algorithms, this fact greatly increases its usefulness in problems where security and performance are needed, especially when large amounts of data have to be processed.

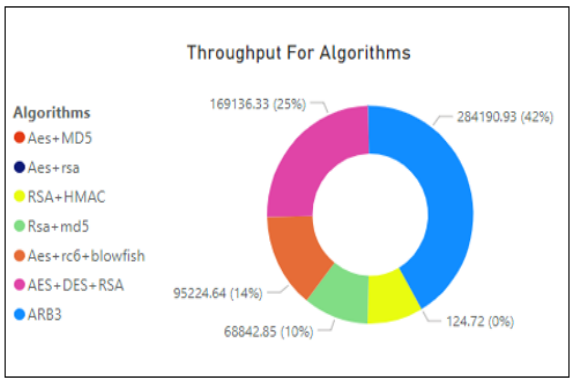

Fig 7: Throughput for Algorithms

4.1 Final Output with Results: -

The encryption processing via the ARB3EncryptorApp results in three outputs files. Nc,. hash and. mail invariably. The .enc file is the safe encryption of the original content utilizing AES encryption, guaranteeing confidentiality. The .hash file holds the Blake3 hash of the original content, which serves as a digital fingerprint for proving the integrity of data. The .mail file would store the email of the intended individual to receive the information, for utilization in the OTP-based two-factor authentication as shown in Fig 8. It subsequently employs the AES key one is able to extract from the users and decrypts the contents, before writing out to a .dec file that contains the recovered fully restored original file. A hash comparison is done between the .hash file and the freshly decrypted contents to verify data integrity as shown in Fig 9.

A)EncryptionProcess

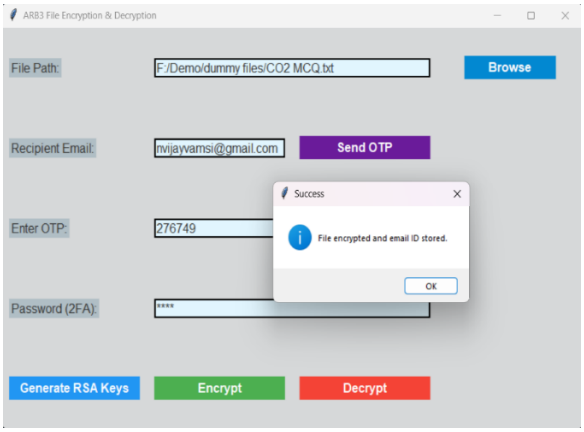

Fig 8: Encryption Time

B)DecryptionProcess

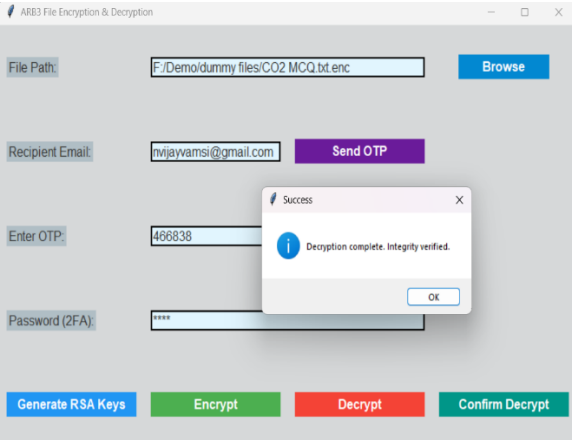

Fig 9: Decryption Time

VI. CONCLUSION AND FUTURE SCOPE

The proposed hybrid ARB3 cryptographic model-an amalgamation of the AES for high-speed encryption, the RSA for secure key exchange, and BLAKE3-for fast, reliable hashing, combined with email-based two-factor authentication-forms an effective and secure solution for the modern needs of data protection. With regard to performance in cloud-based and high-throughput applications, this model promotes the data's confidentiality, integrity, and user authentication. The experimental results confirm the ARB3's higher speed of encryption and decryption, along with its ability to resist against the common cyber threats, well proving, therein, its scalability and applicability in the real world.

One way of enhancing ARB3 in the future will be to add some biometric-based authentication so that the security layer becomes ever-stronger for identity verification. The other possible line of future extension for ARB3 could be applicable towards real-time implementation in IoT and edge computing devices, giving life to light and strong cryptographic applications. The model can also be validated against quantum computing threats, with possible infusion of post-quantum cryptographic approaches to safeguard it into the future. This further implementation of ARB3 in secure messaging platforms, cloud APIs, and secure file sharing systems would help in gaining publicity and enhancing its effect in varied industries.

REFERENCESS

[1] H. Sharma, R. Kumar and M. Gupta, "A Review Paper on Hybrid Cryptographic Algorithms in Cloud Network," 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India, 2023, pp. 1-5, doi: 10.1109/INOCON57975.2023.10101044.

[2] V. Verma, P. Kumar, R. K. Verma and S. Priya, "A Novel Approach for Security in Cloud Data Storage Using AES-DES-RSA Hybrid Cryptography," 2021 Emerging Trends in Industry 4.0 (ETI 4.0), Raigarh, India, 2021, pp. 1-6, doi: 10.1109/ETI4.051663.2021.9619274.

[3] K. Tajane, R. Pitale, S. Zambre, H. Huda, A. Utage and V. Dhar, "Efficient Cloud Data Deduplication with Blake3 and Secure Transfer using AES," 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, 2024, pp. 572-579, doi:10.1109/ICPCSN62568.2024.00096.

[4] Z. Li, "Exploration on Accounting Data Encryption Processing System Based on DES Algorithm," 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE), Ballari, India, 2023, pp. 1-6, doi: 10.1109/AIKIIE60097.2023.10389923.

[5] C. Susmitha, S. Srineeharika, K. S. Laasya, S. K. Kannaiah and S. Bulla, "Hybrid Cryptography for Secure File Storage," 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, 2023, pp. 1151-1156, doi: 10.1109/ICCMC56507.2023.10084073.

[6] Murad, Sherief & Rahouma, Kamel. (2022). Hybrid Cryptography for Cloud Security: Methodologies and Designs. 10.1007/978-981-16-2275-5_7.

[7] K. Jaspin, S. Selvan, S. Sahana and G. Thanmai, "Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm," 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, 2021, pp. 791-796, doi: 10.1109/ESCI50559.2021.9397005.

[8] V. Sharma, A. Chauhan, H. Saxena, S. Mishra and S. Bansal, "Secure File Storage on Cloud using Hybrid Cryptography," 2021 5th International

Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2021, pp. 1-6, doi: 10.1109/ISCON52037.2021.9702323.

[9] E. Jintcharadze and M. Iavich, "Hybrid Implementation of Twofish, AES, ElGamal and RSA Cryptosystems," 2020 IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, 2020, pp. 1-5, doi: 10.1109/EWDTS50664.2020.9224901.

[10] S. A. Ahmad and A. B. Garko, "Hybrid Cryptography Algorithms in Cloud Computing: A Review," 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, 2019, pp. 1-6, doi: 10.1109/ICECCO48375.2019.9043254.

[11] Od, Zina & Derdour, Makhlouf & Bouhamed, Mohammed Mounir. (2024). Improving Resource Security by Integrating Authentication and Cryptography. 1-8. 10.1109/ECTE-Tech62477.2024.10851191.

[12] omwoyo, R., Kamau, J., & Mgala, M. (2022). A review of Two Factor Authentication Security Challenges in the Cyberspace. International Journal of Advanced Computer Technology, 11(5), 1-6. Retrieved from https://ijact.org/index.php/ijact/article/view/112

[13] Y. Sharma, H. Gupta and S. K. Khatri, "A Security Model for the Enhancement of Data Privacy in Cloud Computing," 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, 2019, pp. 898-902, doi: 10.1109/AICAI.2019.8701398.

[14] P. V. Maitri and A. Verma, "Secure file storage in cloud computing using hybrid cryptography algorithm," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2016, pp. 1635-1638, doi: 10.1109/WiSPNET.2016.7566416.

[15] B. P. Gajendra, V. K. Singh and M. Sujeet, "Achieving cloud security using third party auditor, MD5 and identity-based encryption," 2016 International Conference on Computing, Communication and Automation (ICCCA), Greater Noida, India, 2016, pp. 1304-1309, doi: 10.1109/CCAA.2016.7813920.

[16] B. Swathi, S.D Bhaludra, S. Raveendranadh, "Secure File Storage In Cloud Computing Using Hybrid Cryptography Algorithm", International Journal of Advance Research in Science and Engineering 6(11), 2017.

[17] Kumar, Sanjeev & Karnani, Garima & Gaur, Madhu & Mishra, Anju. (2021). Cloud Security using Hybrid Cryptography Algorithms. 599-604. 10.1109/ICIEM51511.2021.9445377.

[18] Baqtian, H & Ali Al-Aidroos, Naziha. (2023). Three Hash Functions Comparison on Digital Holy Quran Integrity Verification. 11. 1-7.

[19] F. Kahri, B. Bouallegue, M. Machhout and R. Tourki, "An FPGA implementation of the SHA-3: The BLAKE hash function," 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13), Hammamet, Tunisia, 2013, pp. 1-5, doi: 10.1109/SSD.2013.6564030.

[20] kaur, Sandeep & Kaur, Gaganpreet & Shabaz, Dr. Mohammad. (2022). A Secure Two-Factor Authentication Framework in Cloud Computing. Security and Communication Networks. 2022. 1-9. 10.1155/2022/7540891.

[21] Colnago, Jessica & Devlin, Summer & Oates, Maggie & Swoopes, Chelse & Bauer, Lujo & Cranor, Lorrie & Christin, Nicolas. (2018). It's not actually that horrible: Exploring Adoption of Two-Factor Authentication at a University. 1-11. 10.1145/3173574.3174030.

[22] Lu, C. C., & Tseng, S. Y. "Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Application Specific Systems, Architectures and Processors", 2002. Proceedings. The IEEE International Conference on (pp. 277285).

[23] P. V. Maitri and A. Verma, "Secure file storage in cloud computing using hybrid cryptography algorithm," 2016 International Conference on Wireless Communications, Signal Processing and Networking (WiSPNET), Chennai, India, 2016, pp. 1635-1638, doi: 10.1109/WiSPNET.2016.7566416.

[24] Kumar, Sanjeev & Karnani, Garima & Gaur, Madhu & Mishra, Anju. (2021). Cloud Security using Hybrid Cryptography Algorithms. 599-604. 10.1109/ICIEM51511.2021.9445377.

[25] F. J. Aufa, Endroyono, and A. Affandi, "Security System Analysis in Combination Method: RSA Encryption and Digital Signature Algorithm," Proc. - 2018 4th Int. Conf. Sci. Technol. ICST

2018, vol. 1, pp. 1–5, 2018, doi: 10.1109/ICSTC.2018.8528584.

[26] F. Kahri, B. Bouallegue, M. Machhout and R. Tourki, "An FPGA implementation of the SHA-3: The BLAKE hash function," 10th International Multi-Conferences on Systems, Signals & Devices 2013 (SSD13), Hammamet, Tunisia, 2013, pp. 1-5, doi: 10.1109/SSD.2013.6564030.

[27] Baqtian, H & Ali Al-Aidroos, Naziha. (2023). Three Hash Functions Comparison on Digital Holy Quran Integrity Verification. 11. 1-7.

[28] Melo, Laerte & Amaral, Dino & Albuquerque, Robson & de Sousa Junior, Rafael & Sandoval Orozco, Ana & García Villalba, Luis. (2024). A Secure Approach Out-of-Band for e-Bank with Visual Two-Factor Authorization Protocol. Cryptography. 8. 51. 10.3390/cryptography8040051.

[29] I. J. Mathematical Sciences and Computing, 2020, 4, 35-41 Published Online August 2020 in MECS (http://www.mecs-press.org/) DOI: 10.5815/ijmsc.2020.04.04