International Journal of Safety and Security Engineering
Vol. 15, No. 7, July, 2025, pp. 1509-1518
Journal homepage: http://iieta.org/journals/ijsse

A Hybrid Encryption Method for Data Security Based on ARB3 Algorithm

Vijay Vamsi Nadakuditi*, Radhika Rani Chintala

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Guntur 522302, India

Corresponding Author Email: nvijayvamsi123@gmail.com

https://doi.org/10.18280/ijsse.150718

Received: 21 May 2025
Revised: 25 June 2025
Accepted: 15 July 2025
Available online: 31 July 2025

Keywords:
ARB3 algorithm, encryption performance optimization, data integrity, hybrid cryptography

ABSTRACT

In a world where cloud storing and online correspondence will increase in a faster pace, it will be very important that the cryptographic safeguarding will be such that will be solid and powerful. Even where traditional hybrid techniques of encryption have been found to be effective, it will not be spared of the problem of throughput bottleneck, performance pitfalls, and security vulnerabilities. To ensure synergy we will solve these problems using ARB3, a new algorithm to implement a hybrid cryptography approach to merge the strengths of Rivest-Shamir-Adleman, BLAKE3 hashing, and Advanced Encryption Standard into a new algorithm ARB3. The ARB3 Algorithm will take advantage of the high-performance and symmetric encoding of the AES, the key exchange process of RSA, and, high performance hashing of BLAKE 3 to develop an inclusive security protocol. Using testing, we shall have the advantage of showing that ARB3 will be better in terms of processing speed and provide an innocuous mix of security characteristics and that it will surpass traditional hybrid models in encryption and decryption. The overall increase in throughput in the ARB3 will ensure that it accepts large quantities of data without necessarily compromising their security or their speeds. As the proposed solutions will not only show the effectiveness of the ARB3 in eradicating the security threats, they will also contribute remarkably to the enhancement of the performance metrics that will become imperative in future data security. Since the restrictions of the existing hybrid cryptographic systems will be reconsidered, ARB3 is going to provide a more promising and effective way of converting and processing data in the framework of many applications. This will be the greatest advance in the cryptography field. The paper will also point out the topicality of the new hybrid mechanism of encryption that will be adapted to the shifting needs of the data security.

1. INTRODUCTION

Cryptography systems have relevance in the process of guaranteeing security of information. Cryptography form's part and parcel of modern protection of information. Economics describes cryptography as the process of modifying the readable messages to unreadable messages in a way that the true receiver of the message can read the message. The cryptograph is a term that is made using Greek terms kryptos, meaning hidden or secret and graphia meaning writing [1]. Cryptographic algorithms are very instrumental in maintaining confidentiality, authenticity, and integrity of data [2].

Cryptographic systems are commonly classified into symmetric, asymmetric and the hash- based cryptographic systems. Both of them include advantages and disadvantages:

1.1 Classification of cryptography

1.1.1 Symmetric cryptography

Symmetric encryption is one whose encryption and decryption are carried out using a shared key. Many well-known symmetric algorithms are AES, DES, and Blowfish [3]. The most commonly used AES is fast allowing a time complexity of about O(1) per block when used normally. Symmetric cryptography, however, is blind to the secure distribution of keys, and that becomes a big problem in an untrusted environment.

1.1.2 Asymmetric cryptography

Public-key cryptography or more generally asymmetric cryptography, utilizes two different keys: an encryption key, which is called a public key, and a decryption key, which may be called a secret key or a private key [4]. As an example, RSA offers good security features like authentication, non-repudiation and integrity [5]. But the operations of RSA are very computationally heavy, with a time complexity of O(n 3) measured in terms of key size. This is not efficient in encrypting big volumes of data.

1.1.3 Hash function

A hash function is a function that turns input content into a fixed-size digest and provides properties of determinism, efficiency, and resistance to pre-image and collision attacks [6]. Algorithms that are now considered insecure, such as MD5, have been retired; however, SHA-256 is considered secure. Also present is a more recent alternative, BLAKE3 which is significantly faster than classical hash functions (and reaches a speed of greater than 4 GB/s on a modern CPU) due to its optimizations and offers the same cryptographic guarantees [7].

1.1.4 Hybrid cryptography

The concept of mixed cryptography is characterized by various cryptographic schemes implementation and the possibility to achieve a performance-rich solution with an insignificant sacrifice of efficiency versus security. Encrypted data that is securely carried by using an asymmetric key would be usually offered in a symmetric key. Hybrid cryptography is the technique for better overall compute costs and usage of a secure key exchange.

Hybrid cryptography is currently being implemented much more widely as hybrid cryptography describes some of the following examples:

- HTTPS/SSL/TLS - based on the asymmetric key exchange, and a symmetric encryption of a session.
- E-mail Encryption (PGP/S-MIME) - having the symmetric keys are sad secured with an asymmetric methodology.
- Cloud Storage - provided many companies fast encryption and secured key [8].
- VPN - used an RSA algorithm to exchange and the key to enter a tunnel and associate the tunnel with AES.
- E-commerce - secured highly sensitive information with bulk encryption and key management process through the use of hybrid cryptography.

1.1.5 Why prefer hybrid cryptography

The combination of asymmetric and symmetric encryption enables hybrid models as shown in Table 1 to outperform individual techniques in both speed and security [9].

- The speed of data encryption is relatively high with AES encrypting approximately 1.5 GB/s as opposed to RSA whose encryption is fast through small keys.
- RSA-2048 key operations can take tens of milliseconds per transaction and AES-128 can operate on blocks at microsecond speeds.
- Hybrid systems lose major key management risks while enabling large scale fast data encryption.

1.2 Advanced encryption standard

The enhanced encryption standard is called AES [10]. This one is a member of the symmetric block cipher algorithm family. It encrypts and decrypts by means of a 128-bit, 192-bit, or 256-bit key. As seen in Figure 1, this technique makes use of an SP network with various rounds, which again rely on the size of the key being used [11]. In every round, there are four stages:

1. In the initial phase, we replace the bytes.
2. In the second phase, we move the rows.
3. We then combine the columns.
4. After that, we give it the round key.

1.3 Rivest-Shamir-Adleman (RSA)

RSA is asymmetric, means it uses two distinct keys: a private key and a public key. The private key is confidential and the public key is passed on to everybody. Rivest, Shamir and Adleman are attributed to the invention of RSA algorithm. The most famous aspect of RSA is its encryption method that is secure. The exponential usage of positive integer prime integers for encryption and decryption is the core idea behind RSA [12]. The RSA algorithm works in the following manner as shown in Figure 2.

This method applies two variables exponents namely ‘e’ the exponent of public and ‘d’ the exponent of the private. As plaintext, M, and ciphertext, C define the specified encryption, and decryption process as follows: the encryption and decryption procedure [13].

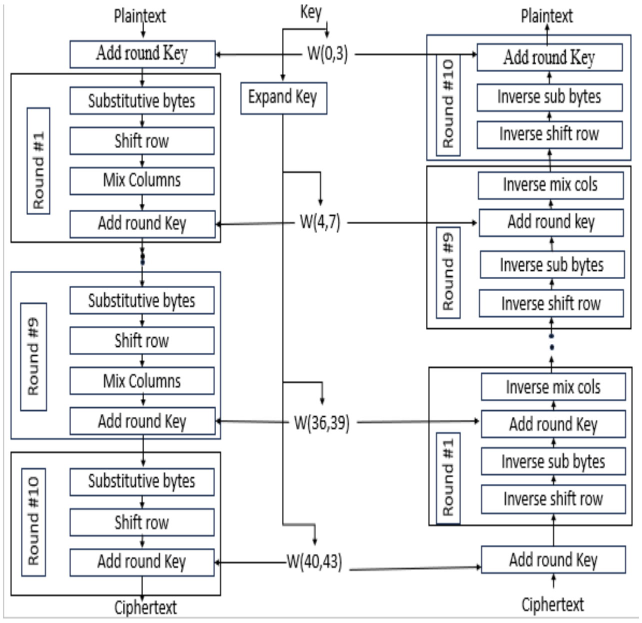

Figure 1. AES encryption and decryption system

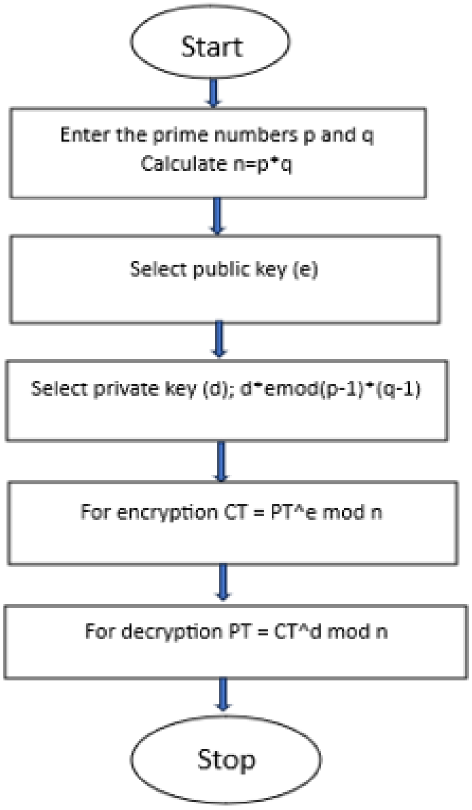
Start
Enter the prime numbers p and q
Calculate n=p*q
Select public key (e)
Select private key (d); d*emod(p-1)*(q-1)
For encryption CT = PT^e mod n
For decryption PT = CT^d mod n
Stop

Figure 2. RSA algorithm

1.4 BLAKE3

It 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 [14], but only 1.26 times better for a 256-bit hash. Performance is expressed in cycles per byte.

The primary steps involved in BLAKE3 operation are as follows:

Chunking: The data text is divided to the pieces of data where the standard size is 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) [15].

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 constructed out of the hash values of the every chunk. This continues up the tree until pairs of chunks have been combined and have their for that combination, and then are squashed into single parent nodes. Due to this architecture, BLAKE3 could efficiently allow parallel computation in that chunks are computed in parallel where they are then merged.

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. It is a procedure supplying security and originality to the ultimate hash. The output is either truncated or expanded to reach a length equivalent to the desired length usually of 256 bits or the average size of output of BLAKE3.

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 [16].

As demonstrated in the architectural diagram of BLAKE3 Figure 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.

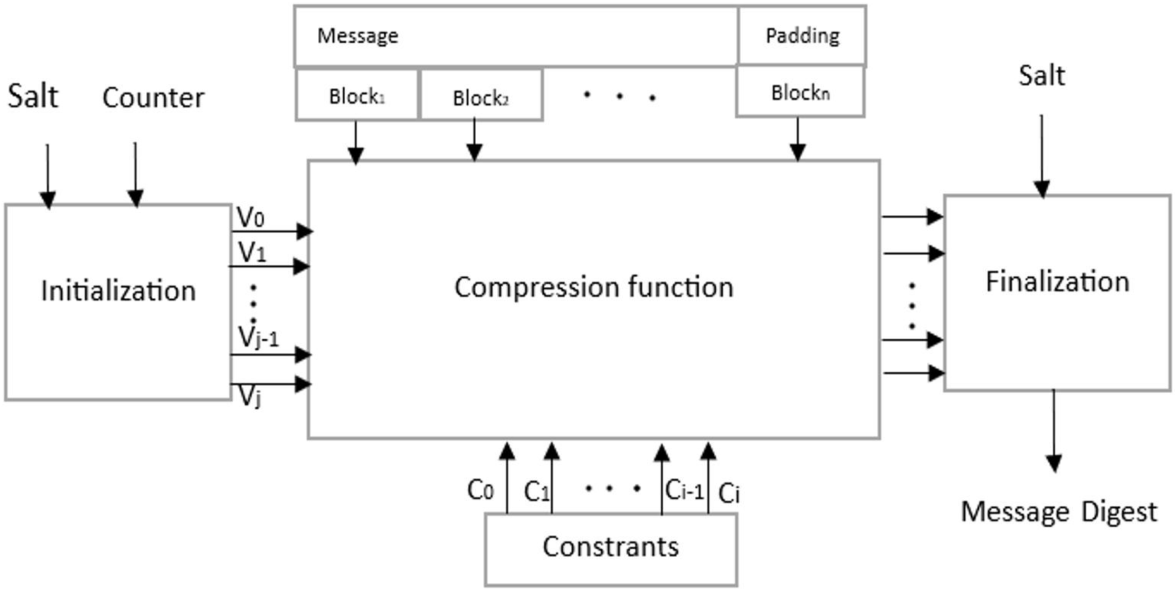

Figure 3. Architecture of the BLAKE3 hash function

The Table 1 shows the comparative analysis of the different hybrid cryptographic algorithms that have been implemented in the past works. This assessment aims at analyzing the performance of these algorithms in solving the cloud security problems where efficiency and reliability are equally significant. Parameters taken into consideration include the encryption time, the decryption time, memory used, and the time complexity. It would be necessary to measure the encryptions/ decryptions times that directly impact the speed of data security and retrieval during the real application. On the same note, the memory consumption helps to report on the efficiency of the algorithms in the application of resources, which is essential in computing setups with low affordability. Time complexity further on demonstrates the scalability of these approaches indicated how the algorithms scale with respect to larger datasets or high system loads. In order to have a complete performance analysis, a throughput calculation based on a specific formula, which measures execution speed and the data size, was also carried out. This makes it possible to compare the efficiency of the varying models of cryptography directly. The consolidation of these metrics gives a clear picture of the trade-offs between the strength of security and the computational performance of the hybrid cryptographic schemes.

Table 1. The comparison table presents an analysis of various hybrid cryptographic algorithms

Combination	Enc/Dec Time (s)	Memory Usage (MB)	Time Complexity	Throughput (Bytes/Sec)	Cryptographic Strength
AES + DES + RSA [17]	16.93 / 11.71	20.51 / 16.28	AES:O(1), RSA: O(n³), DES: O(1)	16,936,330.00	AES-128, RSA-2048, DES-56 (vulnerable)
AES + RSA [18]	18.48 / 13.08	12.01 / 16.00	AES: O(1), RSA: O(n³)	13,285,474.46	AES-128, RSA-2048
AES + RC6 + Blowfish [18]	17.00 / 27.00	14.08 / 12.89	AES: O(1), RC6: O(1), Blowfish: O(1)	9,532,509.00	All 128-bit symmetric keys; moderate resistance
RSA + MD5 [19]	30.69 / 30.25	10.35 / 16.74	RSA: O(n³), MD5: O(n)	6,881,289.00	RSA-2048, MD5 (128-bit; known collisions)
RSA + HMAC [20]	38.02 / 33.15	12.66 / 15.79	RSA: O(n³), HMAC: O(n)	5,892,200.49	RSA-2048, HMAC-SHA2 (256-bit; strong)
AES + MD5 [21]	18.93 / 14.65	12.73 / 16.09	AES: O(1), MD5: O(n)	12,486,131.99	AES-128, MD5 (128-bit; weak)
Blowfish + SRNN [22]	11.93 / 11.84	13.00 / 9.00	Blowfish: O(1), SRNN: varies	10,048,576.00	Blowfish-128, SRNN (uncertain; depends on model)

2. PRELATED WORK

This study proposes Advanced Encryption Standard encryption with file deduplication in order to enhance the security property of a file. Deduping was achieved through the BLAKE3 hashing system that enables one to eliminate the data block duplicates in a bid to maximize the available storage capacities. AES encryption is a sound method of protection of the distinctive files. In the given case, it was the balance between strong encryption and the optimization of data. This approach has been more appropriate in dealing with the issues of cloud computing whose primary requirement is security and efficiency. This research involves his investigation of the state of the art in these areas, focusing on the necessity of combining encryption and deduplication in order to realize an effective, but secure cloud architecture [23].

In such a situation the work concentrates on a more complex encryption processing system which employs a modified algorithm of DES to process an accounting information. The authors save much time in encryption processes by engineering the system architecture in terms of hardware acceleration points and incorporating a safer algorithm (AES) as compared to DES [24]. This shows that, beyond better security training there is need to have proper key management, backup and recovery procedures. Enhancements in systems are supposed to give the businesses reliable and efficient systems to safeguard the integrity and confidentiality of accounting information.

Data security is the main focus of this paper to protect against input of the unauthorized access and destruction. It has been a say it is a blend of the Pros of the public-key and symmetric-key encoding; this is the effective means of keeping the digital data secured. 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 need to provide safe data storage has been growing as the amount of data, and the increase in malware and phishing attempts contribute to the importance of the topic. It also addresses the problem of inability of their local servers to operate commands on large amounts of data which promote the use of cloud computing despite its security disadvantages. Researchers have suggested a novel technique to protect important data files after criticizing well-known algorithms like AES, DES, and RSA [25].

The article [26] discusses various strategies to the hybridization of cryptographic mechanisms with the aim to improve cloud security. It studies different hybrid cryptological models of 2013-2020 and provides the information about its applicable side, limitations, constructions, and implementations. It establishes that symmetric and asymmetric systems perform enhanced when they collaborate and that the security and performance can be augmented. Despite the fact that hybrid cryptography is restricted in several aspects based on the effective distribution of keys and the computational cost, the study concludes with an effective solution of cloud hybrids that offers security to sensitive data. I would support studies that try to close the gaps and improve Protected measures.

Increased security was also needed in every sector to ensure data secrecy. The robust hybrid encryption provided a good solution where both asymmetric and symmetric cryptography algorithms were integrated to code the messages. Symmetric algorithms which used encryption like the AES and asymmetric ones that used secure key exchange mechanism like the RSA were considered as they were being used to encrypt large databases. Relatively new research reports stated a usage of a twofold encryption method to ensure files safety in the cloud. Data Security Protection of keys management with the help of RSA-algorithms, rapid encryption and decryption of data using AES technologies, were involved in data security. The paper [27] asserts that there will be a major gain to the confidentiality and integrity of the data through the use of two various encryption protocols, AES, and RSA.The main management and calculation overhead problems were still open problems. It has been determined that optimized hybrid systems having an acceptable performance-security trade-off remain an area of research that needs to be studied in the future more so in association with emerging technologies like blockchain and the Internet of Things.

In paper [28] hybrid cryptography methods are reviewed to increase the security of text entered into the cloud storage. The suggested method encrypts and decrypts data using 3DES in conjunction with Blowfish encryption. 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. Besides the revisiting of the pertinent works in the domain this study describes the various methods of encryption and their effectiveness. The details of the implementation of the proposed system such as all the benefits of it namely the data and its integrity, extremely high security, its authentication and confidentiality is also discussed in detail.

High data security demands that researchers investigate hybrid cryptographic systems. Providing input text security by establishing hybrid cryptographic systems which are the product between symmetric and asymmetric encodings systems might be hard but need to be done. They can use asymmetric algorithms such as ElGamal and RSA to give secure key exchange mechanisms; symmetric algorithms such as Twofish and AES are fast and effective in encrypting large datasets. Integration studies are being carried out in these capabilities. An example is the hybrid procedure of RSA and Twofish that has been demonstrated to not only shrink the length of the ciphertext but also leading to an enhanced computing result. AES and RSA combine to afford greater security in general. The outward application of hybrid cryptosystems, as revealed in study [29], holds much promise in boosting the data security of cloud computing systems through the association between the positive level of encryption and a well-balanced key. They also discussed how ElGamal may be combined with symmetric primitives to give quantum error resistant security. Despite this being realized, other areas which needed research remained opened such as computing overhead and the complexity in key management. Therefore, the next developments would most likely pay attention to the scalability and optimum performance of such hybrid systems.

The article [30] elaborated that with the expanding use of cloud storage as a mode of storing data, there has been the development of a data security problem. The authors have highlighted the vitality of cryptography to safeguard information that is kept on third party servers. They examined a number of hybrid cryptographic algorithms that appeared since 2015 and up to early 2019. Their paper pointed out many benefits of using a combination of a number of encryption methods to enhance security. It categorized literature into tabular reviews and comprehensive ones, including issues that are majorly encountered, that is, user authentication can be ignored and hybrid algorithms may not be implemented. The authors concluded that the area of hybrid cryptography seemed to be in a favorable network security position; nevertheless, the user authentication and the viability of creations of such algorithms were subject to additional improvement in the cloud computing framework.

The paper [31] discussed the issues that need to be handled with regards to data security in cloud computing system. The authors have pointed out that there is a great necessity in ensuring the privacy of data and assuring safety. The organizations and consumers were increasingly becoming dependent on the cloud services. They looked into security technologies that already existed and referred to the shortcomings of cloud storage e.g. loss of information and unlawful access. In order to increase data security they introduced a paradigm of multi-level encryption of AES and RSA algorithms. This was a safe way since information would not be read by the unauthorized person as it was either open or covertly locked in some places. The authors affirm that with this procedure, data security, confidentiality, and integrity can be enhanced to great extent in cloud computing. They also gave the possible subject areas of research in cloud security which raise the issue of data lineage and remanence.

3. PROPOSED HYBRID ALGORITHM

Hybrid cryptography combines the elements of symmetric and asymmetric encoding and hashing; it exploits the superior security of AES, RSA, and BLAKE3 to securely protect massive sets of data with low performance overhead. At the start of the process, the primary data is encapsulated by an encryption method known as symmetric encryption of AES. Since AES is classified as a block cipher that ensures very high level of security at minimal performance reductions, it was chosen due to its high processing speed to encode masses of input texts.

The AES key itself must next be encoded using an asymmetric algorithm in this case, RSA. RSA uses a pair of keys, one a particular key that is used to decrypt and a second key may be shared which is used to encrypt. Due to this, the recipient of an AES key who knows of the corresponding private RSA key only will be capable of decrypting the key obtained with RSA, obtains the AES key, and unlocks the encrypted data. The key and block sizes of AES, RSA and BLAKE3 are represented in Table 2.

The last step is to create a tamper-proof hash of the original data using the cryptography hash function BLAKE3. Because BLAKE3 creates a distinct fingerprint of the data, unwanted modification and alteration are easily identified and prevented, allowing integrity testing. It is believed to be among the keenest ways of making sure that integrity is not compromised at the expense of speed because of its hashing speed and security.

Together, this hybrid approach strengthens security by transferring, as noted above, the speed of AES, the RSA benefit on key management, and the integrity guarantees of BLAKE3 into a powerful and secure cryptographic solution for modern applications.

After the user calls the application, it opens and the user may then choose a file to be encrypted through the use of the choose file button. The files path where it is chosen is shown in the entry box. When the user clicks the button labelled with Encrypt File, the actual encryption properly is begun by recording the time it begins and the memory usage. It is followed by the AES encoding where salt is randomly generated, and finally, the AES key would be derived with the help of PBKDF2HMAC. The chosen file is encoded using this AES key. The second procedure is RSA encryption protocol, in which the generation of keys is put into effect including the generation of the private and public key along with the RSA public key encryption employed to encrypt the AES key. BLAKE3 must then be used to hash the data and the AES key encoded. You have the option of storing the encrypted file and this will make changes to the time and memory utilization of the encryption. As much as the Tkinter window shows a graph between encryption time and memory, all this data will be plotted on the GUI.

Table 2. Cryptographic algorithm characteristics: Key size and block size for AES, RSA, and BLAKE3

Algorithm	Type	Key Size	Block Size
AES	Symmetric Block Cipher	256 bits (32 bytes)	128 bits(16 bytes)
RSA	Asymmetric (Public-Key)	2048 bits	Variable (depends on key size)
BLAKE3	Cryptographic Hash Function	256 bits	512 bits (64 bytes)

Similar to decryption, decryption is started by clicking the "Decrypt File" button. After the duration and memory is recorded, a copy against the BLAKE3 hash that generates the

RSA-encrypted AES key and AES-encrypted file contents is recorded. The RSA secret key is used to decode the final AES key so as to decrypt the contents of the encoded file. Then the encrypted file is requested to be saved and the memory consumption and the descrambling time is updated and displayed. Consequently, the graph of the decryption time and memory usage is shown in a Tkinter window. Consequently, such procedures would complete the encrypting task and offer the user encryption decryption information.

Flowchart: A multi-step process to encode and decode a file using the combination of the AES, RSA, and BLAKE3 hashing algorithms.

1. Start and Upload File: Here is the start of the uploading file and will be encrypted. It then undergoes several stages of the process of encoding.
2. AES Encoding: The algorithm Advanced Encryption Standard carries out the first level of encryption once after the file has been uploaded. Often referred to as the first level of encryption, AES encodes file data and produces a partially encrypted output.
3. A Random Key the RSA algorithm is used for RSA encryption. In order to attempt to secure the AES key, the RSA is used to encipher the AES key found with the first layer. The second-layer encoding will result from this phase.
4. BLAKE3 Hashing: Then the file data encoded by AES and AES key encoded by RSA are fed together into the BLAKE3 hashing algorithm. BLAKE3 is fast and secure hashing algorithm that has added an additional layer of the encryption.
5. Save Encrypted File: Finally, the place where the last encrypted data is written is discovered when the hash is complete. Here the encryption is complete and the file is encrypted and safe.
6. Decryption Process - Upload encrypted file: The first stage of decryption process is uploading the encrypted file once again.
7. BLAKE3 Decryption: Reverse BLAKE3 hashing to recover RSA-encrypted AES key along with AES-encrypted file data.
8. RSA Decryption: It employs secret RSA key that decrypts the AES key in a decryption process needed in the decryption of the file data.
9. AES Decryption: Data of the files that are encrypted at the initial level of the encryption is recoded with the help of the retrieved AES key.
10. Save decrypted file: Then the decrypt file is saved that an end.

In the whole workflow, multi-layered encryption security could be achieved using symmetric (AES) encryption techniques along with asymmetric (RSA) encryption. These would come in the context of integration with the integrity and hashing capability of BLAKE3. It will enhance data confidentiality and protection against unauthorized access as shown in Figure 4.

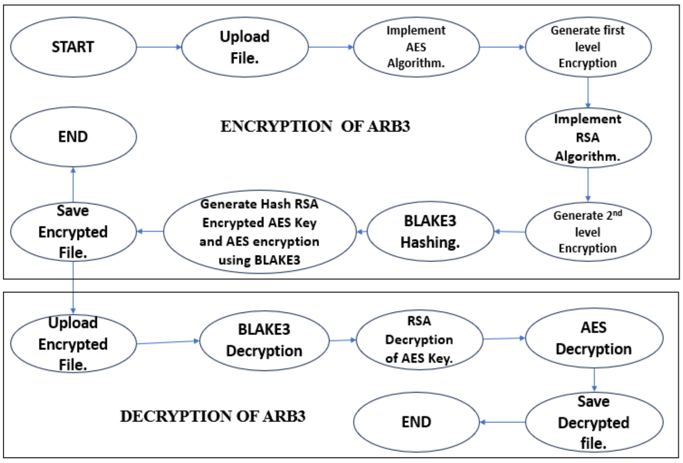

Figure 4. Proposed ARB3 algorithm flowchart

Algorithm 1. ARB3 Hybrid encryption and decryption

```
INPUT ← Plaintext file F
SECRET_KEY ← Static password string
SALT ← Static or randomly generated value
RSA_Keys ← Generate RSA public/private key pair
AES_Key ← Derive using PBKDF2HMAC from SECRET_KEY + SALT
Encrypted_F ← AES.encrypt(F, AES_Key)
Encrypted_AES_Key ← RSA.encrypt(AES_Key, RSA_Keys.public)
Hash ← BLAKE3(Encrypted_F + Encrypted_AES_Key)
OUTPUT ← Save (Encrypted_F, Encrypted_AES_Key, Hash) to file
Decryption:
INPUT ← (Encrypted_F, Encrypted_AES_Key, Hash) from file
AES_Key' ← RSA.decrypt(Encrypted_AES_Key, RSA_Keys.private)
Valid ← (BLAKE3(Encrypted_F + Encrypted_AES_Key) == Hash)
if Valid:
    Decrypted_F ← AES.decrypt(Encrypted_F, AES_Key')
    OUTPUT ← Save Decrypted_F
else:
    OUTPUT ← Integrity verification failed
```

4. RESULTS OF EXPERIMENTAL RESEARCH

4.1 Experimental setup

All experiments were conducted under the following hardware and software environment:

- Processor: Intel Core i7-11800H (2.30 GHz, 8 cores).

- RAM: 16 GB DDR4.
- OS: Windows 11 64-bit.
- Programming Language: Python 3.9.
- Libraries: cryptography, psutil, blake3, tkinter.
- Dataset: Uniform 4 MB plaintext files (text and log formats).
- Repetitions: Algorithm was tested 5 times, and the average value was taken to ensure consistency and reliability of results.

4.2 Encryption and decryption time in seconds

Encryption and decryption times, as shown in Table 3, are critical metrics that represent the responsiveness and speed of an encryption model. As can be seen in Figure 5, the overall performance of the ARB3 algorithm is the shortest at 7 seconds per encryption and decryption.

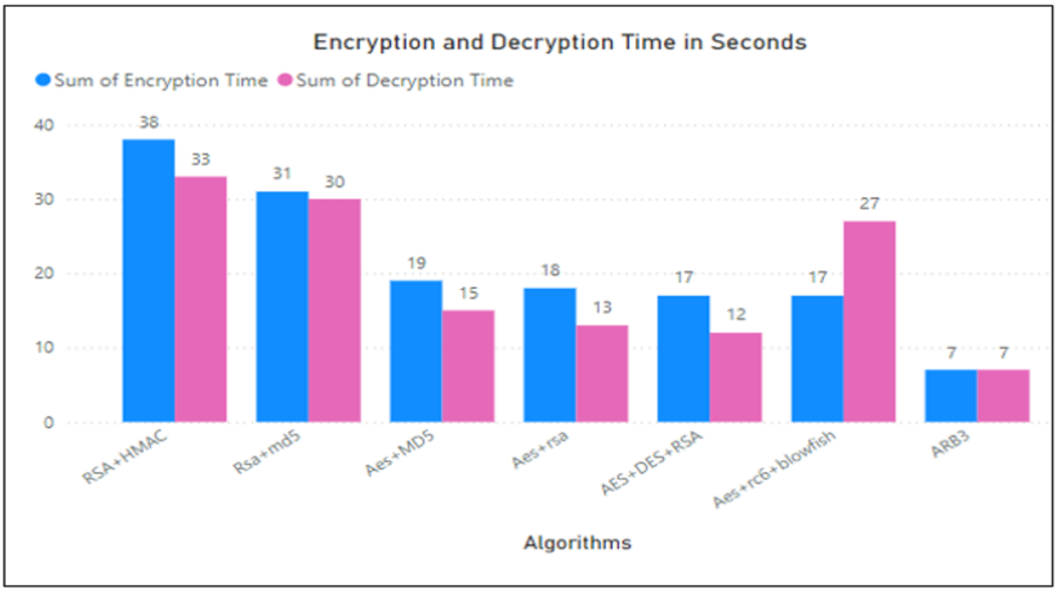

Figure 5. Encryption and decryption time in seconds

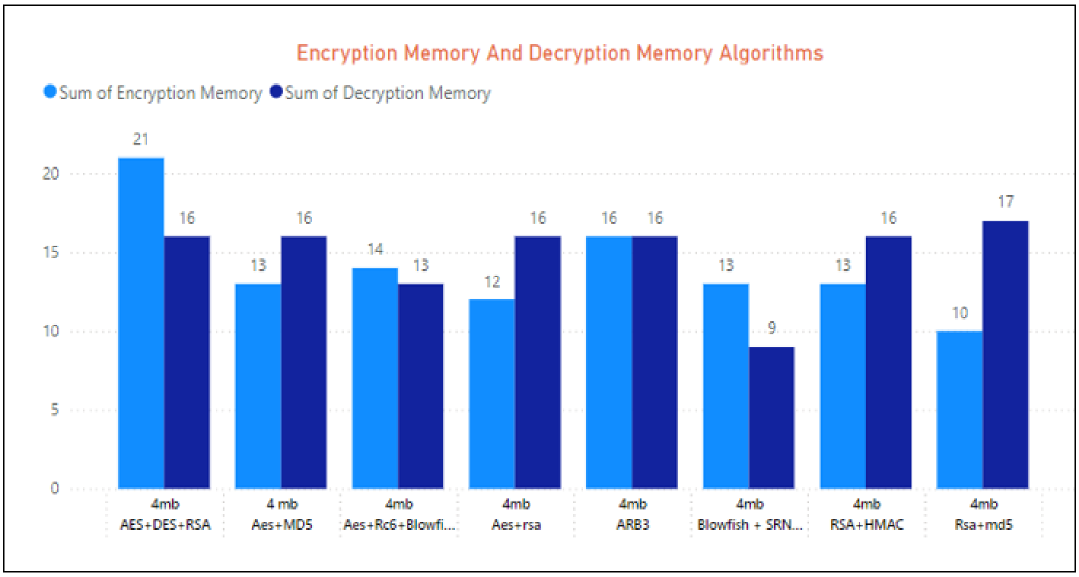

Figure 6. Encryption and decryption memory consumption

Table 3. Encryption and decryption time (in seconds)

Algorithm	Encryption Time (s)	Decryption Time (s)
RSA + HMAC	38.02	33.15
RSA + MD5	30.69	30.25
AES + MD5	18.93	14.65
AES + RSA	18.48	13.08
AES + DES + RSA	16.93	11.71
AES + RC6 + Blowfish	17.00	27.00
ARB3 (Proposed)	7.46	7.29

4.3 Memory consumption during encryption and decryption

The Table 4 presents the memory usage of encryption and decryption process of different hybrid algorithms. Figure 6 shows the summary of the amount of memory (in MB) used on the encryption and decryption works. The values indicate is the peak of memory recorded in execution through psutil library.as shown in Table 4.

Table 4. Encryption & decryption memory usage

Algorithm	Encryption Memory (MB)	Decryption Memory (MB)
AES + DES + RSA	21.00	16.00
AES + MD5	13.00	16.00
AES + RC6 + Blowfish	14.00	13.00
AES + RSA	12.00	16.00
RSA + HMAC	13.00	16.00
Blowfish + SRNN	13.00	9.00
RSA + MD5	10.00	17.00
ARB3 (Proposed)	16.00	16.00

4.4 Throughput

Throughput is the volume of data that may be processed in every second [32]. It is calculated using the formula:

$$Throughput = \left(\frac{T_p}{E_t + D_t}\right)$$

where, Tp = Enter plain text (Bytes). Et and Dt = Encryption-Decryption Time (seconds).

Table 5. Throughput for each algorithm

Algorithm	Throughput (Bytes/sec)	Share (%)
ARB3 (Proposed)	284,190.93	42%
RSA + HMAC	58,922.00	9%
AES + RC6 + Blowfish	95,224.64	14%
AES + RSA	132,854.74	19%
AES + MD5	124,861.32	18%
RSA + MD5	68,842.85	10%
AES + DES + RSA	16,936.33	2%
Blowfish + SRNN	10,048.57	1%

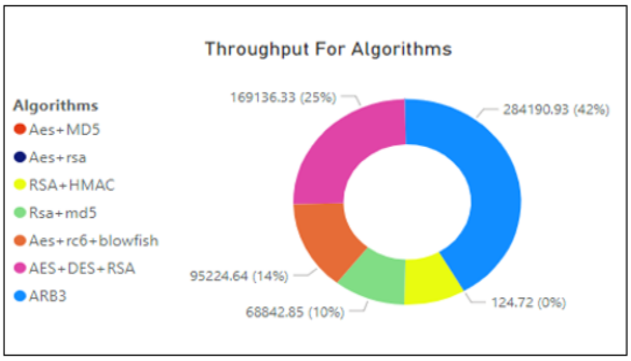

Figure 7. Throughput for algorithms

ARB3 far exceeds everything in its class by a wide margin. The RSA component of ARB3 has little overhead (relative to RSA's cryptographic capabilities of asymmetric encryption).as shown in Figure 7. This directly means performance improvements happen by greatly relieving the burden of processing from system overhead of asymmetric encryption. AES processing is trivial and defines little overhead with high speed. To assist data integrity for consistency and reliability, ARB3 uses the BLAKE3 hashing algorithm which is ultra-fast and designed to scale across multiple cores quickly in the Table 5. ARB3 provides such a powerful and effective hybrid with modern cryptographic features.

5. CONCLUSION AND FUTURE WORK

The proposed hybrid encryption system ARB3 is optimum and convenient in storing and transmitting data securely. Being a hybrid cryptographic algorithm, ARB3 has the advantage of AES to encrypt its data, the speed at which RSA can execute secure key distributions, and the high-speed and parallelity of checking integrity offered by BLAKE3, thus avoiding the drastic inefficiencies inherent in the traditional hybrid encryption processes, and, especially, the inefficiencies linked to throughput, latency and responsiveness of the system. It is evident that experimental outcomes show that ARB3 is better than typical combinations of RSA + HMAC, RSA + MD5, AES + RC6 and Blowfish. It had an encryption time of 7.46 seconds, decryption time of 7.29 seconds and greatest throughput amongst all the tested algorithms of 28,445,023.15 bytes per second and is particularly quality to real-time and high-performance applications.

ARB3 is structurally sound and fast enough to find a utility within real-world operations. In cloud storage applications, ARB3 acts to encrypt sensitive files before they can be offloaded to remote servers while BLAKE3 assures the integrity of the data during storage and synchronization. In secure communications, email systems, chat, or file transfer applications, ARB3 can encrypt messages quickly enough to deliver securely and in time. For Virtual Private Networks (VPNs), ARB3 provides a strong cryptographic basis from RSA for negotiating session keys, AES for encrypted tunnels, and BLAKE3 for detecting tampering. And, low memory profile and fast execution speed make it possible to deploy ARB3 to IoT, edge systems such as industrial sensors, medical devices. In addition, given its high integrity guarantee, ARB3 has a suitable category of applications for government and military systems where secure transfer of data is critical for operational success. As a conclusion forming the prospective planning, despite the good results that ARB3 has demonstrated, more has to be done. One path will be inclusion of post-quantum cryptographic cryptography with hardened resistance to quantum-computed attacks including Kyber and Dilithium. Another will be the design of lightweight versions of ARB3 for restricted-resource environments such as embedded system and low-power IoT solutions.

Also, the parallel processing optimization could be considered on multicore processors or graphical processing units (present GPUs) to shorten encryption and decryption time even more. Enlarging ARB3 trust model through secure multi-party computation (SMPC) and blockchain-based key verification technique would encompass its decentralized system use. To sum up, ARB3 is effective in filling the gaps inherent in existing methods of hybrid encryption owing to its high-performance, secure and scalable character. It shows good promise as a next generation cryptographic system applicable in a number of fields such as cloud computing, network security, embedded systems and governmental infrastructures among others. It has also acted as a forward-looking approach in ensuring the protection of sensitive details in the modern digital world that is now connected.

REFERENCES

[1] Sharma, H., Kumar, R., Gupta, M. (2023). A review paper on hybrid cryptographic algorithms in cloud network. In 2023 2nd International Conference for Innovation in Technology (INOCON), Bangalore, India,

pp. 1-5. https://doi.org/10.1109/INOCON57975.2023.10101044

[2] Fernando, E., Agustin, D., Irsan, M., Murad, D.F., Rohayani, H., Sujana, D. (2019). Performance comparison of symmetries encryption algorithm AES and DES with raspberry PI. In 2019 International Conference on Sustainable Information Engineering and Technology (SIET), Lombok, Indonesia, pp. 353-357. https://doi.org/10.1109/SIET48054.2019.8986122

[3] Zhang, Q. (2021). An overview and analysis of hybrid encryption: The combination of symmetric encryption and asymmetric encryption. In 2021 2nd International Conference on Computing and Data Science (CDS), Stanford, CA, USA, pp. 616-622. https://doi.org/10.1109/CDS52072.2021.00111

[4] Meng, Z., Wang, Y. (2022). Asymmetric encryption algorithms: Primitives and applications. In 2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI), Changchun, China, pp. 876-881. https://doi.org/10.1109/ICETCI55101.2022.9832032

[5] Dijesh, P., Babu, S., Vijayalakshmi, Y. (2020). Enhancement of e-commerce security through asymmetric key algorithm. Computer Communications, 153: 125-134. https://doi.org/10.1016/j.comcom.2020.01.033

[6] Jirwan, N., Singh, A., Vijay, S. (2013). Review and analysis of cryptography techniques. International Journal of Scientific & Engineering Research, 4(3): 1-6.

[7] Timothy, D.P., Santra, A.K. (2017). A hybrid cryptography algorithm for cloud computing security. In 2017 International conference on microelectronic devices, circuits and systems (ICMDCS), Vellore, India, pp. 1-5. https://doi.org/10.1109/ICMDCS.2017.8211728

[8] Kumar, A., Jain, V., Yadav, A. (2020). A new approach for security in cloud data storage for IoT applications using hybrid cryptography technique. In 2020 International Conference on Power Electronics & IoT Applications in Renewable Energy and its Control (PARC), Mathura, India, pp. 514-517. https://doi.org/10.1109/PARC49193.2020.236666

[9] Hyseni, D., Luma, A., Selimi, B., Cico, B. (2018). The proposed model to increase security of sensitive data in cloud computing. International Journal of Advanced Computer Science and Applications, 9(2): 203-210.

[10] Lu, C.C., Tseng, S.Y. (2002). Integrated design of AES (Advanced Encryption Standard) encrypter and decrypter. In Proceedings IEEE International Conference on Application-Specific Systems, Architectures, and Processors, San Jose, CA, USA, pp. 277-285. https://doi.org/10.1109/ASAP.2002.1030726

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

[12] Kumar, S., Karnani, G., Gaur, M.S., Mishra, A. (2021). Cloud security using hybrid cryptography algorithms. In 2021 2nd International Conference on Intelligent Engineering and Management (ICIEM), London, United Kingdom, pp. 599-604. https://doi.org/10.1109/ICIEM51511.2021.9445377

[13] Aufa, F.J., Affandi, A. (2018). Security system analysis in combination method: RSA encryption and digital signature algorithm. In 2018 4th International Conference on Science and Technology (ICST), Yogyakarta, Indonesia, pp. 1-5. https://doi.org/10.1109/ICSTC.2018.8528584

[14] Ciocan, I.T., Kelesidis, E.A., Maimuţ, D., Morogan, L. (2021). A modified Argon2i using a tweaked variant of Blake3. In 2021 26th IEEE Asia-Pacific Conference on Communications (APCC), Kuala Lumpur, Malaysia, pp. 271-274. https://doi.org/10.1109/APCC49754.2021.9609933

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

[16] Baqtian, H.S., Al-Aidroos, N.M. (2023). Three hash functions comparison on digital holy quran integrity verification. International Journal of Scientific Research in Network Security and Communication, 11(1): 1-7.

[17] Verma, V., Kumar, P., Verma, R.K., Priya, S. (2021). A novel approach for security in cloud data storage using AES-DES-RSA hybrid cryptography. In 2021, Emerging Trends in Industry 4.0 (ETI 4.0), Raigarh, India, pp. 1-6. https://doi.org/10.1109/ETI4.051663.2021.9619274

[18] Chinnasamy, P., Deepalakshmi, P. (2018). Design of secure storage for health-care cloud using hybrid cryptography. In 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), Coimbatore, India, pp. 1717-1720. https://doi.org/10.1109/ICICCT.2018.8473107

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

[20] Sajay, K.R., Babu, S.S., Vijayalakshmi, Y. (2024). RETRACTED ARTICLE: Enhancing the security of cloud data using hybrid encryption algorithm. Journal of Ambient Intelligence and Humanized Computing, 15(Suppl 1): 51. https://doi.org/10.1007/s12652-019-01403-1

[21] Matte, S., Dubey, A., Shirsat, N., Kale, A. (2018). Hybrid model for securing E-commerce transaction. International Journal of Scientific and Engineering Research, 9(4): 25-26.

[22] Sharma, M., Sharma, V. (2016). A Hybrid Cryptosystem approach for file security by using merging mechanism. In 2016 2nd international Conference on Applied and Theoretical Computing and Communication Technology (iCATccT), Bangalore, India, pp. 713-717. https://doi.org/10.1109/ICATCCT.2016.7912092

[23] Tajane, K., Pitale, R., Zambre, S., Huda, H., Utage, A., Dhar, V. (2024). Efficient cloud data deduplication with blake3 and secure transfer using AES. In 2024 4th International Conference on Pervasive Computing and Social Networking (ICPCSN), Salem, India, pp. 572-579. https://doi.org/10.1109/ICPCSN62568.2024.00096

[24] Li, Z. (2023). Exploration on accounting data encryption processing system based on DES algorithm. In 2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics

(AIKIIE), Ballari, India, pp. 1-6. https://doi.org/10.1109/AIKIIE60097.2023.10389923

[25] Susmitha, C., Srineeharika, S., Laasya, K.S., Kannaiah, S.K., Bulla, S. (2023). Hybrid cryptography for secure file storage. In 2023 7th International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, pp. 1151-1156. https://doi.org/10.1109/ICCMC56507.2023.10084073

[26] Murad, S.H., Rahouma, K.H. (2021). Hybrid cryptography for cloud security: Methodologies and designs. In Digital Transformation Technology: Proceedings of ITAF 2020. Springer, Singapore, pp. 129-140. https://doi.org/10.1007/978-981-16-2275-5_7

[27] Jaspin, K., Selvan, S., Sahana, S., Thanmai, G. (2021). Efficient and secure file transfer in cloud through double encryption using AES and RSA algorithm. In 2021 International Conference on Emerging Smart Computing and Informatics (ESCI), Pune, India, pp. 791-796. https://doi.org/10.1109/ESCI50559.2021.9397005

[28] Sharma, V., Chauhan, A., Saxena, H., Mishra, S., Bansal, S. (2021). Secure file storage on cloud using hybrid cryptography. In 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, pp. 1-6. https://doi.org/10.1109/ISCON52037.2021.9702323

[29] Jintcharadze, E., Iavich, M. (2020). Hybrid implementation of Twofish, AES, ElGamal and RSA cryptosystems. In 2020 IEEE East-West Design & Test Symposium (EWDTS), Varna, Bulgaria, pp. 1-5. https://doi.org/10.1109/EWDTS50664.2020.9224901

[30] Ahmad, S.A., Garko, A.B. (2019). Hybrid cryptography algorithms in cloud computing: A review. In 2019 15th International Conference on Electronics, Computer and Computation (ICECCO), Abuja, Nigeria, pp. 1-6. https://doi.org/10.1109/ICECCO48375.2019.9043254

[31] Sharma, Y., Gupta, H., Khatri, S.K. (2019). A security model for the enhancement of data privacy in cloud computing. In 2019 Amity International Conference on Artificial Intelligence (AICAI), Dubai, United Arab Emirates, pp. 898-902. https://doi.org/10.1109/AICAI.2019.8701398

[32] Subedar, Z., Araballi, A. (2020). Hybrid cryptography: Performance analysis of various cryptographic combinations for secure communication. International Journal of Mathematical Sciences and Computing, 6(4): 35-41. https://doi.org/10.5815/ijmsc.2020.04.04