Enhancing Cloud Storage Privacy (CSP) Based on Hybrid Cryptographic Techniques

Mohammed Y. Shakor 1,2, Mustafa I. Khaleel 2, Fadhil S. Abed 3

1 Research Center, University of Garmian
2 Department of Computer, College of Science, University of Sulaimani
3 Department of Information Technology, Sulaimani Polytechnic University

Abstract
Cloud computing has emerged as a new technology model for providing customized, high quality and reliable services for clients over the Internet. These clients can rent the services as long as they need rather than owning and maintaining large expensive IT infrastructure. One of the most used cloud services is the cloud storage as it provides storage on demand service. Nevertheless, due to the structure of cloud services which may not fully be managed by the cloud clients, the privacy and security of cloud storage become a significant challenge. Most of the previous works on the cloud security trying to solve the security challenges with data encryption techniques. However, these solutions do not protect data from being exposed or read by the server administrator or owner. In this paper, we have proposed a hybrid novel solution referred to as Cloud Storage Privacy (CSP) paradigm to enhance the privacy policies for cloud storage. This framework solution combines three of the superior cryptographic techniques: RSA, AES and Block-Chain. The solution not only secures the asset in the cloud but also protects the asset from unauthorized server access and improving user password protection. We have compared the proposed solution with the best based on our knowledge cloud storage services which are Google Drive and Dropbox in terms of performance and security. The result shows that the proposed solution outperforms existing services.

Introduction
There has been rapid progress in cloud computing since the last few years. Cloud computing provides users with a wide range of resources, such as web services, computer platforms, storage and applications over the Internet. In 2017, the internet records about 7.5 billion users and this number is growing rapidly [9]. Therefore, many companies redirect their works to be as cloud services providers such as Microsoft with Office 365 and Google with Google Docs and Drive. Nowadays, storage service over the Internet, which named cloud storage, is gaining more
popularity as an efficient solution for mobile storage [28]. Due to the lossing of control over data in the cloud server and the importance of client data, the privacy and security issue of this service is very important and discussed in many papers in order to propose optimal security solution [13, 21]. As there are many unsolved issues in cryptographic schemes such as the key management issue in symmetric cryptography algorithms [14] and the performance of encryption in the asymmetric algorithms [22]. Many solutions tend to use a hybrid solution by combining two or more algorithms [1, 8].

The proposed solutions are tending to use RSA and AES algorithm as they are considered most secure algorithms [12]. Furthermore, some works start using block-chain technique to improve the security of the cloud [19] such as improving data integrity [4]. The main issue in these solutions is the data protection in the server. The proposed algorithms are allowed to the server or services providers to read the data as the encryption/decryption keys stored in the server itself. Moreover, the solutions do not provide an efficient solution to provide a dedicated key for each user in the cloud due to the difficulty of that.

In this paper, we have proposed a hybrid novel solution heuristic namely Cloud Storage Privacy (CSP) in order to improve the confidentiality of the data in the cloud by combining three algorithms which are: RSA encryption, AES encryption and Block-Chain techniques. In section 2, we have explained the literature reviews of the latest proposed solutions in the cloud storage security in order to address the pros and cons of each solution. In section of state of art, we have formulated the environment that we used to apply our proposed solution and how we have calculated the performance of the framework. Furthermore, we have described in details the proposed solution concept. In Section 4, we have described the main component of the proposed solution with a flowchart for each and provide algorithm description for the login, encryption and decryption. For section of experimental results, the comparison between our paradigm and other algorithms have been discussed in terms of performance and security.

2. Literature Review

There are numerous studies and researches executed to beautify the security of cloud computing services and surroundings using encryption techniques and other methods. However, there was a mild development within the effects of these works evaluating with the fast increase of cloud computing communications. Normally, protection models in cloud-based environments are divided to authentication models which includes [17], facts protection fashions consisting of [23], and access management fashions which include [27]. Combining RSA and AES algorithms is one of the possible protection solutions for securing cloud storage services. Venkatesh et al. [26] delivered RSA based storage safety (RSASS) technique by means of improving existing RSA based signature era to apply on the public auditing of the far-flung data. The method the use of RSA a public key cryptography to offer strong safety which could ensure the statistics storage correctness and also identity of misbehaving server with excessive opportunity. Fatemi Moghaddam
et al. [16] compares 6 uneven key algorithms in cloud computing environment to research capability and efficiency of each algorithm. The evaluation of this comparison became completed primarily based on time, key length, and protection parameters. In step with the nature of data encryption service, three factors had been considered to pick the maximum suitable statistics encryption method in consumer aspect: acceleration, accuracy, and protection. the key generation process is performed in keys cloud server and it turned into completely separated from the encryption and decryption manner, but the cryptography methods are completed in a purchaser aspect with restricted processors and memories, and numerous software program. In 2018, [15] have been working on securing the healthcare data in cloud using a hybrid scheme to protect the data between sender and receiver. The proposed algorithm has used linear network coding and re-encryption based on ElGamal cryptography. The solution used linear network coding mechanism to provide a fault tolerance for cloud storage. Furthermore, the ElGamal re-encryption scheme is used to exchange the encoding key. However, the solution has a drawback which is using the same key to encrypt all the data in the storage and does not provide a server protection for the data in the cloud which means that the cloud services providers can read and edit the data. [1] have proposed a hybrid solution to secure file storage in cloud. The solution is combining between AES, BlowFish, RC4 and BRA algorithm to provide block wise security. It uses 128-bit key size for encryption and LSB algorithm to hide the key with steganography technique. The proposed solution protects the file in cloud but it has many draw backs. The key size is relatively small compared with the current trend of key size. Furthermore, LSB algorithm can be easily broken with brute force attack.

In average, the main intention of all researches is to investigate feasible methods to decorate the safety of cloud computing services. Therefore, on this work, a relaxed framework has been proposed for more securing personal tasks being saved in cloud systems the use of hybrid of AES and RSA encryption methods.

3. State of The Art

The proposed solution is focusing on improving the security of the cloud storage by combining RSA, AES and block-chain technique which are considered to be the most well-known algorithms in the security field. RSA is successful to guide encryption and digital signatures [20]. It gives the first-class safety coverage by encrypting the records which is private. That is the cause why the big carrier providers like Google mail, Yahoo mail etc. are the use of this set of rules to present their users the insurance of confidentiality in using their services. AES algorithm stated that up till nowadays, no possible attack against AES exists and it use in many important applications such as cloud and Internet of Things (IoT) [24]. As a consequence, governments, banks and high protection structures round the world are favored the usage of AES for the encryption trendy. Blockchain technology is not always just only single one approach, but consists of cryptography, arithmetic, algorithm and economic version, combining peer-to-peer networks and the use of allotted consensus algorithm to solve conventional disbursed
database synchronize trouble, it is an incorporated multi-eld infrastructure production [5, 10, 6].

One of our major contribution includes: incorporating the private block-chain to improve both user authentication and data integrity via dual steps. The block-chain technology considers as high secure technique to protect user identity in the cloud and IoT [3] because the ability to protect the privacy and managing the clients over the network. Moreover, some research start studying and addressing the advantage of using block-chain technique for a high security and performance cloud and distributed system then they conclude that block-chain technique is very efficient to use to protect records and data [25]. First, we improve the security of user authentication by replacing the old and insecure algorithm which is MD5 with RSA algorithm. The user password will encrypt and store in the database using the RSA algorithm with 1024-bit key size. This will ensure that the passwords will be protected even from the user admin. One of the best well-known application of block-chain which present how the block-chain protect the user identity over the internet is Bitcoin cryptocurrency [2].

In the second step, which is the main security improvement in the proposed solution, we focused on data security. The problem with data security in the cloud that the cloud administrator can read the information of the users in the cloud which includes the user files. For that, we have used AES algorithm to encrypt the user files but on the client side not on the server side. With that, the server will receive encrypted data and will not be able to decrypt it. At that point, we have found the issue of key-chain. The problem is how the user can save the AES key to encrypt and decrypt the own data. Using 256-bit key is impossible to remember by the user and saving key as a file in the user device will lead to two issues.

![Figure 1: Proposed Framework Flow Diagram](image)

First, the user will not be able to use another device to get the data from the cloud and this against the cloud concept. The second issue is that the use needs extra technique to protect the key file in his/her device. In this point, the idea of using the new block-chain technique is coming. Block-chain is a high secure algorithm to generate data as a chain and enable to validate and get the same data if we apply the same chain series. We have used the user password with a specific block-chain series for each user in order to generate the AES key for encryption and decryption at the client side. First, the user will input his/her password then the password will be
use as chain data with a different block series for each user to generate a unique and secure password. The user will not need to remember the key and just need to enter his/her password. Using different block numbers in the block-chains will provide a secure and unpredictable key for AES. Furthermore, using a hash algorithm in the block-chain will provide a right key-size every time. Figure 1 explains the entire levels of our heuristic which includes both the encryption and decryption process.

4. System Architecture and Algorithm

- Cloud User: In the proposed solution, the user has to send the plain information at the registration stage to be saved in the cloud database and give the user authentication and authorization to use the cloud services. When user login, the plain password with login username is send to the server (as shown in Figure 2). The password is encrypted using RSA algorithm then the saved password which is located in database is read to match the input password with database password. If the user is valid, the server will generate a user token which is a long hexadecimal number to be used in client server requests.

- Cloud Scheduler: The cloud computing is attracting an increased number of applications to run in the remote data centers. Many complex applications require parallel processing capabilities. Some of the parallel applications show a decrease in utilization of CPU resources whenever there is an increase in parallelism if the jobs are not schedule correctly then it reduces the computer performance [7, 11]. These process is called cloud scheduler. The cloud that are used to deploy the cloud storage web application of the proposed solution used a simple python program to schedule all the process automatically. In our proposed solution, the system has two main tasks: encryption (Figure 3) and decryption (Figure 4). These two tasks are managed and balanced automatically by the server to be processed in best performance.
Cloud Infrastructure: The cloud application has been hosted on a dedicated server with dual processing unit. The database also hosted on the same server but with a backup file automatically created for the changed on a different server which is connected to the internet when need to preserve the database content. Furthermore, a python application with Nginx and Gunicorn manager is set to manage these infrastructures and connect the real IP with the cloud storage application. Files of the user are located on the server SSD drive.

Additionally, we need to explain the pseudo code of the framework as an algorithm. Accordingly, the credential encryption, file encryption and file decryption steps are explained as follows.

**Algorithm 1 Credential Encryption**

1: Input: Credential information
2: Output: User Cross-Site Request Forgery (CSRF) token
3: Initialize Flag = 1
4: for each server $\in$ avail_server do
5:   if (server; == Flag) then
6:     Server; $\leftarrow$ user_cred
7:     RSA_encrypt $\leftarrow$ user_pass
8:     if (server; == Flag) then
9:       compare(user_pass, db_pass)
10:      if (user_pass $== db_pass$) then
11:        return web token
12:      end if
13:    end if
14:  end if
15: end for

**Algorithm 2 File Encryption / Uploading**

1: Input: user credential
2: Output: message authenticated
3: block – chain miner $\leftarrow$ user_pass
4: for Each block to block – chain size do
5:   generate: block miner
6: end for
7: AES_k $\leftarrow$ latest_block.data
8: initiate encryption process
9: cloud_resource $\leftarrow$ user_asset
10: if ((user_per) & (user_auth) is validated) then
11:   redirect encrypted message to local host
12: end if

**Algorithm 3 File Decryption / Downloading**

1: Input: user credential
2: Output: message decrypted
3: block – chain miner $\leftarrow$ user_pass
4: for Each block to block – chain size do
5:   generate: block miner
6: end for
7: AES_k $\leftarrow$ latest_block.data
8: server $\leftarrow$ request
9: if ((user_per) & (user_auth) = cloud_db) then
10:  retrieve encrypted message from local host
11: end if
12: return encrypted data
13: Initiate decryption process

---

Figure 4: Decryption Flowchart
5. Experimental Results

5.1. Simulation
For this experiment, we have simulated our algorithm based on multi-paradigm programming language, python, with incorporating open-source web framework, Django, to evaluate the secrecy performance of our underlying cloud servers. Furthermore, we have installed a real cloud server within Ubuntu operating system and capacity of three GBs of RAM and 2 CPUs. We also reserved Digital Ocean Platform (DOP) as a cloud infrastructure provider to initiate our proposed algorithm, Cloud Storage Privacy (CSP), for the sake of data encryption accuracy. On the other hand, we have used open-source programming language, Typscrip, with open-source front-end web application framework, Angular, to develop the client end and process the operations in order to maintain the highest security requirements. Following screen shots (5 and 6) show the cloud tenant’s registration and sign-in page and cloud storage privacy web page.

![User Sign-In](image1.png) ![User Registration](image2.png)

**Figure 5: Tenants Sign-In and Registration Pages**

![Cloud Storage Privacy](image3.png)

**Figure 6: Cloud Storage Privacy Page**

5.2. Result Discussion
We have tested three metrics to evaluated the performance of our proposed algorithm. These metrics include encryption and decryption executing speed, memory usage and the level of CPU allocated. Moreover, we have compared our heuristic with standard traditional AES and Block-Chain techniques.
Table 1: Features Comparison

| Feature                  | Google Drive | Dropbox |
|--------------------------|--------------|---------|
| Dedicated Key            | No           | No      |
| Client Side Encryption   | No           | No      |
| File-in-Server Protection| No           | No      |
| Password Protection      | Yes          | Yes     |
| Key Distribution         |             |         |
| Encryption               |             |         |
| Protection               |             |         |

which additionally provide the third feature, file-in-server protection. The last feature is the password protection. Our proposed solution provides an additional layer of security by using encryption algorithm instead of hashing algorithm in order to protect the user password.

The reason of using block-chain technology in server-side to generate the key of user was because of the key distribution issue. On the other hand, it does provide an optimum result because generating asymmetric keys is a time consuming. After calculating the performance of key generation for the user we got the results as shown in Table 2.

Table 2: Block-Chain Generation Speed

| No. Of CPUs | CSP Generating Speed Per Second |
|-------------|---------------------------------|
| 1           | 2.5                             |
| 2           | 6                               |

The speed of block-chain mining is calculated and compared with the speed of block-chain mining of the most popular block-chain application which is Bitcoin to show the simplicity of the Block-chain of the proposed solution. In Bitcoin mining and based on [18], we need enormous computation power and energy to mining one block due to the hashing condition and complexity of the bitcoin mining. The block-chain requires performing millions of hash function to get a correct block-chain data [18]. However, the proposed solution is using the base block-chain concept therefore is can perform fast enough to use for key generation. Furthermore, increasing number of CPUs can significantly increase the speed.

Table 1 explains the first comparison stage of our heuristic versus characteristics such as dedicated key, file-in-server protection, and password protection. Dedicated key for each user which is provided by the block-chain technology helps to give each user a different key for file encryption. The second feature is client-side encryption which protect the data from being seen and read by the server side.
of block generation. Therefore, if the system uploaded to a high-performance cloud server, the block-chain generating speed will be very high and as a result the file uploading speed will increase. The third part of the results is the time of encryption and decryption of the proposed solution. This result contains also the time of key generation for encryption and decryption. As describe before, the proposed solution edits the part of key generation in the AES encryption process. Therefore, the encryption speed which results in file uploading should be high. We test the speed of file encryption and decryption with different file size and the results shown in Table 3.

| File Size | Encryption Time (s) | Decryption Time (s) |
|-----------|---------------------|---------------------|
| 10 KB     | 1.5                 | 1.7                 |
| 100 KB    | 2.5                 | 3.2                 |
| 1 MB      | 3.6                 | 4.5                 |
| 10 MB     | 4.1                 | 5.3                 |

We calculate the time of encryption and decryption with key generation in order to subtract these result from uploading and downloading time to get the exact time for file uploading and downloading compared with Google Drive and DropBox. The speed of uploading and downloading has been calculated on the same network speed with the same files.

Table 3: Encryption/Decryption Speed

| File Size | Proposed Solution | Google Drive | DropBox |
|-----------|-------------------|--------------|---------|
| 10 KB     | 2.5               | 3.8          | 4.3     |
| 100 KB    | 5.5               | 8            | 12.6    |
| 1 MB      | 10.3              | 13.4         | 16.7    |
| 10 MB     | 48.6              | 76.3         | 95.3    |

Table 4: Uploading Speed (in second)

| File Size | Proposed Solution | Google Drive | DropBox |
|-----------|-------------------|--------------|---------|
| 10 KB     | 0.5               | 1.2          | 2.3     |
| 100 KB    | 1.5               | 2.1          | 3.4     |
| 1 MB      | 2.1               | 2.4          | 4.7     |
| 10 MB     | 4.6               | 6.3          | 9.3     |

As shown in Table 4 and Table 5, the uploading speed of the proposed solution is much better than Google Drive and DropBox after subtracting the encryption time duration. This show how light the cloud application and how efficient the combination of the algorithms. The downloading speed in present the time between send request and start downloading the file. We can notice that the speed of downloading in all platform is much better than uploading because of the saving file time, encrypting the file in server (in Google Drive and DropBox) and the feedback message from the server which is the message of successfully uploading. Because the system is working on both server side and client side, the processes are divided between them. Most of the operations are on the client side therefore, we need to calculate the memory usage of the system of the client side to ensure that it will work on any client device. The results are compared with Google Drive and DropBox cloud storage.
Cloud Storage Privacy (CSP) v.s. Memory Consumption

Figure 7: Cloud Client Memory Usage

Table 6: Memory Usage

| File Size | Proposed Solution | Google Drive | DropBox |
|-----------|-------------------|--------------|---------|
| 10 KB     | 340 KB            | 530 KB       | 640 KB  |
| 100 KB    | 560 KB            | 780 KB       | 860 KB  |
| 1 MB      | 600 KB            | 900 KB       | 1100 KB |
| 10 MB     | 750 KB            | 1360 KB      | 1450 KB |

The results show that the proposed solution keep memory usage minimal. Furthermore, memory usage is much better than Google Drive and DropBox and that enable the system to work with low memory space. The CPU usage of the system need to be calculated on the client side to ensure that it will work on limited processing device. The results are compared with Google Drive and DropBox cloud storage.

The result as shown in Figure 8 indications that the proposed solution utilizes minimal CPU usage. Furthermore, the usage is much better than Google Drive and DropBox instead of encryption/decryption.

Table 7: CPU Usage

| File Size | Proposed Solution | Google Drive | DropBox |
|-----------|-------------------|--------------|---------|
| 10 KB     | 10%               | 15%          | 20%     |
| 100 KB    | 15%               | 25%          | 35%     |
| 1 MB      | 30%               | 40%          | 45%     |
| 10 MB     | 35%               | 55%          | 65%     |

Cloud Storage Privacy (CSP) v.s. CPU Allocated

Figure 8: Cloud Client CPU Allocated

process on the client side. All in all, compared to existing method, a Multi-Level technique of encryption is more secure to use. The mentioned drawbacks must be considered in future works to enhance the security of cloud computing services.

Randomness Test with NIST CrypTool

There are four criteria need to be considered when we check the randomization of any number value. These criteria are frequency, serial, poker, and runs test. The frequency test represents both numbers: zero and one in the file. The serial test indicates to the number of zero and one in each pair of the file and the poker test represents the number of zero and one in each sequence of five.
number of that file. However, the run test explains the pattern of the file. To test these parameters via our proposed solution, we have applied the CrypTool from NIST. The following tables illustrate the superior that our paradigm has achieved. As we can observe that our paradigm with file type (docx) has superior to the standard AES algorithm in terms of frequency, poker, runs, and serial tests which takes 3 out of 4. However, for PDF files, our proposed solution has improved in terms of frequency with up to 100 times better than the others. Furthermore, it gains the same exact outcome for the test of poker testbed. For the ZIP files, our heuristic was higher than the standard AES in terms of both frequency and serial testbed with the same exact result in the poker test. Moreover, for the jpg file type our method obtained the same results for both the runs and poker testbed and overcome the standard AES in terms of frequency.

### Table 8: Standard AES Test with 256-bit Key Size

| File Type | Frequency Test | Poker Test | Run Test | Serial Test |
|-----------|----------------|------------|----------|-------------|
| Docx      | 0.950          | 3.841      | -        | 2.275       |
|           | 5.99           | 2.993      | 9.488    |
| Pdf       | 1.415          | 3.841      | -        | 1.228       |
|           | 5.99           | 1.228      | 9.488    |
| Zip       | 2.176          | 3.841      | -        | 4.317       |
|           | 5.99           | 4.317      | 9.488    |
| Jpg       | 0.815          | 3.841      | -        | 3.527       |
|           | -683.2         | 3.841      | 9.488    |

6. Conclusion and Future Work

In this paper, the problem of facts safety in cloud data storage changed into investigated, which is basically a dispensed storage machine. Cryptography approach frequently used to comfortable the information transmission and storing among person and cloud storage offerings. The focal point of this paper was on providing comfortable files transmission among these entities. An aggregate of uneven and symmetric encryption strategies (i.e. RSA and AES encryption strategies) became proposed in this approach to achieve the assurances of cloud information safety. The point of interest was on RSA encryption to provide trouble for attackers to get the person password and using AES to guard consumer files. Furthermore, we’ve got used block-chain method to defend and generate a unique AES key for every consumer. The procedure of sending the documents to the cloud and retrieving the files from the cloud was finished with the aid of symmetric and block-chain encryption respectively. The results show that the proposed solution
provides a high security and flexible technique to protect data in the cloud. However, because of the rapid development of security field and computer performance, many algorithms are being proposed every day therefore the proposed technique can take the advantage of that and can be improved. The encryption techniques can be replaced with any new algorithm that may be proposed in the future. Moreover, increasing the key size and block-chain in order to elevate the security of encryption by taking the advantage of processing power.

| File Type | File Frequency Test Value | Poker Test Value | Run Test Value | Serial Test Value |
|-----------|--------------------------|------------------|----------------|------------------|
| Docx      | 0.9877                   | 3.841            | 14.070         | 2.760            |
| Pdf       | 0.001                    | 3.841            | 14.070         | 2.760            |
| Zip       | 1.522                    | 3.841            | 14.070         | 2.760            |
| Jpg       | 0.709                    | 3.841            | 14.070         | 2.760            |

Table 9: Blockchain-Key AES Encryption (with 3-chain and 256-bit key)

### References

1. Bala, B., Kamboj, L., Luthra, P., 2018. Secure file storage in cloud computing using hybrid cryptography algorithm. International Journal of Advanced Research in Computer Science 9 (2).
2. Dorri, A., Kanhere, S. S., Jurdak, R., 2016. Blockchain in internet of things: challenges and solutions. arXiv preprint arXiv:1608.05187.
3. Dorri, A., Kanhere, S. S., Jurdak, R., Gauravaram, P., 2017. Blockchain for iot security and privacy: The case study of a smart home. In: Pervasive Computing and Communications Workshops (PerCom Workshops), 2017 IEEE International Conference on. IEEE, pp. 618–623.
4. Gaetani, E., Aniello, L., Baldoni, R., Lombardi, F., Margheri, A., Sassone, V., 2017. Blockchain-based database to ensure data integrity in cloud computing environments.
5. Garay, J., Kiayias, A., Leonardos, N., 2015. The bitcoin backbone protocol: Analysis and applications. In: Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, pp. 281–310.
6. Gervais, A., Karame, G. O., Capkun, V., Capkun, S., 2014. Is bitcoin a decentralized currency? IEEE security & privacy 12 (3), 54–60.
7. Ghanbari, S., Othman, M., 2012. A priority based job scheduling algorithm in cloud computing. Procedia Engineering 50 (1), 778–785.
8. Goyal, V., Kant, C., 2018. An effective hybrid encryption algorithm for ensuring cloud data security. In: Big Data Analytics. Springer, pp. 195–210.
9. Jain, P., Pandey, U. K., 2018. Trending security mechanism in cloud computing. Technology 9 (3), 250–258.
10. Khaleel, M., Zhu, M., 2018. Efficient and fair bandwidth scheduling in cloud environments. ARO-TE SCIENTIFIC JOURNAL OF KOYA UNIVERSITY 6 (2), 20–26.
11. Khaleel, M., Zhu, M. M., Sep. 2015. Energy-aware job management approaches for workflow in cloud. In: 2015 IEEE International Conference on Cluster Computing. pp. 506–507.

12. Khanezaei, N., Hanapi, Z. M., 2014. A framework based on rsa and aes encryption algorithms for cloud computing services. In: Systems, Process and Control (ICSPC), 2014 IEEE Conference on. IEEE, pp. 58–62.

13. Lopez-Falcon, E., Tchernykh, A., Chervyakov, N., Babenko, M., Nepretimova, E., Miranda-López, V., Drozdiv, A. Y., Radchenko, G., Avetisyan, A., 2018. Adaptive encrypted cloud storage model. In: Young Researchers in Electrical and Electronic Engineering (EiConRus), 2018 IEEE Conference of Russian. IEEE, pp. 329–334.

14. Mahajan, P., Sachdeva, A., 2013. A study of encryption algorithms aes, des and rsa for security. Global Journal of Computer Science and Technology.

15. Modi, K. J., Kapadia, N., 2019. Securing healthcare information over cloud using hybrid approach. In: Progress in Advanced Computing and Intelligent Engineering. Springer, pp. 63–74.

16. Moghaddam, F. F., Karimi, O., Alrashdan, M. T., 2013. A comparative study of applying real-time encryption in cloud computing environments. In: Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on. IEEE, pp. 185–189.

17. Moghaddam, F. F., Khanezaei, N., Manavi, S., Eslami, M., Samar, A., 2014. Uaa: user authentication agent for managing user identities in cloud computing environments. In: Control and System Graduate Research Colloquium (ICSGRC), 2014 IEEE 5th. IEEE, pp 208–212.

18. O’Dwyer, K. J., Malone, D., 2014. Bitcoin mining and its energy footprint.

19. Park, J. H., Park, J. H., 2017. Blockchain security in cloud computing: Use cases, challenges, and solutions. Symmetry 9 (8), 164.

20. Roy, D., Das, P., 2018. A modified rsa cryptography algorithm for security enhancement in vehicular ad hoc networks. In: Proceedings of the International Conference on Computing and Communication Systems. Springer, pp. 641–653.

21. Selvanayagam, J., Singh, A., Michael, J., Jeswani, J., Secure file storage on cloud using cryptography. International Research Journal of Engineering and Technology.

22. Singh, S., Maakar, S. K., Kumar, D. S., 2013. A performance analysis of des and rsa cryptography. International Journal of Emerging Trends & Technology in Computer Science (IJEETCS) 2 (3).

23. Sood, S. K., 2012. A combined approach to ensure data security in cloud computing. Journal of Network and Computer Applications 35 (6), 1831–1838.

24. Stergiou, C., Psannis, K. E., Kim, B.-G., Gupta, B., 2018. Secure integration of IoT and cloud computing. Future Generation Computer Systems 78, 964–975.

25. Tosh, D. K., Shetty, S., Liang, X., Kamhoua, C. A., Kwiat, K. A., Njilla, L., 2017. Security implications of blockchain cloud with analysis of block withholding attack. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing. IEEE Press, pp. 458–467.

26. Venkatesh, M., Sumalatha, M., Selvakumar, C., 2012. Improving public auditability, data possession in data storage security for cloud computing. In: Recent Trends In Information Technology (ICRTIT), 2012 International Conference on. IEEE, pp. 463–467.

27. Wang, C., Wang, Q., Ren, K., Lou, W., 2009. Ensuring data storage security in cloud computing. In: Quality of Service, 2009. IWQoS. 17th International Workshop on. Ieee, pp. 1–9.

28. Xu, J., Chang, E.-C., Zhou, J., 2013. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In: Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security. ACM, pp. 195–206.