Available online at http://scik.org

J. Math. Comput. Sci. 11 (2021), No. 5, 5525-5535

https://doi.org/10.28919/jmcs/6072

ISSN: 1927-5307

SECURE AND EFFICIENT RETRIEVAL OF VIDEO FILE USING BLOOM FILTER AND HYBRID ENCRYPTION ALGORITHMS

K. MOHAMED SAYEED KHAN[†,*], S. SHAJUN NISHA, M. MOHAMED SATHIK

PG & Research Department of Computer Science, [5]Sadakathullah Appa College, Tirunelveli, India

Affiliation of Manonmaniam Sundaranar University, Abishekapatti, Tirunelveli 627 012, Tamil Nadu, India

Abstract: Video files are the important source for the big data. Storing, managing and processing the video files are becoming more challenging day by day. Securing and retrieving the video files from the cloud storage has become one of the hot research topic as the vulnerability remains high. Usually, the confidential video files such as CCTV monitoring, OTT videos are transferred to the cloud storage using an insecure communication network. To protect the video file, existing technique uses a symmetric or single key encryption, which uses only one key to perform encryption and decryption. Symmetric key encryption reduces the work load key management and computation time. However, symmetric algorithms are vulnerable to various attacks as they use only single key to perform encryption and decryption. Likewise, existing video file encryption are inefficient to retrieve a particular block of data. In this research work, we create a hybrid model, which integrates the symmetric and asymmetric key encryption to tighten the security. Likewise, to retrieve a particular block of video file from the cloud storage, a bloom filter based index structure is maintained by the cloud service provider. The experimental result shows that the proposed model reduces the computation time and improves the security of the multimedia files stored in the cloud.

Keywords: video encryption; data security; hybrid encryption; data recovery.

2010 AMS Subject Classification: 68P25.

[*]Corresponding author

E-mail address: mrkhan1031@gmail.com

[†]Reg. No: 18211192281016, Research Scholar PhD

Received May 19, 2021

1. INTRODUCTION

With the fast growth of social media, online gaming, OTT platform, IoT devices and CCTV cameras, the multimedia data generation rate has increased exponentially in recent years. According to the 'Big Data Analytics Industry Report 2020' survey, over 2.5 quintillion bytes of data are generated each day. In addition, it also estimates that the amount of digital data produces rate are daily will increase up to 3.5 quintillion bytes in 2025[1]. Securely transferring, storing and managing the multimedia data in the cloud storage remains as a challenging and tedious tasks to the Cloud Service Providers (CSP). Recently, a video messaging app named 'Dubsmash' announced that hackers nabbed nearly 162 million users account holder names, email addresses, passwords and the video contents from their servers. Likewise, in 2017, the HBO OTT contents were hacked and around 1.5terabytes of multimedia files were illegally circulated among the internet [2]. By encrypting the user's confidential multimedia data, the vulnerability towards data alteration and unauthorized access can be stopped in the cloud environment. However existing single key encryption techniques such as AES and DES creates a large amount of cipher data upon encryption [3][4]. As far as cloud is concerned, it allows multiple authorized user to approach the data stored in the cloud though internet. Since, the symmetric keyis shared to multiple authorized users in the cloud environment, the possibility of key leakage is high [5].

On the other hand, it is important also to reduce the computation overhead on the device while performing encryption [6], because nowadays devices with limited resources, such as, mobile phones, connected cameras and IoT devices are used as an end device to capture and transfer the data the cloud storage servers. Therefore, taking into consideration of limited resources, a video encryption algorithms need to be developed. For realworld applications, a video encryption algorithm has to take into account various parameters like security, computational efficiency, compression efficiency and so on. On the other hand, existing algorithm takes too much of time in retrieving the video files. Because, they doesn't use any index based structure to retrieve the multimedia data blocks.

2. RELATED WORKS

This section discuss about the several existing woks. Jagadeeshwari et al have proposed a Dynamic Bloom Filter Hashing based Cloud Data Storage (DBFH-CDS) Technique [7] for improving the security as well as confidentiality of the data storage in a cloud environment. They

have used the data fragmentation model for fragmenting the large cloud datasets. They have also employed Bloom Filter in DBFH-CDS Technique for storing the fragmented sensitive data increased security. Sourin Chakrabarti [8] have proposed an efficient and modified approach for image retrieval using multiple neural hash codes. They have limited the number of queries by identifying the false positives with the use of bloom filter. They have used the local deep convolutional neural network that combines the powers of both the features of lower and higher layers. They create the feature maps which are further compressed with the use of PCA. This is then fed to the bloom filter after doing the binary sequencing with the use of a modified multi k-means approach. The feature maps obtained are additionally utilized in the image retrieval process. They first compares the images in the higher layers for finding the images that are semantically similar and gradually moves towards the lower layers for searching structural similarities. Mai Jiang et al [9] have developed an improved algorithm based on Bloom filter application for bar code processing and recognition. The bit vector of Bloom filter is divided into two parts. Every element is mapped to the part of the bit vector with the use of hash functions. Each element is amplified and is mapped to another part of the bit vector by using the hash function. They states that their algorithm reduces the false positive rate of the Bloom filter also it does not increase the time and space costs. Thilina et al have proposed two techniques [10] [11] that can be applied on Bloom Filter encoding for improving the privacy against attacks. They use neighboring bits in a BF that generates new bit values. They have made an empirical study on large databases and are compared with the proposed techniques and states that the proposed model provides high security against privacy attacks, and achieve better similarity computation accuracy. Raghavendra et al [12] have made a survey and have investigated the various aspects of data sharing in different manner such as user revocation, encryption techniques, identity privacy, competency and key distribution. Different schemes such as Plutus, Sirius, Secure scalable data access schemes are analyzed and found that they have improved proxy encryption. The Multi-owner DataSharing is also discussed based on the above mentioned significant parameters.

3. SYSTEM MODEL

The proposed model intends to create a secure framework for protecting the user’s multimedia data from unauthorized users in the cloud storage servers. In the proposed work,

before encrypting the data using hybrid algorithm, two important operations are performed, such as, data chunking and data hashing. Data chunking helps the data owner to split the multimedia data into several blocks and makes encryption easy. It also facilitate the user to decrypt a particular block of data upon retrieval. Likewise, data hashing is used to produce the message digest for each block and create a hash index.

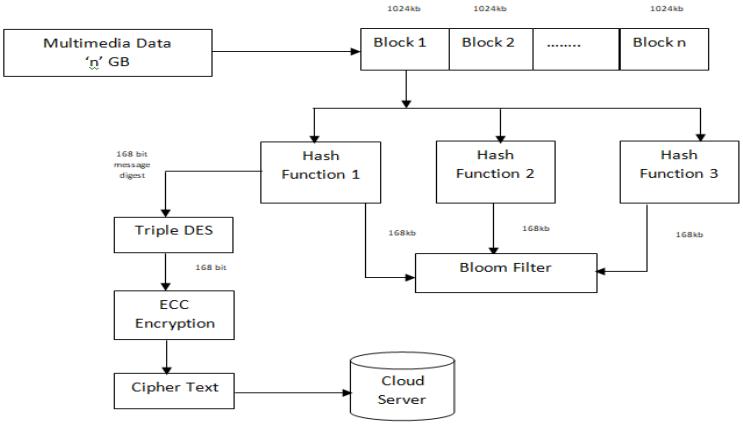

FIGURE 1: Architecture of the proposed work.

3.1 Data Chunking

Data chunking is an important process in data management while outsourcing the user data. The plain text (unencrypted user data) can be of any size ranging from 1KB to 'n' TB. Chunking is a process of breaking down the larger user data into smaller data chunks before encrypting. The proposed scheme uses fixed size data chunking mechanism to create even sized data chunks. The chuck size is fixed to 1024Kb. Algorithm 1 explains the fixed size data chunking method below,

Algorithm 1: Data chunking

Inputs :Multimedia input file f, Required size of a single block.

Output : Data blocks of the input multimedia file 'f'

```
Begin
if (new Multimedia File == yes)
        int offset value = 0;
        List blockList = 0;
        Chunk = getBlock (f, offset, Blocksize)
        While Blocksize> 0 do
                BlockList.add (block);
                Offset value = offset value + Blocksize;
                Block= getBlock (f, offset, Blocksize);
        End while
return chunkList;
End
```

3.2 Data hashing and creation of Bloom filter

To create the hash table in the cloud service provider, Bloom filter is used It is a space-efficient probabilistic data structure used to test whether an element is a member of a set. In the proposed work, after splitting the multimedia data into several small-sized data blocks, ‘n’ number of hash functions are applied onthese multimedia data blocks. Hash functions and their related hash tables are used in data storage and retrieval applications to access data in a small and almost constant time per retrieval. As a result, it produces ‘n’ number of message digest ‘Mi’with the size of 168bits. Later, these message digest ‘Mi’ are stored in the Bloom hash table present in the Cloud Service Provider (CSP). Bloom hash tableeffectively reduces the retrieval time and allows the user toretrieve any random data block from the cloud storage server. The process of hashing the data blocks and storing the corresponding hash bits is explained in Algorithm 2.

Algorithm 2: Hashing and creating the hash tables
Inputs : Data block to be hashed; No. of hash functions. **Output :** Message digest and creation of bloom filter

Begin
S=Size of the bloom filter
N= Number of hashing functions used
if (new data block == yes&& S !=0)
a. Message digest MD = HF (data block)
b. Create corresponding hash bits for each data block
c. Update Bloom filter and change the bits to 1 from 0
else if (new data block == yes && S==0)
a. All locations in BF is occupied
b. Alert the CSP
c. Increase the size of the bloom filter
else
a. Reject new data block from entering the cloud storage
b.Exit
End

3.3 Hybrid encryption

The proposed hybrid algorithm combines the convenience of a public-key cryptosystem with the efficiency of a symmetric-key cryptosystem. To improve the security while transferring the data to the cloud Triple DES algorithm is used [15]. Likewise, to protect the data from internal and external attacks in cloud storage server, a public key cryptosystem called ECC (Elliptic Curve Cryptography) is used to encrypt the data blocks[13].Triple DES is an encryption technique which uses three instance of DES on same plain text. It uses there different types of key choosing technique in first all used keys are different and in second two

keys are same and one is different and in third all keys are same. The working process of Triple DES algorithm is shown in Fig. 2.

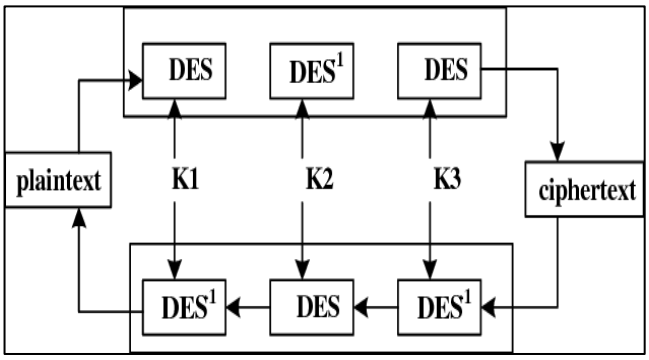

FIGURE 2: Encrypting the data block using Triple DES

Once after the message digest MDi are encrypted using Triple DES algorithm, it will produce a 168 bit cipher text. Later, these cipher text will be considered as a plain text and re-encryption will be performed using ECC algorithm.

Elliptic Curve Cryptography (ECC) is a key-based technique used for encrypting data. ECC focuses on pairs of symmetric and asymmetric keys for decryption and encryption.

3.3 Retrieving a particular data block using Bloom Filter

Reducing the time of the retrieval and allowing the authorized users to directly access a particular block of multimedia file is important. The proposed work uses a Bloom Filter based approach to retrieve the data block in a fast manner. To retrieve the particular data block, user proves the ownership and checks the corresponding hash location in the hash table. If all the bits are set to 1, then the data block is retrieved from the cloud server. Algorithm 3 explains the retrieval of particular data block from the cloud storage server

Algorithm 3: Retrieving the data block from cloud storage
Inputs : User's signature and hash bit locations **Output :**Retrieved data block

Begin
a. CSP verifies the user signature before letting the user to access the data block
b. Checks the corresponding hash bits
c. If (all the hash bit ==1)
Retrieve the data
Else
Return 'No such data'

4. IMPLEMENTATIONS AND RESULT DISCUSSION

The proposed hybrid encryption algorithm is analyzed by executing a set of experiments. The experiment iscarried out in a eucalyptus private cloud which includes cloud controller and walrus as storage controller. Private cloud was installed on a server with the specification of Intel Xeon processor, which has a processing speed of 2.1GHz, 64GB of RAM memory and 4TB of storage space. The test used 500 files of real video data set, uploaded into the storage anddownloaded based on the user's requirement. The experiment result clearly shows that the hybrid encryption algorithm provides secureness for the cloud users to store the data in the cloud. Results demonstrate that the hybrid encryption algorithmis highly complex in nature and the time taken for the encryption and decryption operation is reduced in a higher rate. Performance analysis metrics are done based on the experimental setup.

To analyze the performance of the proposed technique, avalanche effect is measured and compared with other existing algorithms. Avalanche effect is a property in which the change in output with respect to the input can be measured.It will be hard to perform any kind of analysis in the cipher text. Hence avalanche effect has to be measured for observing the level of change[14]. An effective security algorithm is a one which has more avalanche effect. From table.1, the avalanche effect of the ACS cryptosystem is high compare due to the CBC operations.

TABLE 1: Avalanche Effect of the proposed hybrid encryption algorithm

File Size	RSA Asymmetric	DES Symmetric	Proposed Hybrid Encryption Algorithm
1024kb	0.62	0.63	0.70
8072kb	0.43	0.44	0.55
12086kb	0.56	0.53	0.68
16087kb	0.65	0.55	0.67
18726kb	0.57	0.51	0.66
32728kb	0.52	0.57	0.62
45726kb	0.49	0.48	0.58

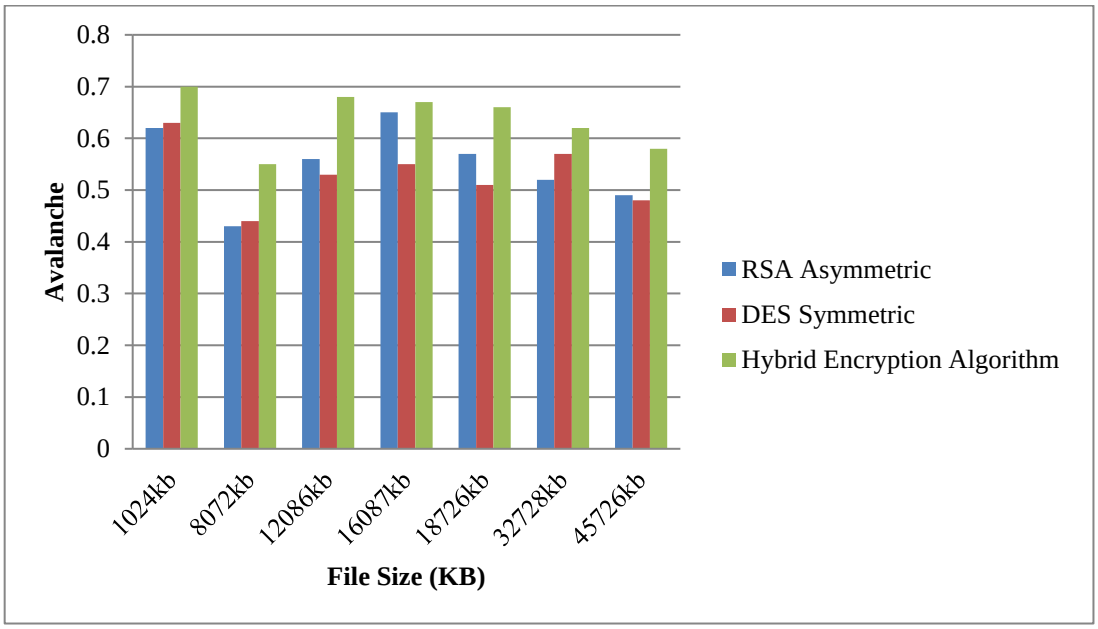

FIGURE.3 Comparison of Avalanche Effect

Table 2, represents the time taken for hybrid encryption algorithm to perform encryption. The derived values are compared with the existing algorithms.

TABLE 2: Time taken to perform hybrid encryption algorithm

File Size	Blowfish	DES	CS	Proposed Technique
1024kb	411	444	341	414
8072kb	606	542	526	535
12086kb	912	539	677	699
16087kb	1045	1063	1024	1011
18726kb	1070	1079	1114	1058
32728kb	1983	2007	1989	1939
45726kb	2534	2761	2646	2273

Figure 4, clearly shows, how far the proposed hybrid encryption algorithm reduces the time of the encryption. As the file size increase, the system performs faster. So the system will give predominant results against the large file, which is feasible to be implemented in cloud environment.

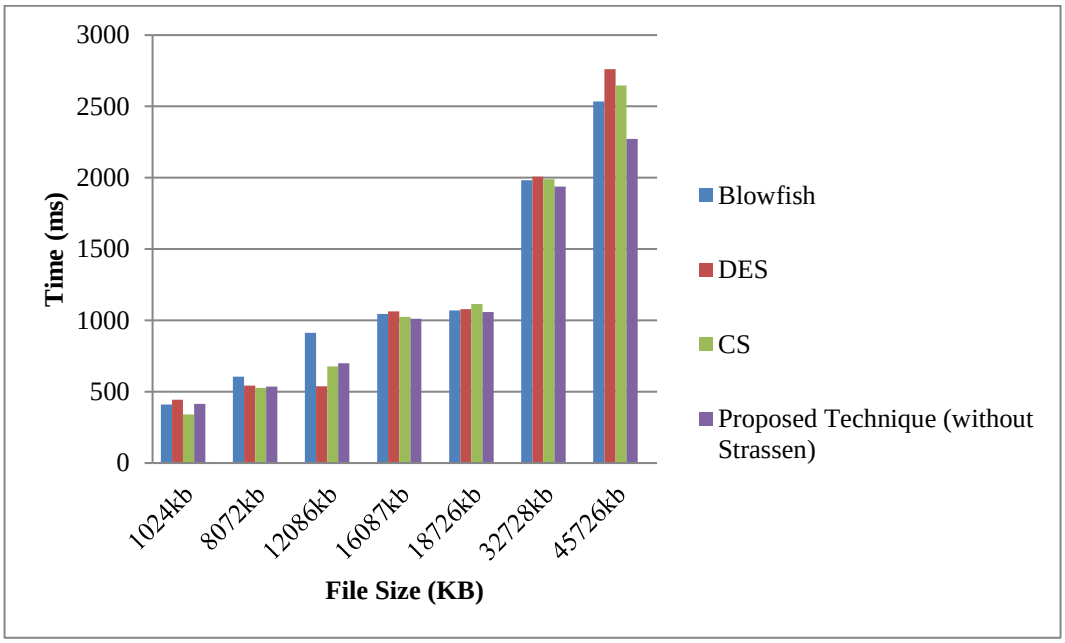

FIGURE.4 Time taken to perform encryption

We can observe that encryption time is reduced to a greater extent. It is also observed that the proposed technique takes too much of time to encrypt without pre-processing module.

TABLE 3: Time taken to perform decryption

File Size	Blowfish	DES	CS	Proposed Technique
1024kb	382	404	353	326
8072kb	513	525	397	332
12086kb	615	543	524	480
16087kb	747	842	695	616
18726kb	798	919	713	695
32728kb	853	969	784	649
45726kb	1148	1258	998	946

From Figure 4 and Figure 5, it is observed that the time take for encrypting as well as decrypting using proposed system is less when compared to other existing public key algorithms.

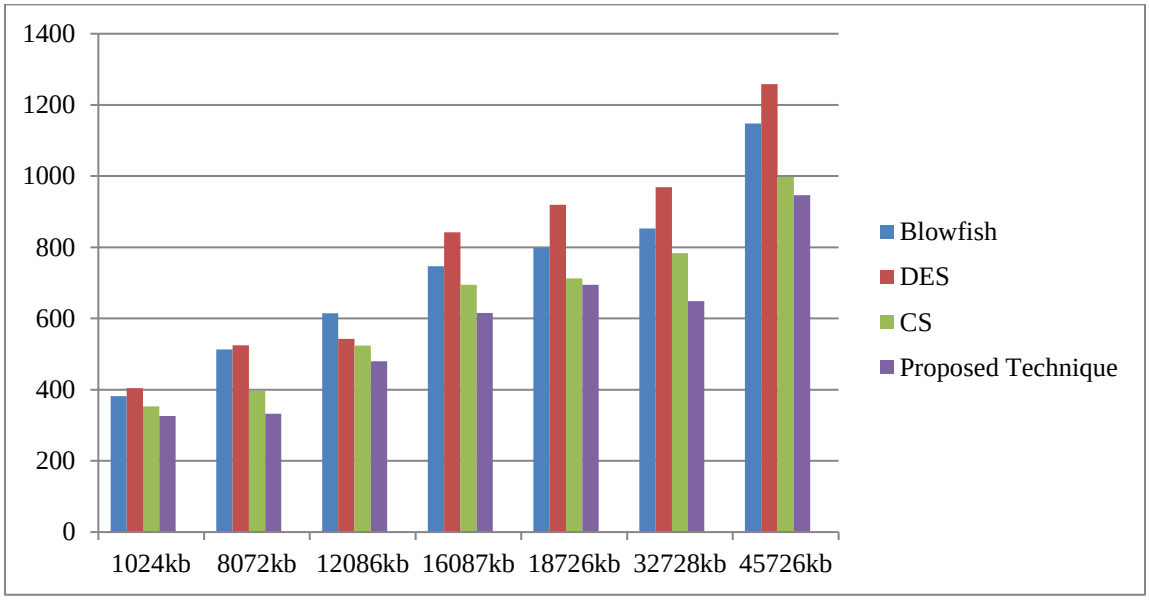

FIGURE.5 Time taken to perform decryption

The proposed pre-processing model reduces the encryption time and makes the system work faster. Even though the time taken for pre-processing the user data is high, it will highly reduce the working time of the encryption module.

5. CONCLUSION

Thus a hybrid model is created that integrates both the asymmetric and symmetric key encryption to tighten the security. A Bloom filter based index structure is created and maintained by the cloud service provider for retrieving the video blocks from the cloud storage. The experimental result shows that the proposed method reduces the computation time and improves the security of the video files stored in the cloud. The proposed pre-processing model reduces the encryption time and makes the system work faster. Though the time taken is high for pre-processing the user data, the working time of the encryption module is reduced much.

CONFLICT OF INTERESTS

The author(s) declare that there is no conflict of interests.

REFERENCES

[1] Global Big Data Analytics Market Size, Market Share, Application Analysis, Regional Outlook, Growth Trends, Key Players, Competitive Strategies and Forecasts, 2019 To 2027, ID: 4992328, January 2020.

[2] https://ciso.economictimes.indiatimes.com/news/sony-netflix-and-now-hbo-hackers-are-threatening-the-way-that-hollywood-does-business/60054215

[3] J. Thakur, N. Kumar, DES, AES and Blowfish: Symmetric Key Cryptography Algorithms Simulation Based Performance Analysis, Int. J. Emerg. Technol. Adv. Eng. 1 (2011), 6-12.

[4] W. Stallings, Cryptography and network security: principles and practice, 5th ed, Prentice Hall, Boston, 2011.

[5] X. Fan, Faster Dual-Key Stealth Address for Blockchain-Based Internet of Things Systems, in: S. Chen, H. Wang, L.-J. Zhang (Eds.), Blockchain – ICBC 2018, Springer International Publishing, Cham, 2018: pp. 127–138.

[6] T. Jiang, X. Chen, J. Li, D.S. Wong, J. Ma, J.K. Liu, Towards secure and reliable cloud storage against data re-outsourcing, Future Gen. Computer Syst. 52 (2015), 86–94.

[7] S. Jegadeeswari, P. Dinadayalan, D. Gnanambigai, Efficient Dynamic Bloom Filter Hashing Fragmentation for Cloud Data Storage, Cybern. Inform. Technol. 19 (2019), 53–72.

[8] S. Chakrabarti, Efficient image retrieval using multi neural hash codes and bloom filters. In 2020 IEEE International Conference for Innovation in Technology (INOCON), pp. 1-6, (2020).

[9] M. Jiang, C. Zhao, Z. Mo, J. Wen, An improved algorithm based on Bloom filter and its application in bar code recognition and processing, J. Image Video Proc. 2018 (2018), 139.

[10] P. Christen, T. Ranbaduge, D. Vatsalan, R. Schnell, Precise and fast cryptanalysis for Bloom filter based privacy-preserving record linkage. IEEE Trans. Knowl. Data Eng. 31 (2018), 2164-2177.

[11] P. Christen, R. Schnell, D. Vatsalan, T. Ranbaduge, Efficient Cryptanalysis of Bloom Filters for Privacy-Preserving Record Linkage, in: J. Kim, K. Shim, L. Cao, J.-G. Lee, X. Lin, Y.-S. Moon (Eds.), Advances in Knowledge Discovery and Data Mining, Springer International Publishing, Cham, 2017: pp. 628–640.

[12] S. Raghavendra, C.S. Reddy, C.M. Geeta, et al. Survey on data storage and retrieval techniques over encrypted cloud data, Int. J. Computer Sci. Inform. Secur. 14 (2016), 718.

[13] V.S. Miller, "Use of Elliptic Curves in Cryptography" in Advances in Cryptology - CRYTO '85, Springer-Verlag, vol. 128, pp. 417-426, (1985).

[14] K.M.S. Khan, Encryption and decryption techniques of text, image and video files in image processing: a survey, Int. J. Eng. Res. Technol. 9 (2020).

[15] A.K. Mandal, C. Parakash, A. Tiwari, Performance evaluation of cryptographic algorithms: DES and AES, in: 2012 IEEE Students' Conference on Electrical, Electronics and Computer Science, IEEE, Bhopal, India, 2012: pp. 1–5.