PRAGUE UNIVERSITY
OF ECONOMICS
AND BUSINESS

Acta Informatica Pragensia
2022, Volume 11, Issue 3, pp. 361–379
Special Issue: Sustainable Solutions for Internet of Things Using
Artificial Intelligence and Blockchain in Future Networks
https://doi.org/10.18267/j.aip.182

Article Open Access

Improving Privacy-preserving Healthcare Data Sharing in a Cloud Environment Using Hybrid Encryption

Insaf Boumezbeur, Karim Zarour

LIRE Laboratory, Software and Information Systems Technologies Department, Faculty of Information and Communication Technology, Constantine 2 University – Abdelhamid Mehri, Nouvelle ville Ali Mendjli BP67A, Constantine, Algeria

Corresponding author: Insaf Boumezbeur (insaf.boumezbeur@univ-constantine2.dz)

Abstract

In recent years, cloud computing has been widely used in various fields and is gaining importance in healthcare systems. Patients' health data are outsourced to cloud storage, enabling healthcare professionals to easily access health information from anywhere and at any time to improve health services. Once patient data are stored in the cloud, they are vulnerable to attacks such as data loss, denial of service (DoS), distributed denial of service (DDoS) and other sorts of cyberattacks. Data confidentiality and patient privacy are more of a problem in the cloud computing context due to their public availability. If a patient's personal information is stolen, he or she may face a range of problems. These are concerns that necessitate more security. The transmission of this sensitive information over the internet is always susceptible to hacking. Therefore, the privacy of patients' data is considered one of healthcare organizations' main issues. To overcome this problem, encryption mechanisms that place a significant emphasis on securing data within the cloud environment are used to preserve sensitive health data. A hybrid cryptography approach is employed in this paper to ensure the secure sharing of health data over the cloud. To maintain data privacy and secrecy, a hybrid cryptography mechanism for storing and transporting data to and from the cloud is used. To protect data from malevolent insiders, the encryption key is separated into two halves, controlling access to patient records via a specific technique. This paper shows the implementation and performance evaluation of the proposal as a functional system prototype. The evaluation is based on the key generation time, the record encryption time, the record decryption time, the record upload time and the record download time for different user numbers and different file sizes varying from 0.1 MB to 500 MB. The findings show that the proposal performs better than other state-of-the-art systems and can practically share secure health data in cloud environments.

Keywords

Cloud computing; Encryption; Healthcare; Privacy; Confidentiality.

Citation: Boumezbeur, I., & Zarour, K. (2022). Improving Privacy-Preserving Healthcare Data Sharing in a Cloud Environment using Hybrid Encryption. *Acta Informatica Pragensia*, 11(3), 361–379. https://doi.org/10.18267/j.aip.182

Special Issue Editors: Venkatachalam Kandasamy, University of Hradec Kralove, Czech Republic
Mohamed Abouhawwash, Mansoura University, Egypt
Nebojsa Bacanin, Singidunum University, Serbia

Academic Editor: Zdenek Smutny, Prague University of Economics and Business, Czech Republic

1 Introduction

In today's digital world, communication technologies offer an efficient and fast way to share data and resources. These technologies have led to strong growth in many areas, including the healthcare field. Due to cloud computing benefits (Low and Chen, 2012; Poulymenopoulou et al., 2012; Kuo, 2011), most healthcare organizations are prompted to move their healthcare service and storage towards the cloud. Recently, the COVID-19 pandemic has strengthened the worldwide healthcare sector of cloud computing technologies, which is estimated to grow by $25.54 billion between 2020 and 2024 (Technavio, 2020). Cloud computing helps manage, share, protect and store electronic health records (EHRs), medical images, pharmacy information systems and laboratory information systems. Besides, patients will benefit from better care through up-to-date health records and ongoing interactions among various healthcare providers (Al-Issa, 2019). Sharing healthcare information could provide practical solutions to improve accessibility to public information, facilitate the healthcare management process for the public and provide ongoing healthcare reports. For example, in cases where the providers and the patients are not in the same area, telemedicine permits surveillance of vital stats and provision of healthcare services, utilizing information and communication technology.

Despite all these benefits, some barriers still need to be managed (Anderson et al., 2007; Svantesson and Clarke, 2010). Sharing electronic health records from an organization's environment in the cloud always involves several privacy risks (Boumezbeur and Zarour, 2022a; Rajakumar et al., 2010), knowing that health data privacy is a significant problem that needs particular consideration. Numerous regulations and standards such as openEHR, HL7 CDA, HITECH Act, HIPAA and EHRcom have proposed guidelines and frameworks to share and exchange health information between various entities across healthcare communities. HIPAA (ACT, 1996) is one of the regulations that control access to health data in the professional medical field. The ISO 18308 standard (ISO, 2011) defines the security and privacy challenges for EHRs. In Australia, a personally controlled EHR (PCEHR) (Andrews et al., 2014) has adopted solutions to develop an IT infrastructure for sharing health information. The US Department of Health and Human Services (HHS) (HHS, 2006) has published a report concerning personal health records (PHRs). The Integrating Healthcare Enterprise (IHE) developed the so-called Cross-Enterprise Document Sharing (XDS) (Noumeir, 2010) to address the needs for distribution, registration and access to patient clinical information across healthcare organizations.

Though there are various concerns in the medical care area, such as accessibility and privacy, the creation of technologies to treat health problems has progressed quickly thus far. To protect the integrity of health data and patients' privacy, electronic health data must be shared securely through networks. An encryption system is often needed as a specific device to provide data confidentiality and privacy services and prevent electronic health data disclosure in communications. Generally speaking, cryptography techniques encrypt data before being stored in the cloud so that even a cloud service provider (CSP) cannot access them. In cryptography, the key is vital and is one of the most challenging tasks to manage. It is legally responsible for any loss that could occur during the encryption and decryption processes. It is a fact that the key used to encrypt and decrypt needs itself to be protected. The issue emerges when the key is to be stored. Diverse works (Boumezbeur and Zarour, 2022b; Pugazhenthi and Chitra, 2019; Rezaeibagha, 2019; Singh, 2018; Suresh and Florance, 2019; Zhang et al., 2016; Chen et al., 2012) have applied several standard cryptographic techniques in various applications to store data in an encrypted form. Despite all these regulations, standards and works, the problem of securing shared electronic health data persists.

This paper proposes a healthcare record system that combines cryptographic techniques to preserve sensitive and personal health information privacy and confidentiality. The system utilizes two encryption algorithms to guarantee a better level of confidentiality to healthcare data. This work uses the Advanced Encryption Standard (AES) to encrypt healthcare records with a single key and Rivest-Shamir-Adleman

(RSA) to secure the AES key. Three entities are used in the suggested methodology. The data owner sends the data, the user list and the parameters needed to generate an access control list to a trusted third-party cryptographic server that handles key management, encryption, decryption and access control. The third-party generates the symmetric key and uses it to encrypt the data. The third party then generates asymmetric keys for each user and divides the private key into two pieces, preventing a single part from regenerating the key. One part of the key is sent to the associated user, while the other is kept by a third party as part of the data file access control. The data owner submits a parameter, which generates the access control. After that, the encrypted health data are uploaded to the cloud for storage. A download request is sent to the system by the user who wishes to obtain the data. After verifying the requesting user, the third party receives the user's portion of the private key and then downloads the data file from the cloud. The key is regenerated by performing the exclusive OR on the user component of the key. The information is decrypted and returned to the user.

The contribution of the present paper can be summarized mainly as:

- A hybrid encryption method to store/transmit data to and from the cloud to guarantee data confidentiality and privacy.
- The encryption key is divided into two segments to secure data against malicious insiders.
- The access to medical records is limited to authorized users who can only perform the necessary verification and ensure data integrity.

We compared our method to other relevant works to enhance the varied times of encryption, decryption, key generation, uploading and downloading.

The remainder of this paper is organized as follows. An overview of the previous works in the domain is given in Section 2. Section 3 presents the details of the proposed system and its architecture. Section 4 demonstrates the performance of the proposed architecture. The discussion of the proposed system is presented in Section 5. Finally, Section 6 concludes the paper.

2 Literature Review

Bentajet et al. (2019) proposed a design and implemented a fully featured prototype to accomplish guaranteed deletion in cloud storage based on identity-based cryptography (IBE) with lightweight key management. Moreover, the suggested scheme permits key delegation and revocation.

An efficient multi-level data encryption scheme is presented by Jana et al. (2017). The proposed approach enables double authentication in cloud security using the ECC algorithm. It is a strong public-key cryptosystem with reduced computational complexity, encrypts the key, and uses the AES algorithm which is a fast symmetric algorithm.

Secure Data Sharing in Clouds (SeDaSC) is a security scheme for cloud storage proposed for group data (Ali et al., 2015). SeDaSC relies on symmetrical encryption to access control malicious attacks, provide data confidentiality, share secure data without re-encryption, and provide access control. The encryption and decryption processes are conducted by the cryptographic server, which is a trusted third party in the SeDaSC.

Hama and Kesavan (2019) provided a security framework for the secure sharing of medical data from health centres via the health cloud system. For safe sharing, the proposed health cloud architecture uses an ECC-based cryptographic scheme. Furthermore, the TP-CS level handles the encryption and decryption processes. The key generation time, file encryption time, file decryption time, file upload time, file download time, uploading speed and security overhead were all considered when evaluating the model.

The AC-AC protocol, introduced by Oliveira et al. (2021), is a dynamic revocable access control system that allows acute care teams to access patients' electronic medical records (EMR) during an emergency session. AC-AC is a hybrid encryption system that combines dynamic index-based symmetric searchable encryption with ciphertext-policy attribute-based encryption. The proposed protocol extends the MicroSCOPE protocol (Michalas et al., 2019) allowing access privileges to be granted and revoked to healthcare practitioners who are part of an acute care team utilizing scope values. According to the acute care timeline, AC-AC algorithms allow the treating team to add a new team to the emergency session. A team can also withdraw the access rights of another team that has previously finished its task using AC-AC.

3 System Model

This work proposes a scheme (HDaSC) for healthcare data sharing in a cloud environment. The HDaSC is meant to secure privacy, confidentiality and integrity of sharing health data through the cloud environment. The proposed scheme is based on a hybrid encryption scheme that utilizes the advantages of both AES and RSA. The HDaSC scheme is detailed in the following subsections.

3.1 System architecture

The proposed HDaSC provides confidentiality, privacy and integrity to ensure the secure sharing of health data via the cloud environment. The architecture of the proposed system consists of three entities, as shown in Figure 1. Each entity has its specific functionalities. Table 1 lists the notations used along with the paper.

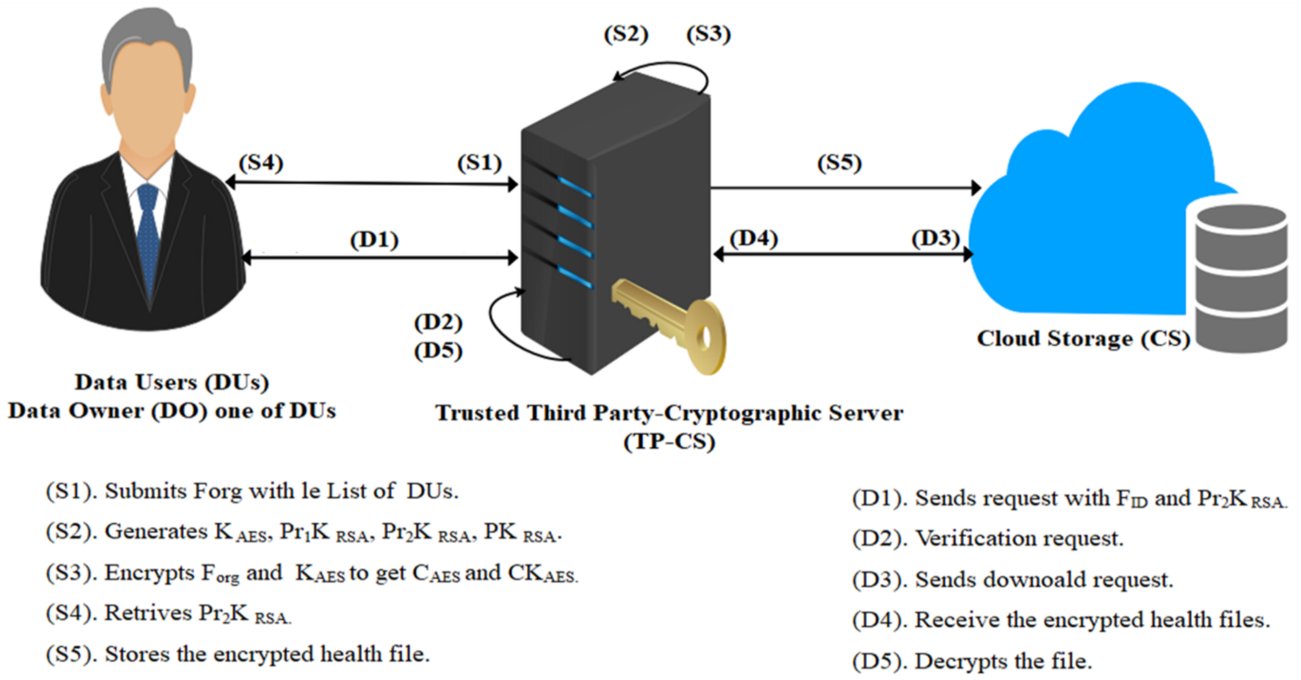
A full appraisal of each entity is given below.

- *Cloud storage (CS):* The cloud storage module provides storage facilities concerning data that must be protected against privacy violations. The system includes only basic operations for uploading and downloading healthcare records.
- *Trusted third party-cryptographic server (TP-CS):* TP-CS is a trusted third party among system users and cloud storage that functions under specific government authority. In the HDaSC system, the TP-CS is responsible for security operations, including integrity, data confidentiality, key, access control (ACL) management, digital signature, encryption and decryption keys to ensure the security and sharing of sensitive health data.
- *Data users (DUs):* They form an entity that requests access to the healthcare record data with permission from the corresponding data owner (DO), which is one of the DUs who is the only one responsible for creating, managing and granting access controls to his/her healthcare record (F_{org}). To realize security services, the DUs must be registered with the TP-CS.

Table 1. *Notations and definitions.*

Notations	**Description**
$\mathbf{F_{org}}$	Original record
$\mathbf{PrK_{RSA}}$	RSA Private Key
$\mathbf{Pr1K_{RSA}}$	RSA Private Key part 1
$\mathbf{Pr2K_{RSA}}$	RSA Private Key part 2
$\mathbf{PK_{RSA}}$	RSA Public Key
$\mathbf{K_{AES}}$	AES Key
$\mathbf{F_{ID}}$	Record ID
$\mathbf{C_{AES}}$	Cipher-text AES

Notations	Description
CKAES	Cipher Key AES
UL	User list



(S1). Submits Forg with le List of DUs.

(S2). Generates K_{AES}, Pr_1K $_{RSA}$, Pr_2K $_{RSA}$, PK $_{RSA}$.

(S3). Encrypts F_{org} and K_{AES} to get C_{AES} and CK_{AES}.

(S4). Retrives Pr_2K $_{RSA}$.

(S5). Stores the encrypted health file.

(D1). Sends request with F_{ID} and Pr_2K $_{RSA}$.

(D2). Verification request.

(D3). Sends downoald request.

(D4). Receive the encrypted health files.

(D5). Decrypts the file.

Figure 1. *Architecture of proposed HDaSC.*

3.2 System workflow

The workflow of the HDaSC system is shown in Figure 2. This model upholds symmetric and asymmetric encryption to secure each health data record sent to and from the cloud. The motivation to combine the strengths of AES and RSA encryption is to achieve the security of RSA with the performance of AES. Based on their results, Bibtra and Babu (2016) concluded that the AES algorithm is much better than the RSA algorithm. However, since the AES encryption poses a major problem in that as a symmetric algorithm, it requires the encryptor and decryptor to use the same key. This poses a crucial key management issue. The data are encrypted and decrypted with the AES, a fast algorithm, and only the key itself utilizes the slower RSA algorithm.

Initially, the healthcare record owner must register to the TP-CS by creating an account with login information. The DO transmits the health record, the list of DUs and their permissions to the TP-CS. Then, the TP-CS generates the symmetric key (KAES) for each health data record using a hash function (SHA-256) of a random number RN. KAES is the hash function output used in the Advanced Encryption Standard (AES) to secure the healthcare record to get ciphertext (CAES). For each user in the system, the TP-CS generates the public key (PKRSA) and the private key (PrKRSA) through hashing a random number (RN) utilizing the 256-bit SHA-1 hash function. The PKRSA and PrKRSA are used in the asymmetric key encryption Rivest-Shamir-Adleman (RSA) to secure the AES key and generate the ciphertext key (CKAES).

The generated PKRSA can be freely distributed and used in the encryption process. The PrKRSA is sub-divided into two parts to be used for the decryption process. It is stored in different locations to avoid reconstituting even if one part is intercepted and analysed. The system generates two random strings and gives one to each party, and then it decrypts the data with the exclusive OR of the two random strings. The section below explains the healthcare record storing, downloading and updating operations.

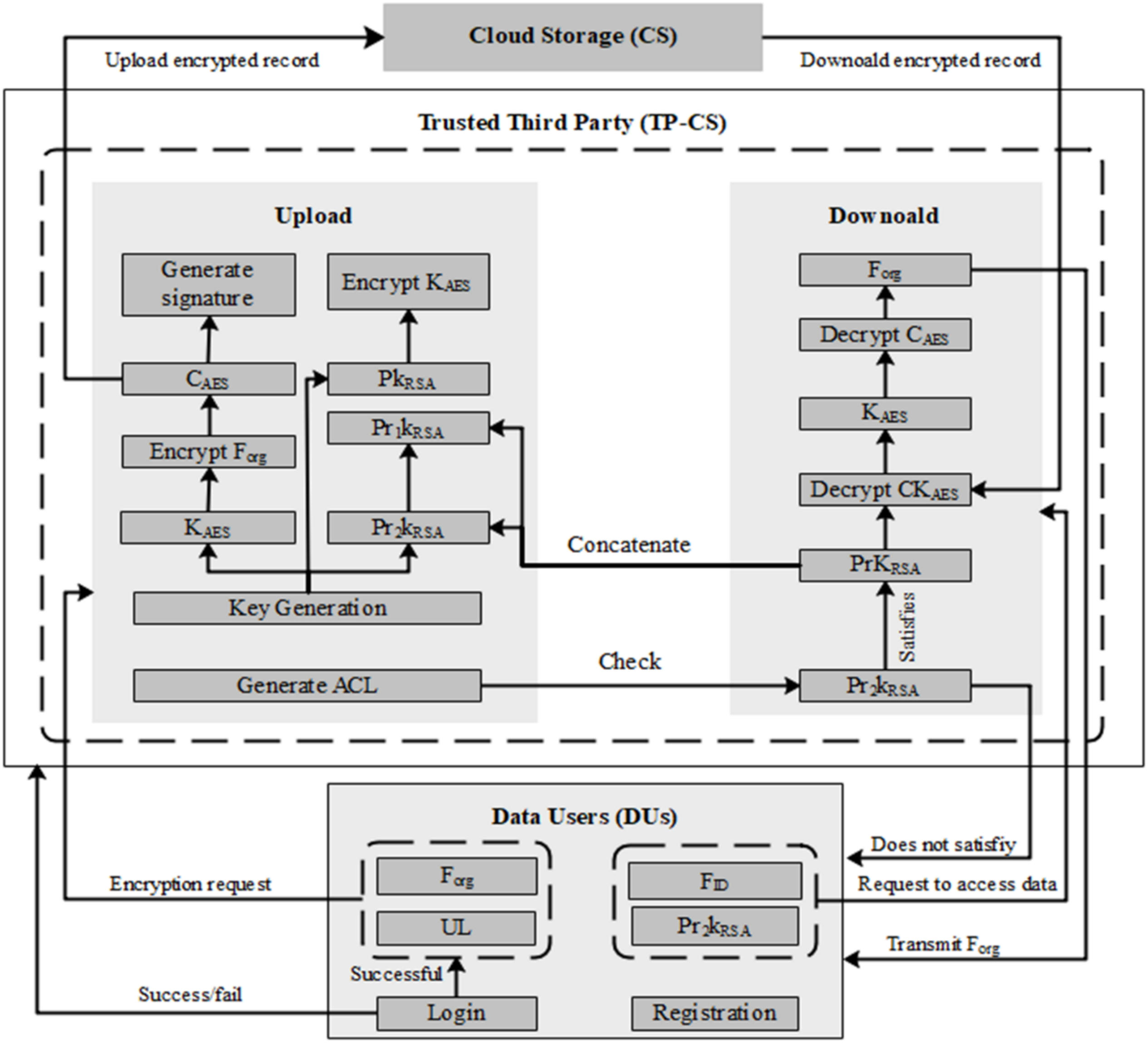


Figure 2. *Workflow of proposed HDaSC.*

3.2.1 Healthcare records stored in the cloud

When the data owner wants to upload his/her healthcare data to the CS, he/she transmits the encryption demand to the TP-CS. The request is attached with the health data record (F_{org}) and the users' list (UL) containing the users' access rights. Access privileges are provided and revoked depending on the owner's decisions.

Users can own read-only or both read-and-write access permissions to the medical record. The TP-CS uses the UL to generate the ACL containing the health record information. Then, the TP-CS generates the K_{AES} for each healthcare record and a PK_{RSA} and PrK_{RSA} for every CU. The TP-CS subdivides the PrK_{RSA} into two parts ($Pr1K_{RSA}$ and $Pr2K_{RSA}$); one part is sent to the user and the other is kept with TP-CS. The remaining DU gets only the $Pr2K_{RSA}$ via the secure socket layer (SSL). Then, the F_{org} is encrypted using the AES encryption technique to obtain an encrypted record (C_{AES}) uploaded directly to the cloud. After that, the PK_{RSA} is used to secure the K_{AES} to form the ciphertext key (CK_{AES}). Ultimately, the $Pr2K_{RSA}$ and K_{AES} are separated from the TP-CS after the secure overwrite encryption process. The TP-CS also computes the digital signature and its key on every encrypted record to protect the health record integrity. Algorithms 1 and 2 present the process of the proposed key generation and encryption technique. Figures 3 and 4 present the healthcare record encryption process.

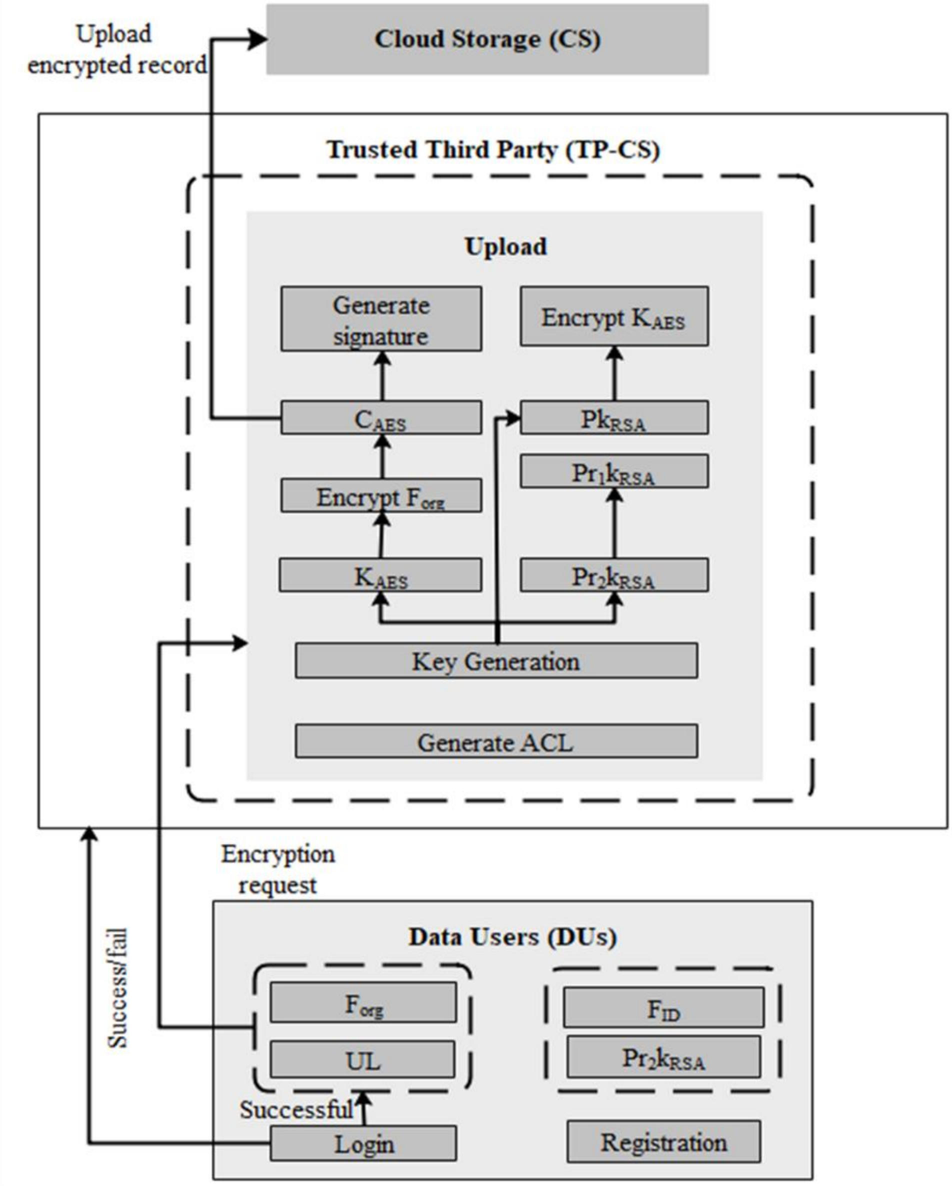


Figure 3. *Healthcare record uploading process.*

Algorithm 1: Key Generation

```
Input: PWD, RSA.
USE PBKDF2 to generate EncPwd;
Use RSA to generate $PK_{RSA}$ and $PrK_{RSA}$;
Divide $PrK_{RSA}$ to $Pr1K_{RSA}$ and $Pr2K_{RSA}$ using XOR;
Keep $Pr1K_{RSA}$, $PK_{RSA}$, and EncPwd for DU in TP-CS;
Transmit $Pr2K_{RSA}$ to DU;
Output: EncPwd, $PK_{RSA}$, $Pr1K_{RSA}$, $Pr2K_{RSA}$.
```

Algorithm 2: Encryption

```
Input: $F_{org}$, RSA, AES, and ACL.
For each health record do
    Use AES to generate $k_{AES}$;
    $C_{AES}$= ($F_{org}$, $k_{AES}$);
    Calculate Digital signature;
        For each DU in the ACL do
        $CK_{AES}$ = RSA ($K_{AES}$, $PK_{RSA}$);
        End for
    Upload the $C_{AES}$ to the cloud;
End for
Output: $C_{AES}$, $CK_{AES}$, $PK_{RSA}$, $Pr1K_{RSA}$, $Pr2K_{RSA}$.
```

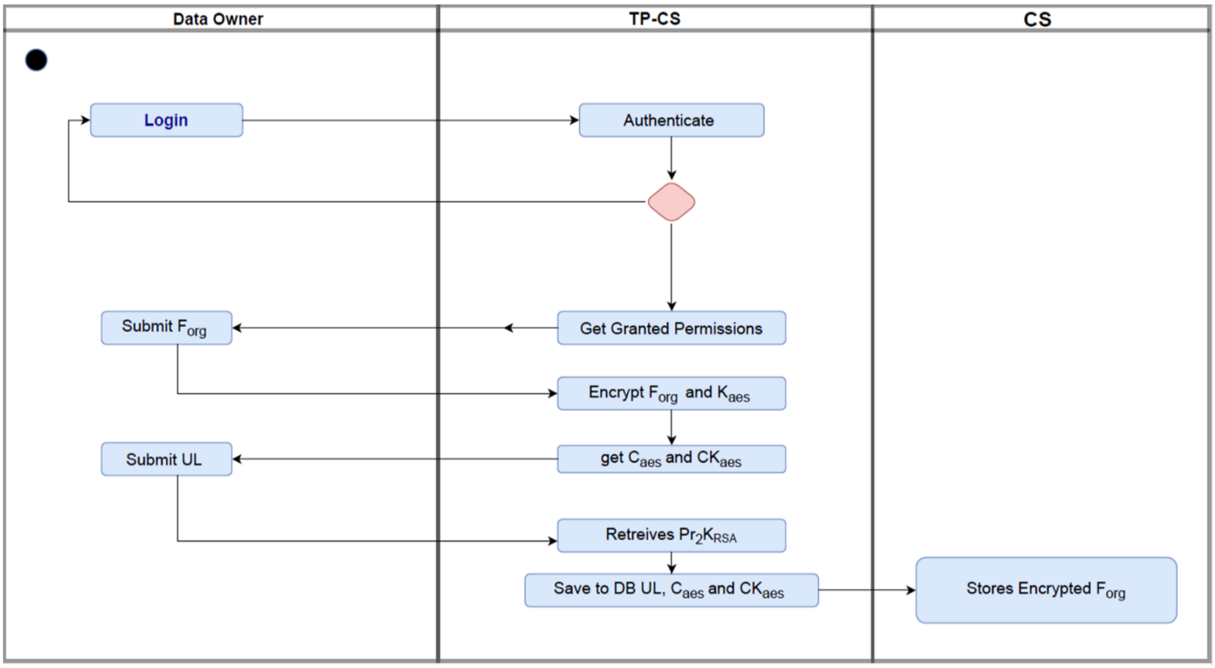

Data Owner
TP-CS
CS
Login
Authenticate
Submit F_{org}
Get Granted Permissions
Encrypt F_{org} and K_{aes}
Submit UL
get C_{aes} and CK_{aes}
Retreives Pr_2K_{RSA}
Save to DB UL, C_{aes} and CK_{aes}
Stores Encrypted F_{org}

Figure 4. *Block diagram of encryption process.*

3.2.2 Healthcare record download from the cloud

To download a healthcare record from the cloud storage, the user has to transmit a request with the necessary authentication information (F_{ID} and $Pr2K_{RSA}$) to the TP-CS. First, the TP-CS checks the user permission details associated with the F_{ID}. If the list includes the user information and access authorizations, the TP-CS recreates the private key by merging the parts, the one received from the user and the second one at its disposal. At that time, the TP-CS downloads the record in the encrypted form (C_{AES}) from cloud storage. It uses the PrK_{RSA} to decrypt the CK_{AES} and get the session key K_{AES}. The C_{AES} is then decrypted using the AES key (K_{AES}) to generate the original record F_{org}. Finally, the F_{org} is transmitted to the corresponding DU through an SSL channel after a successful AES decryption operation, and PrK_{RSA} and K_{AES} are detached from the TP-CS. Algorithm 3 overviews the downloading process. Figures 5 and 6 present the healthcare record decryption process.

Algorithm 3: Decryption algorithm

```
Input: $C_{AES}$, ACL, AES, and RSA.
Obtain $Pr2K_{RSA}$ from the requesting DU;
Obtain $Ck_{AES}$ from TP-CS;
Obtain $C_{AES}$ from the CS;
If $Pr2K_{RSA}$ does not exist in the ACL then
    'Resend denied access message to DU';
Else
    $PrK_{RSA}$ = concatenate ($Pr1K_{RSA}$, $Pr2K_{RSA}$);
    $K_{AES}$ = RSA ($CK_{AES}$, $PrK_{RSA}$);
    $F_{org}$ = AES ($C_{AES}$, $K_{AES}$);
    Transmit $F_{org}$ to the user;
Endif;
Remove $PrK_{RSA}$ and $K_{AES}$;
Output: $F_{org}$.
```

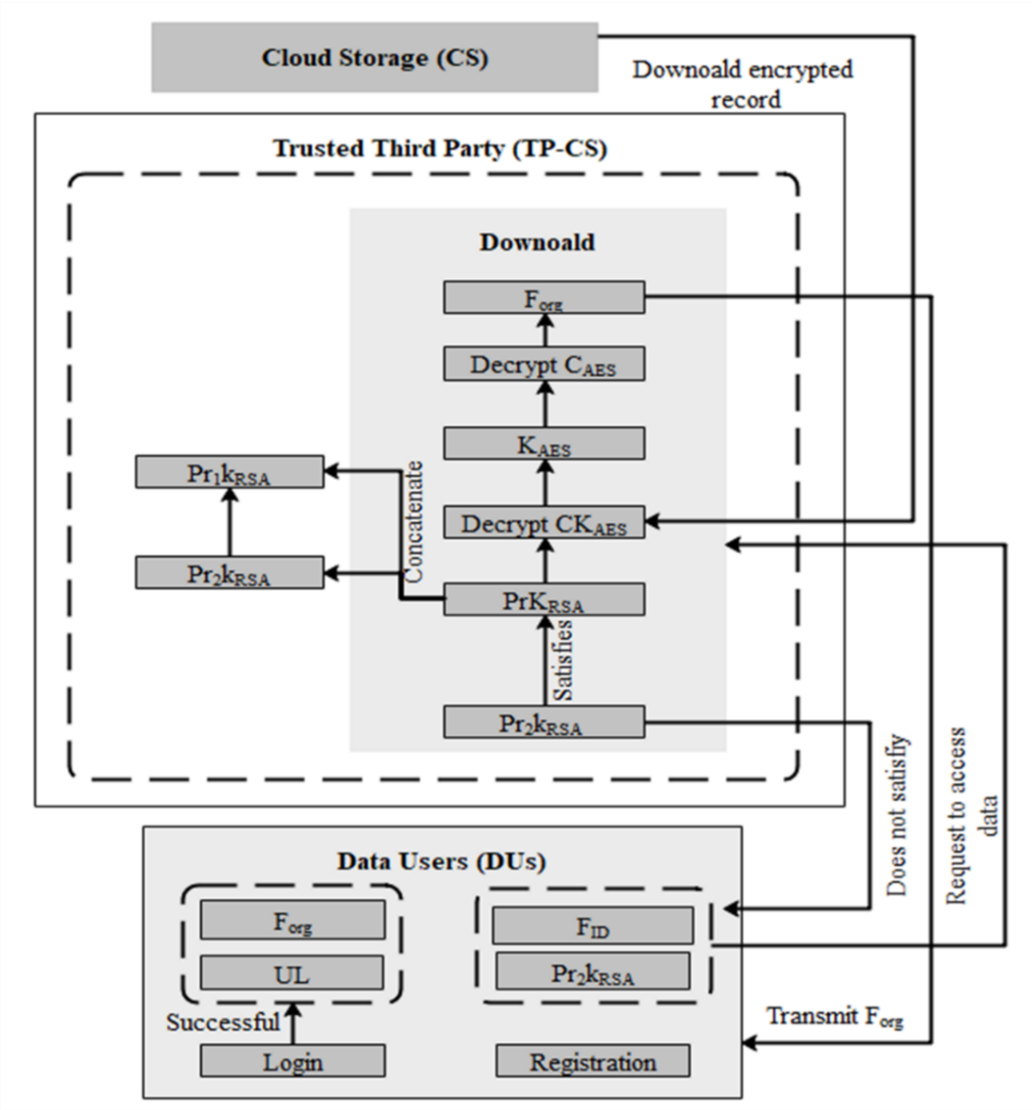


Figure 5. *Healthcare record downloading process.*

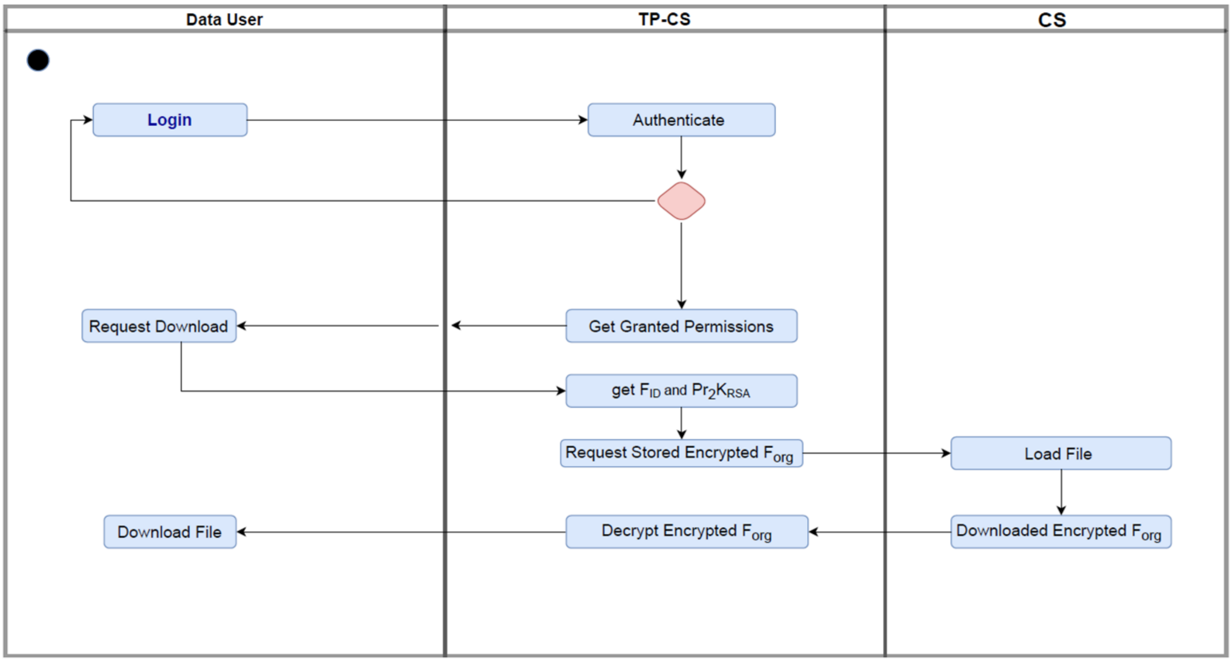

Data User
TP-CS
CS
Login
Authenticate
Request Download
Get Granted Permissions
get F_{ID} and Pr_2K_{RSA}
Request Stored Encrypted F_{org}
Load File
Download File
Decrypt Encrypted F_{org}
Downloaded Encrypted F_{org}

Figure 6. *Block diagram of decryption process*

3.2.3 Healthcare record updating

Updating the health record follows the same procedure as record storage. The significant dissimilarity is that, while updating, all the key generation activities are not performed again. Once the user has downloaded the healthcare record and made a change, he/she sends a restore request to the TP-CS. The demand consists of the F_{ID} and $Pr2K_{RSA}$, as well as the record to be encrypted after modifications. The TP-CS checks whether the user possesses write access to the record. If it is an adequate update request, the TP-CS encrypts the updated record and calculates the digital signature. Finally, the encrypted health record is uploaded to the cloud, and the $Pr2K_{RSA}$ and K_{AES} are ignored. Otherwise, the DU receives an access denial message. Figure 7 illustrates the process of updating a healthcare record.

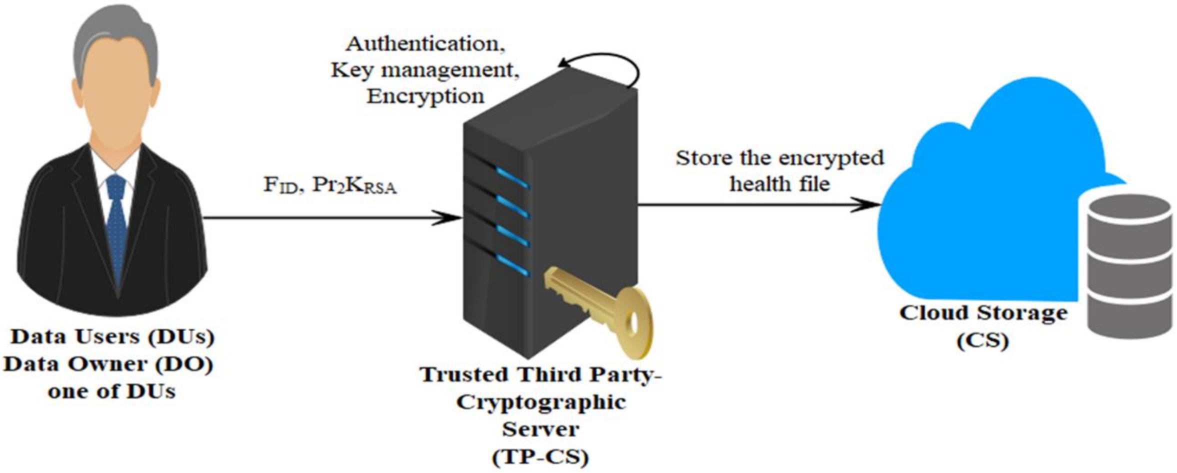


Figure 7. *Healthcare record updating process.*

4 Performance Evaluation

This section covers the experimental setup and findings of the HDaSC system.

4.1 Experimental setup

The proposed system is implemented on the Windows 10 system on an AMD Ryzen 3 2300 CPU @ 2 GHz and 4.00 GB RAM using Java. The application programming interfaces communicate with Google Cloud Platform Storage (GCP), serving as the cloud storage server in the implementation. As already indicated in Section 3, the system comprises three primary entities: the CS, the TP-CS and the DUs. In addition, SSL secures communication. Experiments are conducted using public datasets downloaded from HealthIt.gov, provided by the Office of the National Coordinator for Health IT (ONC) (HealthData.gov, 2021). Medical data, various measures, financial data, statistical data, demographics of certain groups and insurance data are all included in healthcare data sets.

Electronic health records, administrative data, claims data, patient/disease registries, health surveys and other datasets are examples. Patient data in our case include information about a person's past and current health or sickness, treatment history, lifestyle choices and genetic information.

The experimental study has two objectives. Firstly, it aims to analyse the performance of the proposed scheme for different user numbers and different file sizes. Secondly, it aims to compare the proposed scheme to the best state-of-the-art algorithms. The evaluation is based on the following parameters: the key generation time, the record encryption time, the record decryption time, the record upload time and the record download time. Figure 8 shows the implementation architecture of the HDaSC application.

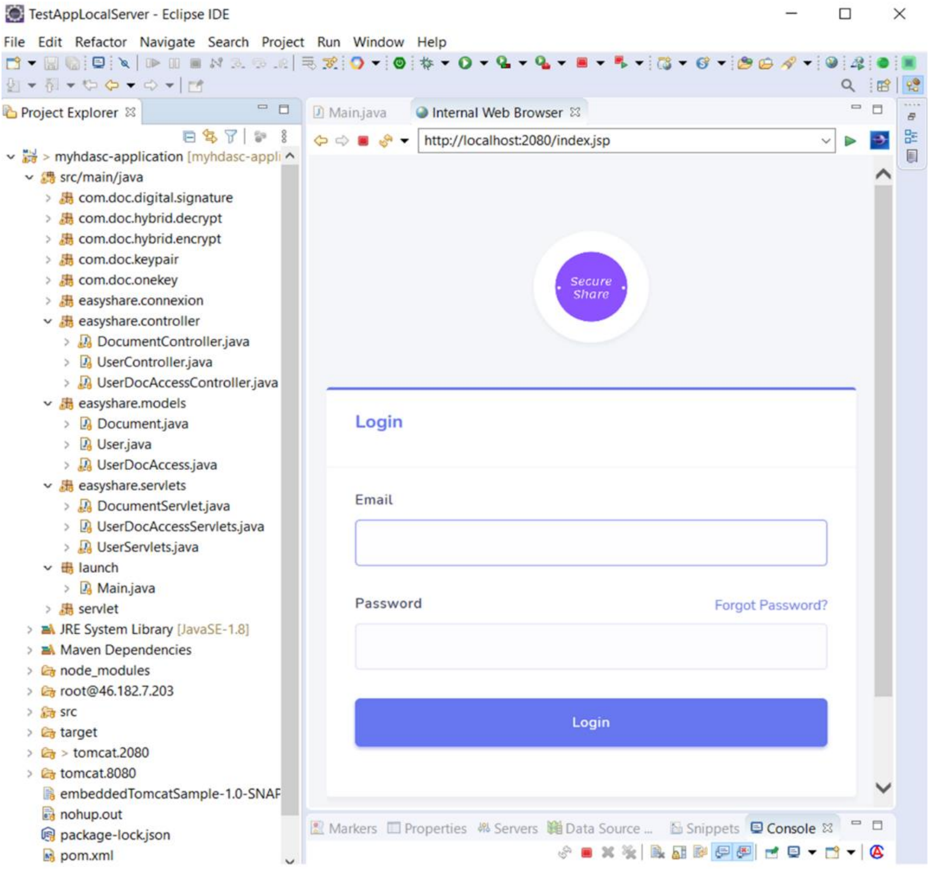


Figure 8. Implementation architecture of HDaSC application.

4.2 Experimental results

The HDaSC system was assessed for the following three parameters.

4.2.1 Key generation time

The key generation time is an important process when it comes to encryption and decryption. As described in Section 3, a symmetric key is generated for each record, and an asymmetric key is generated for each user. In the proposed system, the key generation time is calculated for diverse numbers of users. This number varies from 10 to 100. The graph in Figure 9 shows that the asymmetric key generation time increases as DUs increase, as expected.

As can be observed, the analysis of time consumption for the key generation discloses that the increase in time consumption is not uniformly proportionate to user number growth. For example, it takes 0.0006 s for key generation for 10 users and increases to 0.00066 s for 50 users. Consequently, the time does not increase at a rate similar to the number of users; the time needed for key generation varies only by 0.000138 s when the number of users increases from 10 to 100. As a result, the key generation time varies from 0.000642 s to 0.00078 s; this could be explained by the independence of the users from each other, which makes possible the parallel generation of their encryption keys.

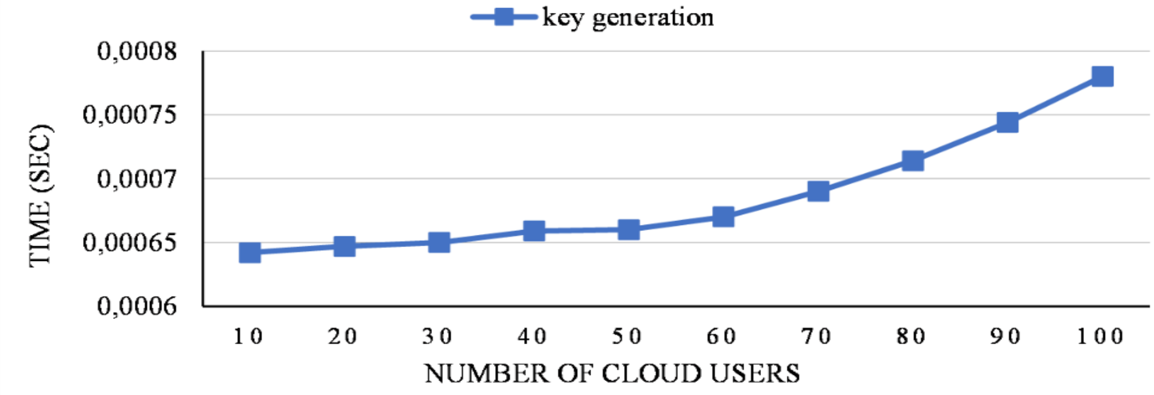


Figure 9. *Key generation time.*

Table 2 provides a detailed comparison of the key generation time using different competitive methods (Bentajar et al., 2019; Jana et al., 2017; Ali et al., 2015; Hema and Kesavan, 2019). According to the observed results summarized in Table 2 and Figure 9, it is obvious that HDaSC is more efficient than all other methods under comparison; it is at least three times faster than the best one among them.

Table 2*. Comparison of key generation times.*

No. of users	Ali et al. (2015)	Hema and Kesavan (2019)	HDaSC
10	0.004	0.00212	0.000642
20	0.00425	0.00235	0.000647
30	0.00476	0.00286	0.00065
40	0.005	0.00302	0.000659
50	0.00512	0.00328	0.00066
60	0.0055	0.0035	0.00067
70	0.00598	0.00398	0.00069
80	0.00632	0.00427	0.000714
90	0.00664	0.00463	0.000744
100	0.00697	0.00499	0.00078

4.2.2 Health record encryption and decryption time

In HDaSC, the encryption time is set as the TP-CS time for encrypting healthcare records according to the DU's encryption request. In HDaSC, different healthcare file sizes ranging from 0.1 to 500 megabytes (MB) are used to estimate encryption and decryption times, in contrast to Bentajar et al. (2019) and Jana et al. (2017), who use limited file sizes outside the healthcare domain.

The graph in Figure 10 shows that the time for the key computation stays stable with a minor variation, possibly because of the treatment conditions at that time. It should also be mentioned that the total time to compute the encryption key varies between 0.0037 s and 0.0029 s, while the total time to compute the decryption key varies between 0.0046 s and 0.0029 s. This is because the health record size is independent of the main calculation time.

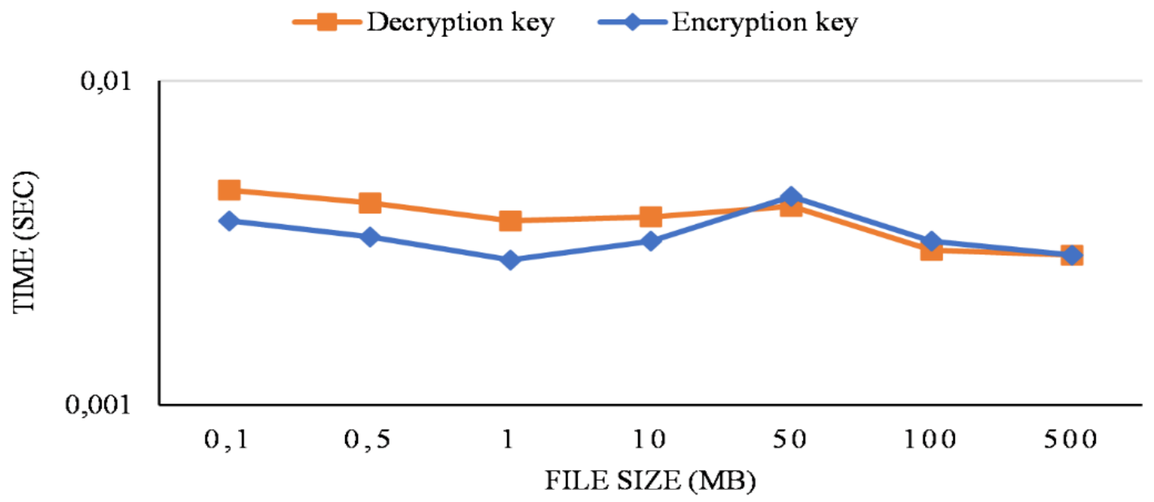


Figure 10. *Performance of key encryption time for proposed HDaSC.*

The time necessary to encrypt a health record based on the owner's encryption request is referred to as record encryption time, while the time necessary to decrypt a health record based on the user's decryption request is referred to as record decryption time.

The plot for encryption and decryption times in Figure 10 exhibits a steady increase in time consumption as the record size increases. This means that the time it takes to encrypt and decrypt a file is proportional to its size.

Figure 11 illustrates the same trend for decryption as it did for the encryption process; the decryption time has a sharp rise proportional to the changing record size. The time consumption increases from 0.009 to 0.3 seconds for encryption, while it varies from 0.0092 to 0.79 seconds for the decryption process.

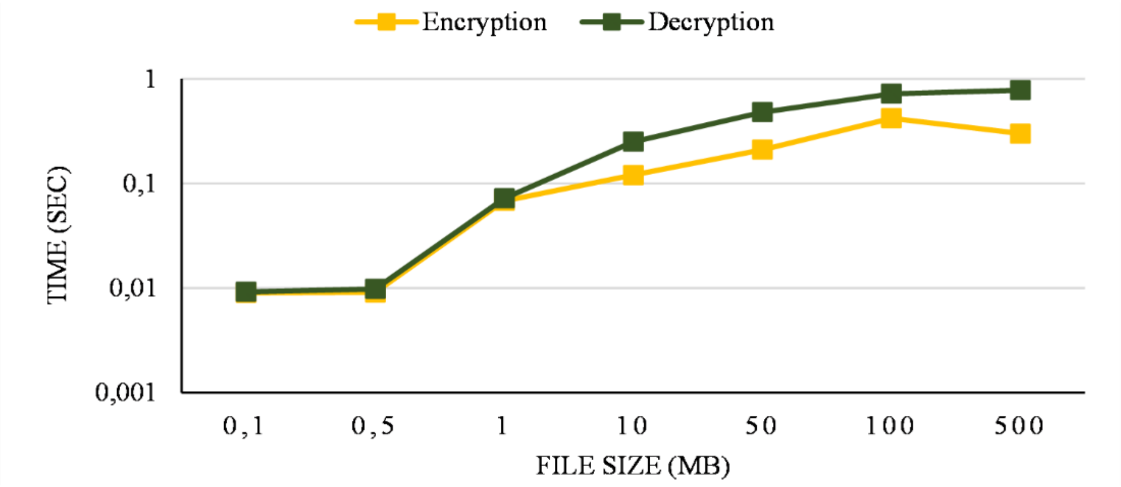


Figure 11. *Performance of encryption and decryption time for proposed HDaSC.*

The time difference between key encryption/decryption and file encryption/decryption is shown in Figure 12, which combines the two preceding plots.

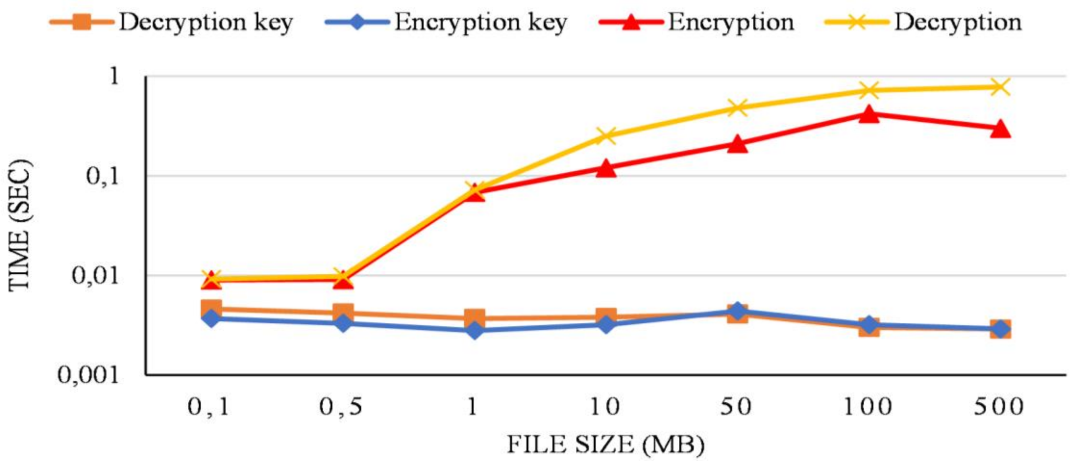


Figure 12. *Performance of record encryption and decryption for proposed HDaSC.*

4.2.3 Health record upload and download time

The system evaluation is based on the overall time required to upload and download a health record to/from the CS. The time needed to transfer the healthcare record from the TP-CS to the cloud environment is the record upload time. Figure 13 shows the performance of health record upload to the CS for HDaSC with different record sizes. Separate bar charts represent all the constituent times. The time taken for uploading a health record to the CS increases proportionally to the increase in the health record size. Concerning the calculation time of the key, we can notice that it remains stable by specifying that it is independent of the size of the file.

Nonetheless, the marginal increase was small in the record upload time at times, which could be due to the network state at different times. The encryption time represents 0.67% of the total encryption time in the case of 0.1 MB. In comparison, the percentage rose to 8.89% with the 10 MB record size, while the 50 MB record size requires 15.56% of the overall encryption time; but when the trend continues with a larger record size of 500 MB, the percentage is 37.78%. In this case, the record encryption time increases as the record size increases.

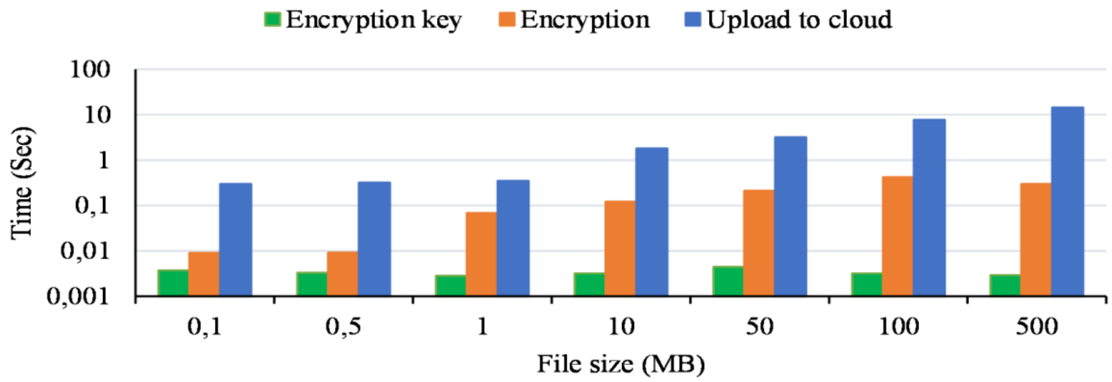


Figure 13. *Performance analysis of record upload to CS for HDaSC with varying record sizes.*

In the download process, the original health record received from the CS is the opposite of the upload process. Figure 14 illustrates the results for the download process from the CS and the decryption process for HDaSC. As the record size grows from 0.1 to 500 MB, the download time steadily increases. It requires only 0.32 seconds with the smallest record size of 0.1 MB, although the greater record computation of size 500 MB greatly raises the download time to 18.07 s. It follows that the trend of results is the same as the uploading process. Moreover, the times in decryption, as well as download, are changing.

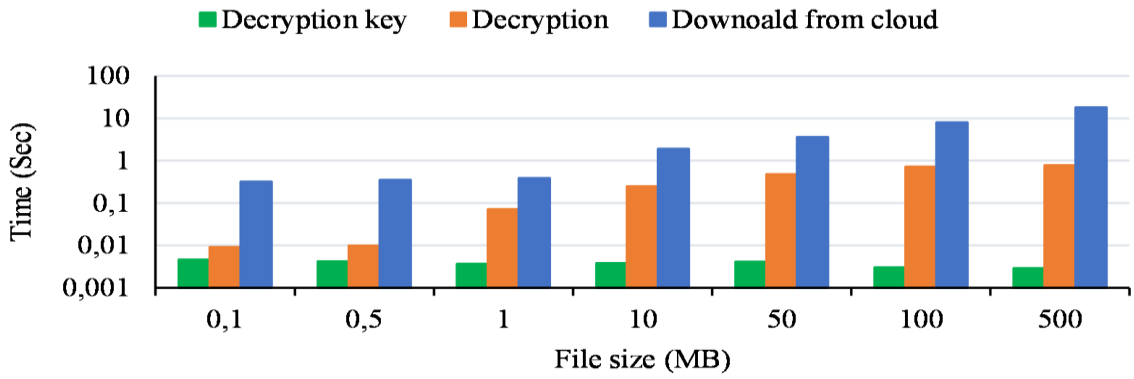


Figure 14. *Performance of record download from CS for HDaSC with varying record sizes.*

5 Discussion

The evaluation is based on the key generation time, record encryption time, record decryption time, record upload time and record download time for different user numbers and different file sizes ranging from 0.1 MB to 500 MB to prove the usefulness of our proposal. A comparison of our suggested system to other nearby works based on a set of security criteria such as integrity, authentication, privacy, access control, encryption and key encryption is also included.

Table 3 and Table 4 show the upload and download process response times according to the difference in file size in diverse current methodologies alongside the HDaSC system, where ‘FS’ indicates the record size, and ‘UP’ and 'DN' refer to ‘upload’ and ‘download', respectively. As a result, the HDaSC system outperforms the other existing methods. It offers better upload and download performance, where the values might vary considerably based on the internet connection speed and record size.

Comparisons of response times for encryption and decryption provided in Table 3 and Table 4 reveal that the proposed HDaSC outperforms other methods thanks to the absence of heavy calculations. The proposed method is at least twice faster than the best one among the others. This could not be explained by exceptionally favourable internet connection speed, especially when the superiority is stable over different runs.

***Table 3**. Comparison of turnaround times for small files.*

FS (MB)	Methodology time in seconds					
	Jana et al. (2017)		**Hema and Kesavan (2019)**		**HDaSC**	
	UP	DN	UP	DN	UP	DN
0.1	0.2024	0.422	0.70	0.70	0.24	0.32
0.5	0.8116	1.6806	0.80	0.82	0.32	0.35
1	1.6222	3.362	1.20	1.24	0.35	0.39

***Table 4**. Comparison of turnaround times for large files.*

FS (MB)	Methodology time in seconds					
	Bentajar et al. (2019)		**Hema and Kesavan (2019)**		**HDaSC**	
	UP	DN	UP	DN	UP	DN
10	12.8	4.552	5.60	5.68	1.82	1.89
50	39.941	14.528	8.25	8.78	3.2	3.58
100	76.554	25.492	16.35	18.98	7.69	8.01
500	310.402	137.145	32.10	38.22	14.36	18.03

The principal objective of adopting a hybrid encryption algorithm (RSA and AES) to secure a health record in a cloud environment is that it offers three keys, i.e., a secret key for encryption of the healthcare record, the public key for encryption of the secret key, and a private key which is divided into two segments and stored in two different locations for decryption. Thus, the health record after uploading is stored in an encrypted form and can only be decrypted with the secret key, which is decrypted by concatenating the two segments of the user's private key to ensure copyright protection and integrity of the health record. Both algorithms are used to protect health data. The RSA algorithm is excellent for key exchange, but it is painful to use. While the AES algorithm is extremely fast, it is vulnerable to security problems associated with key exchange.

In addition to the results shown above, most of the works (Bentajar et al., 2019; Jana et al., 2017; Michalas and Sachdeva, 2013; Mahalle and Shahade, 2019) do not address medical data, especially since sensitive information requires special attention.

On top of the fact that medical data are considered sensitive and need to be carefully secured, they are also characterized by their large size, which requires taking this aspect seriously in terms of encryption and decryption time of each medical record according to its size, which does not exist in some other studies (Yang et al., 2019; Oliveira et al., 2020).

The HDaSC guarantees the confidentiality of data during the transmission and storage of health records. The HDaSC system ensures data confidentiality during storage by encrypting the health data records with hybrid encryption (AES and RSA) by the TP-CS before outsourcing. The system also maintains confidentiality when exchanging data between the components of the framework. There are two types of data transmission: the first is an encrypted form between the TP-CS and the cloud storage. The second is via an SSL channel between the TP-CS and the users.

The proposed system ensures the integrity of health data. It verifies that no unauthorized person knowingly or unknowingly alters patient data. It also ensures that the data are not altered once they have been created. With integrity, it is impossible to prevent information from being changed. It simply provides evidence to determine whether the information has been altered or not. The healthcare field is very important in terms of security, especially when dealing with data in the cloud, where additional protection is needed. Restricting access to the health record to authorized users only protects health data from access, use and modification.

The HDaSC system ensures the privacy of private health data. Health information identified as private in the proposed system is preserved by granting permission for these data to users who need them. As such, it protects health data records from insider threats. The access list limits user access, thus protecting health data from insider attacks.

Moreover, the proposed architecture is an open (not closed), extensible and easily maintainable architecture using advanced technologies such as blockchain, unlike some works that are often closed and not very extensible (Jana et al., 2017; Seo et al., 2013; Khan et al., 2014). Table 5 presents how HDaSC effectively ensures the security objectives. The ✗ and ✓ indicate whether the literature supports this feature or not.

***Table 5**. Comparison of HDaSC system with related works.*

Functions	**Ali et al. (2015)**	**Jana et al. (2017)**	**Seol et al. (2018)**	**Michalas et al. (2019)**	**Hema and Kesavan (2019)**	**Babrahem and Monowar (2021)**	**HDaSC**
Privacy	✓	✓	✓	✓	✓	✓	✓
Authentication	✓	✗	✓	✓	✓	✓	✓
Integrity	✓	✓	✓	✓	✓	✓	✓
Access control	✓	✗	✓	✓	✓	✓	✓
Non-repudiation	✗	✓	✓	✗	✗	✗	✓
Encryption	✓	✓	✓	✓	✓	✓	✓
Hybrid encryption	✓	✓	✗	✓	✗	✗	✓
Key encryption	✗	✓	✗	✗	✗	✗	✓

6 Conclusion

The privacy of patients' personal information when sharing health data in the cloud is a major concern. The primary goal of our research was to develop a cloud-based health data sharing system that ensures access control, patient privacy, data confidentiality and data integrity. This paper has proposed a privacy-preserving sharing and storage of health data in the cloud based on symmetric and asymmetric encryption named HDaSC (Healthcare Data Sharing in the Cloud). The system leverages AES and RSA cryptography benefits to secure the sharing of health records in the cloud. Besides, the encryption and decryption processes are performed on the TP-CS cryptographic server. To obtain the ciphertext, the AES technique is employed to secure the health record. The AES key is secured and the ciphertext key is generated using the RSA technique. In addition, the TP-CS cryptographic server handles the encryption and decryption processes. The private key is split into two halves, one of which is used for decryption. Even if one portion is intercepted and processed, it is stored in various locations to avoid reconstituting it. The system generates two random strings and distributes one to each side, after which the data are decrypted using the exclusive OR of the two random strings.

Moreover, the HDaSC system was evaluated considering the time consumed during key generation, record encryption and decryption time, record upload and download time. The findings show that the proposed HDaSC system performs better than other state-of-the-art systems and can practically share secure health data in a cloud environment. Based on the results given in this paper, we plan to expand the designs proposed while retaining the security and availability of health data. As future work, we are considering many prospective extensions or improvements to our system, finding a link between our concept and blockchain to achieve a hybrid system that ensures the continuation of health data exchange and moving towards a machine-learning-based anomaly detection system during the health data sharing procedure.

Additional Information and Declarations

Funding: This research is partially supported by the Projets de Recherche-Formation Universitaire (PRFU) under the number C00L07UN250220200002.

Conflict of Interests: The authors declare no conflict of interest.

Author Contributions: I.B.: Conceptualization, Methodology, Writing – original draft, Writing – review & editing. K.Z.: Supervision, Writing – review & editing.

References

ACT. (1996). Health insurance portability and accountability act of 1996. Public law. https://aspe.hhs.gov/report/health-insurance-portability-and-accountability-act-1996

Ali, M., Dhamotharan, R., Khan, E., Khan, S. U., Vasilakos, A. V., Li, K., & Zomaya, A. Y. (2017). SeDaSC: Secure Data Sharing in Clouds. *IEEE Systems Journal*, *11*(2), 395–404. https://doi.org/10.1109/jsyst.2014.2379646

Al-Issa, Y., Ottom, M. A., & Tamrawi, A. (2019). eHealth Cloud Security Challenges: A Survey. *Journal of Healthcare Engineering*, *2019*, Article ID 7516035. https://doi.org/10.1155/2019/7516035

Anderson, N. R., Lee, E. S., Brockenbrough, J. S., Minie, M. E., Fuller, S., Brinkley, J., & Tarczy-Hornoch, P. (2007). Issues in Biomedical Research Data Management and Analysis: Needs and Barriers. *Journal of the American Medical Informatics Association : JAMIA*, *14*(4), 478–488. https://doi.org/10.1197/jamia.M2114

Andrews, L., Gajanayake, R., & Sahama, T. (2014). The Australian general public's perceptions of having a personally controlled electronic health record (PCEHR). *International Journal of Medical Informatics*, *83*(12), 889–900. https://doi.org/10.1016/j.ijmedinf.2014.08.002

Babitha, M., & Babu, K.R. (2016). Secure cloud storage using aes encryption. In *2016 International Conference on Automatic Control and Dynamic Optimization Tech- niques (ICACDOT),* (pp.859–864). IEEE. https://doi.org/10.1109/ICACDOT.2016.7877709

Babrahem, A. S., & Monowar, M. M. (2021). Preserving confidentiality and privacy of the patient's EHR using the OrBAC and AES in cloud environment. *International Journal of Computers and Applications*, 43(1), 50-61. https://doi.org/10.1080/1206212X.2018.1505025

Bentajer, A., Hedabou, M., Abouelmehdi, K., Igarramen, Z., & El Fezazi, S. (2019). An IBE-based design for assured deletion in cloud storage. *Cryptologia*, *43*(3), 254–265. https://doi.org/10.1080/01611194.2018.1549123

Boumezbeur, I., & Zarour, K. (2022a). Privacy-Preserving and Access Control for Sharing Electronic Health Record using Blockchain Technology. *Acta Informatica Pragensia*, 11(1), 105-122. https://doi.org/10.18267/j.aip.176

Boumezbeur, I., & Zarour, K. (2022b). EMR Sharing with Privacy Preservation Using Blockchain Technology. In *Proceedings of the The 1st national Conference on Information and Communication (CICT),* (pp.41-43). Tamanrasset.

Chen, Y.-Y., Lu, J.-C., & Jan, J.-K. (2012). A Secure EHR System Based on Hybrid Clouds. *Journal of Medical Systems*, *36*(5), 3375–3384. https://doi.org/10.1007/s10916-012-9830-6

HealthData.gov. (2021). HealthIT. https://healthit.gov

Hema, V., & Kesavan, R. (2019). ECC Based Secure Sharing of Healthcare Data in the Health Cloud Environment. *Wireless Personal Communications*, *108*(2), 1021–1035. https://doi.org/10.1007/s11277-019-06450-7

HHS. (2006). Personal health records and personal health record systems. A Report and Recommendations from the National Committee on Vital and Health Statistics. US Department of Health & Human Services. https://ncvhs.hhs.gov/wp-content/uploads/2014/05/0602nhiirpt.pdf

ISO. (2011). *ISO 18308:2011, Health Informatics: Requirements for an Electronic Health Record Architecture*. International Organization for Standardization.

Jana,B., Poray, J., Mandal. T., & Kule, M. (2017). A multilevel encryption technique in cloud security. In *2017 7th International Conference on Communication Systems and Network Technologies (CSNT),* (pp. 220-224). IEEE. https://doi.org/10.1109/CSNT.2017.8418541

Khan, A. N., Kiah, M. L. M., Madani, S. A., Ali, M., Khan, A. ur R., & Shamshirband, S. (2013). Incremental proxy re-encryption scheme for mobile cloud computing environment. *The Journal of Supercomputing*, *68*(2), 624–651. https://doi.org/10.1007/s11227-013-1055-z

Kuo, A. M.-H. (2011). Opportunities and Challenges of Cloud Computing to Improve Health Care Services. *Journal of Medical Internet Research*, *13*(3), e67. https://doi.org/10.2196/jmir.1867

Low, C., & Hsueh Chen, Y. (2012). Criteria for the Evaluation of a Cloud-Based Hospital Information System Outsourcing Provider. *Journal of Medical Systems*, *36*(6), 3543–3553. https://doi.org/10.1007/s10916-012-9829-z

Mahalle, V.S. & Shahade, A.K. (2014). Enhancing the data security in cloud by implementing hybrid (rsa & aes) encryption algorithm. In *2014 International Conference on Power, Automation and Communication (INPAC),* (pp.146–149). IEEE. https://doi.org/10.1109/INPAC.2014.6981152

Michalas, A., Bakas, A., Dang, H.V., & Zalitko, A. (2019). MicroSCOPE: Enabling Access Control in Searchable Encryption with the Use of Attribute-Based Encryption and SGX. In *Nordic Conference on Secure IT Systems,* (pp. 254–270). Springer. https://doi.org/10.1007/978-3-030-35055-0_16

Noumeir, R. (2011). Sharing Medical Records: The XDS Architecture and Communication Infrastructure. *IT Professional*, *13*(4), 46–52. https://doi.org/10.1109/mitp.2010.123

Oliveira, M. T., Bakas, A., Frimpong, E., Groot, A. E. D., Marquering, H. A., Michalas, A., & Olabarriaga, S. D. (2020). A break-glass protocol based on ciphertext-policy attribute-based encryption to access medical records in the cloud. *Annals of Telecommunications*, *75*(3-4), 103–119. https://doi.org/10.1007/s12243-020-00759-2

Oliveira, M. T., Dang, H.-V., A. Reis, L. H., Marquering, H. A., & D. Olabarriaga, S. (2021). AC-AC: Dynamic revocable access control for acute care teams to access medical records. *Smart Health*, *20*, 100190. https://doi.org/10.1016/j.smhl.2021.100190

Poulymenopoulou, M., Malamateniou, F., & Vassilacopoulos, G. (2011). Emergency Healthcare Process Automation Using Mobile Computing and Cloud Services. *Journal of Medical Systems*, *36*(5), 3233–3241. https://doi.org/10.1007/s10916-011-9814-y

Pugazhenthi, A., & Chitra, D. (2019). Data Access Control and Secured Data Sharing Approach for Health Care Data in Cloud Environment. *Journal of Medical Systems*, *43*(8). https://doi.org/10.1007/s10916-019-1381-7

Rajakumar, M., Ramya, J., Sonia, R., & Uma Maheswari, B. (2021). A Novel Scheme for Encryption and Decryption of 3D Point and Mesh Cloud Data in Cloud Computing. *Journal of Control Engineering and Applied Informatics*, 23(1), 93–102.

Zhang, L., Hu, G., Mu, Y., & Rezaeibagha, F. (2019). Hidden Ciphertext Policy Attribute-Based Encryption With Fast Decryption for Personal Health Record System. *IEEE Access*, *7*, 33202–33213. https://doi.org/10.1109/access.2019.2902040

Seo, S.-H., Nabeel, M., Ding, X., & Bertino, E. (2014). An Efficient Certificateless Encryption for Secure Data Sharing in Public Clouds. *IEEE Transactions on Knowledge and Data Engineering*, *26*(9), 2107–2119. https://doi.org/10.1109/tkde.2013.138

Seol, K., Kim, Y.-G., Lee, E., Seo, Y.-D., & Baik, D.-K. (2018). Privacy-Preserving Attribute-Based Access Control Model for XML-Based Electronic Health Record System. *IEEE Access*, *6*, 9114–9128. https://doi.org/10.1109/access.2018.2800288

Singh, N., & Singh, A. K. (2017). Data Privacy Protection Mechanisms in Cloud. *Data Science and Engineering*, *3*(1), 24–39. https://doi.org/10.1007/s41019-017-0046-0

Suresh, D., & Florence, M. L. (2019). Securing Personal Health Record System in Cloud Using User Usage Based Encryption. *Journal of Medical Systems*, *43*(6). https://doi.org/10.1007/s10916-019-1301-x

Svantesson, D., & Clarke, R. (2010). Privacy and consumer risks in cloud computing. *Computer Law & Security Review*, *26*(4), 391–397. https://doi.org/10.1016/j.clsr.2010.05.005

Technavio. (2020). COVID-19 Impact and Recovery Analysis- Global Healthcare Cloud Computing Market 2020-2024| Increasing Cloud Assisted Medical Collaborations to Boost Market Growth. https://www.technavio.com

Yang, Y., Zheng, X., Guo, W., Liu, X., & Chang, V. (2019). Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Information Sciences*, *479*, 567–592. https://doi.org/10.1016/j.ins.2018.02.005

Zhang, L., Wu, Q., Mu, Y., & Zhang, J. (2016). Privacy-Preserving and Secure Sharing of PHR in the Cloud. *Journal of Medical Systems*, *40*(12). https://doi.org/10.1007/s10916-016-0595-1

Editorial record: The article has been peer-reviewed. First submission received on 12 April 2022. Revision received on 24 May 2022. Accepted for publication on 13 June 2022. The editors coordinating the peer-review of this manuscript were Venkatachalam Kandasamy, Mohamed Abouhawwash, Nebojsa Bacanin. The editor in charge of approving this manuscript for publication was Zdenek Smutny.

Special Issue: Sustainable Solutions for Internet of Things Using Artificial Intelligence and Blockchain in Future Networks.

Acta Informatica Pragensia is published by Prague University of Economics and Business, Czech Republic.

ISSN: 1805-4951