Hybrid and Secure E-Health Data Sharing Architecture in Multi-Clouds Environment

Tayssir Ismail[1,2(✉)], Haifa Touati[1,2], Nasreddine Hajlaoui[1,2], and Hassen Hamdi[3]

[1] Hatem Bettahar Research Unit IResCoMath, Gabes, Tunisia
taissirism88@gmail.com, haifa.touai@cristal.rnu.tn, hajlaoui.ing@gmail.com
[2] Faculty of Science of Gabes, Gabes, Tunisia
[3] MIRACL Laboratory, FSEG-Sfax, Sfax, Tunisia
hhassen2006@yahoo.fr

Abstract. Healthcare is among the sectors showing efforts in adopting cloud computing to its services considering the provided cost reduction and healthcare process efficiency. However, outsourcing patient's sensitive data increases the concerns regarding security, privacy, and integrity of healthcare data. Therefore, there is a need for building a trust relationship between patients and e-health systems. In this paper, we propose a privacy-preserving framework, called Hybrid and Secure Data Sharing Architecture (HSDSA), to secure data storage in e-health systems. Our approach improves security in healthcare by maintaining the privacy and confidentiality of sensitive data and preventing threats. In fact, in the upload phase, Multi-cloud environment is used to store Rivest–Shamir–Adleman (RSA) encrypted medical records. We adopt a Shamir's secret sharing approach for the distribution of shares to different independent cloud providers. In the retrieval phase, the reconstruction operation is based on the (t, n) strategy. To check the requester identity and to prove the hash possession, we used a zero-knowledge cryptography algorithm, namely the Schnorr algorithm. The patient has a total control over the generation and management of the decryption keys using Diffie-Hellman algorithm without relying on a trusted authority.

Keywords: E-health system security · Privacy preservation · Multi-cloud · Data storage · Data share · Data encryption

1 Introduction

Cloud computing is a new promising technology that leverages the user from the burden of hardware maintenance and offers dynamically flexible and scalable computational resources accessible from any place where a network is available. The emergence of this paradigm has deeply influenced many domains and especially the healthcare sector. However, the usage of this model in the healthcare

M. Jmaiel et al. (Eds.): ICOST 2020, LNCS 12157, pp. 249–258, 2020.
https://doi.org/10.1007/978-3-030-51517-1_21

domain needs the reinforcement of security measures because data are susceptible to lose, leakage or theft. Therefore, confidentiality and integrity of the stored Electronic Health Records (EHR) are deemed as one of the major challenges elevated by the external storage. Besides, the privacy of sensitive data must be guaranteed. To overcome the above cited challenges, cryptographic techniques for securing e-health systems are widely adopted. But the reliance on a single cloud storage provider has shown many drawbacks like a single point of failure, vendor lock-in and malicious insiders. To narrow down the listed disadvantages, it is advisable to use multi-cloud architecture. One of the key concepts of this model is to store data on different cloud server providers where an insider is not able to reconstruct the original data from a single share [1].

In this context, several solutions have been proposed in the literature to ensure secure multi-cloud storage in e-health systems [2–5]. They mainly have two phases: *storage* and *retrieval*. They also all use cryptographic primitives to ensure EHRs security. Authors of [2] use an Attribute Based-Encryption (ABE) for selective access authorisation and cryptographic secret sharing. The EHRs split and reconstruction is done through a proxy. In [3], ABE is used for selective data sharing with physicians without allowing them to know the precise description of the patient's illnesses. Biometrics based authentication and Kerberos tickets session are used in [4] to guarantee secure interaction with the EHR system. In addition, a steganographic technique is used to store EHR. In [5], authors propose the use of Shamir's Secret Sharing not only to distribute EHR shares among cloud servers but to retrieve the requested EHR from partial cloud servers. In summary, the main drawback of [2–5] is the reliance on a trusted third party which may not be adequate for practical use as they show security risks. Hence, a secure privacy-preserving data storage solution is still needed to improve the patient role to monitor his data on the cloud.

In this paper, we present a Hybrid and Secure Data Sharing Architecture (HSDSA), for secure and privacy-preserving storing and sharing of patient's sensitive data in a Multi-cloud environment without relying on a trusted third party. In HSDSA, cloud providers are assumed to be semi-trusted: honest but curious. HSDSA gives the patient total control over the generation and management of the decryption keys without relying on a trusted authority and thus it is more applicable for public cloud environments. To protect the data from external attackers, Rivest–Shamir–Adleman (RSA) encryption is applied before outsourcing EHR. To secure data against cloud providers curiosity, Shamir's secret sharing is adopted. The resulted shares are distributed to multiple clouds. To download an EHR, HSDSA recovers its shares using an outsourcing reconstruction operation based on the (t, n) strategy. To complete the file decryption, a Schnorr-based technique is used to prove data possession and to verify the requester identity. Then a session, using the Diffie Hellman (DH) algorithm, is created to securely exchange the decryption key. Finally, the key is extracted and the original EHR could be recovered. Outsourcing reconstruction operation based on the (t, n) strategy is used.

The remainder of the paper is organised as follows. Section 2 gives an overview of the overall architecture of the proposed framework and its components. Sections 3 and 4 detail the different techniques used in the storage and retrieval processes. Finally, Sect. 5 concludes the paper and highlights some open issues.

2 Architecture Overview of the Proposed Scheme

We recall that our goal is to securely store EHRs in multi-cloud environment and to securely share them among healthcare organisations staff. In the following, we will give an overview of the HSDSA framework in which we focus on the context of medical data storage, share and retrieval. The basics of HSDSA are shown in Fig. 1. Our system compromises three different entities: Data Owner (DO), Data Requester (DR) and n CSP (CSP_1,...,CSP_n). The key components of the HSDSA framework include:

- **The storage process** which is composed of two phases, namely the **registration** phase and the **storage** phase.
- **The retrieval process** which is composed of two phases, namely the EHR **reconstruction** and the EHR **recovery** phases.

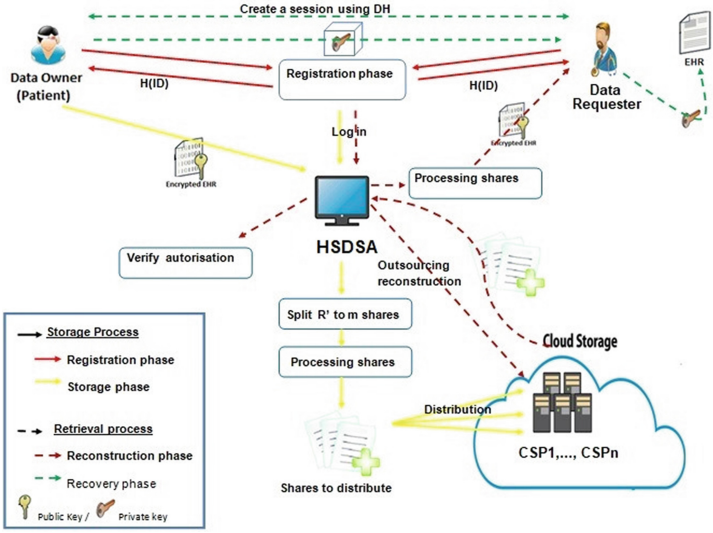

Fig. 1. Architecture overview

As shown in Fig. 1, the workflow of HSDSA is as follows:

- **The registration phase** starts when a DO or a DR signs in to the Framework Interface. After being signed to HSDSA, a DO or a DR receives in response the hash of his identity H($ID_{DO,DR}$).

- **The storage phase** starts when a DO wants to store his EHRs, he calculates the digital signature of his EHR (R), encrypts R using Rivest–Shamir–Adleman (RSA) algorithm for both and logs in to the HSDSA. Then, the DO uploads his encrypted record ($R^{'}$) and the hash value of the original record H(R). HSDSA generates a unique identifier (ID_R) of the EHR to guarantee the anonymity of stored data in Cloud servers and stores H(ID_{DO}), H(R) and ID_R. The framework calculates the hash of the uploaded EHR ($R^{'}$) and splits it into m shares. Then, it performs an exclusive OR operation between each share ($S_i^{'}$) and the hash value of $R^{'}$. The distribution is done using Shamir's secret sharing algorithm and the resulted shares are sent to n different Cloud Server Providers CSP_1, ... CSP_n.
- **The reconstruction phase** starts when a DO or an authorised user DR wants to get the EHR, he sends a request to the framework. After confirming the request, HSDSA assigns a CSP (CSP_R) to perform the reconstruction step. The CSP_R gets t shares or more from CSP_1, ... CSP_n. Once the reconstruction is done, the CSP_R returns the resulted shares to the framework.
- Finally, **the recovery phase**, when the DR wants to get the DO's private key, he has to prove that he is the right DR and he has the correct hash value of R. To this end, the Schnorr algorithm is used. Once, the DO makes sure that the DR is an authorised requester and that he possesses the encrypted version of the EHR ($R^{'}$), then the DR and the DO try to establish a session using Diffie-Hellman (DH) algorithm to exchange decryption key securely. Once they agree on a session key, K_s. The DO encrypts his private key using K_s and sends it to the DR. Once the private key is extracted, the DR can finally recover the desired EHR (R).

3 Analysis of the Proposed Storage Process

In the following, we detail the techniques used in the two phases of the storage process: the registration phase and the storage phase.

3.1 The Registration Phase

As recommended in cloud-based storage solutions, building a trust relationship between partners is a necessity. To achieve this goal, the first step is to make sure that all users are registered to the framework. If a new user wants to benefit from services provided by the HSDSA framework, he must be correctly authenticated. Once he registers, he receives a value containing the hash of his identity H($ID_{DO,DR}$) in order to maintain the anonymity of user identities.

3.2 The Storage Phase

HSDSA acts as an intermediary between DO and CSPs. Our goal is to provide a secure storage facility to authorised users. This phase involves Shamir's secret sharing technique to make sure that the multi-cloud environment, used to

store shares, is a collusion-safe. By collusion-safe we mean that if two or more CSPs combine their keys, they cannot decrypt the data. Ten steps, illustrated in Table 1, describe the storage phase.

When a DO wants to store an EHR, he calculates the digital signature of the original EHR (R). Then the RSA is used to split the selected EHR into blocks and encrypt them. Sequential execution of RSA needs a lot of calculation. Therefore, we use the enhancement proposed in [6] where authors have parallelized the process of encryption and decryption of a large number of data blocks. The resulted file $R^{'}$ and H(R) are sent to HSDSA.

$$R^{'} = E_{PK_R}(R) \tag{1}$$

When the framework receives $R^{'}$ and H(R), it generates a unique identifier ID_R corresponding to the file $R^{'}$. This is used to guarantee the unlinkability between DO and EHR. After that, HSDSA computes the hash of $R^{'}$ (H($R^{'}$)) and stores ID_R, H(R) and H($R^{'}$). Next the framework splits $R^{'}$ into m shares [S_1, ... ,S_m], performs the exclusive OR operation of each split of $R^{'}$ with H($R^{'}$).

$$\begin{aligned}[S_1^{'}, ..., S_m^{'}] &= R^{'} \bigoplus H(R^{'}) \\ &= [S_1, ..., S_m] \bigoplus H(R^{'}) \\ &= [S_1] \bigoplus H(R^{'}), .., [S_m] \bigoplus H(R^{'})\end{aligned} \tag{2}$$

[$S_1^{'}$, ... ,$S_m^{'}$] are the shares to be stored in independent CSPs. To securely distribute the shares, we adopted Shamir's secret sharing protocol. It represents a so-called (t,n) threshold scheme with $1 \leq t \leq n$. This mechanism permits the distribution of a document among n parts in a way that reconstruction is possible if at least t shares are present. Suppose a share $S_i^{'}$ (for i i = 1 ... m), Shamir's secret sharing algorithm sets $a_{i0} = S_i^{'}$, chooses a_{i1}, ..., a_{it-1} at random, takes distinct values x_1, x_2,..., x_m with $m \geq t$-1 and computes the shares to distribute, as follows:

$$\begin{cases} S_{1i} = (x_i, f_1(x_i)) \\ \quad .... \\ S_{mi} = (x_i, f_m(x_i)) \end{cases}, \ for\ i = 1..n$$

In the proposed architecture, HSDSA selects m polynomials.

$$\begin{cases} f_1(x) = a_{10} + a_{11}x + a_{12}x^2 + ... + a_{1t-1}x^{t-1}\ mod\ p \\ \quad ... \\ f_m(x) = a_{m0} + a_{m1}x + a_{m2}x^2 + ... + a_{mt-1}x^{t-1}\ mod\ p \end{cases}$$

Where

$$\begin{bmatrix} a_{11}, ..., a_{1t-1} \\ ... \\ a_{m1}, ..., a_{mt-1} \end{bmatrix} \in \quad \mathbb{Z}_{\text{ı}}$$

The HSDSA computes n shares S_{1i}, ..., S_{mi} (i = 1, ..., n) and distributes them to CSP_1, ..., CSP_n.

Table 1. Scenario of the storage phase

User	**HSDSA**	CSP_1, ... CSP_n
1.Calculate H(R) = E_{PU_R}(R)		
2. Encrypt R: $R^{'} = E_{PK_R}$(R)		
3.Log in		
$\xrightarrow[H(R)]{\text{upload } R^{'}}$		
	4.Generate an identifier of $R^{'}$: ID_R	
	5.Calculate H($R^{'}$)	
	6.Split $R^{'}$ to m shares: $R^{'}$=S_1, ... ,S_m	
	7.Calculate $S_i^{'}$ for i=(1,..,m)	
	$S_i^{'} = S_i \bigoplus$ H($R^{'}$)	
	8. Select polynomials:$f_i(x)$ i=(1,..,m)	
	9. Compute shares:S_{1i},..,S_{mi}	
	$\xrightarrow[ID_R]{\text{Send shares}}$	
		10.Store shares and ID_R

4 Analysis of the Proposed Retrieval Process

File retrieval, also known as file reconstruction, is the reversal process of file distribution and file slicing. In this framework the retrieval process starts when a data requester DR needs to get an EHR. He must log in and submit the EHR identifier (ID_R). HSDSA checks if the DR has the right to get the requested EHR. If the authorisation succeeded, then the reconstruction phase starts.

4.1 The Reconstruction Phase

Since the reconstruction of $R^{'}$ requires a massive amount of computation and that client resources are limited, we will use the reconstruction outsourcing scheme proposed in [5]. The framework assigns a CSP (CSP_R) to reconstruct shares $S_j^{'}$. The reconstruction is considered successfull only if CSP_R gets at least t shares from CSP_1, ..., CSP_n. We assume that CSP_R gets k shares.

$$\begin{bmatrix} S_{11}, ...S_{m1} \\ ... \\ S_{1k}, ..., S_{mk} \end{bmatrix}, (k \geq t)$$

CSP_R computes $S_j^{'}$ for $j = (1, ..., m)$ using Lagrange interpolation polynomial and sends them to HSDSA:

$$S_j^{'} = \sum_{i=1}^{k} S_{ji} \prod_{l=1, l \neq i}^{k} \frac{x_i}{x_i - x_l} \ mod\ p\ (j = 1, ..., m) \tag{3}$$

To make sure that CSP_R could not reveal any useful information, knowing that he is a curious and dishonest party, doing an exclusive OR operation helps to blind the content. Upon receipt of the shares, the HSDSA framework performs the exclusive OR operation between $S_j^{'}$ and H(R') to get S_j.

4.2 The Recovery Phase

This phase aims to securely transfer the Data Owner's private key to the right Data Requester. Table 2 illustrates the main steps related to this phase.

Once the reconstruction phase is done, HSDSA sends ID_{DO}/ID_{DR} to the Data Owner and to the Data Receiver. First the DR has to prove to the DO that he holds the right hash value of the original file $(H(R))^{'}$ and that he is the correct DR. For this purpose, a Schnorr's identification protocol [7] is used not only to prove the hash possession but also to verify the DR identity. In the process of the latter algorithm the DO checks if H(R) $\stackrel{?}{=}$ $(H(R))^{'}$), and verifies if the n first bits of the DR identity match the ID_{DR} previously sent by FI. Then the DO and the DR must establish a secure connection for key exchange, based on Diffie-Hellman (DH) scheme Ephemeral version [8], reinforced with the hash value of the original file (H(R)). Establishing a session means that the two partners have agreed on session key (K_s) that will be used to crypt partners

Table 2. Scenario of the recovery phase

Data Requester		Data Owner
Schnorr algorithm		
$\alpha = (H(R))^{'}$, s: secret key $v = \alpha^{-s}$ 1. Choose r and calculate $x = \alpha^{r} \bmod p$		
	$\xrightarrow{x}$	
		2.Choose e, n $(3 \leq n \leq 20)$
	$\xleftarrow{e,\ n}$	
3.Calculate $y = r + es \bmod p$ Calculate $F = n$ first bits of ID_{DR} Calculate $Y = y \bigoplus F$		
	$\xrightarrow{Y}$	
		4.Calculate $y = Y \bigoplus F$ 5.Verify $x \stackrel{?}{=} H(R)^{y}.v^{e} \bmod p$
Diffie Hellman algorithm		
s, p, g		b
6.Calculate $A = g^{s} \bmod p$ F_1 = premier n bits de H(R) $A_1 = A \bigoplus F_1$		
	$\xrightarrow{A_1,p,g}$	
		7.Calculate $B = g^{b} \bmod p$ $B_1 = B \bigoplus F_1$
	$\xleftarrow{B_1}$	
	$\xleftrightarrow{\text{Calculate } K_s}$	
$B = B_1 \bigoplus F_1$ $K_s = B^{s} \bmod p$	8. Calculate	$A = A_1 \bigoplus F_1$ $K_s = A^{s} \bmod p$
		9.Encryt private key PU $PU^{'} = E_{K_s}(\text{PU})$
	$\xleftarrow{PU^{'}}$	
10.Extract PU PU $= D_{K_s}\ (PU^{'})$ 11. Recover R $= D_{PU_R}\ (R^{'})$		

metadata. Next, the DO encrypts his private key (PU) using K_s and sends the resulted value $PU^{'}$ to the DR.

$$PU^{'} = E_{K_s}(PU) \tag{4}$$

The DR decrypts $PU^{'}$ using K_s to extract the DO private key. Possessing PU, the DR decrypts $R^{'}$ to finally recover the original EHR (R).

$$PU = D_{K_s}(PU^{'}) \tag{5}$$

$$R = D_{PU}(R^{'}) \tag{6}$$

5 Conclusion

In this paper, we presented HSDSA, a novel architecture for secure EHR. HSDSA includes several techniques, namely (i) RSA algorithm to guarantee the security of outsourced data, (ii) Shamir's secret sharing to securely distribute data across multiple clouds, (iii) a secure outsourcing reconstruction based on the (t, n) strategy, (iv) a Schnorr-based technique to prove data possession and to verify the requester identity and (v) a Diffie-Hellman algorithm to securely exchange decryption key. The proposed scheme allows the patient to get total control over the generation and management of the decryption keys without relying on a trusted authority. In a future work, we plan to add governmental organisation as Data Requester. These latter need to access data without the Data Owner authorisation. Hence, we aim to protect privacy while giving them access to EHR.

References

1. AlZain, M.A., et al.: Cloud computing security: from single to multi-clouds. In: 45th International Conference on System Sciences (2012)
2. Fabian, B., Ermakova, T., Junghanns, P.: Collaborative and secure sharing of healthcare data in multi-clouds. Inform. Syst. **48**, 132–150 (2015)
3. Xhafa, F., Li, J., Zhao, G., Li, J., Chen, X., Wong, D.S.: Designing cloud-based electronic health record system with attribute-based encryption. Multimed. Tools Appl. **74**(10), 3441–3458 (2015)
4. Premarathne, U.S.: Hybrid cryptographic access control for cloud based electronic health records systems. IEEE Cloud Comput. **2**, 1–7 (2017)
5. Zhang, H., Yu, J., Tian, C., Zhao, P., Xu, G., Lin, J.: Cloud storage for electronic health records based on secret sharing with verifiable reconstruction outsourcing. IEEE Access **6**, 40713–40722 (2018)
6. Gupta, P., Verma, D.K., Singh, A. K.: Improving RSA algorithm using multithreading model for outsourced data security in cloud storage. In: 8th International Conference on Cloud Computing, Data Science & Engineering, pp. 14–15 (2018)
7. Smart, N.P.: Zero-knowledge proofs. In: Smart, N.P. (ed.) Cryptography Made Simple. Information Security and Cryptography, pp. 425–438. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-21936-3_21
8. Blake-Wilson, S., Menezes, A.: Authenticated Diffe-Hellman key agreement protocols. In: Tavares, S., Meijer, H. (eds.) SAC 1998. LNCS, vol. 1556, pp. 339–361. Springer, Berlin (1998). https://doi.org/10.1007/3-540-48892-8_26