CLOUD STORAGE FOR ELECTRONIC HEALTH RECORDS BASED ON SECRET SHARING

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

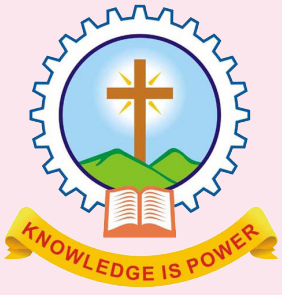
COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

NISHWA FATHIMA



Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

CLOUD STORAGE FOR ELECTRONIC HEALTH RECORDS BASED ON SECRET SHARING

Seminar Report

Submitted in partial fulfillment of the requirements for the award of degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

of

APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

Submitted By

NISHWA FATHIMA

Department of Computer Science & Engineering
Mar Athanasius College Of Engineering Kothamangalam

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
MAR ATHANASIUS COLLEGE OF ENGINEERING
KOTHAMANGALAM

CERTIFICATE

This is to certify that the report entitled **Cloud Storage for Electronic Health Records Based on Secret Sharing With Verifiable Reconstruction Outsourcing** *submitted by* **Mrs. NISHWA FATHIMA**, *Reg.No.***MAC15CS042** *towards partial fulfillment of the requirement for the award of Degree of Bachelor of Technology in Computer science and Engineering from APJ Abdul Kalam Technological University for December 2018 is a bonafide record of the seminar carried out by her under our supervision and guidance.*

.....	.....	.....
Prof. Joby George	**Prof. Neethu Subash**	**Dr. Surekha Mariam Varghese**
Faculty Guide	***Faculty Guide***	***Head of the Department***

Date: Dept. Seal

ACKNOWLEDGEMENT

First and foremost, I sincerely thank the God Almighty for his grace for the successful and timely completion of the seminar.

I express my sincere gratitude and thanks to Dr. Solly George, Principal and Dr. Surekha Mariam Varghese, Head Of the Department for providing the necessary facilities and their encouragement and support.

I owe special thanks to the staff-in-charge Prof. Joby george, Prof. Neethu Subash and Prof. Joby Anu Mathew for their corrections, suggestions and sincere efforts to co-ordinate the seminar under a tight schedule.

I express my sincere thanks to staff members in the Department of Computer Science and Engineering who have taken sincere efforts in helping me to conduct this seminar.

Finally, I would like to acknowledge the heartfelt efforts, comments, criticisms, co-operation and tremendous support given to me by my dear friends during the preparation of the seminar and also during the presentation without whose support this work would have been all the more difficult to accomplish.

ABSTRACT

Deploying electronic health records (EHR) is now an undisputable trend in healthcare systems. Storing sensitive information such as health records on the cloud incurs severe security and privacy risk. The work related to a novel cloud storage system for EHR which fully ensure the data privacy by employing Shamirs secret sharing. EHR is divided into multiple segments by a healthcare center, and the segments are distributed to numerous cloud servers. When retrieving the EHR, the healthcare center capture segments from partial cloud servers and reconstructs the EHRs. Meanwhile, in reality the reconstruction of a shared EHR could be much burdensome for a healthcare center or a patient, thus it proposes a practical cloud storage scheme which outsources the reconstruction of a shared EHR to a cloud computing service provider, and the result of outsourcing reconstruction can be verified by healthcare center or patients.

Contents

List of Figures

LIST OF ABBREVIATION

EHR	Electronic Health Record
HC	Health Record
CSP	Cloud Service Providers
ECDH	Elliptic Curve Diffie-Hellman
SLSE	Sparse Linear Systems of Equations
SCC	Secure Cloud Computing

Introduction

Over the past decade, the deployment of electronic health records in health care institutions has increased widely. Compared with the conventional health records,EHR shave many considerable advantages such as economy, normativity, efciency and accessibility. The electronic approach provides an easy and ubiquitous access to health data and enhances medical services and researches.The storage of EHRs should guarantee that massive data can be saved and accessed easily for each authorized health care center (HC) or patient. With the advance of information communication technology, the Internet of Things (IoT) is benetting our daily lives. To adapt the IoT applications, a cloud service provider offers rapid access to exible, low-cost resources. Known for these benets, cloud computing has been increasingly adopted in many elds, including health care. With the development of large-scale, on-demand, exible storing and computing infrastructures provided by cloud computing services, HCs could avoid the burden of data management, reduce the cost of storing massive data by themselves and achieve universal data access with location independence

EHRs are usually private and condential since they include patient identiers and highly sensitive information. However, when using the cloud computing services, users do not have physical control over their data. And cloud service providers are not completely trusted though the infrastructures under the cloud are much more reliable than personal computing devices.The curious clouds may have various incentives to be unfaithful toward the cloud users . They can either maliciously tamper the data or spy on some sensitive information. Therefore, cloud computing also brings security challenges while its advantages are appealing for EHRs storage .

There have been numerous approaches discussing data security and privacy protection issues in cloud computing environments .Encryption is a traditional method to protect the privacy of sensitive data stored in clouds. However, the storage of encryption key is a high-risk target and any negative incident may jeopardize all encrypted health records.Aiming to limit as in single point failure problem, schemes based on secret sharing are proposed Nonetheless, all

the existing cloud storage solutions for EHRs based on secret sharing have some common disadvantages:rst,the management of encrypted key is too complicated;second,it is not practical for patients or HCs to execute the expensive reconstruction of shared EHRs. The encrypted key is an obvious and weak target for the whole actual implementation regardless of what symmetric or public-key encryption algorithm is employed. Moreover, it demands tremendous time and resources of clients (patients or HCs) to perform the EHR reconstruction that involves time-consuming modular exponentiation operations, especially when a HC deal with a enormous amount of EHRs. In order to avoid the complicated key management problem, we plan to preprocess the original EHR before its shares are stored indifferent clouds.Also,we tend to outsource the EHR reconstruction to a cloud computing provider to reduce the local computation burden.

It is our essential goal to propose a practical cloud storage solution for EHRs by using secret sharing to solve the single point failure problem. Unavoidably though, the reconstruction of EHR can become cubersome for personal or health care hosts. As a response, we outsource the reconstruction operation to a cloud computing provider, which brings some other challenges. First, the cloud computing server might be curious on users sensitive data, so we have to make sure that cloud computing server executing the reconstruction outsourcing cannot obtain the original EHRs. Second, the cloud server might return incorrect results intentionally or unintentionally, so HCs have to be able to verify the correctness of the reconstructed EHR. Also, the recovery operation of the original EHR by client hosts should be simpler than the reconstruction operation. To address all the above challenges, we propose our cloud storage for electronic health records based on secret sharing with veriable reconstruction outsourcing.

Our contribution can be summarized as follows:

1) We introduce a cloud storage system architecture for EHRs which employs the secret sharing algorithm.

2) We propose a practical cloud storage scheme which satises the architecture. As far as we know, our scheme is the rst to have dened reconstruction outsourcing concept in all cloud storage schemes for EHRs based on secret sharing, which can signicantly improve the client side efciency.

Existing works

Generally the cloud storage solution for EHRs utilizes several cryptographic cloud solutions for data storage.However, the inherent properties of EHRs make it different from traditional cloud storage schemes. The design of cloud storage solutions for EHRs should consider unique access control policies and exceptions. Specically, the break the glass mechanism and a complete and widely-used health care system are needed to guarantee well-functional access of EHRs in any case.

In case of emergency, Break the glass mechanism is needed to protect the privacy and condentiality of EHRs. Previous work has shown break the glass mechanism has been abused. With such a mechanism in place, the threats in cloud storage solution for EHRs shift from unauthorized access to abuse a policy that allows any valid authenticated user to access any record. This fact indicates that the procedure of handling necessary privilege escalations must be considered at the design stage of an EHRs access control system, strictly enforcing audit processes to prevent abuse. Conversely, regular accesses to the records should not dilute the auditing mechanism to facilitate easier analysis of emergency escalations.

In our approach, we assume that all participants in the cloud storage solution for EHRs, including healthcare centers (HCs), cloud service providers (CSPs) and data owners (patients), have a common interest in securing the infrastructure and data against external adversaries (Note that the cloud computing server who helps user for the EHRs reconstruction is not considered as a fully-trusted party). All the participants need to register in healthcare system with valid

2.1 Secret sharing

Secret sharing (also called secret splitting) refers to methods for distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types, of shares are combined together; individual shares are of no use on their own. n one type of secret sharing

scheme there is one dealer and n players. The dealer gives a share of the secret to the players, but only when specific conditions are fulfilled will the players be able to reconstruct the secret from their shares. The dealer accomplishes this by giving each player a share in such a way that any group of t (for threshold) or more players can together reconstruct the secret but no group of fewer than t players can. Such a system is called a (t, n)-threshold scheme (sometimes it is written as an (n, t)-threshold scheme).

Blakley's scheme

Two nonparallel lines in the same plane intersect at exactly one point. Three nonparallel planes in space intersect at exactly one point. More generally, any n nonparallel (n 1)-dimensional hyperplanes intersect at a specific point. The secret may be encoded as any single coordinate of the point of intersection. If the secret is encoded using all the coordinates, even if they are random, then an insider (someone in possession of one or more of the (n 1)-dimensional hyperplanes) gains information about the secret since he knows it must lie on his plane. If an insider can gain any more knowledge about the secret than an outsider can, then the system no longer has information theoretic security. If only one of the n coordinates is used, then the insider knows no more than an outsider (i.e., that the secret must lie on the x-axis for a 2-dimensional system). Each player is given enough information to define a hyperplane; the secret is recovered by calculating the planes' point of intersection and then taking a specified coordinate of that intersection.

Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Aiming to limit a single point failure problem,schemes based on secretsharing were proposed utilized a secret sharing scheme to distribute the encryption key among a number of cloud nodes. Ermakova and Fabian proposed a scheme based on Shamirs(t,n)secret sharing which devided the encrypted EHRs into shares to be stored in different cloud service providers. This approach guaranteed less than t cloud service providers colluding to break the privacy cannot obtain EHRs, even if the encryption of EHRs is broken. Besides being used in EHRs cloud storage schemes, secret sharing is also adopted in many cryptographic schemes in cloud computing environment

Takahashi and Iwamura proposed a system named CloudStash,which applied the secret-sharingscheme directly on the le to store multi-shares of a le into multi-clouds. Compared with the traditional cloud storage, the proposed scheme achieved improvement on condentiality, availability, performance and fault tolerance. Yang and Lai proposed a new secret sharing scheme that can reduce the amount of shares, which is suitable for cloud systems, and they proved that the scheme is computationally secure.Alsolami and Boult designed the secure cloud computing (SCC) by employing Elliptic Curve Dife-Hellman (ECDH) and symmetric bivariate polynomial based secret sharing. Zhu et al.proposed a robust and simple N-Party entangled authentication cloud storage protocol based on secret sharing scheme.

2.2 Secure outsourcing

There have been a number of research efforts on securely outsourcing computations in the past decades. These research works mainly focused on the secure outsourcing algorithms for basic cryptographic operations and largescale scientic computation[1]s. The basic cryptographic operations are sometimes too expensive for resource-constrained devices. Many researchers investigated how to securely outsource cryptographic computations. Chaum and Pedersent rst brought up the idea of Wallet with Observers in 1992. It allowed a service provider install a piece of secure hardware on the clients device to carry out expensive computations for the client. In 2005, Hohenberger and Lysyanskaya [2] proposed the rst formal security model for outsourcing cryptographic computations.They also presented practical secure outsourcing schemes for modular exponentiation and CCA2-secure encryption. Besides basic cryptographic operation, scientic computing is another area that researchers focus on. These computations are usually complicated, and involve some sensitive data. In 2001, Atallah et al.rst presented a generic framework for secure outsourcing of scientic computations. However, the proposed framework cannot detect the misbehaviour of cloud server. Chen et al.leveraged the sparse matrix to develop a secure outsourcing algorithm for the large-scale linear equations. Their algorithm needs only one round communication between the client and server, and the proposed algorithm is able to detect them is behaviour of cloud server with probability 1. Salinas et al. [3] developed an efcient secure outsourcing algorithm for solving large-scales parse linear systems.

2.3 Cloud Security

Cloud consumers face security challenges from both external and internal attacks . Many of the security matters involved in protecting the cloud from external threats are related to those already facing large data centres. However, in the cloud, this information security responsibility is shared among many parties. These parties include the cloud user, the CSP, and any other service provider that consumers depend on for sensitive security software and configurations. The cloud consumer is in charge of application-level security. The CSP is in charge of physical security and applying external firewall policies. Security for the middle level of software load is distributed between the consumer and the CSP; the lower the levels of abstraction open to the consumer, the more the responsibilities that accompany it The consumer responsibility, in turn, can be subcontracted to other service providers who trade in special security services. The uniformity and standardised interfaces of systems platforms, example EC2, increases the possibility for an institution to provide services in configuration management and firewall-rule analysis. CSPs must guard against theft and denial-of-service attacks by consumers. Consumers must be protected from each other. There are some organisations and international bodies drafting cloud standards and application programming interfaces (APIs) [8]. Some of the risks that are seen by most consumers are that the CSP have to manage possibly millions of clients and this may present a challenge . This shows that many people are concerned that the CSPs will not be competent enough to manage the enormous scale of or that the infrastructure may not be able to balance correctly with enormous amounts of usage. Confidentiality and privacy is essential for institutions, especially when personal information or sensitive information is being kept. It is not yet entirely understood whether the cloud computing infrastructure will be capable to support the storage of sensitive information without making institutions responsible for breaking privacy regulations . It is believed that cloud authorisation systems are not tough enough. With a username and password, one is given access to the system. In many private clouds, users can have similar usernames, debasing the authorisation measures further. When sensitive information is stored on a private cloud, there is a high probability that somebody can view the information easier than many might believe. The client is counselled to only give their

data or use the CSP system if they trust them. Encryption can help secure health data but what come along with the benefits of encryption are the drawbacks as encryption can be processor exhaustive. Encrypting is not always the best for protecting data. Thus, combining different security measures to protect health data is the best way to guard sensitive data against unauthorised access and use. There can be times when little The resources of the cloud can also be misused as CSPs reassign IP addresses when a client no more needs the IP address. Once an IP address is no more needed by one client after a period of time, it then becomes accessible to another client to use. CSPs save money by reusing IP addresses. Many of these idle or used IP addresses can make the CSP open to misuse of its resources. Another client of the same CSP can possibly get access to another clients resources by routing through the CSPs networks, if no or little security measures are put in place. Data or information is like money for cyber criminals. Clouds can hold vast amounts of data and this is making clouds an attractive target for these cyber criminals. Therefore, cloud security must have a high standard and should not be ignored

Proposed scheme

3.1 System architecture

The notations of the proposed scheme are represented in Table 1.The system architecture is designed to store EHRs on different HCs using secret sharing. As shown in gure 1, the key components include:

1)The preprocessing of EHRs

2)Distribution of preprocessed EHRs

3)Reconstruction outsourcing

4)Verication and recovery of EHRs. Additionally, our cloud storage solution for EHRs handles necessary privilege escalations of EHRs access control to satisfy the break the glass mechanism. But the details of how to identify emergency situations are not discussed in this paper.We just concentrate on the key design components that are directly related with our designing goals of system architecture. We now describe the workow of the system.

Assuming that the EHRs are created by a healthcare center named HC A. After HC A uploads the EHR to a healthcare system, the healthcare system generates an unique identier for it, which is relevant to patients ID, HC A s ID, time stamp, etc. In the preprocessing phase, the healthcare system performs a bitwise exclusive OR operation between the EHR and its hash value. Then healthcare system distributes the preprocessed EHR into n shares using Shamirs threshold secret sharing algorithm and sends the n shares to n different cloud service providers (CP_1),..., (CP_n) according the protocols between healthcare institutions and (CP_s). When the owner of the EHR or an authorized healthcare center HC B wants to get the EHR, they send a request through the healthcare system. After conrming the r(CP_{re})equest, healthcare system assigns a cloud service provider $(CP_{re}^{'})$ to do the reconstruction operation. The assigned outsourcing cloud service provider (CP_{re}) ($(CP_{re}^{'})$) gets*t* or more shares from (CP_1),..., (CP_n) .After finishing the reconstruction,(CP_{re}) ($(CP_{re}^{'})$) returns the result s0 to HCB (or the patient) through healthcare system.Atlast,HCB (or the patient) can recover the original EHR with its

hash values to redin healthcare system by simply performing a bitwise exclusive OR operation. Inaddition , the validity of the recovered EHR can be veried. figure 3.1 illustrates the system architecture.

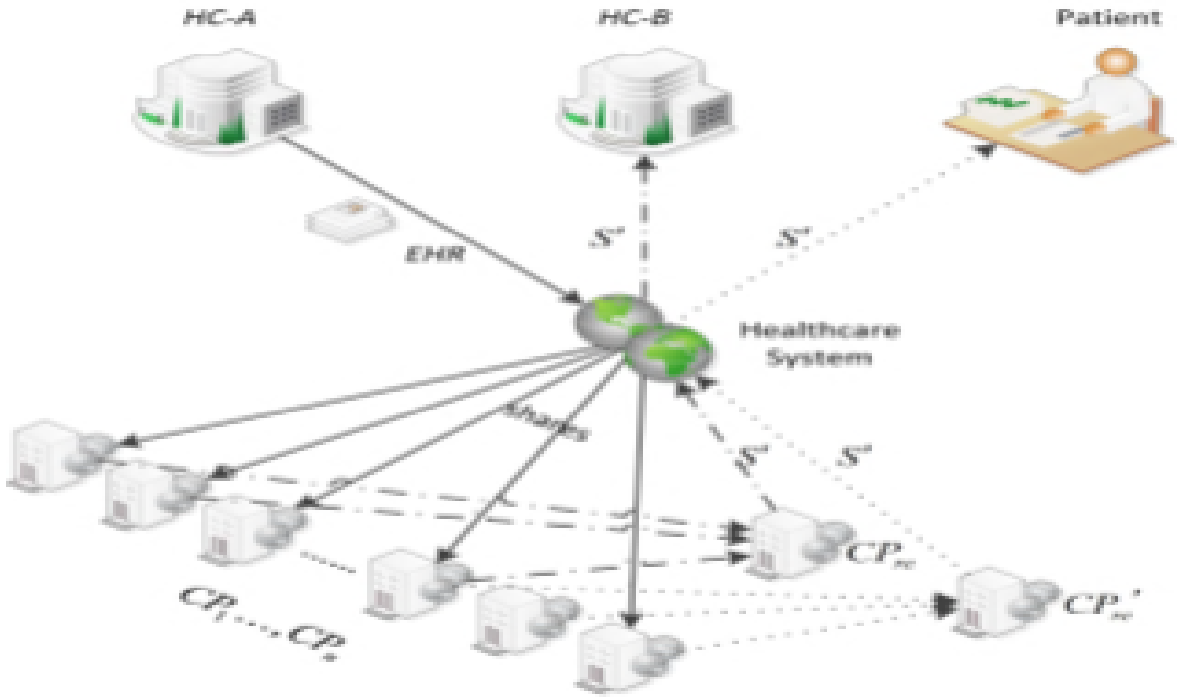


Fig. 3.1: System architecture.

To enable the privacy and condentiality of EHRs and practical performance of the cloud storage solution under aforementioned system architecture, our designed scheme should achieve the following security and performance goals: EHRs are distributed into shares that are stored on different CPs respectively, then the EHRs are accessible even if a few CPs break down.

The curious cloud or colluded clouds cannot acknowledge the real contents.

The reconstruction operation of shared EHRs is outsourced to a cloud computing provider, so that no complicated computation is needed for an authorized party, such as a healthcare center or the owner of the EHR, to reconstruct the EHRs.

Shamir's secret sharing

divide data D into n pieces in such a way that D is easily reconstructable from any k pieces[4], but even complete knowledge of k - 1 pieces reveals absolutely no information about D. This technique enables the construction of robust key management schemes or cryptographic schemes that can function securely and reliably even when misfortunes destroy half the pieces and security breaches expose all but one of the remaining pieces.

we generalize the problem to one in which the secret is some data D (e.g., the safe combination) and in which non mechanical solutions (which manipulate this data) are also allowed. Our goal is to divide D into n pieces D1, ..., Dn in such a way that:
(1) knowledge of any k or more Di pieces makes D easily computable.
(2) knowledge of any k - 1 or fewer Di pieces leaves D completely undetermined
Such a scheme is called a (k, n) threshold scheme.
Efficient threshold schemes can be very helpful in the management of cryptographic keys. In order to protect data we can encrypt it, but in order to protect the encryption key we need a different method (further encryptions change the problem rather than solve it). The most secure key management scheme keeps the key in a single, well-guarded location (a computer, a human brain, or a safe). This scheme is highly unreliable since a single misfortune (a computer breakdown, sudden death, or sabotage) can make the information inaccessible. An obvious solution is to store multiple copies of the key at different locations, but this increases the danger of security breaches (computer penetration, betrayal, or human errors). By using a (k, n) threshold scheme with n = 2k - 1 we get a very robust key management scheme: We can recover the original key even when floor(n/2) = k - 1 of the n pieces are destroyed, but our opponents cannot reconstruct the key even when security breaches expose floor(n/2) = k - 1 of the remaining k piece

In other applications the tradeoff is not between secrecy and reliability, but between safety and convenience of use. Consider, for example, a company that digitally signs all its checks (see RSA [5]). If each executive is given a copy of the company's secret signature key, the system is convenient but easy to misuse. If the cooperation of all the company's executives is necessary in order to sign each check, the system is safe but inconvenient. The standard solution requires at least three signatures per check, and it is easy to implement with a (3, n) threshold scheme.

Each executive is given a small magnetic card with one Di piece, and the company's signature generating device accepts any three of them in order to generate (and later destroy) a temporary copy of the actual signature key D. The device does not contain any secret information and thus it need not be protected against inspection. An unfaithful executive must have at least two accomplices in order to forge the company's signature in this scheme.

Threshold schemes are ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate. Ideally we would like the cooperation to be based on mutual consent, but the veto power this mechanism gives to each member can paralyze the activities of the group. By properly choosing the k and n parameters we can give any sufficiently large majority the authority to take some action while giving any sufficiently large minority the power to block it. *A Simple (k, n) Threshold Scheme* Given any subset of k of these Di values (together with their identifying indices), we can find the coefficients of q(x) by interpolation, and then evaluate D = q(O). Knowledge of just k - 1 of these values, on the other hand, does not suffice in order to calculate D.

make this claim more precise, we use modular arithmetic instead of real arithmetic. The set of integers modulo a prime number p forms a field in which interpolation is possible. Given an integer valued data D, we pick a prime p which is bigger than both D and n. The coefficients a[1], ..., a[k-1], in q(x) are randomly chosen from a uniform distribution over the integers in [0, p), and the values D1, ..., Dn are computed modulo p Let us now assume that k - 1 of these n pieces are revealed to an opponent. For each candidate value D' in [O, p) he can construct one and only one polynomial q'(x) of degree k - 1 such that q'(0) = D' and q'(i) = Di for the k - 1 given arguments. By construction, these p possible polynomials are equally likely, and thus there is absolutely nothing the opponent can deduce about the real value of D. Efficient O(n log2 n) algorithms for polynomial evaluation and interpolation are discussed in [11 and [3], but even the straightforward quadratic algorithms are fast enough for practical key management schemes. If the number D is long, it is advisable to break it into shorter blocks of bits (which are handled separately) in order to avoid multiprecision arithmetic operations. The blocks cannot be arbitrarily short, since the smallest usable value of p is n + 1 (there must be at least n + 1 distinct arguments in [0, p) to evaluate q(x) at). However, this is not a severe limitation since

sixteen bit modulus (which can be handled by a cheap sixteen bit arithmetic unit) suffices for applications with up to 64,000 Di pieces. Some of the useful properties of this (k, n) threshold scheme (when compared to the mechanical locks and keys solutions) are:

(1) The size of each piece does not exceed the size of the original data.

(2) When k is kept fixed, Di pieces can be dynamically added or deleted (e.g., when executives join or leave the company) without affecting the other D, pieces. (A piece is deleted only when a leaving executive makes it completely inaccessible, even to himself.)

(3) It is easy to change the Di pieces without changing the original data D – all we need is a new polynomial q(x) with the same free term. A frequent change of this type can greatly enhance security since the pieces exposed by security breaches cannot be accumulated unless all of them are values of the same edition of the q(x) polynomial.

(4) By using tuples of polynomial values as Di pieces, we can get a hierarchical scheme in which the number of pieces needed to determine D depends on their importance. For example, if we give the company's president three values of q(x), each vice-president two values of q(x), and each executive one value of q(x), then a (3, n) threshold scheme enables checks to be signed either by any three executives, or by any two executives one of whom is a vice-president, or by the president alone.

3.2 Assumption

The EHR file *R* uploaded to healthcare system is divided into *m* blocks

$$(b_1).....(b_i).....,(b_m) \in (Z_p) \quad \text{(Equ: 3.2.1)}$$

where *p*is a big prime. *H*(.) is a collision-resistant one-way hash function, which satises

$$H() : \{0,1\}^* \rightarrow (Z_p) \quad \text{(Equ: 3.2.2)}$$

For example the SHA-3 can be one candidate. Also note that the size of each block is equal with the hash size. As a result, the expression R$\bigotimes H(R)$

3.3 Proposal

The proposed cloud storage scheme for EHRs consists of four phases,namely the preprocessing phase,the distribution phase, the reconstruction outsourcing phase, and the recovery and verication phase. Before elaborating the proposed scheme, we rst give the denition of reconstruction outsourcing.

Reconstruction outsourcing is a processing method of reconstruction in a cloud storage solution based on secret sharing. Unlike a conventional way, the reconstruction of stored data in different cloud service providers is outsourced to a cloudcomputing provider,so that the computing resources of client hosts can be saved. In our case, the reconstruction outsourcing of preprocessed EHRs must make sure that the outsourcing cloud service provider cannot obtain any content of the EHRs during the reconstruction.

Without the loss of generality, we assume HC A is the generator of an EHR. We will show how the proposed cloud storage scheme works by taking the EHR generated by HC A as an example. HC A denes a policy for the storage and retrieval of the EHR before uploading it to the healthcare system.And the policy isused to guide the for(cp_s) the distribution and reconstruction of the EHR. For instance, the values of n and t are decided by the policy.

3.3.1 Preprocessing phase

The preprocessing operation of EHRs is executed by a healthcare system.After HCA uploads the EHR,denoted as a *R*=$\{(b_1)...(b_m)\}$ $((b_1)..,\in(b_m))\in(z_p)$ to the healthcare system, the healthcare system generates a unique identifier for the EHR and computes the hash value of R. The identifier and H(R) are both stored in the healthcare system. Then the healthcare system performs the preprocessing of R by making each block of R do the bitwise exclusive OR operation with H(R):

$$[s_1, ..., s_m] = R \bigotimes (R) \quad = [b_1, ..., b_m] \bigotimes H(R) = [b_1, ..., b_1] \bigotimes H(R) \qquad \text{(Equ: 3.3.1)}$$

$[s_1, \ldots, s_m]$ is the result of the preprocessing with R, each of which will be divided and distributed to *n* different cp_s by secret sharing. The bitwise exclusive OR operation protects the privacy of the EHR from the cloud service provider to execute the reconstruction outsourcing.

3.3.2 Distribution phase

The healthcare system is in charge of distribution of the preprocessed EHR.First, healthcare system selects *m* polynomials $\begin{bmatrix} f_s(x) = a_{10} + a_{11}x + \quad + a_{1t-1}x^t 1 mod p \\ f_m(x) = a_{m0} + a_{m1}x + \quad + a_{mt-1}x^t 1 mod p \end{bmatrix}$ where $\begin{bmatrix} a_{11}, \ldots, a1_{t-1} \\ a_{m1}, \ldots, a_{mt-1} \end{bmatrix} \in z_p$ and $[a_{10}, \ldots, a_{m0}] = [s_1, \ldots, s_m]$. Then the healthcare system computes n shares of $s1_i, \ldots, s_{mi}$(i = 1, . . . , n) and distributes them to $cp_1, \ldots, cp_n$ respectively. The shares $s1_i = f1_i$ (i), . . . ,$s_{mi} = f_{mi}$(i) and the identifier of the EHR are uploaded to CP by healthcare system. The identifier can be used to retrieve the preprocessed EHR when a reconstruction is needed

3.3.3 Reconstruction outsourcing phase

The reconstruction of a secret requires a massive amount of computations since it involves solving large-scale system of linear equations. The computation workload for a resource limited client is burdensome. To improve the efficiency on the client side, we propose the reconstruction

outsourcing scheme. The detailed process is discussed as follows:

Assuming another healthcare provider HC B needs to get the EHR. The healthcare system first verifies the authorization of HC B to check if it has the right to get the requested EHR. If the authorization passes, healthcare system outsources the reconstruction to a cloud service provider CP_{RE}. Notice that here we consider the CP_{RE} to be a curious and dishonest party, which is the strongest threat model. In other words, the cloud server has the potential to return incorrect computation results or steal useful information from the inputs. CPRE gets no less than t shares from CP_1, . . .CP_n , to reconstruct the preprocessed EHR, i,e.R. Without loss of generality, we assume CP_{RE} gets k shares:

[s_{11}.....s_{m1}

s_{1k}.....s_{mk}] (k$\geq t$)

from CP_1, . . . ,CP_k . CP_{RE}computes s_1, . . . , s_m using Lagrange interpolation polynomial and sends them to HCB Notic that by doing the bitwise exclusive OR operation with the hash value of R, the content of each block could be blinded. Consequently, although the cloud service provider to execute the reconstruction outsourcing recovered the [s_1, . . . , s_m] out of the data from . . . ,cp_n, he cannot reveal any information useful on the original EHR. Thus, reconstruction outsourcing process is secure against the curious cloud server.

3.3.4 Recovery verification phase

After receiving froms_1, . . . , $s0_m$ CP_{RE},and getting H(R) healthcare system,HC B can recover the result $R^{'}$) by calculating each block and then connecting them in series. Each block of $(R')$is (b'_l)=$(s_l^{'}) \bigotimes H(R) where l = 1....m$

b_1, . . . , b_m known, the EHR is recovered as $R^{'}$=b_1.....b_m

Then HC B checks the following equation: H((R') = H (R)

If it holds, the recovered $R^{'}$is the real original EHR. Otherwise,the recovered $(R'$ is not the real EHR and HC B reflects to healthcare system. The verification process ensures that both the cloud service providers to store the EHR and the cloud service provider to execute the reconstruction outsourcing behave honest.

Security analysis

We analyze the security of the proposed scheme based on the design goals of system architecture

*Theorem 1:*The EHRs stored on the cloud service providers under our proposed cloud storage scheme can be reconstructed with t or more valid shares.

prof: Our proposed cloud storage scheme for EHRs utilizes Shamirs (t, n) secret sharing algorithm to distribute shares of the preprocessed EHRs to different cp_s. According to Lagrange interpolation theorem, to reconstruct a polynomial with t 1 degree needs at last t points on the polynomial curve. Shamirs (t, n) secret sharing algorithm applies Lagrange interpolation polynomial to reconstruct the secret, and the shares are actually the points on the selected polynomial curve.

Theorem 2: case 1: The probability of leaking the EHRs to cp_s or cp_{RE} is negligible

prof: : The CPs are not colluded. When the cp_s are non-colluded, for each block of EHR, s_l, each cp only has one share of data, which is one *x, fl(x)* pair

case 2:t or more cp_s are colluded. When t or more cp_s are colluded, they can exchange shares they have. Consequently, for each block of the EHR, s_l, the cp_s can get at least t shares of data.The value of each s_l can be obtained by the cp_s. Notice that the preprocessing of EHRs in our proposed scheme uses the hash value H(R) to blind each block of R. To recover the EHRs, the CPs need the value of s_l and the value of H(R).Thus, the problem of recovering EHRs can be transformed to finding the hash value H(R), in which the probability is negligible. The same logic applies to the cp_{RE} executing the outsourcing reconstruction operation, it can not obtain the EHR with the preprocessing results reconstructed. Thus, the probability of leaking the EHRs to CPs or CPRE is negligible

Theorem 3:The result of reconstruction outsourcing process can be verified by the health center or patient *proof:* : In the recovery and verification phase of our proposed scheme, the result of reconstruction is verify by checking the equationH(R)= H(R). That means the correct results can pass the verification), the patient or the HC who requests the EHR can verify the

result reconstructed by $cp_{RE}$is valid or

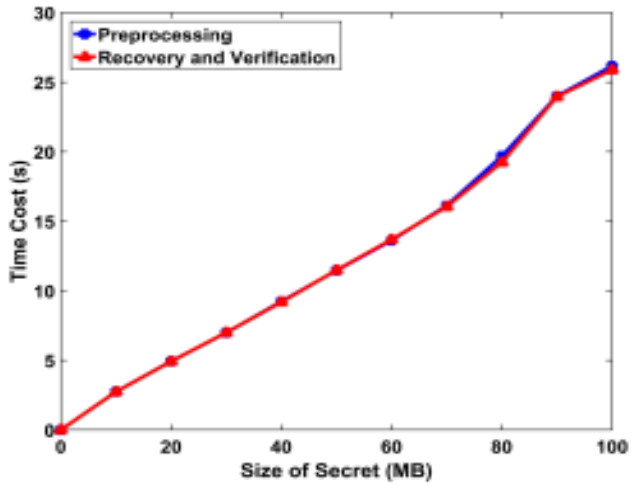
3.4 Performance evaluation

ted a performance evaluation to demonstrate the effectiveness of our proposed strategies. In the following, we first present evaluation methodology and then describe the evaluation results

To evaluate the practical efficiency of our proposed scheme, we developed an application with a user friendly interface, and conducted comprehensive experiments that simulate the outsourcing process. The experiment was carried out on the Windows 10 on Intel Pentium processor of 2.70 GHz with 4 GB memory. We implemented our proposed scheme in Python with secret sharing which is a free library for securely splitting secrets with Shamirs Secret Sharing Scheme. In our experiments, MD-5 is used as the hash function. Please kindly notice that we picked MD-5 for the experiments just for simplicity purpose. We conducted the proof of concept experiments to show the effectiveness of the proposed scheme.In our developed application, users are allowed to define the number of servers n that the secret is distributed to, which ranges from 20-200. The threshold t is set by users as well, which ranges from 20 to 200. The experimental results are calculated as the average value of 10 executions of the algorithm

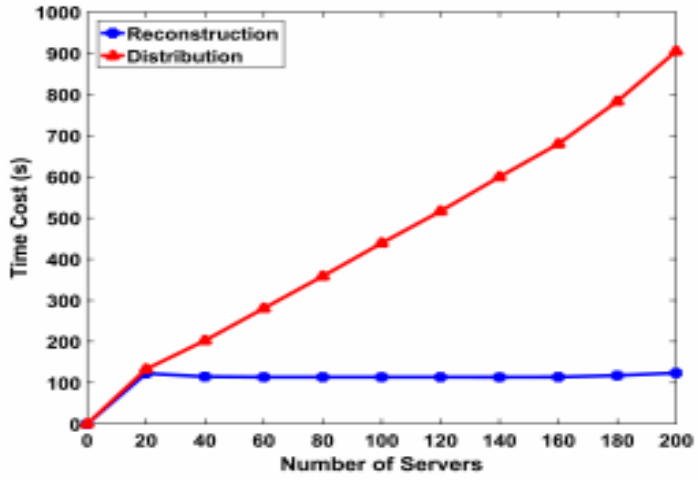
In the first experiment[5], we evaluate the preprocessing phase and the recovery and verification phase. Note that these two phases are irrelevant to the number of servers and the threshold. fig. 3.2 indicates the variation of the time cost versus the size of secret, which ranges from 20 to 200 MB. As we can observe from the figure, the time cost increases as the size of secret grows, which is obvious and sound. Also, there is no tmuch time cost difference between the preprocessing phase and the recovery and verification phase.

In our second attempt[6], we set the threshold t statically as Fig. 3.3 compares the client side time cost of the proposed scheme between with and without outsourcing. It shows the relationship between the time cost and the number of servers n. The client side operations include the reconstruction phase and the recovery and verification phase. Note that the preprocessing phase and the distribution phase only have to been executed once, so we does not take them



$e.png.$

Fig. 3.2: The time cost comparison among preprocessing, recovery and verification phase

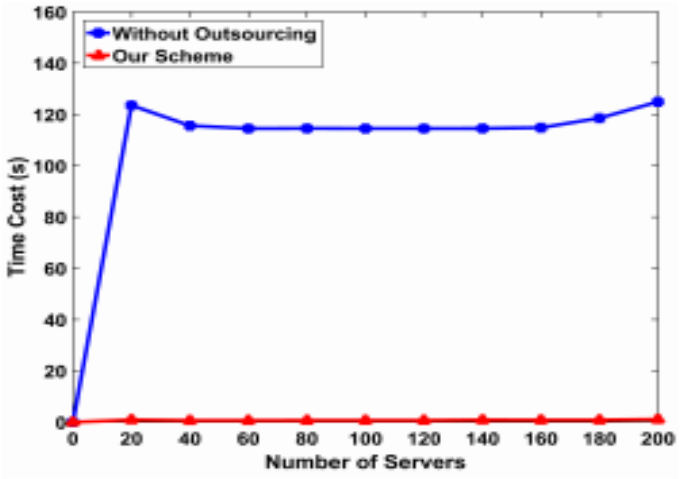


$.png.png$

Fig. 3.3: The time cost comparison between our scheme and reconstruction without outsourcing with static threshold

into concern when evaluating the client side time cost. As we can see from the figure, the time cost increases when the number of servers grows. In addition, the time cost of our outsourcing approach is much less than that without outsourcing. The reconstruction phase is the most time-consuming part since it involves solving large-scale system of linear equation compared with the reconstruction phase, the time cost of the recovery and verification phase is negligible. Without outsourcing, the client side needs to execute the reconstruction phase as well as the recovery and verification phase. In our scheme with reconstruction outsourcing, the client side only needs to perform the recovery and verification phase. Thus, the time cost of our outsourcing approach is much less than that without outsourcing, The reconstruction phase computes the results using Lagrange interpolation polynomial, which are lighter computations compared with modular exponentiations. Second, the time cost of distribution increases with the number of servers, while the time cost of the reconstruction remains invariable when the number of servers grows. In the distribution phase, the heath center needs to calculate n shares of the secret, so it is rational that the time cost increases with n. Meanwhile, the calculation in reconstruction phase is independent of the number of servers n

figure 3.3 compares the time cost between reconstruction phase and distribution phase. We have the following two observations: First, the distribution phase costs much more time than the reconstruction phase. The underling reason is that the distribution phase involves massive modular exponentiations when computing the shares, which are expensive computations. The reconstruction phase computes the results using Lagrange interpolation polynomial, which are lighter computations compared with modular exponentiations. Second, the time cost of distribution increases with the number of servers, while the time cost of the reconstruction remains invariable when the number of servers grows. In the distribution phase, the heath center needs to calculate n shares of the secret, so it is rational that the time cost increases with n. Meanwhile, the calculation in reconstruction phase is independent of the number of servers n. To present statistical results, we also evaluate the standard deviation of the data corresponding to fig. 3. As we can observe from table 4, the standard deviation for time cost without outsourcing is around 1, while the standard deviation for time cost of our scheme is around 0.1



.png.png

Fig. 3.4: The time cost comparison between reconstruction and distribution phases with static threshold.

In the third experiment, we set a static number of servers n to 200 and observe the variation of the time cost to the threshold t. figure 3.4 compares the client side time cost between with and without outsourcing. We have the following two observations from the figure. First, the time cost increases when the number of servers grows. In addition, the time cost of our outsourcing approach is much less than that without outsourcing[7]. figure 3.5 compares the time cost between reconstruction phase and distribution phase. As we can see, the distribution phase costs much more time than the reconstruction phase. The above analysis applies here as well. Also, the time cost of both distribution and reconstruction remains increasing when the threshold t grows. In the distribution phase, the heath center needs to calculate n shares of the secret. In each share, the number of modular exponentiations is decided by the threshold t, so it is rational that the time cost increases with t. For reconstruction phase, the scale of the system of equations is in connection with t, so the time cost increases with t.

We also evaluate the time cost of the scheme with variational file size. We set the number of servers as 100 and the threshold as 50, the file size ranges from 256 to 2048KB[8]. figure 3.5 shows the time cost comparison between with and without outsourcing. Note that these two curves show the subtotal time cost of reconstruction and verification. figure 3.5 compares the time cost between reconstruction phase and distribution phase. Our first observation is that the

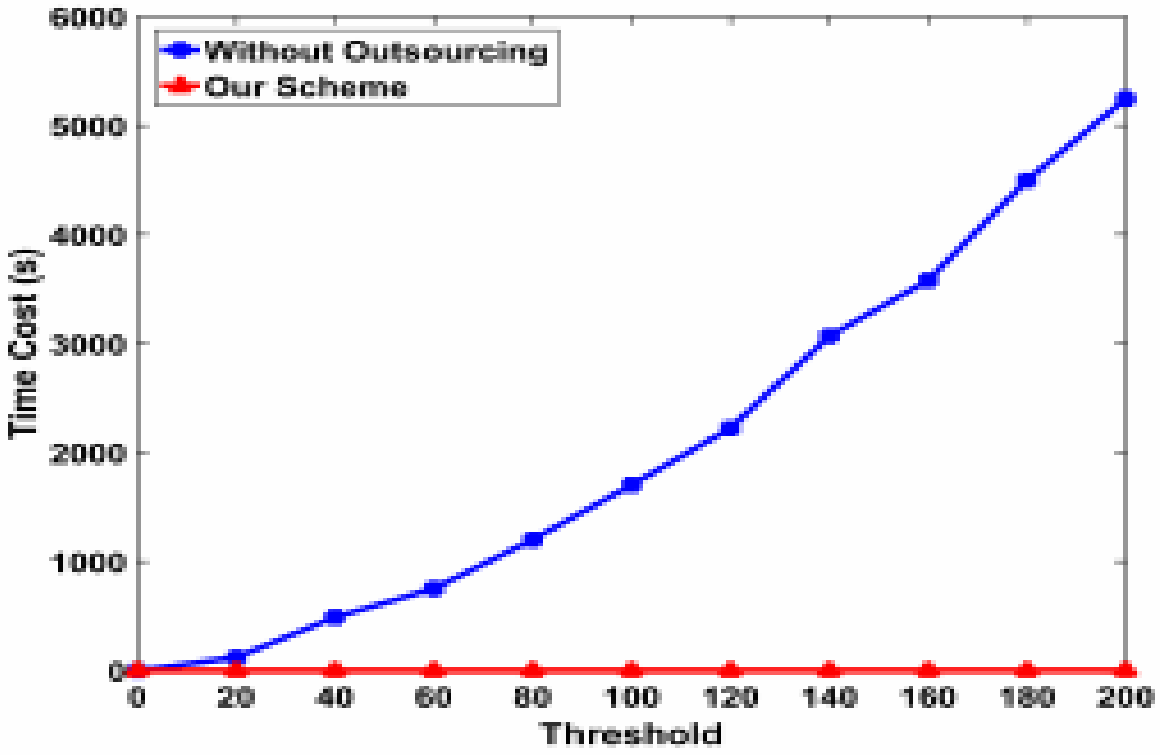


Fig. 3.5: The time cost comparison between our scheme and reconstruction without outsourcing with static number of servers.

distribution phase costs much more time than the reconstruction phase. And the above analysis on these two phases also applies here. Also, the time cost increases with the file size, which is obvious and sound.

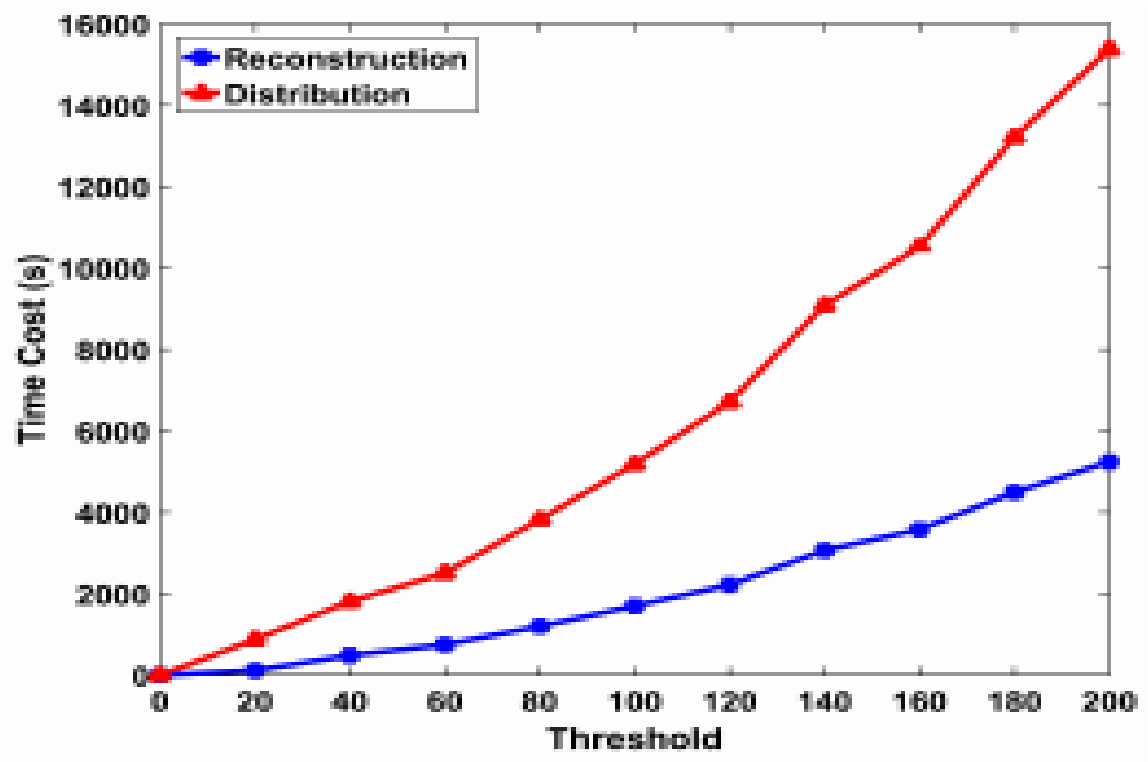
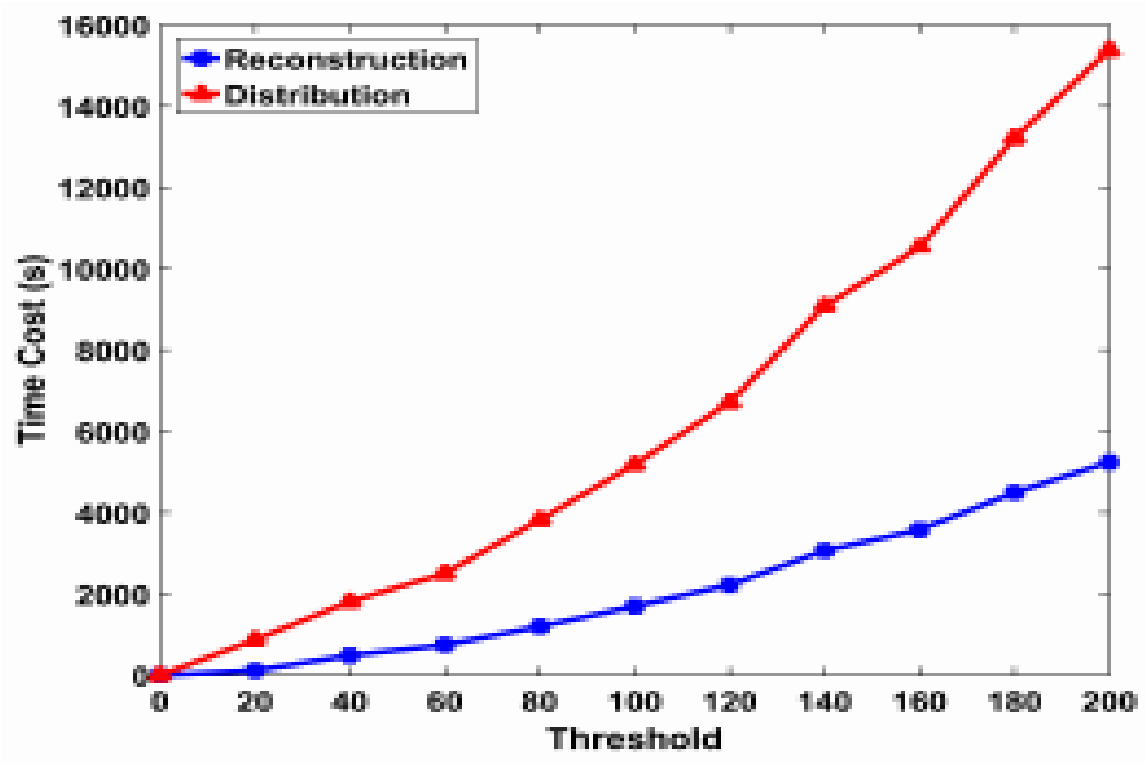


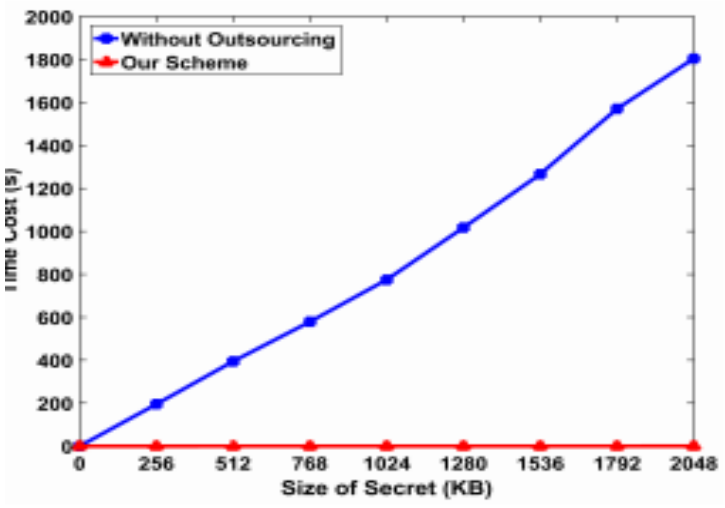
Fig. 3.6: The time cost comparison between reconstruction and distribution phases with static number of servers.

As a conclusion, both the distribution phase and the reconstruction phase are time consuming. Our proposed outsourcing approach effectively reduces the burden for the reconstruction phase. As for the distribution phase, since the user only needs to execute it once, we did not take it into concern when we design the scheme. Therefore, the outsourcing for distribution was not designed.



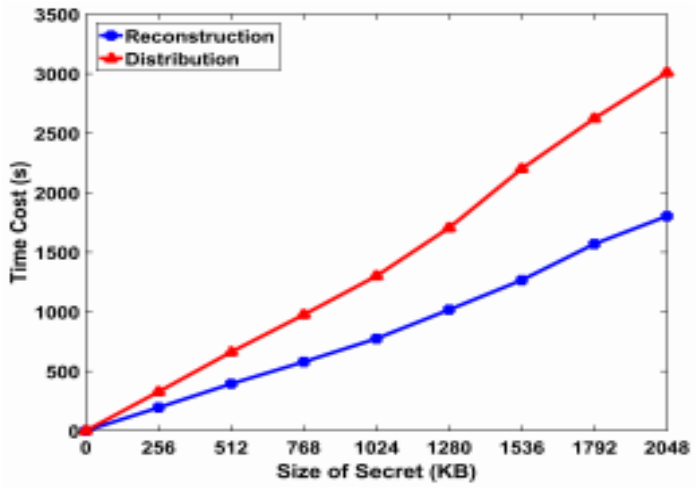
e.p

Fig. 3.7: The time cost comparison between reconstruction and distribution phases with static number of servers.



.pnge.pnge.png.

Fig. 3.8: The time cost comparison between our scheme and reconstruction without outsourcing with variational file size.



$.png.pnge.pnge.png.$

Fig. 3.9: The time cost comparison between reconstruction and distribution phases with variational file size.

Conclusion

Novel privacy-preserving cloud storage scheme for electronic health records based on Shamirs Secret Sharing. To address the problem that the reconstruction of shared EHR is burdensome for a healthcare center or a patient in a real-world application, we proposed a secure outsourcing approach for the secret reconstruction of shared EHR leveraging a computational powerful cloud service provider. Through theoretical analysis, we demonstrated that the proposed scheme satisfies the security requirements. We also conducted experiments on real documents, and the results show that, when our proposed reconstruction outsourcing approach is in place, the operation cost for healthcare centers and patients can be reduced significantly

REFERENCES

[1] F. Alsolami and T. E. Boult, CloudStash: Using secret-sharing scheme to secure data, not keys, in multi-clouds, in Proc. Int. Conf. Inf. Technol., New Gener., 2014, pp. 315320

[2] M. J. Atallah, K. N. Pantazopoulos, J. R. Rice, and E. E. Spafford, Secure outsourcing of scientific computations, Adv. Comput., vol. 54, pp. 215272, 2002.

[3] K. Wang, R. He, W. Wang, L. Wang, T. Tan, Learning coupled feature spaces for cross-modal matching, in: Proceedings of the IEEE International Conference on Computer Vision, 20882095, 2013.

[4] A. Shamir, How to share a secret, Commun. ACM, vol. 22, no. 11K.

[5] H. Lhr, A. R. Sadeghi, and M. Winandy, Securing the e-health cloud, in Proc. ACM Int. Health Inform. Symp., 2010, pp. 220229

[6] J. Wang, Y. He, C. Kang, S. Xiang, C. Pan, Image-text cross-modal retrieval via modality-specific feature learning, in: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, 347354, 2015.

[7]] Wang, R. He, L. Wang, W. Wang, T. Tan, Joint feature selection and subspace learning for cross-modal retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 38 (10) (2016) 20102023

[8] J. Gill, J. Alberto, J. A. L. Hinojosa, and I. Svecs, SYSTEM: Secure cloud storage, auditing, and access control for electronic health records, Dept. Comput. Sci., Univ. Illinois Urbana-Champaign, Champaign, IL, USA, Tech. Rep., 2012..