ISSN 2349-0292
Impact Factor 2.365

Global Journal of Advanced Engineering Technologies and Sciences

SECURING PERSONAL HEALTH RECORDS STORED IN THE PUBLIC CLOUD USING IMPROVED BLOWFISH ALGORITHM

Mrs.K.Vidhya*, S.Mohanapriya, C.Priyanka, V.Sangeetha
*M.E., (PhD) Assistant Professor (Sr.G), Dept of Computer Science and Engineering, KPR Institute of Engineering and Technology, Arasur, Coimbatore.
UG Students, Dept of Computer Science and Engineering KPR Institute of Engineering and Technology, Arasur, Coimbatore.

Abstract
Personal Health Record (PHR) service is the most important and emerging model for health information sharing. It allows the patients to create, manage, control and share their health information with other users like their family, friends and also the health care providers. In recent days, a PHR service is to be hosted by third-party cloud service provider's to enhance its performance. There have been privacy issues while outsourcing patient's health data to cloud servers, that is not only because cloud providers 8, but due to an enormous number of cloud data issues happened in lat few years. In this paper, we use blowfish algorithm in order to avoid these security issues. An elevated scale of patient's privacy is guaranteed. It provides the owners a full control over his/her health records.

Keywords: The Blowfish Algorithm, Feistel Networks & Encryption using improved Blowfish, Decryption using improved Blowfish.

Introduction
Cloud computing is a computing terminology or metaphor based on utility and consumption of resources and utilities like electricity, water system and software that allows data storage and to access computer services and resources online. In the foundation of cloud computing there is a broader concept of infrastructure, services and shared resources. Cloud typically provides model that is the amount paid is upto the amount used.

A Personal Health Record is a health record where health data and information related to the care of the patient is maintained by the patient. Patients can control the information in PHR and can get it anywhere at any time with internet access. The recent development of the cloud computing has lead to personal health record (PHR) that has driven a great attention of many researchers all over the world recently. PHR acts as the tool that is used used to collect, tracks and for sharing past and current data about personal health records. A personal health empowers people to manage the health records and share information between the visits. Personal health record enable people to track and assess their health and manage their health between families and get organized such as Tracking appointments, medication, and preventive measures or services. A study found that when the parents used personal health records for their children, the children were more likely to get their preventive well-child checkups on time.

In order to provide security for this health records we use blowfish algorithm. It is included in the class file of encryption symmetric cryptosystem, the method of encryption is similar to DES was created by a President Bruce Schneier Counterpane and published in the year 1994. This algorithm was developed to meet the criteria in the designing a fast implementation in which upto 26 clock cycles per byte, optimal condition is reached, can run on less than 5 KB of memory, and simpler to determine the security which the variable key length varies (varying lengths of 32 bits, 448 bits maximum, Multiples of 8 bits, 128 bits default). In Blowfish key does not change frequently in encrypted files for optimized application. Blowfish has been proved to be much faster than DES and other algorithms.

Literature Review
Revocable ABE is a most challenging issue to take back or withdraw attributes efficiently. This is frequently done by the broadcasting authority periodically by updating the key to users frequently, which does not achieve security and it is less efficient. Recently CP-ABE schemes with immediate attribute revocation capability, instead of periodically revoking. However, there is no design for MA-ABE [2].

In Authentication-Based PHR system, some of the PHR systems select an attribute-based access control (ABAC) scheme or a role-based access control (RBAC) scheme to manage user's access rights of the data. The above mentioned type of system normally ensures full trust on the cloud server where the PHRs are placed. A similar example of authentication-based PHR system is the Indivo X platform. Indivo is a open-source open standard personally controlled health record (PCHR) system that enables users to own and manage their personal health records. Indivo platform to provide the patients sharing ability to their records with doctors, pharmacists, hospitals and clinics while providing the properties of access control on the patient's health records. Access control mechanism is established by the Indivo server in accordance with Institutional policies and patient specified policies [1].

RSA algorithm is based on the issues of factorizing the large prime numbers that is the numbers that have 2 factors only. This system works based on the public and private key system in which the private key is made as secret key. The public key will be made visible to everybody as it is not made as the secret key. By using this key a user will be given rights to encrypt the data but not to decrypt the data, the person who will be able decrypt it is the one who has the private key. Even though this process is theoretically possible, it is a difficult process to extract public key for private key, which is major advantage to RSA algorithm that makes it more popular in the process of encrypting data [5].

In the enhanced method of data encryption : AES encryption, HIBE-based access control and TPA access control and TPA (third party auditing) with public data auditing the patients or the users can use their personal health record details from the database [6].Before outsourcing the PHR data into cloud sever. Using AES algorithm PHR owner will encrypt the PHR data. Then the original data can be transformed into encrypted data. This encrypted data can be transmitted in the cloud server. Directly we cannot find the original data from the encrypted data. In that case our system provides permission to the users such as physician, insurance company and users (friend). Usually many physicians are present in the hospital and many insurance companies who are in need of those data. Based on their requirements the user can query the cloud server and get the data. But user cannot query the original data directly from the cloud server.

Existing System

The rapid development of the cloud computing leads to personal health record (PHR) that has driven a great attention of many developers all over the world. Even though PHR is frequently outsourced to be stored at a third party server and thus has many security issues. Therefore, the study of secure and efficient Personal Health Record Scheme is used to protect user's privacy in PHR files that is of great significance. Personal health record (PHR) is an emerging model for personal health information sharing, that is often outsourced and that are to be stored at a third party servers, such as cloud providers. Though, there have been many privacy concerns not in order to expose to the third party severs and people who are unauthorized. In order to assure the users control over access to their own health records, it is a convincing method for encrypting the health records before outsourcing to the cloud sever. Yet, there are many issues regarding the privacy exposure, scalability in the management of key, access flexibility, and user revocation efficiency have been remained the most important challenges in achieving fine-grained access data control. Thus a novel patient-centric framework and a suite of mechanisms for data access control to PHRs stored in semi trusted TPA servers. In order to achieve fine-grained and scalable data access control for PHRs, we introduce blowfish algorithm techniques to encrypt every patient's PHR file and to ensure high security against the attacks. By improving this algorithm security is guaranteed and this method reuses key components [4].

In ABE, attributes of the users that selects the access policies which enables the patients to share their health details among the group of people, encrypting the file with a set of attributes, without the need to know the complete details of users [7].

The Blowfish Algorithm

Blowfish is a symmetric block cipher that is commonly used for encryption, to secure data and decryption. Blowfish was designed in 1993 by Bruce Schneier as a fast and easy for encryption of data. Blowfish Algorithm is a feistel network that is iterating a simple encryption function 16 rounds. The block size is 64 bits and the key size can be of variable length of 32 to 448 bits. Eventhough there is a complex initialization phase required before the encrypting the data, the data encryption process is an efficient one on large microprocessors with caches of larger size [8].

Feistel Networks

It is a recently used method for transforming any function called F converting into a permutation. It was invented by

Horst Feistel and has been used in designing a block cipher. A large proportion of block ciphers uses the encryption technique such as blowfish, Data Encryption Standard (DES). The Feistel structure has the major advantage that both the encryption and decryption of data are similar process, identical and requiring only a reversal of the scheduled key. Feistel networks and similar constructions are product ciphers, and this process also combines multiple rounds of repeated functions, such as:

1. Bit-shuffling also called as permutation boxes or P-boxes
2. Simple non-linear functions are often called substitution boxes or S-boxes
3. Linear mixing in the sense of modular algebra using XOR operation.

The working procedure of Feistel Network is as follows:

1. Splits each of the data blocks into two halves.
2. Right half of the data becomes the new left half.
3. New right half becomes the final result while the left half is XORed with the result by applying function f to the right half and the key.
4. Notice that the above rounds can be obtained even if the function f is not invertible.

The basic operation is as follows: Split the plaintext block into two equal parts, (L0, R0)
For each round of encryption, i=1, 2 …n. calculate
$Li = Ri - 1$
Where f is the round function and Ki is the sub-key. Then the cipher text is (Ln, Rn).Decryption is accomplished through,
$Ri - 1 = Li$

One of the advantages of this method is that the round function f used need not be invertible. Observe that the reversal of the sub key order is used for decryption of data that this is the only difference between encryption and decryption process. A Feistel cipher that is unbalanced uses a modified structure where left half and right half are not of equal lengths. The Skipjack encryption algorithm is one of the best examples of such a cipher. The Texas Instruments Digital Signature Transponder uses a Feistel cipher of proprietary unbalanced lengths to perform challenge-response authentication.

Blowfish has 16 rounds of encryption and decryption. Each round consists of a dependent key permutation and dependent data substitution. All operations are XORed and additions on 32-bit data. The only additional operations are four indexed array data lookups per round of encryption.

1. Sub keys: Blowfish algorithm uses a more number of sub keys. These keys should be precompiled before any of the data encryption or decryption process. The permutation P array consists of 18 number of 32-bit sub keys that are P1, P2... P18. It also four 32-bit substitution S-boxes with 256 entries such as S1,0,S1,1,..., S1,n; S2,0, S2,1,...,, S2,n; S3,0, S3,1,..., S3,n; S4,0, S4,1,...,, S4,n(n=0-255).
2. Encrypting and decrypting the data: Blowfish has 16 rounds of encryption and decryption of data. The 64-bit data element x is the input. Divide x block into two halves of 32-bit xL, xR. Then, the value for i = 1 to 16.

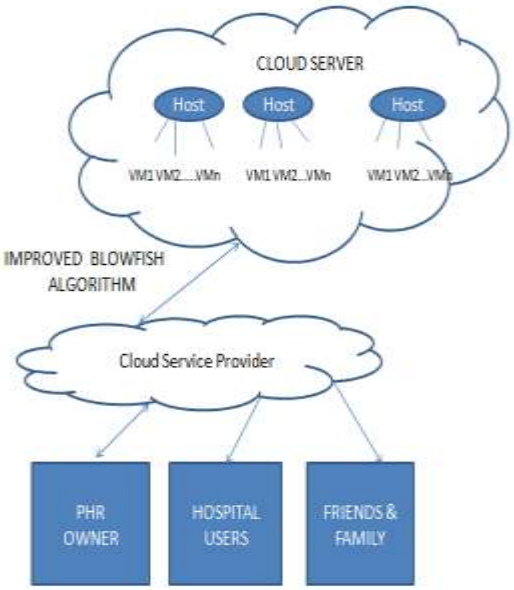
xL = xL XOR Pi
xR = F(xL) XOR xR And
Swap xL and xR

After completing all the sixteen rounds of encryption, swap xL and xR again to undo the last swap. Then, xR = xR XOR P17 and xL = xL XOR P18.
At last, recombine xL and xR to get the cipher text.
In Function F, divide xL into four eight-bit quarters such as a, b, c, and d. Then, F (xL) = ((S1, a + S2, b mod 232) XOR S3, c) + S4, d mod 232. Decryption is exactly the similar process as encryption, except that the sub keys P1, P2... P18 that are used in the reverse order [3].

Proposed System

The Personal Health Records are now a day maintained in a data server under the cloud infrastructure. A basic framework for secured sharing of personal health records has been proposed in this model. Access models such as the public and personal are designed with security and privacy concerned mechanism. The framework illustrates the most important challenges brought by the multiple PHR owners and users, in which the complexity of managing keys is greatly reduced. This System is enhanced to the model that supports dynamic policy management. The security and privacy concerns are maintained for the records. In future, to provide high security and privacy for Personal Health

Record (PHR) we enhance the blowfish algorithm



ARCHITECTURE

To assure the patients control over their access to their own PHR data's, it is a promising method to encrypt the PHRs before outsourcing to the third parties. Though, issues such as risks of privacy exposure, key management scalability, access flexibility, and revocation of user efficiently, are the most important challenges toward achieving fine-grained data access control. We propose a novel patient-centric framework and a suite of mechanisms for controlling data access to PHR dates stored in semi trusted servers. Inorder to achieve fine-grained and data access control scalability for PHRs, we introduce blowfish techniques to encrypt each patient's PHR data and by using enhanced Blowfish algorithm we ensure high security by overcoming those attacks. By enhancing this algorithm by increasing the security key sizes, guarantees high security and also this method reuses key computing components.

Architecture Flow

1. Cloud Architecture
2. Task creation
3. Encryption using improved improved Blowfish algorithm.
4. Decryption using improved Blowfish algorithm.
5. Performance Evolution.

1. Cloud Architecture

In this module cloud architecture is created. The user will create a user space for his access in cloud space provided by a concern. Creation of cloud architecture involves allocation of virtual machine which is an mediator between user and the actual service. In this module user creates his cloud space by specifying number of virtual machine needed to act on his behalf.

2. Task Creation

In this module, task is created. Since interval scheduling mechanism is used, each task requests service from the cloud platform with a specific start and finish time. Task which has been created is stored for allocation process.

3. Encryption using improved Blowfish

Encryption is the process that is used to encrypt a plain text into a cipher text. Cipher text is created along with a single key. The data is encrypted with a single key using random key generation method. Every time when the users encrypt the original data, each time a new key is generated. Storing data in a third party server does not provide confidentiality in cloud storage. Confidentiality of data is provided by re-encryption proxy method. This is used to generate more than 10,000 key at the time.

4. Decryption using improved Blowfish
Decryption is the same process but in the reverse order of encryption. Input given for decryption is cipher text and the output is plain text.

5. Performance Evolution
As the cloud performs various operations, it is On-demand revocation. Every time a user's attribute is no longer valid, the user should not be able to access future PHR files using that attribute.This process is commonly called as revocation attributes, and the corresponding property security is forward secrecy. There is also user revocation, where access privileges for all the users are revoked.

System Initialization
For the system initialization the administrator is responsible

1. Master key is created by the adminstrator.
2. This Master key is used to access the PHR files.

Consider the revocation of a user attributes/access data reader or privileges. There are many possible ways:

1. Revocation of one or more attributes of a public domain.
2. Revocation of a public user domain which is equivalent to revoking that entire user's attributes to improve efficiency.
3. Revocation of a personal domain user's access Privileges to the authorized user.
4. The PHR owner's client application can be initiated in a similar way.

User Registration

1. The users who are new are registered in the cloud.
2. With the registered user, the cloud architecture is created by allocating the virtual machines.
3. This private key is associated with the set of attributes.
4. The files can be decrypted by separate users if and only if they are similar to the set of attributes matches with the access policy.

The users can only decrypt the files if and the access policy with the set of attributes matches. The key strength is enhanced by increasing the key size 1024 from 512. Only the authorized user with the right key can able to access the file.

Generating The Subkeys

The sub keys are created using the Blowfish algorithm:

1. First initialize the P-array and then the four S-boxes, with a fixed string. The hexadecimal digits of pi is consisted in this string.
2. P1 is XORed with the first 32 bits of the key, P2 is XOR with the next 32-bits of the key, and consecutive process for all bits of the key. Until the entire P-array has been XORed with key bits repeatedly cycle through the key bits. Every smaller sized key, there will be at least one equivalent longer key. For example, if A is a smaller key of size 64-bit, then its equivalent keys are AA, AAA, etc., .
3. Encrypt the all-zero string with the blowfish encryption algorithm, using the sub keys in the above steps (1) and (2).
4. Replace the sub keys P1 and P2 with the output of step (3).
5. Output of step (3) is encrypted using the Blowfish encryption algorithm with the altered sub keys.
6. Replace the output of step (5) with P3 and P4.
7. Continue the same process, replacing all the entries of the P-array, and then all four S-boxes, with the output of Blowfish encryption algorithm.

There are 521 iterations in total that are required to generate the entire required sub keys. Rather than executing this derivation process multiple times the application can store the sub keys which reduce the time span [10].

Conclusion And Futher Idea

The process results shows that blowfish has a better performance than other encryption algorithms. Enhanced blowfish

is supposed to be better algorithm when compared to the original Blowfish Algorithm. By increasing the key size from 448 to 1024 bits security is highly ensured.This advance Blowfish Algorithm is more efficient that ensure confidentiality by increasing the key strength. In the new proposed model of Blowfish algorithm by further increasing the length of the key, Blowfish will provide the better results.

References

1. T.Baba,K. Jeevan Pradeep "Securing Personal Health Records in Cloud Server", August 2014.
2. Bharti Ratan Madnani, Sreedevi.N, "Attribute Based Encryption for Scalable and Secure Sharing of Medical Records in Cloud Computing Design and Implementation
3. Gurjeevan Singh, Ashwani Kumar, K. S. Sandha, "A Study of New Trends in Blowfish Algorithm", 2012.
4. Ming Li Member, IEEE, Shucheng Yu,Member, IEEE, Yao Zheng, Student Member, IEEE, Kui Ren, Senior Member, IEEE, and Wenjing Lou, Senior Member, IEEE "Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute based Encryption", 2012.
5. Nishitha Ramakrishnan, Sreerekha.B "Enhancing Security of Personal Health Records in Cloud Computing by Encryption", April 2015.
6. Priyanka Korde, Vijay Panwar, Sneha Kalse, "Securing Personal Health Records in Cloud using Attribute Based Encryption", April 2013.
7. Satheesh .K, Ram kumar.A,"Scalable And Secure Sharing Of Personal Health Records In Cloud Computing Using Multi Authority Attribute-Based Encryption", March 2014.
8. Saikumar Manku and K.Vasanth, "Blowfish encryption algorithm for information security",June 2015.
9. Shaheen Taj S.A, Prathibha Kiran, Elavarasi, "A Novel Method for Patient Centric Secure and Scalable sharing of PHR in Cloud Computing using Encryption", May-2013.
10. Tanjyot Aurora, Parul Arora," Blowfish Algorithm", 2013.