Volume: 10 Issue: XI Month of publication: November 2022

DOI: https://doi.org/10.22214/ijraset.2022.47256

Mobile Finger Print Verification and Automatic Log in Platform Using Blockchain

Dr. Sameena Banu[1], Bibi Hajra Umm E Hani[2]
[1]*Head of Department, Department of computer science and Engineering, Khaja Bandanawaz University, Gulbarga, Karnataka, India*
[2]*PG Student, Department of Computer Science and Engineering, Khaja Bandanawaz University, Gulbarga, Karnataka, India*

Abstract: Biometric identification has become increasingly popular in recent years. With the development of cloud computing, database owners are motivated to outsource the large size of biometric data and identification tasks to the cloud to get rid of the expensive storage and computation costs, which however brings potential threats to users' privacy. In this paper, we propose an efficient and privacy-preserving biometric identification outsourcing scheme. Specifically, the biometric data is encrypted and outsourced using Block Chain. To execute a biometric identification, the database owner encrypts the query data and submits it to the cloud. The Block Chain performs identification operations over the encrypted database and returns the result to the database owner. A thorough security analysis indicates the proposed scheme is secure even if attackers can forge identification requests and collude with the cloud.

I. INTRODUCTION

With the growth of Technology, the efficient and privacy-preserving is the major issue so here by the use of Biometric identification for an individual verification. The detailed security analysis shows that the proposed scheme can achieve a required level of privacy protection. Specifically, our project is to secure under the biometric identification outsourcing model and can also resist the attack proposed by the proposed system. Compared with the existing biometric verification schemes, the performance analysis shows that the proposed work provides a lower computational cost in both preparation and identification procedures. An efficient and privacy preserving biometric identification scheme which can resist the collusion attack launched by the users. Attackers can only observe the encrypted data stored in the cloud. In order to avoid, the well-known cipher text-only model has been implemented. The integration of blockchain technologies with the fingerprint verification along with the enhancement of blockchain features, improves security . The blocks contain data, hash, random number, previous block, timestamp, transactions, and each block is intertwined so that it is impossible to interconnect the data found in the block chain difficult to encrypt. The information is stored in the cloud to protect the user's data so that it is not easily hacked.

II. LITERATURE SURVEY

Title: On Implementing Deniable Storage Encryption for Mobile Devices.Authors: Adam Skillen and Mohammad Mannan. Year: Feb. 2013 Data confidentiality can be effectively preserved through encryption. In certain situations, this is inadequate, as users may be coerced into disclosing their decryption keys. In this case, the data must be hidden so that its very existence can be denied. Stefano graphic techniques and deniable encryption algorithms have been devised to address this specific problem. Given the recent proliferation of smart phones and tablets, we examine the feasibility and efficacy of deniable storage encryption for mobile devices. We evaluate existing, and discover new, challenges that can compromise plausibly deniable encryption (PDE) in a mobile environment. To address these obstacles, we design a system called Mobiflage that enables PDE on mobile devices by hiding encrypted volumes within random data on a device's external storage. We leverage lessons learned from known issues in deniable encryption in the desktop environment, and design new countermeasures for threats specific to mobile systems. Key features of Mobiflage include: deniable file systems with limited impact on through-put; efficient storage use with no data expansion; and restriction/prevention of known sources of leakage and disclosure. We provide a proof-of-concept implementation for the Android OS to assess the feasibility and performance of Mobiflage. We also compile a list of best practices users should follow to restrict other known forms of leakage and collusion that may compromise deniability. However it includes some of the drawbacks:Efficient storage use with no data expansion and restriction/prevention of known sources of leakage and disclosure.

Title: Data leakage mitigation for discertionary access control in collaboration clouds Authors: Qihua Wang, Hongxia Jin Year: Jun.

2011 With the growing popularity of cloud computing, more and more enterprises are migrating theircollaboration platforms from in-enterprise systems to Software as a Service (SaaS) applications. While SaaS collaboration has numerous advantages, it also raises new security challenges. In particular, since SaaS collaboration is increasingly used across enterprise boundaries, organizations are concerned that sensitive information may be leaked to outsiders due to their employees' inadvertent mistakes on information sharing. In this article, we propose to mitigate the data leakage problem in SaaS collaboration systems by reducing human errors. Built on top of the discretionary access control model in existing collaboration systems, we have designed a series of mechanisms to pro-vide defense in depth against information leakage. First, we allow enterprises to encode their organizational security rules as mandatory access control policies, so as to impose coarse-grained restrictions on their employees' discretionary sharing decisions. Second, we design an attribute-based recommender that suggestsand prioritizes potential recipients for users' files, reducing errors in the choices of recipients. Third, our system actively examines abnormal recipients entered by a file owner, providing the last line of defense before a file is shared. We have implemented a prototype of our solution and performed experiments on data collected from real world collaboration systems. The growing popularity of SaaS collaboration raises new security challenges. However it includes some of the drawbacks: Abnormal recipients entered by a file owner, and their they have to providing the last line of defense before a file is shared.

3. Title: How to build a trusted database system on untrusted storage. Authors: Maheshwari U, Vingralek R, Shapiro W Year: 2000 Some emerging applications require programs to maintain sensitive state on untrusted hosts. This paper presents the architecture and implementation of a trusted database system, TDB, which leverages a small amount of trusted storage to protect a scalable amount of un- trusted storage. The database is encrypted and validated against a collision-resistant hash kept in trusted storage, so untrusted programs cannot read the database or modify it undetectably. TDB integrates encryption and hashing with a low-level data model, which protects data and metadata uniformly, unlike systems built on top of a conventional database system. The implementation exploits synergies between hashing and log-structured storage. Preliminary performance results show that TDB outperforms an off-the-shelf embedded database system, thus supporting the suitability of the TDB architecture. The party hosting the database storage has the opportunity to alter its state for unauthorized benefits. For example, a consumer could save a copy of the local data-base, purchase some goods, then replay the saved copy ,thus eliminating payments for the purchased goods. It is difficult to secure a trusted program and its data-base because the hosting party ultimately controls the underlying hardware and the operating system. How-ever, a number of emerging trusted platforms provide a processing environment that runs only trusted programs and resists reverse engineering and tampering. However it includes some of the drawbacks: It is difficult to secure a trusted program and its data-base because the hosting party ultimately controls the underlying hardware and the operating system.

III. SYSTEM ARCHITECTURE

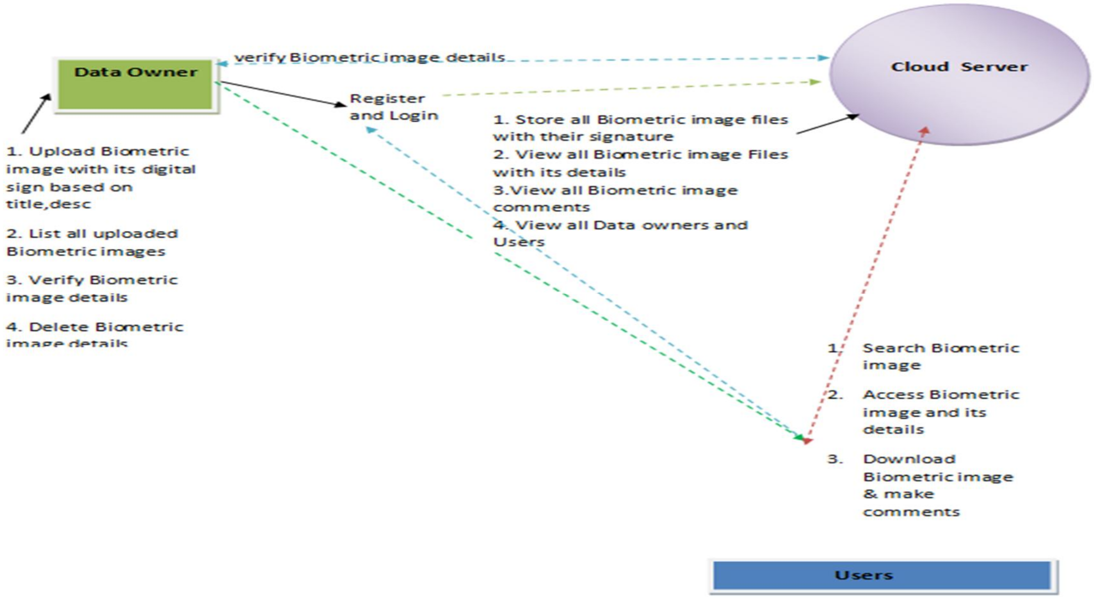
verify Biometric image details
Data Owner
Register and Login
Cloud Server
1. Store all Biometric image files with their signature
2. View all Biometric image Files with its details
3.View all Biometric image comments
4. View all Data owners and Users
1. Upload Biometric image with its digital sign based on title,desc
2. List all uploaded Biometric images
3. Verify Biometric image details
4. Delete Biometric image details
1. Search Biometric image
2. Access Biometric image and its details
3. Download Biometric image & make comments
Users

Figure 1:Architecture Diagram of Proposed Work

The figure 1 show the Architecture Diagram of the proposed system where the all the interfaces between the User, Data Owner, Cloud are been displayed and explained below.

1) *Data Owner:* In this module, the data owner uploads their Biometric images with their contents data to the Cloud server. For the security purpose the data owner assigns the digital sign and then store in the Cloud and also performs the following operations such as Upload Biometric image with its digital sign based on title, desc, List all uploaded Biometric images, Verify Biometric image details, and Delete Biometric image details
2) *Cloud Server:* The Cloud service provider manages a Cloud to provide data storage service. And performs the following operations such as Store all Biometric image files with their signature, View all Biometric image Files with its details, View all Biometric image comments, View all Data owners and Users, and View all attackers
3) *Users:* The Cloud User who has a large amount of data to be stored in Cloud Servers and have the permissions to access and manipulate stored Biometric image and its data. The consumer will search the data and accessing the Biometric image data if he is authorized and performs the following operations such as Search Biometric image , Access Biometric image and its details, Download Biometric image & make comments

As shown in Fig.2 three types of entities are involved in the system including the database owner, users and the cloud. The database owner holds a large size of biometric data (i.e., fingerprints, irises, voice, and facial patterns etc.), which is encrypted and transmitted to the cloud for storage. When a user wants to identify himself/herself, a query request is be sent to the database owner. After receiving the request, the database owner generates a cipher-text for the biometric trait and then transmits the cipher- text to the cloud for identification. The cloud server figures out the best match for the encrypted query and returns the related index to the database owner. Finally, the database owner computes the similarity between the query data and the biometric data associated with the index, and returns the query result to the user.

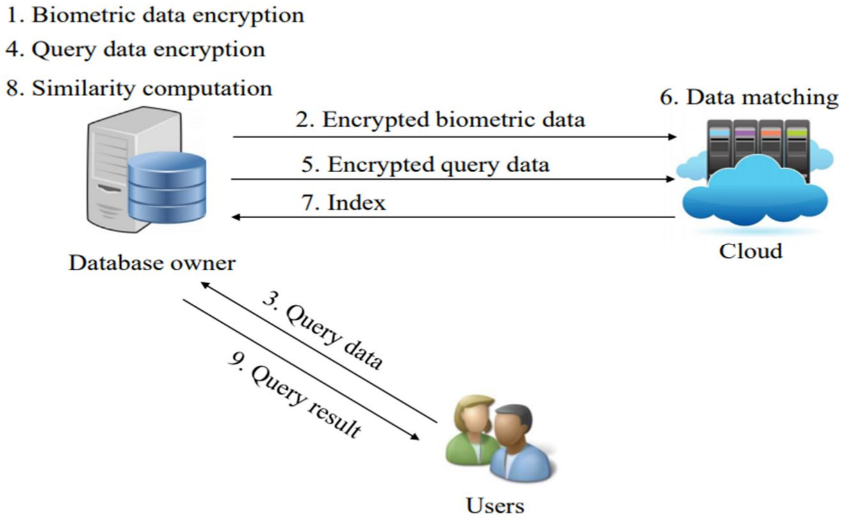

Figure 2: Work flow diagram.

The SHA256 algorithm is used to hash the public information. A digital fingerprint is created by applying a hash to the file and then using an encryption algorithm to encrypt it via private and public keys. The private key is used when the fingerprint of owner signs the document; the public key is used by the recipient to decrypt the message on their end. But all of that would mean nothing without verification, which is where SHA-256 comes in. Hashing ensures that the digital fingerprint hasn’t been altered since it was signed or register .The recipient’s system runs the hashing algorithm on its end and uses the public key to decrypt the message. If it matches, then it knows the data is unaltered and authentic. In our scheme, we assume that the biometric data has been processed such that its representation can be used to execute biometric match. Without loss of generality, similar , we target fingerprints and use Finger Codes to represent the fingerprints. More specifically, a Finger Code consists of n elements and each element is a l-bit integer (typically n = 640 and l = 8). Given two Finger Codes x = [x1, x2, · · · , xn] and y = [y1, y2, · · · , yn], if their Euclidean distance is below a threshold , they are usually considered as a good match, which means the two fingerprints are considered from the same person

IV. IMPLEMENTATION

The development of the project is divided into several stages, allowing developers to easily build software. One of the most important stages is the realization stage. The programmer is involved in coding purposes, and the designer is involved in the design of the front view. Which language and architecture should be used for programming purposes. At the end of this phase, all requirements will be verified, regardless of whether they are covered or not. Eclipse Environment (IDE) is used to develop applications. Java is used as a programming language.

1) *Data Owner:* In this module, The database owner holds a large size of biometric data (i.e., fingerprints, images and facial patterns etc.), which is encrypted and transmitted to the cloud for storage. When a user wants to identify himself/herself, a query request is be sent to the database owner. After receiving the request, the database owner generates a cipher text for the biometric trait and then transmits the ciphertext to the cloud for identification. the data owner uploads their Biometric images with their contents data to the Cloud server. For the security purpose the data owner assigns the digital sign and then store in the Cloud and also performs the following operations such as Upload Biometric image with its digital sign based on title, desc, List all uploaded Biometric images, Verify Biometric image details, and Delete Biometric image details.
2) *Cloud Server:* The Cloud service provider manages a Cloud to provide data storage service .When the cloud receives the cipher text from the owner it first cloud server figures out the best match for the encrypted query and returns the related index to the database owner. Here just the Cloud storage is a model of computer data storage in which the digital data is stored.And performs the following operations such as Store all Biometric image files with their signature, View all Biometric image Files with its details, View all Biometric image comments, View all Data owners and Users, and View all attackers.
3) *User:* The Cloud User who want to access a large amount of data from the owner is stored in Cloud Servers and have the permissions to access and manipulate stored Biometric image and its data. The users first need to get authorized by the server the query request is send to the owner then from the owner it is send to the cloud when he/ she gets authorized then consumer will search the data and accessing the Biometric image data and performs the following operations such as Search Biometric image , Access Biometric image and its details, Download Biometric image & make comments

V. CONCLUSION

We have designed a new encryption and cloud authentication certification to understand the efficiency and secure requirements. The thorough review reveals that possible threats can be resisted. In addition, we also showed by performance reviews that the proposed framework satisfies the need for efficacy well. Also the outsource this computationally expensive scanning while protecting the privacy of both the database and the computation. Exploiting the inherent structures of biometric data and the properties of identification operations, we first present a privacy-preserving biometric identification scheme which uses a single server.

VI. FUTURE SCOPE

In the future its can be applied to a scheme which can vulnerable to an attacker who operates with a cloud server by introducing statistical-inference attack algorithm, also it can further consider its extensions in the two-server model. It achieves a higher level of privacy than our single-server solution assuming two servers are not colluding. A part from somewhat homomorphic encryption, our second scheme uses batched protocols for secure shuffling and minimum selection.

REFERENCES

[1] On Implementing Deniable Storage Encryption for Mobile Devices Adam Skillen and Mohammad Mannan Concordia Institute for Information Systems Engineering Concordia University, Montreal, Canada {a skil, mmannan}@ciise.concordia.ca .Related from https://users.encs.concordia.ca/~mmannan/publications/mobiflage-ndss2013.pdf

[2] Higgins S (2016) Hours after launch, OpenBazaar sees first drug listings. CoinDesk. Retrieved from http://www.coindesk.com/drugs-contraband-openbazaar/. Accessed 1 June 2018 Data leakage mitigation for discretionary access control in collaboration clouds January 2011

[3] How to Build a Trusted Database System on Untrusted Storage.January 2000,Author Maheshwari u, Vingralek R ,Shapiro Retrived from https://www.researchgate.net/publication/220851794_How_to_Build_a_Trusted_Database_System_on_Untrusted_Storage

[4] Attribute-based fine-grained access control with efficient revocation in cloud storage systems. Authors: Kan Yang, XiaohuaJia, KuiRen Year: 2013 Retrived from https://ece.uwaterloo.ca/~kan.yang/paper/C10-AsiaCCS.pdf

[5] Secured Electronic Voting System Using the Concepts of Blockchain Authors: Sudharsan B ; Rishi Tharun V ; Nidhish Krishna M P ; Boopathi Raj J ; Surya Arvindh M ; Dr. M. Alagappan Year:Oct. 2019 . Retrived from https://ieeexplore.ieee.org/document/8936310

[6] Fully secure key-policy attribute-based encryption with constant- size ciphertexts and fast decryption. Authors: Junzuo Lai, Robert H. Deng ,YingjiuLi ,et al Year: Jun. 2014 Retrived from https://www.researchgate.net/publication/266656648_Fully_secure_key-policy_attribute-based_encryption_with_constant-size_ciphertexts_and_fast_decryption

[7] SDSM: a secure data service mechanism in mobile cloud computing. Authors: Jia W, Zhu H, Cao Z, et al Year: 2011 Retrived from https://www.researchgate.net/publication/224243124_SDSM_A_secure_data_service_mechanism_in_mobile_cloud_computing

[8] Automatic Mediation of Privacy-Sensitive Resource Access in SmartphoneApplications. Authors: Benjamin Livshits, Jaeyeon Jung Year: Aug. 2013.Retrived from https://www.researchgate.net/publication/261849235_Automatic_mediation_of_privacy-sensitive_resource_access_in_smartphone_applications