

Cross-Platform Secure Data Sharing through Web-Based Cloud Storage

¹K.Jaya Krishna, ²Shaik.Suhana,

¹Associate Professor, Department of Master of Computer Applications,
QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

²PG Scholar, Department of Master of Computer Applications,
QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

ABSTRACT:With an ever increasing number of information moving to the cloud, protection of client information have raised incredible worries. Client-side encryption/decoding is by all accounts an appealing answer for safeguard information security, be that as it may, the current arrangements experienced three significant difficulties: low security as a result of low-entropy PIN encryption, inconvenient data sharing as a result of traditional encryption algorithms, and poor usability as a result of specialized software and plugins that require particular kinds of terminals. Utilizing cutting-edge Web technologies, this work develops and implements WebCloud, a practical browser-side encryption solution. It accomplishes several remarkable feats in addition to resolving all three of the aforementioned issues: strong and prompt client denial, quick information handling with disconnected encryption and re-appropriated decoding. Our solution, in particular, is usable on any device that has a Web user agent installed, including desktop and mobile applications as well as

Web browsers. We use WebAssembly and the Web Cryptography API to integrate complex cryptographic operations, and we implement WebCloud, a file management utility, based on ownCloud. Finally, extensive tests with a wide range of popular browsers, Android applications, and PC software demonstrate that WebCloud is cross-platform and effective. WebCloud's design naturally includes a dedicated and useful ciphertext-policy attribute-based key encapsulation mechanism (CP-AB-KEM) scheme, which can be useful in other applications. This is an interesting byproduct..

1.INTRODUCTION

The public cloud storage service is becoming more and more popular because it is less expensive and makes it easy for users to use their data. This pattern has provoked clients and companies to store (decoded) information on open cloud, and offer their cloud information with others. The user must have faith in the server's ability to safeguard high-value data when

..ISSN: 2040-0748

using the cloud. This trust is frequently misplaced due to the numerous methods by which confidential data can be leaked, as evidenced by the data breaches that have been reported [1], [2], [3], [4], [5], and [6]. To check information spillage, one of the most encouraging methodologies is client-side encryption/unscrambling. Specifically, senders can encrypt data before sending it to clouds and decrypt it after downloading it from clouds with client-side encryption. Clouds can only access encrypted data, making server-side data exposure more challenging or even impossible. At the same time, flexible file sharing between multiple users or a group of users is an essential feature of cloud storage and must be fully supported. However, the security, efficiency, and ease of use of the currently available client-side encryption solutions vary. Common methods for client-side encryption. We examine existing solutions and highlight their drawbacks.

Insufficient support or none at all. Client-side encryption is not supported by many cloud storage providers, such as Google Drive and Drop Box. They use two-factor authentication for user authentication, server-side encryption for stored files, and TLS for data in transit. End-to-end encryption is available for sensitive data stored in Apple I Cloud, such as Wi-Fi passwords and I Cloud Keychain. Server

encryption is used only for other data that is uploaded to I Cloud.

Solutions That Rely on Passwords Users' data is encrypted using symmetric encryption, typically AES, in some products, which then upload ciphertexts to clouds. The cryptographic keys, on the other hand, are derived from a password or passphrase or even a 4-digit PIN in these schemes. It is dangerous to rely on such low entropy [10]. Worse still, the majority of password-based solutions only support file encryption and decryption for a single user and do not offer a file sharing feature. Particularly noteworthy is that [7] permits users to generate a share link for each password-protected file. In any case, clients should physically send the offer connection through one channel, and secret key to all recipients through another solid channel, which is badly designed and fragile.

Scheme for Hybrid Encryption The KEM-DEM setting is made up of a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM). The RSA-AES paradigm is used by many public cloud service providers, such as Mega [13], Tresor it [11], and Amazon [12]. Users create RSA key pairs and apply for certificates from providers, who build and maintain PKIs. Data is encrypted using fresh sampled AES keys that are then encrypted using the RSA public keys of all

..ISSN: 2040-0748

recipients. This record sharing component is unyielding and wasteful. During encryption, a sender must obtain and specify the public keys of all recipients. Even worse, the number of recipients and the size of the cipher text and encryption workload are inversely proportional, increasing bandwidth and storage costs and user expenditure.

Problems with the Currently Available Solutions Three disadvantages exist in previously mentioned arrangements: 1) similarly unfortunate security, 2) coarse-grained admittance control, unyielding and wasteful document sharing, and 3) unfortunate convenience. The initial two are not difficult to see and we presently elaborate the convenience issue. When uploading files, users typically use a variety of terminals, including desktop, Web, and mobile applications [14]. However, almost all of the currently available solutions necessitate additional software or plugins, restricting users' platforms and devices. Users must repeat the tedious installation process when switching to a new device, which significantly increases user burden and reduces usability..

2.LITERATURE SURVEY

2.1 Ciphertext-policy attribute based encryption

AUTHORS: J. Bethencourt, A. Sahai, and B. Waters

In several distributed systems a user should only be able to access data if a user possesses a certain set of credentials or attributes. Currently, the only method for enforcing such policies is to employ a trusted server to store the data and mediate access control. However, if any server storing the data is compromised, then the confidentiality of the data will be compromised. In this paper we present a system for realizing complex access control on encrypted data that we call Ciphertext-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover, our methods are secure against collusion attacks. Previous Attribute-Based Encryption systems used attributes to describe the encrypted data and built policies into user's keys; while in our system attributes are used to describe a user's credentials, and a party encrypting data determines a policy for who can decrypt. Thus, our methods are conceptually closer to traditional access control methods such as Role-Based Access Control (RBAC). In addition, we provide an implementation of our system and give performance measurements.

..ISSN: 2040-0748

2.2 Securing communications between external users and wireless body area networks

AUTHORS: C. Hu, F. Zhang, X. Cheng, X. Liao, and D. Chen, Wireless Body Area Networks (BANs) are expected to play a crucial role in patient-health monitoring in the near future. Establishing secure communications between BAN sensors and external users is key to addressing the prevalent security and privacy concerns.

In this paper, we propose the primitive functions to implement a secret-sharing based Ciphertext-Policy Attribute-Based Encryption (CP_ABE) scheme, which encrypts the data based on an access structure specified by the data source. We also design two protocols to securely retrieve the sensitive patient data from a BAN and instruct the sensors in a BAN. Our analysis indicates that the proposed scheme is feasible, can provide message authenticity, and can counter possible major attacks such as collusion attacks and battery-draining attacks.

2.3 Exploiting prediction to enable secure and reliable routing in wireless body area networks

AUTHORS: X. Liang, X. Li, Q. Shen, R. Lu, X. Lin, X. Shen, and W. Zhuan

In this paper, we propose a distributed Prediction-based Secure and Reliable

routing framework (PSR) for emerging Wireless Body Area Networks (WBANs). It can be integrated with a specific routing protocol to improve the latter's reliability and prevent data injection attacks during data communication. In PSR, using past link quality measurements, each node predicts the quality of every incidental link, and thus any change in the neighbor set as well, for the immediate future. When there are multiple possible next hops for packet forwarding (according to the routing protocol used), PSR selects the one with the highest predicted link quality among them. Specially-tailored lightweight source and data authentication methods are employed by nodes to secure data communication. Further, each node adaptively enables or disables source authentication according to predicted neighbor set change and prediction accuracy so as to quickly filter false source authentication requests. We demonstrate that PSR significantly increases routing reliability and effectively resists data injection attacks through in-depth security analysis and extensive simulation study.

3.1 METHODOLOGY

• Data Owner

In this module, the data provider uploads their encrypted data in the Cloud server. For the security purpose the data owner encrypts the data file and then store in the

..ISSN: 2040-0748

server. The Data owner can have capable of manipulating the encrypted data file and performs the following operations Register and Login, Attackers, Upload File, View Files, Verify data(Verifiability), View and Delete Files, View All Transactions.

Cloud Service Provider

The **Cloud** server manages which is to provide data storage service for the Data Owners. Data owners encrypt their data files and store them in the Server for sharing with data consumers. To access the shared data files, data consumers download encrypted data files of their interest from the Server and then Server will decrypt them. The server will generate the aggregate key if the end user requests for file authorization to access and performs the following operations such as Login, View and Authorize Users, View and Authorize Owners, View Files, View All Search Transactions, View All File Transactions, View All Top Searched, View Attackers, Search Requests, View Time Delay, View Throughput.

- **User**

In this module, the user can only access the data file with the secret key. The user can search the file for a specified keyword. The data which matches for a particular keyword will be indexed in the cloud server and then response to the end user and performing the following operations

Register and Login, My Profile, View Files, Search Files, Search Ratio, Top K Search, Req Search Control.

- **PKG**– responsible for viewing Files and Generate Key.

3.2 ALGORITHMS

3.2.1 Purpose of CP-AB-KEM

Ciphertext-Policy Attribute-Based Key Encapsulation Mechanism (CP-AB-KEM) is a cryptographic scheme designed to enable fine-grained access control over encrypted data. The primary purpose is to allow a data owner to specify an access policy and encrypt data such that only users whose attributes satisfy the policy can decrypt the data. This is particularly useful in scenarios where data needs to be shared among a large group of users with varying levels of access rights, such as in cloud storage, secure data sharing, and digital rights management.

3.2.2 Functionality of CP-AB-KEM

Key Components

1.**Attributes:** Descriptive properties that can be assigned to users. For example, "Role: Manager" or "Department: HR".

2.**Access Policy:** A logical expression that specifies which combinations of attributes can decrypt the ciphertext. For example, "Role: Manager AND Department: HR".

..ISSN: 2040-0748

3.Key Generation: The process of generating keys associated with specific attributes.

4. Encryption: The process of encrypting data such that only users with attributes satisfying the access policy can decrypt it.

5.Decryption: The process of decrypting the ciphertext using the keys associated with attributes that satisfy the access policy.

Process Overview

1.Setup: The system generates public parameters and a master secret key. The public parameters are used for encryption and key generation, while the master secret key is used to generate attribute keys for users.

2. Key Generation: When a user registers with the system, they receive attribute keys generated based on their attributes using the master secret key.

3. Encryption:

- The data owner defines an access policy and uses it to encrypt the data.

- A symmetric key is generated and used to encrypt the actual data.

- The symmetric key is then encrypted using the public parameters and the access policy, creating a ciphertext that encapsulates the symmetric key.

Decryption:

- A user attempts to decrypt the ciphertext using their attribute keys.

- If the user's attributes satisfy the access policy, they can recover the symmetric key.

- The user then uses the symmetric key to decrypt the actual data.

Detailed Steps

Setup Phase:

Generate the public parameters (PP) and the master secret key (MSK).

$$(PP, MSK) \leftarrow \text{Setup}()$$

Key Generation Phase:

For each user, generate a private key (SK) based on their attributes.

$$SK \leftarrow \text{KeyGen}(MSK, \text{Attributes})$$

Encryption Phase:

Define an access policy (AP).

Generate a random symmetric key (K).

Encrypt the data (D) with K using symmetric encryption.

Encrypt K using the access policy (AP) and public parameters (PP) to produce the ciphertext (CT).

$$CT \leftarrow \text{Encrypt}(PP, AP, K)$$

$$\text{Encrypt}(PP, AP, K)$$

The final ciphertext is the encrypted data along with CT.

Decryption Phase:

- o A user with attribute keys (SK) attempts to decrypt CT.
- o If the user's attributes satisfy the access policy (AP), they can decrypt and recover K.
- o $K \leftarrow \text{Decrypt}(CT, SK)$
- o $\text{Decrypt}(CT, SK) \leftarrow \text{Decrypt}(CT, SK)$
- o Use K to decrypt the actual data (D).

Example Use Case

Imagine a company where only managers in the HR department are allowed to access certain sensitive documents. Using CP-AB-KEM:

1. **Setup:** The company's IT department sets up the CP-AB-KEM system, generating the necessary public parameters and master secret key.

2. **Key Generation:** Employees are issued attribute keys based on their roles and departments, such as "Role: Manager" and "Department: HR".

3. **Encryption:** The document is encrypted with an access policy that requires "Role: Manager AND Department: HR". The symmetric key used to encrypt the document is encapsulated with this policy.

4. **Decryption:** Only employees who have the attributes "Role: Manager" and "Department: HR" can decrypt the encapsulated symmetric key and subsequently the document.

By using CP-AB-KEM, the company ensures that only authorized personnel can access sensitive information, enhancing security and access control.

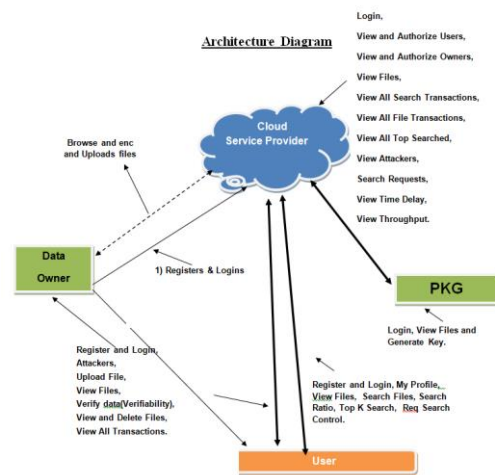


Fig 1:Architecture

4.RESULTS

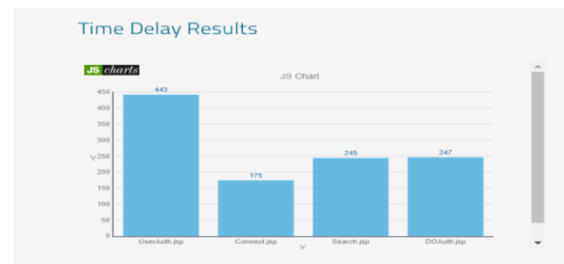


Fig 2:Time Delay Results

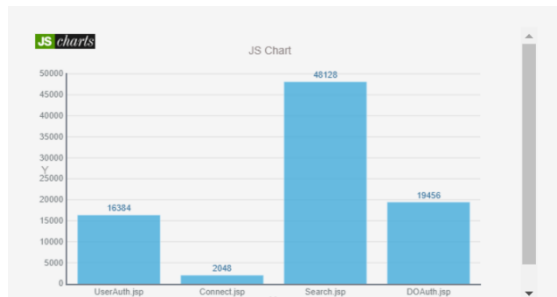


Fig 3: Throughput Results

5.CONCLUSION

We suggest Web Cloud, a workable client-side encryption option for public cloud storage in a Web environment where users only use browsers to perform cryptography. We assess the security of Web Cloud, put it into practise using our own cloud, and evaluate its performance thoroughly. The outcomes of the experiment demonstrate the viability of our solution. The design of Web-Cloud naturally incorporates a specific CP-AB-KEM scheme, which is helpful in many other applications, as an interesting by-product.

REFERENCES

[1] “Vulnerability and threat in 2018,” Skybox Security, Tech.Rep., 2018. [Online]. Available: <https://lp.skyboxsecurity.com/WICD-2018-02-Report-Vulnerability-Threat-18 Asset.html>

[2] D. Lewis, “icloud data breach: Hacking and celebrityphotos,” Duo Security, Tech. Rep., September 2014. [Online]. Available: <https://www.forbes.com/sites/davelewis/2014/09/>

02/icloud-data-breach-hacking-and-nude-celebrity-photos

[3] T. Hunt, “Hacked dropbox login data of 68 million users is now forsale on the dark web,” Tech. Rep., September 2016. [Online].

Available:<https://www.troyhunt.com/the-dropbox-hack-is-real/>

[4] “Amazon data leak,” ElevenPaths, Tech. Rep., November2018. [Online]. Available:

<https://www.elevenpaths.com/amazon-data-leak/index.html>

[5] K. Korosec, “Data breach exposes trade secrets of carmakersgm, ford, tesla, toyota,” TechCrunch, Tech. Rep., July2018. [Online]. Available:

<https://techcrunch.com/2018/07/20/data-breach-level-one-automakers/>

[6] M. Grant, “\$93m class-action lawsuit filed against cityof calgary for privacy breach,” Tech. Rep., October 2017.[Online]. Available:<http://www.cbc.ca/news/canada/calgary/city-calgary-class-action-93-million-privacy-breach-1.4321257>

[7] (2020, April) Secure file transfer — whispily.[Online]. Available:<https://whispily/en>

..ISSN: 2040-0748

[8] (2020, April) Cryptomator: Free cloud encryption for dropbox and others.

[Online]. Available: <https://cryptomator.org/>

[9] (2020, April) Whitepapers from spideroak. [Online].

Available: <https://spideroak.com/whitepapers/>

[10] W. Ma, J. Campbell, D. Tran, and D. Kleeman, "Password entropy and password quality," in Fourth International Conference on Network and System Security, NSS 2010, Melbourne, Victoria, Australia, September 1-3, 2010, Y. Xiang, P. Samarati, J. Hu, W. Zhou, and A. Sadeghi, Eds. IEEE Computer Society, 2010, pp. 583–587.

[Online]. Available: <https://doi.org/10.1109/NSS.2010.18>

AUTHOR'S PROFILE:

[1] Mr. K. Jaya Krishna, currently working as an Associate Professor in the Department of Master of Computer Applications, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He did his MCA from Anna University, Chennai, M.Tech (CSE) from JNTUK, Kakinada. He published more than 10 research papers in reputed peer reviewed Scopus indexed journals. He also attended and presented research papers in different national and international journals and

the proceedings were indexed IEEE. His area of interest is Machine Learning, Artificial intelligence, Cloud Computing and Programming Languages.

[2] Miss. Shaik Suhana, currently pursuing Master of Computer Applications at QIS College of engineering and Technology (Autonomous), Ongole, Andhra Pradesh. She Completed B.Sc. in Computer Science from S.K.P.R Degree College, Parchur, Andhra Pradesh. Her areas of interests are Cloud Computing & Machine learning.