# SECURE DATA SHARING USING WEB CLOUD STORAGE PLATFORM

**B. UMA MAHESWARI [1], L. SRIVIDYA[2]**

[1]Assistant Professor, Dept of MCA, Audisankara College of Engineering and Technology (AUTONOMOUS), NH-5 Bypass Road, Gudur, Tirupati – 524101.

[2]PG Scholar, Dept of MCA, Audisankara College of Engineering and Technology (AUTONOMOUS), NH-5 Bypass Road, Gudur, Tirupati – 524101.

**ABSTRACT**-With more and more data moving to the cloud, privacy of user data have raised great concerns. Client-side encryption/decryption seems to be an attractive solution to protect data security, however, the existing solutions encountered three major challenges: low security due to encryption with low-entropy PIN, inconvenient data sharing with traditional encryption algorithms, and poor usability with dedicated software/plugins that require certain types of terminals. This work designs and implements WebCloud, a practical browser-side encryption solution, leveraging modern Web technologies. It solves all the above three problems while achieves several additional remarkable features: robust and immediate user revocation, fast data processing with offline encryption and outsourced decryption. Notably, our solution works on any device equipped with a Web user agent, including Web browsers, mobile and PC applications. We implement WebCloud based on own Cloud for basic file management utility, and utilize Web Assembly and Web Cryptography API for complex cryptographic operations integration. Finally, comprehensive experiments are conducted with many well-known browsers, Android and PC applications, which indicates that WebCloud is cross-platform and efficient.

Index terms—key encapsulation mechanism, data encapsulation mechanism, Public Key Infrastructures

## I. INTRODUCTION

PUBLIC cloud storage service becomes increasingly popular due to cost reduction and good data usability for users. This trend has prompted users and corporations to store (unencrypted) data on public cloud, and share their cloud data with others. Using a cloud for high-value data requires the user to trust the server to protect the data from unauthorized disclosures. This trust is often misplaced,

because there are many ways in which confidential data leakage may happen, e.g. these data breaches reported to counteract data leakage, one of the most promising approaches is client-side encryption/ decryption. Concretely, client-side encryption allows senders to encrypt data before transmitting it to clouds, and decrypt the data after downloading from clouds. In this way, clouds only obtain encrypted data, thus making server-side data exposure more difficult or impossible. At the same time, as a crucial functionality of cloud storage, flexible file sharing with multiple users or a group of users must be fully supported. However, existing client-side encryption solutions suffer from more or less disadvantages in terms of security, efficiency and usability. Limited support or no support. Many cloud storage providers, including Google Drive and Drop box, do not provide support for client-side encryption. They adopt server-side encryption for files stored, TLS for data at transit, and two-factor authentication for user authentication. Apple I Cloud supports end-to end encryption for sensitive information, e.g., I Cloud Keychain, Wi-Fi passwords. For other data uploaded to I Cloud, only server encryption is adopted.

Password-Based Solutions. Some products use symmetric encryption (typically AES) to encrypt users' data and then upload ciphertexts to clouds. However, in these schemes, the cryptographic keys are derived from a password/ passphrase or even a 4-digit PIN. Relying on such low entropy is considered unsafe [10]. Worse still, most password-based solutions only deal with the case of single-user file encryption and decryption, and do not provide any file sharing mechanism. Notably, [7] allows users to generate a share link for each password-protected file. However, users must manually send the share link through one channel, and password to all receivers through another secure channel, which is inconvenient and brittle.

Hybrid Encryption Scheme. The cloud adopts a key encapsulation mechanism (KEM) and a data encapsulation mechanism (DEM), so called the KEM-DEM setting. Many public cloud service providers, including Amazon Tresorit and Mega adopt the RSA-AES paradigm. Users generate RSA key pairs and apply for certificates from the providers, who build and maintain a Public Key Infrastructures(PKI). Users encrypt data under fresh sampled AESkeys, which are further encrypted under all recipients' RSA public keys. This file sharing mechanism inflexible and inefficient. A sender needs to obtain and specify the public keys of all receivers during

encryption. Even worse, the size of the cipher text and encryption workload are proportional to the number of recipients, resulting in greater bandwidth and storage costs and more user expenditure.

## II. LITERATURE SURVEY

In-Browser Cryptography. Both the Web community and security researchers understand the importance and usefulness of in-browser cryptography and have made remark able efforts in this area. JavaScript cryptographic libraries were developed for ease of use of cryptography on browsers, for instance Many of these libraries have a large number of downloads, e.g., 423;368 for OpenPGP.js in total. The World Wide Web Consortium (W3C) noticed this trend of using in-browser cryptography and as a solution proposed a standard called Web Cryptography API. The standard supports a few widely adopted standard algorithms ,e.g., AES and ECDSA, which is convenient for building several secure Web applications [28] including authenticated video services and encrypted communications via Web mail. Meanwhile, there are researches in the literature having explored the idea of running cryptographic algorithm Web browsers. focused on using Identity-Based Cryptography for client side security in Web applications and presented a JavaScript implementation of their scheme. They selected Combined Public Key cryptosystem as the encryption scheme to avoid complex computations involved in bilinear pairing and elliptic curve. ShadowCrypt [30] allows users to transparently switch to encrypted input/output for text-based Web applications. It requires a browser extension, replacing input elements in a page with secure, isolated shadow inputs and encrypted text with secure, isolated clear text. [26] implemented several Lattice-based encryption schemes and showed the speed performance on four common Web browsers on PC. Their results demonstrated that some of today's Lattice-based cryptosystems can already have efficient JavaScript implementations. Recently, constructed an efficient two-level homomorphic public-key encryption in prime-order bilinear groups and presented a high-performance implementation using WebAssembly that allows their scheme to be run very fast on any popular Webbrowser, without any plugins required. Attribute-Based Encryption. Attribute based encryption(ABE) was first introduced by Sahai and Waters under the name fuzzy identity-based encryption. Goyal et al. extended fuzzy IBE to ABE. Up to now, there are two forms of ABE: key-policy ABE (KP-ABE) here the key is assigned to an access

policy and the ciphertext to a set of attributes, and ciphertext-policy ABE(CP-ABE) [17], [37], [38], where the ciphertext is assigned to an access policy and the key to a set of attributes. A user can decrypt a ciphertext if the set of attributes satisfies the access policy. In this work, CP-ABE is adopted as a building block of WebCloud: each file has an access policy to indicate the allowed receivers. The complex pairing and exponentiation operations in ABE are migrated by many works. Green et al. introduced outsourced decryption into ABE systems such that the complex operations of decryption can be outsourced toa cloud server, only leaving one exponentiation operation for a user to recover the plaintext. Further, online/offline ABE [20] was proposed by Hohenberger and Waters, which splits the original algorithm into two phases: an offline phase which does the majority of encryption computations before knowing the attributes/access control policy and generates an intermediate ciphertext, and an online phase which rapidly assembles an ABE ciphertext with the intermediate ciphertext after the attributes/access control policy is fixed. Meanwhile, [20] proposed two scenarios about the offline phase: 1) the user does the offline work on his smartphone. 2) A high-end trusted server helps the user with low-end device do the offline work.

## III. PROPOSED SYSTEM

The overview of our proposed system is shown in the below figure.

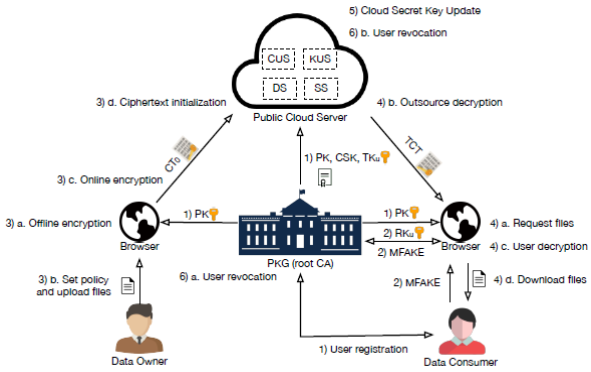


Fig. 1: System Overview

### *Implementation Modules*

### 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 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. PKG generates and distributes system parameters and keys to other entities, and instructs the cloud to revoke a user. PKG maintains a Public Key Infrastructure (PKI) and plays as the root Certificate Authority (CA).

## IV. RESULTS


Fig. 1: Home Page

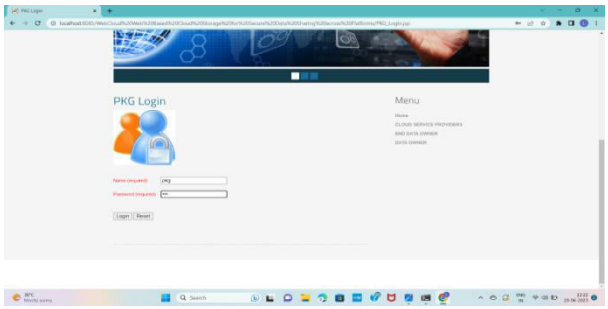


Fig. 2: PKG Login


Fig. 3: Cloud Service Provider Login


Fig. 4: Data Owner Login

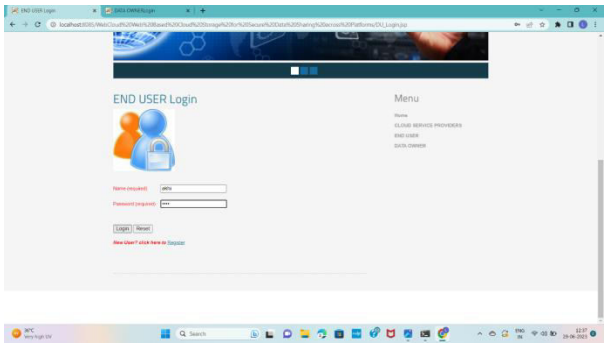


Fig. 5: End User Login

## V. CONCLUSION

We propose WebCloud, a practical client-side encryption solution for public cloud storage in the Web setting, where users do cryptography with only browsers. We analyze the security of WebCloud and implement WebCloud based on ownCloud and conduct a comprehensive performance evaluation. The experimental results show that our solution is practical. As an interesting by-product, the design of Web-Cloud naturally embodies a dedicated CP-AB-KEM scheme, which is useful in many other applications.

## REFERENCES

[1] S. Shaham, M. Ding, B. Liu, Z. Lin, and J. Li, "Machine learning aided anonymization of spatiotemporal trajectory datasets," arXiv preprint arXiv:1902.08934, 2019.

[2] A. Government, "New australian government data sharing and release legislation," 2018.

[3] A. Tamersoy, G. Loukides, M. E. Nergiz, Y. Saygin, and B. Malin, "Anonymization of longitudinal electronic medical records," IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 3,pp. 413–423, 2012.

[4] F. Xu, Z. Tu, Y. Li, P. Zhang, X. Fu, and D. Jin, "Trajectory recovery from ash: User privacy is not preserved in aggregated mobility data," in Proceedings of the 26th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 2017, pp. 1241–1250.

[5] Y. Dong and D. Pi, "Novel privacy-preserving algorithm based on frequent path for trajectory data publishing," Knowledge-Based Systems, vol. 148, pp. 55–65, 2018.

[6] M. Gramaglia, M. Fiore, A. Tarable, and A. Banchs, "Towards privacy-preserving publishing of spatiotemporal trajectory data," arXiv preprint arXiv:1701.02243, 2017.

[7] M. Terrovitis, G. Poulis, N. Mamoulis, and S. Skiadopoulos, "Local suppression and

splitting techniques for privacy preserving publication of trajectories," IEEE Trans. Knowl. Data Eng, vol. 29, no. 7, pp. 1466–1479, 2017.

[8] M. E. Nergiz, M. Atzori, and Y. Saygin, "Towards trajectory anonymization: a generalization-based approach," in Proc. of the SIGSPATIAL ACM GIS. ACM, 2008, pp. 52–61.

[9] S. Gurung, D. Lin, W. Jiang, A. Hurson, and R. Zhang, "Traffic information publication with privacy preservation," ACM Transactions on Intelligent Systems and Technology (TIST), vol. 5, no. 3, p. 44, 2014.

[10] R. Yarovoy, F. Bonchi, L. V. Lakshmanan, and W. H. Wang, "Anonymizing moving objects: How to hide a mob in a crowd?" in Proc. of the 12th International Conference on Extending Database Technology: Advances in Database Technology. ACM, 2009, pp. 72– 83.

## AUTHORS

**B. Uma Maheswari** has received MCA degree from SV University, Tirupati in 2008. She has an experience of 4 years (Teaching) in SVU PG centre, kavali from 2008-2012 in the department of MSC computer science and MCA. Currently she is working as assistant professor in Audisankara engineering College, Guduru, Andhrapradesh, India.

**L . SRIVIDYA** has pursuing his MCA from Audisankara college of Engineering and Technology (AUTONOMOUS), Gudur, affiliated to JNTUA in 2023 . Andhra Pradesh, India.