International Journal for Innovative Engineering and Management Research
PEER REVIEWED OPEN ACCESS INTERNATIONAL JOURNAL
www.ijiemr.org

WEB CLOUD WEB-BASED CLOUD STORAGE FOR SECURE DATA SHARING ACROSS PLATFORMS

[1]M TEJASWI REDDY, [2]OMPRAKASH, [3]CH SANDHYA, [4]K SARALA

[123]ASSISTANCT PROFESSOR, BRILLIANT INSTITUTE OF ENGINEERING & TECHNOLOGY, ABDULLAPURMET(V&M) RANGA REDDY DIST-501505

[4]UG SCHOLAR, DEPARTMENT OF CSE, BRILLIANT INSTITUTE OF ENGINEERING & TECHNOLOGY, ABDULLAPURMET(V&M) RANGA REDDY DIST-501505

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 ownCloud for basic file management utility, and utilize WebAssembly 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.

As an interesting by-product, the design of WebCloud naturally embodies a dedicated and practical ciphertext-policy attribute-based key encapsulation mechanism (CP-AB-KEM) scheme, which can be useful in other applications.

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 [1], [2], [3], [4], [5], [6]. 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. Known Client-Side Encryption Solutions. We review existing solutions and point out their limitations.

_ **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 [7], [8], [9] 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 [11], Tresor it [12], and Mega [13], 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 AES keys, which are further encrypted under all recipients' RSA public keys. This file sharing mechanism is 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.

Limitations of the Existing Solutions. Three drawbacks exist in above-mentioned solutions: 1) comparatively poor security, 2) coarse-grained access control, inflexible and inefficient file sharing, and 3) poor usability. The first two are easy to see and we now elaborate the usability issue. Typically, users use different terminals to upload files, including desktop, Web and mobile applications [14]. However, almost all the existing solutions require additional software or plugins, thus limiting users' devices and platforms. When switching to a new device, users need to repeat the boring installation process, which greatly increases users' burden thus decreases usability.

II.EXISTING SYSTEM

Meanwhile, there are researches in the literature having explored the idea of running cryptographic algorithms on Web browsers. [29] 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-basedWeb applications. It requires a browser extension, replacing input elements in a page with secure, isolated shadow inputs and encrypted text with secure, isolated cleartext. [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, [31] constructed an efficient two-level homomorphic public-key encryption in prime-order bilinear groups and presented a high-performance implementation usingWebAssembly that allows their scheme to be run very fast on any popular Web browser, 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 [32]. Goyal et al. [33] extended fuzzy IBE to ABE. Up to now, there are two forms of ABE: key-policy ABE (KP- ABE) [33], [34], [35], [36], where 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. [19] introduced outsourced decryption into ABE systems such that the complex operations of decryption can be outsourced to a 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.

Disadvantages

1) Comparatively poor security,
2) Coarse-grained access control, inflexible and inefficient file sharing, and
3) Poor usability. The first two are easy to see and we now elaborate the usability issue. Typically, users use different terminals to upload files, including desktop, Web and mobile applications.

III.PROPOSED SYSTEM

We view our contribution as the uniform design, rigorous analysis and efficient

implementation of WebCloud, in particular, it simultaneously achieves the following:

_ **Practical Encryption Solution for Cloud Storage.** We introduce WebCloud, a practical client-side encryption solution for public cloud storage, which effectively combines modern Web techniques and cryptographic algorithms. WebCloud involves of a key management mechanism, a dedicated attribute based encryption scheme and a high-speed implementation. More importantly, WebCloud is crossplatform (including major browsers, Android and PC) and plugin-free.

_ **Fine-Grained Access Control Mechanism with ABE.** It is widely-accepted that attribute-based encryption (ABE) is promising for fine-grained access control of data. However, we find that the existing ABE schemes suffer from high computational overhead, or some vital missing functionalities, e.g., inefficient data encryption, robust and immediate user revocation, offline encryption and outsourced decryption simultaneously. To solve this problem, we propose a dedicated ciphertext-policy attribute-based access control mechanism. The proposed scheme can also be used in other scenarios.

_ **Rigorous Security Analysis.** We present a security model of WebCloud, including the adversarial models for the Web and the cryptographic scheme simultaneously. The security analysis is then done in the proposed model, namely, the provable security of the proposed CP-ABE scheme and the reliability of the key storage in the browser side.

_ **Efficient Operation inside Browsers.** We implement WebCloud based on ownCloud [23]. The functionalities and performances are evaluated in major browsers on many devices, and applications on PC and Android devices. The benchmark result indicates that WebCloud is a practical solution. Most remarkably,
in the Chrome browser on a 4-core 2.2 GHz Macbook machine, encrypting a 1 GB file takes 3.1 seconds, while decryption costs 3.9 seconds.

Advantages

- The proposed system focuses on designing and implementing a practical, secure and cross-platform public cloud storage system. The proposed solution, WebCloud, is a Web-based client-side encryption solution. Users encrypt and decrypt their data using Web agents, e.g., Web browsers.

- The proposed system implemented Multi-Factor Authenticated Key Exchange which gives more security and safe.

IV. IMPLEMENTATION

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.

V.CONCLUSION

We propose Web Cloud, 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 Web Cloud and implement Web Cloud based on own Cloud 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.

VI.REFERENCES

[1] "Vulnearability 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 celebrity photos," 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 for sale 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., November 2018. [Online]. Available: https://www.elevenpaths.com/ amazon-data-leak/index.html

[5] K. Korosec, "Data breach exposes trade secrets of carmakers gm, ford, tesla, toyota," TechCrunch, Tech. Rep., July 2018. [Online]. Available: https://techcrunch.com/2018/07/20/ data-breach-level-one-automakers/

[6] M. Grant, "$93m class-action lawsuit filed against city of 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 — whisply. [Online]. Available: https://whisp.ly/en

[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

[11] (2020, April) Aws sdk support for amazon s3 client-side encryption. [Online]. Available: https://docs.aws.amazon.com/ general/latest/gr/aws sdk cryptography.html

[12] (2020, April) Cloud storage security - secure cloud storage from tresorit. [Online]. Available: https://tresorit.com/security

[13] (2020, April) Mega - secure cloud storage and communication. [Online]. Available: https://mega.nz/

[14] E. Bocchi, I. Drago, and M. Mellia, "Personal cloud storage: Usage, performance and impact of terminals," in 4th IEEE International Conference on Cloud Networking, CloudNet 2015, Niagara Falls, ON, Canada, October 5-7, 2015. IEEE, 2015, pp. 106–111. [Online]. Available: https://doi.org/10.1109/CloudNet.2015.7335 291

[15] "Web cryptography api," the Web Cryptography WG of the W3C, Tech. Rep., January 2017. [Online]. Available: https://www.w3.org/TR/WebCryptoAPI/

[16] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, and J. Bastien, "Bringing the web up to speed with webassembly," in ACM SIGPLAN Notices, vol. 52, no. 6. ACM, 2017, pp. 185–200.

[17] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization," in International Workshop on Public Key Cryptography. Springer, 2011, pp. 53–70.

[18] W. Zhu, J. Yu, T. Wang, P. Zhang, and W. Xie, "Efficient attributebased encryption from r-lwe," Chin. J. Electron, vol. 23, no. 4, pp. 778–782, 2014.

[19] M. Green, S. Hohenberger, B. Waters et al., "Outsourcing the decryption of abe ciphertexts." in USENIX Security Symposium, vol. 2011, no. 3, 2011.

[20] S. Hohenberger and B. Waters, "Online/offline attributebased encryption," in International Workshop on Public Key Cryptography. Springer, 2014, pp. 293–310.

[21] R. Zhang, H. Ma, and Y. Lu, "Fine-grained access control system based on fully outsourced attribute-based encryption," Journal of Systems and Software, vol. 125, pp. 344–353, 2017.

[22] S. Yu, C. Wang, K. Ren, and W. Lou, "Attribute based data sharing with attribute revocation," in Proceedings of the 5th ACM symposium on information, computer and communications security, 2010, pp. 261–270.

[23] (2020, April) owncloud - the leading opensource cloud collaboration platform. [Online]. Available: https://owncloud. org/

[24] (2020, April) Openpgp implementation for javascript. [Online]. Available: https://github.com/openpgpjs/openpgpjs

[25] E. Stark, M. Hamburg, and D. Boneh, "Symmetric cryptography in javascript," in Computer Security Applications Conference, 2009. ACSAC'09. Annual. IEEE, 2009, pp. 373–381.