International Journal on Applications in Engineering and Technology
Volume 9 : Issue 2 : April 2023, pp 19 – 26 www.aetsjournal.com ISSN (Online) : 2455 - 0523

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

M.THANGAVEL , T. ABIRAMI

Abstract— The 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.

Index Terms—Web-Based Cloud Storage, Secure Data Sharing, Cross-Platform Encryption/Decryption Solution, Attribute-Based Encryption

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. Known Client-Side Encryption Solutions. We review existing solutions and point out their limitations.

1) 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.

2) PASSWORD-BASED SOLUTIONS.

Some products [7], [8], [9] use symmetric encryption (typically AES) to encrypt users' data

M.Thangavel , Professor , Department of Computer Applications , Erode Sengunthar Engineering College (Autonomous), Perundurai , Erode.
(Email : thangavelpamu@gmail.com)
T. Abirami, PG Scholar , Department of Computer Applications, Erode Sengunthar Engineering College (Autonomous), Perundurai , Erode.
(Email : Abiramideav@gmail.com)

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.

3) 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.

4) 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. LITERATURE SURVEY

In-Browser Cryptography. Both the Web community and security researchers understand the importance and usefulness of in-browser cryptography and have made remarkable efforts in this area This work designs and implements WebCloud, a practical browser-side encryption solution, leveraging modern Web technologies.[12] 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.

a new methodology for realizing Ciphertext-Policy Attribute Encryption (CP-ABE) under concrete and noninteractive cryptographic assumptions in the standard model. Our solutions allow any encryptor to specify access control in terms of any access formula over the attributes in the system. In our most efficient system, ciphertext size, encryption, and decryption time scales linearly with the complexity of the access formula. The only previous work to achieve these parameters was limited to a proof in the generic group model.

Attribute-based encryption (ABE) has been an active research area in cryptography due to its attractive applications. But almost all attribute-based encryption schemes are based on bilinear maps, which leave them vulnerable to quantum cryptanalysis.

In particular, we resolve this challenging issue by considering more practical scenarios in which semi-trustable on-line proxy servers are available. As compared to existing schemes, our proposed solution enables the authority to revoke user attributes with minimal effort.

the increase in use of digital technology, use of data items in the format of text, image and videos are also increases. To securely send this data, many users and smart applications have adapted the image encryption approach.

III. 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.

1) 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.

IV. PROPOSED SYSTEM

We view our contribution as the uniform design, rigorous analysis and efficient implementation of WebCloud, in particular, it simultaneously achieves the following:

1)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.

2)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.

3)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.

4) 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.

5) IMPLEMENTATION AND EXECUTION

The implementation of this application is split into following modules.

- ❖ Data owner
- ❖ Cloud Service Provider
- ❖ User

A. 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.

B. 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.

C. 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. SYSTEM ARCHITECTURE

Figure1: System Architecture

The above figure represents system architecture of proposed system, where we are showing the process of webcloud based cloud storage for secure data sharing.

USE CASE DIAGRAM

FIGURE 2: usecase diagram

DATA FLOW DIAGRAM

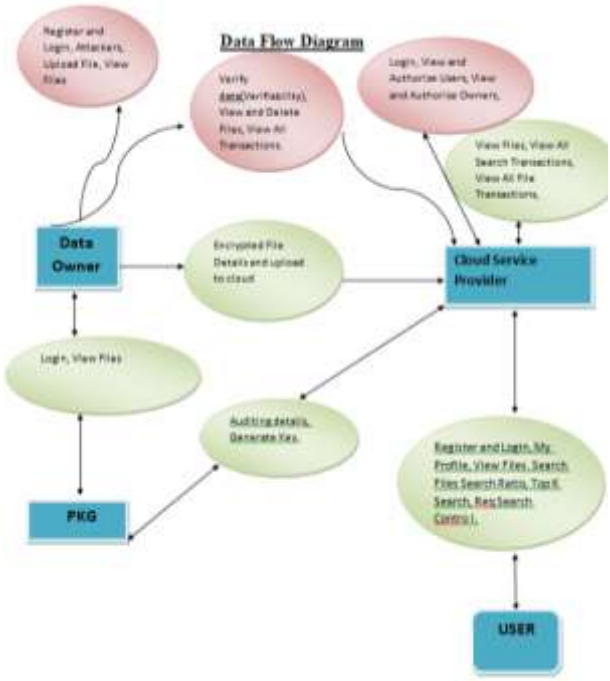


FIGURE 3: dataflow diagram

DESCRIPTION OF ALGORITHMS

We now elaborate the algorithms of WebCloud Some acronyms are listed in Table

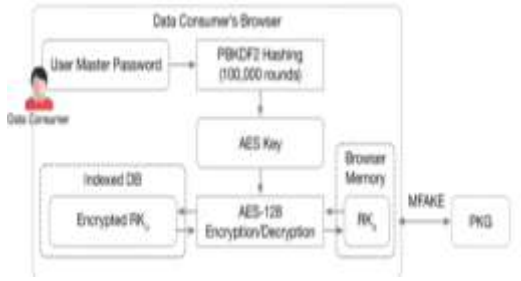
TABLE 1: Acronyms Used in This Paper

Acronym	Description	Acronym	Description
PKG	private key generator	CUS	ciphertext update service
KUS	key update service	DS	outsourced decryption service
SS	storage service	MSK	master secret key
PK	public key	ctr	current time counter
SK_u	a user's secret key	RK_u	a user's retrieval key
CSK	cloud secret key	TCT	transformed ciphertext
TK_u	a user's transformation key	IT	intermediate ciphertext

VI. SYSTEM INITIALIZATION.

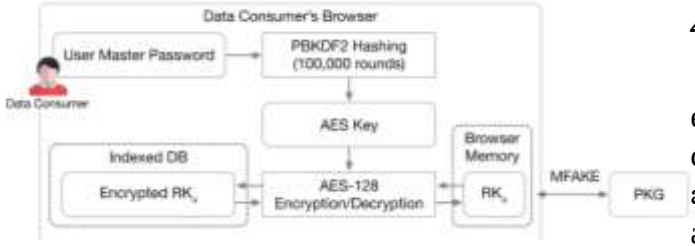
PKG runs the algorithm Setup() to generate a public key PK, a master secret key MSK and a cloud secret key CSK1. All the data consumers register themselves to PKG: 1) run the registration phase of MFAKE protocol [43] where PKG serves as the registration center; and 2) state a set of properties to indicate their identities. Then, PKG runs the algorithms KG() and KG.Random() to generate TKu and RKu for each data consumer. Further, PKG generates a certificate Tcloud for the public cloud server, which is used to establish secure communication between the cloud server and

users. Finally, PKG distributes PK to all the entities, CSK1 and TKu along with Tcloud to the cloud, and keeps RKu for future distributions.



1)KEY MANAGEMENT.

To decrypt data in browsers, data consumers obtain their retrieval keys from PKG as shown in Fig. 3. To this end, the login-authentication phase of the MFAKE protocol [43], is run between a user's browser and PKG to establish a secure communication channel. The consumer's retrieval key RKu is transmitted to the browser through the secure channel and later be used in the browser's memory. If the user remains idle for a specified period of time (e.g., 30 minutes), RKu is automatically erased from the memory and later use of RKu requires running the MFAKE protocol again



2) DATA ENCRYPTION.

To improve performance, data encryp tion procedure is divided into three parts as depicted in Fig. 4. The encryption is in the KEM/DEM setting. Offline encryption in browser (before an access policy is known): This algorithm processes almost all the costly oper- ations in the encryption algorithm of CP-AB-KEM. On open- ing the WebCloud website, a Web worker (cf. Section 2.1) is created in background. During idle time, the worker runs the algorithm Enc.Offline() to generate a few intermediate ciphertexts IT and keys Key. Idle time is defined as: a) no online encryption part is running, b) no user decryption part is running, and c) no AES encryption or decryption is running. We store (IT, Key) in sessionStorage, which is erased automatically by browsers after the Web page is closed. Online encryption in browser (after an access policy string and a file are given): The data owner uploads a file and specifics an access policy string, e.g., "(Employee and IT department) or Manager". Note that the policy string is a flexible logic expression, which supports "and", "or" and "()" operations. The policy string is converted to an access structure (M, ρ). Meanwhile, a pair of (IT, Key) is obtained from sessionStorage. On input the access policy (M, ρ) and the intermediate ciphertext IT, the algorithm Enc.Online() generates ABE ciphertext CT0. The input file is encrypted with AES, using a 128-bit key derived from Key and a random initialization vector (IV). All necessary data including ABE ciphertext CT0 and AES ciphertext, are packed together and forms a new file before uploading to the cloud server.

3) DATA PRIVACY.

In WebCloud, all files are encrypted and decrypted locally, i.e., in browsers. The cloud only sees ciphertext and deals with ciphertext.

4) FLEXIBLE FILE SHARING.

By assigning an access policy to each file, an encrypted file can be decrypted by multiple data consumers as long as their attributes satisfy the access policy. The user only encrypts a file one time and the cloud only stores one copy of each encrypted file. In a corporation scenario, an employee can share a file with all managers of the sales department by setting the access policy to "Sales Department and Manager", without the need to find out who are the concrete receivers or their public keys as in the password-based solutions and RSA-AES paradigm. If a new employee is hired, he can decrypt all the ciphertexts that match his/her attributes immediately. In existing schemes, manually encryption and sharing to the employee are re- quired.

5) USER AND KEY REVOCATION.

The revocation mechanism is ef- ficient and immediately effective. The cloud server revokes a data consumer by adding the consumer to the revocation list L. On receiving a consumer's download request, the cloud checks the list L and rejects revoked consumers' requests. The key revocation is achieved by requiring PKG to regen- erate a new SKu and related (RKu, TKu), and distributes RKu to the user and TKu to the cloud server.

6) USABILITY AND EFFICIENCY.

WebCloud only requires a Web user agent and does not require any additional software, Java applet or browser plugin. WebCloud is fully optimized in two aspects: a) The proposed CP-AB-KEM scheme is very suitable for browser side cryptography, which moves almost all costly computations offline and outsourced, and b) The implementation fully utilize power of modern Web tech- niques, including WebAssembly, Web Cryptography API, Web Workers and Web storage. The functionalities and per- formance are tested in major browsers on different devices, including both laptops and mobiles.

7) CLOUD SERVER KEY-EXPOSURE RESISTANCE.

The cloud secret key CSK is important to the revocation mechanism. If CSK leaks, the revocation functionality is useless. Thus, WebCloud introduces key-exposure resistance property for CSK. Concretely, the cloud updates CSK periodically (or when key is leaked). Meanwhile, it updates all stored ABE ciphertexts and deletes old ciphertexts when the CSK is updated.

VII. 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

VIII. FUTURE ENHANCEMENTS

To protect consumer's private data in distributed cloud storage is challenging task. To share the personal file from one user to another user is most difficult using the cloud computing. To provide more secure to the data using different schemes. We studied different Cloud Computing concepts like mobile cloud computing, integrity and forward security, social cloud computing, framework, resource allocation, data security, privacy and dynamic, IOT and load balancing schemes. While concerning about the security of data we analyze different issues such as low security, high computational complexity, privacy/public file sharing and improper utilization of resources. There are distinctive scheduling strategies used in multiple federated data centers. Aggregate key provides more secure in sending file between sender and receiver. Forward security encryption and re-encryption techniques are used to provide high security for private data. It gives demonstrating and reproduction of substantial scale distributed computing, including data centers. Virtual Machines (VMs) and Physical Machines (PMs) consumes more power. This is the limitation, due to this, the cost of file transfer is increased. A further enhancement to overcome intruder's attacks in private data, to provide more security and more sharing of files, effectively utilization of resources and reducing the energy consumption in VMs and PMs.

IX. 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]. Avail- able: 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.2015.7335291

[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 attribute-based 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 attribute- based 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.

[21] 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.

[22] (2020, April) owncloud - the leading opensource cloud collaboration platform. [Online]. Available: https://owncloud. org/

[23] (2020, April) Openpgp implementation for javascript. [Online]. Available: https://github.com/openpgpjs/openpgpjs

[24] 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.