INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

WEBCLOUD WEB BASED CLOUD STORAGE FOR SECURE DATA SHARING ACROSS PLATFORMS

[1]Ch. KODANDARAMU, Associate Professor

[2]BALUDU VAISHNAVI [3]ANGUDU ALEKHYA

[4]JALAMADULA GANESH [5]KANAPARTHI SASI HARSHA VARDHAN

[1,2,3,4,5]Miracle Educational Society Group of Institutions, Vijayanagaram, Ap, India

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 encryptionwith low-entropy PIN, inconvenient data sharing with traditional encryption algorithms, and poorusability with dedicated software/plugins that require certain types of terminals. This work designs and implements Web Cloud, a practical browser-side encryption solution, leveraging modern Webtechnologies. It solves all the above three problems while achieves several additional remarkablefeatures: robust and immediate user revocation, fast data processing with offline encryption andoutsourced decryption. Notably, our solution works on any device equipped with a Web user agent,including Web browsers, mobile and PC applications. We implement Web Cloud based on ownCloudfor basic file management utility, and utilize Web Assembly and Web Cryptography API for complexcryptographic operations integration. Finally, comprehensive experiments are conducted withmany well-known browsers, Android and PC applications, which indicates that Web Cloud is cross-platform and efficient. As an interesting by-product, the design of Web Cloud naturally embodies adedicated and practical cipher text-policy attribute-based key encapsulation mechanism (CP-AB-KEM) scheme, which can be useful in other applications

1. INTRODUCTION

PUBLIC cloud storage service becomes increasin popular due to cost reduction and good datausabil for users. This trend has prompted users a corporations to store (unencrypted) data onpub cloud, and share their cloud data with others. Usin cloud for high-value data requires theuser to trust t server to protect the data from unauthoriz disclosures. This trust is oftenmisplaced, becau there are many ways in which confidential d leakage may happen, e.g. thesedata breach reported [1], [2], [3], [4], [5], [6]. To counteract data leakage, one of the mostpromising approaches is client-side encryption/decryption. Concretely, client-side encryption allowssenders to encrypt data before transmitting it to clouds, and decrypt the data after downloading fromclouds. In this way, clouds only obtain encrypted data, thus making server-side data exposure moredifficult or impossible. At the same time, as a crucial functionality of cloud storage, flexible filesharing with multiple users or a group of users must be fully supported. However, existing

ISSN: 2057-5688

client-sideencryption solutions suffer from more less disadvantages in terms of security, efficier andusability. Known Client-Side Encrypti Solutions. We review existing solutions and po out theirlimitations.

Limited support or no support. Many cloud storage providers, including Google Driveand Drop box, do not provide support for client-side encryption. They adopt server-side encryptionfor files stored, TLS for data at transit, and two-factor authentication for user authentication. Apple ICloud supports end-to end encryption for sensitive information, e.g., I Cloud Keychain, Wi-Fipasswords. 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 cipher texts to clouds. However, in theseschemes, 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 solutionsonly deal withthe case of single-user file encryption and decryption, and do not provide any file sharingmechanism. 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 allreceivers 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 cloudservice providers, including Amazon [11], Tresor it [12], and Mega [13], adopt the RSA-AESparadigm. Users generate RSA key pairs and apply for certificates from the providers, who build andmaintain 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 isinflexible and inefficient. A sender needs to obtain and specify the public keys of all receivers duringencryption. Even worse, the size of the cipher text and encryption workload are proportional to thenumber 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, and3) poor usability. The first two are easy to see and we now elaborate the usability issue. Typically, usersuse 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 andplatforms. When switching to a new device, users need to repeat the boring installation process, whichgreatly increases users' burden thus decreases usability.

2. LITERATURE SURVEY

In-Browser Cryptography. Both the Web community and security researchers understand the importance andusefulness of in-browser cryptography and have made remarkable efforts in this area. JavaScript cryptographiclibraries were developed for ease of use of cryptography on browsers, for

ISSN: 2057-5688

instance [24], [25], [26]. Many of theselibraries have a large number of downloads, e.g., 423,368 for OpenPGP.js [24] in total. The World Wide WebConsortium (W3C) noticed this trend of using in-browser cryptography and as a solution proposed a standardcalled Web Cryptography API [27], [15]. 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 authenticatedvideo services and encrypted communications via Web mail.

Meanwhile, there are researches in the literature having explored the idea of running cryptographic algorithmson Web browsers. [29] focused on using Identity-Based Cryptography for client side security in Web applicationsand presented a JavaScript implementation of their scheme. They selected Combined Public Key cryptosystem asthe 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 Webapplications. It requires a browser extension, replacing input elements in a page with secure, isolated shadowinputs and encrypted text with secure, isolated cleartext. [26] implemented several Lattice-based encryptionschemes and showed the speed performance on four common Web browsers on PC. Their results demonstratedthat 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 bilineargroups and presented a high-performance implementation using Web Assembly that allows their scheme to berun very fast on any popular Web browser, without any pluginsrequired.Attribute-Based Encryption. Attribute based encryption (ABE) was first introduced by Sahai and Waters under thename fuzzy identity-based encryption [32]. Goyal et al. [33] extended fuzzy IBE to ABE. Up to now, there are twoforms of ABE: key-policy ABE (KP- ABE) [33], [34], [35], [36], where the key is assigned to an access policy and the cipher text to a set of attributes, and cipher text-policy ABE (CP-ABE) [17], [37], [38], where the cipher text is assigned to an access policy and the key to a set of attributes. A user can decrypt a cipher text if the set ofattributes 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 beoutsourced to a cloud server, only leaving one exponentiation operation for a user to recover the plaintext.Further, online/offline ABE [20] was proposed by Rosenberger and Waters, which splits the original algorithminto two phases: an offline phase which does the majority of encryption computations before knowing theattributes/access control policy and generates an intermediate cipher text, and an online phase which rapidly assembles an ABE cipher text with the intermediate cipher text after the attributes/access control policy

ISSN: 2057-5688

is fixed.Meanwhile, [20] proposed two scenarios about the offline phase: 1) the user does the offline work on hissmartphone. 2) A high-end trusted server helps the user with low-end device do the offline work

3. EXISTING SYSTEM

Meanwhile, there are researches in the literature having explored the idea of running cryptographicalgorithms on Web browsers. [29] focused on using Identity-Based Cryptography for client sidesecurity in Web applications and presented a JavaScript implementation of their scheme. Theyselected Combined Public Key cryptosystem as the encryption scheme to avoid complexcomputations involved in bilinear pairing and elliptic curve.

Shadow Crypt [30] allows users to transparently switch to encrypted input/output fortext-basedWeb applications. It requires a browser extension, replacing input elements in a pagewith secure, isolated shadow inputs and encrypted text with secure, isolated clear text. [26]implemented several Lattice-based encryption schemes and showed the speed performance onfour common Web browsers on PC. Their results demonstrated that some of today's Lattice-basedcryptosystems can already have efficient JavaScript implementations. Recently, [31] constructed anefficient two-level homomorphic public-key encryption in prime-order bilinear groups andpresented a high-performance implementation using Web Assembly that allows their scheme to berun 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 toABE. Up to now, there are two forms of ABE: key-policy ABE (KP- ABE) [33], [34], [35], [36], wherethe key is assigned to an access policy and the cipher text to a set of attributes, and cipher text-policy ABE (CP-ABE) [17], [37], [38], where the cipher text is assigned to an access policy and thekey to a set of attributes. A user can decrypt a cipher text if the set of attributes satisfies the accesspolicy. In this work, CP-ABE is adopted as a building block of WebCloud: each file has an accesspolicy 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 decryptioncan be outsourced to a cloud server, only leaving one exponentiation operation for a user to recover theplaintext. Further, online/offline ABE [20] was proposed by Hohenberger and Waters, which splits theoriginal algorithm into two phases: an offline phase which does the majority of encryption computationsbefore knowing the attributes/access control policy and generates an intermediate cipher text, and an online phase which rapidly assembles an ABE cipher text with the intermediate cipher text after the attributes/access control policy is fixed. Meanwhile, [20] proposed two scenarios about the offlinephase: 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.

3.1 LIMITATIONS OF SYSTEM:

ISSN: 2057-5688

The system is implemented by Conventional Machine Learning. The system doesn't implement for

analyzing large data sets.

4. 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 techniquesand cryptographic algorithms. WebCloud involves of a key management mechanism, a dedicatedattribute based encryption scheme and a high-speed implementation. More importantly, WebCloud is cross platform (including major browsers, Android and PC) and plugin-free.

_ Fine-Grained Access Control Mechanism with ABE. It is widely-accepted that attribute-basedencryption (ABE) is promising for fine-grained access control of data. However, we find that theexisting ABE schemes suffer from high computational overhead, or some vital missingfunctionalities, e.g., inefficient data encryption, robust and immediate user revocation, offlineencryption and outsourced decryption simultaneously. To solve this problem, we propose adedicated cipher text-policy attribute-based access control mechanism. The proposed scheme canalso 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 thendone in the proposed model, namely, the provable security of the proposed CP-ABE scheme andthe reliability of the key storage in the browser side.

_ Efficient Operation inside Browsers. We implement WebCloud based on own Cloud [23]. Thefunctionalities and performances are evaluated in major browsers on many devices, andapplications 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 Mac book machine, encrypting a 1 GB file takes 3.1 seconds,while decryption costs 3.9 seconds

4.1 ADVANTAGES OF PROPOSED SYSTEM

The proposed system focuses on designing and implementing a practical, secure and cross-platformpublic cloud storage system. The proposed solution, WebCloud, is a Web-based client-side encryptionsolution. Users encrypt and decrypt their data using Web agents, e.g., Web browsers. The proposedsystem implemented Multi-Factor Authenticated Key Exchange which gives more security andsafe.

5. SYSTEM ARCHITECTURE

System Model As shown in Fig. 1, WebCloud adopts the browser and server (B/S) architecture. There are fourentities involved: a private key generator (PKG), a public cloud server, data owners and data consumers. The rolesof each entity are as follows:

ISSN: 2057-5688

• PKG generates and distributes system parameters and keys to other entities, andinstructs the cloud to revoke a user. PKG maintains a Public Key Infrastructure (PKI) and plays as the rootCertificate Authority (CA). We stress that this only increases PKG's workload marginally since certificate issuanceand key distribution are completed at the same time.

• The public cloud server provides a website for accessingand storing data in Web user agents. Moreover, it runs a few services: – Storage Service (SS) storestransformation keys and encrypted data reliably. – Outsourced Decryption Service (DS) checks whether a dataconsumer has been revoked and preprocesses cipher text to ease computation overhead of decryption for users. –Key Update Service (KUS) updates cloud secret key CSK periodically or when current CSK is leaked. – Cipher text Update Service (CUS) updates cipher texts with new CSK.

• Data owners decide access policies and encrypt dataunder these policies before uploading to the public cloud.

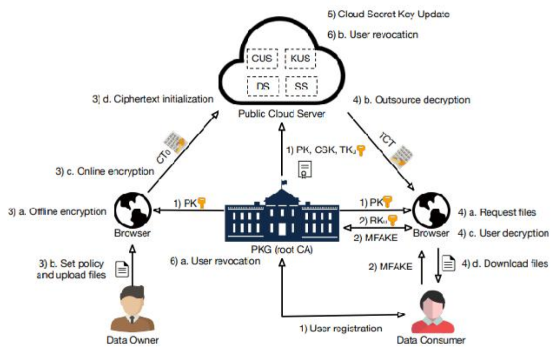
• Data consumers download encrypted data frompublic cloud server and decrypt the data locally. We remark that such a trusted party (serving as PKG) is not hardto find, for instance, e.g., government organizations or major banks.Security Notions The security goal of WebCloud is to protect users' data from disclosure on the server side forcloud storage systems. We assume the cloud is honest-but-curious [45]. The cloud honestly follows the protocol,e.g., provides storage service and outsource decryption service. It does not adversarial modify users' data. Mostof the data consumers are honest, but few of them may be corrupt and share their secret keys in the collusion.

On the contrast, PKG and data owners are assumed to be totally trusted. All the communications are secured byTLS [46]. Following adversary models are considered:Passive Man-in-the-Middle. The adversary reads all network traffic passively, but does not perform any activeattacks, e.g., altering network packets.

Web Attacker Model. This model is the standard security model of Web applications [47]. An adversary in thismodel can access any open Web application, learn its client-side code, send emails and other messages, and canset up their own (malicious) Web applications. The adversary is unable. to forge Web origins [48], since thisundermines the security of any Web application.

Data Security against User Collusion. In this model, some users and the cloud can collude in arbitrary manner,e.g., they can obtain some cryptographic keys RKu,TKu and cloud secret key CSKctr. They try to decrypt files thatbeyond their authorized access rights. We formalize it with Definition 1 in Appendix B.Data Security against Cloud Server. The public cloud, who can obtain CSKctr, the conversion key TKu of all theusers, and all the cipher texts, cannot decrypt the cipher texts. We formalize it with Definition 2 in Appendix B.User Revocation Validity. In this model, a revoked user, who can obtain his/her secret keys (SKu,TKu, RKu), cannot decrypt files within its authorized access rights. We formalize it with Definition 3 in Appendix B.

ISSN: 2057-5688



(a) System Model of WebCloud

Deployment Architecture As shown in Fig. 2, WebCloud consists of four functional modules (M1 to M4).M1. WebCloud Core. This module works in a user's browser and contains crypto and storage modules. The cryptomodule further implements following submodules.

• CP-AB-KEM1 submodule includes offline and onlineencryption, i.e., algorithms Enc.Offline and Enc.Online. Meanwhile, the submodule implements LSSS andconverter from access policy string to access structure. The converter can convert an access policy string, e.g.,"(Employee and IT department) or Manager" to an access structure (M, ρ). To the best of our knowledge, it is thefirst time that LSSS and the converter are implemented in JavaScript environment. This sub module takes advantage of Web Assembly by adopting MCL-WASM [49].

• Utilization submodule packs some useful routines,e.g., encrypting users' retrieval key RKu, deriving AES key using PBKDF2, AES encryption and decryption etc. Theseroutines invoke the Web Cryptography API.

• MFAKE submodule helps to establish a secure channel for retrievalkey distribution. We implement the MFAKE protocol proposed in [43]. The storage module implements a cache storage layer in Indexed, which allows to store and obtain users' encrypted retrieval keys RKu. Meanwhile, italso maintains a cache in sessionStorage, which contains intermediate cipher texts (generated by offline encryption). M2. WebCloud Storage. This module also works in users' browsers and provides access and queryroutines of Web storage, which is used by the storage management submodule of M1. It provides storage in browsers for the WebCloud system, including data consumers' retrieval keys in Indexed and intermediate cipher texts in session Storage. M3. Cloud Crypto Module. This module implements cryptographic routines at the server side, which includes following submodules.

• CP-AB-KEM2 submodule implements the CUS, KUS, DSservices in the cloud. It includes cipher text initialization and outsourced decryption routines, i.e., algorithmsCTInit, CSKUpdate, CTU pdate, and Dec.Out. This submodule relies on MCL-C [49].

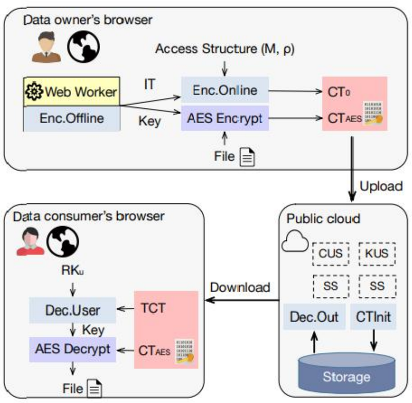
• Utility submodule containsuseful functionalities, including file processing, logging functionality.

• Serialization submodule converts bytes toABE ciphertext CT0 and converts transformed ciphertext TCT to bytes.

• Revocation submodule maintains therevocation list L. It provides add, delete and query functions of the list L to other submodules.M4. Key Management. This module works in PKG and contains two modules. The authentication moduleauthenticates users' identities via the multi-factor authenticated key exchange (MFAKE) protocol [43]. The keystorage

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE
ISSN: 2057-5688

module is responsible to store users' retrieval key securely and return a user's key on receiving anauthenticated user's request



(b) Outline of WebCloud

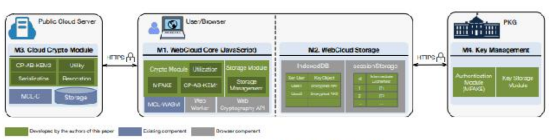


Fig. 2: System Architecture of WebCloud

Description of Algorithms We now elaborate the algorithms of WebCloud (cf. Fig. 1). Some acronyms are

listed in Table 1.

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

1) System Initialization. PKG runs the algorithm Setup() to generate a public key PK, a master secret key MSK anda cloud secret key CSK1. All the data consumers register themselves to PKG: **1) run the registration** phase ofMFAKE protocol [43] where PKG serves as the registration center; and 2) state a set of properties to indicate theiridentities. Then, PKG runs the algorithms KG() and KG.Random() to generate TKu and RKu for each dataconsumer. Further, PKG generates a certificate Tcloud for the public cloud server, which is used to establishsecure communication between the cloud server and users. Finally, PKG distributes PK to all the entities, CSK1and TKu along with Tcloud to the cloud, and keeps RKu for future distributions. The above-mentioned algorithmsare as follows: Setup(λ, U). On input a security parameter λ and an attribute universe U, PKG chooses a bilinearmap D = (G, GT , e, p), where p ∈ Θ(2λ) is the prime order of groups G and GT . The attribute universe U consistsof elements in Z ∗p . It chooses random generators g, h, u, v, w ∈ G, picks two random elements α, β1 ∈ Z ∗p . Itsets a counter ctr = 1. Finally, PKG outputs: PK = (D, g, h, u, v, w, e(g, g) α), MSK = (PK, α) and CSK1 = (csk1 = β1,ctr). KG(S, MSK). On input a master secret key MSK and an attribute set S = (A1, A2, . . . ,Ak) ⊆ Z ∗ p , PKG picks arandom element r ∈ Z ∗ p and computes K0 = g αw r , K1 = g r . For j from 1 to k, it picks random rj∈ Z ∗ p andcomputes Kj,2 = g rj , Kj,3 = (u Aj h) rj · v −r . PKG outputs a secret key SKu = (S, K0, K1, {Kj,2, Kj,3}j∈[1,k]).KG.Random(PK, SKu). On input a public key PK and a secret key SKu, PKG picks a random element τ ∈ Z ∗p , then itcomputes K0 0 = (K0) 1/τ , K0 1 = (K1) 1/τ . For j = 1 to k, it computes: K0 j,2 = (Kj,2) 1/τ , K0 j,3 = (Kj,3) 1/τ . PKG

outputs:RKu = τ,TKu = (S, K0 0 , K0 1 , {K0 j,2 , K0 j,3 }j∈[1,k]).

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

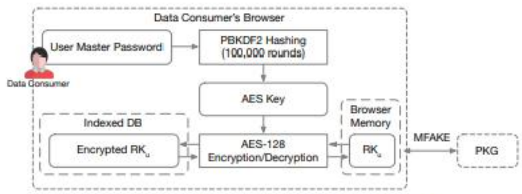


Fig. 3: Key Management of WebCloud

2) Key Management. To decrypt data in browsers, data consumers obtain their retrieval keys from PKG as shownin Fig. 3. To this end, the login-authentication phase of the MFAKE protocol [43], is run between a user's browserand PKG to establish a secure communication channel. The consumer's retrieval key RKu is transmitted to thebrowser through the secure channel and later be used in the browser's memory. If the user remains idle for aspecified period of time (e.g., 30 minutes), RKu is automatically erased from the memory and later use of RKurequires running the MFAKE protocol again. The login-authentication phase of [43] requires a user to enter a fewauthentication factors, which may cause poor usability. For ease of use, another option is provided. We derive a128-bit AES key with PBKDF2 from a user master password. RKu is then encrypted with AES and stored locally inIndexedDB. When necessary, the consumer is required to provide the user master password (with salt ifnecessary) to decrypt the locally encrypted RKu. Following the NIST standard [50], we require that the usermaster password must satisfy three requirements: 1) at least 8 characters in length as a memorized secret,

2) notappear in known dictionaries, and 3) be updated periodically. We remark that in this manner the usage ofpassword does not lead to low-level security as existing password-based solutions. The analysis is postponed toSection 5.1.

3) Data Encryption. To improve performance, data encrypIEEE Transactions on Dependable and SecureComputing,Volume:19,Issue:3,Issue Date:01.May-June.2022 7 tion procedure is divided into three parts asdepicted 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 operations in the encryption algorithm of CP-AB-KEM. On opening the WebCloud website, a Web worker (cf. Section 2.1) is created in background. During idletime, the worker runs the algorithm Enc.Offline() to generate a few intermediate cipher texts IT and keys Key. Idletime is defined as: a) no online encryption part is running, b) no user decryption part is running, and c) no AESencryption or decryption is running. We store (IT,Key) in session Storage, which is erased automatically bybrowsers after the Web page is closed. Online encryption in browser (after an access policy string and a file aregiven): The data owner uploads a file and specifics an access policy string, e.g., "(Employee and IT department) orManager". 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 session Storage. On input the access policy (M, ρ) and the intermediate cipher text IT, the algorithmEnc.Online() generates ABE cipher text CT0. The input file is encrypted with AES, using a 128-bit key derived fromKey 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

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

uploading to the cloud server. Ciphertext initialization in cloud:On receiving the uploaded file, the cloud parses ABE ciphertext CT0 from the file. It processes CT0 to CTctr bycalling the algorithm CTInit() and replaces CT0 with CTctr in the file. Finally, the updated file is stored in the cloud.The above-mentioned algorithms are as follows: Enc.Offline(PK, N0). On input a PK and a maximum bound of N0rows in any LSSS access structure, the Web worker in data owner's browser picks 3N0 + 1 random elements s, {λ 0i , xi , ti}i∈[1,N0] ∈ Z ∗ p , and computes Key = e(g, g) αs , C0 = g s . For i = 1 to N0 , it then computes: Ci,1 = w λ 0 I v ti , Ci,2 = (u xih) −ti , Ci,3 = g ti . The Web worker outputs: IT = (s,Key, C0, {λ 0 i , xi , ti , Ci,1, Ci,2, Ci,3}i∈[1,N0]).Enc.Online(PK,(M, ρ), IT). On input a public key PK, an LSSS access structure (M, ρ) [51], where M is an l×n matrix, an intermediate cipher text IT, data owner's browser picks n − 1 random elements (y2, . . . ,yn) ∈ Z ∗ p andconstructs a vector −→y = (s, y2, . . . , yn) T . It then computes a vector of shares of s as (λ1, . . . ,λl) T = M−→y .For i = 1 to k, compute Ci,4 = λi − λ 0 i , Ci,5 = ti(xi − ρ(i)). The browser outputs: CT0 = ((M, ρ), C0, {Ci,1, Ci,2, Ci,3,Ci,4, Ci,5}i∈[1,l]). CTInit(CT0, CSKctr). On input a ciphertextCT0 and a cloud secret key CSKctr = (cskctr, ctr), thepublic cloud computes C 0 0 = C 1/cskctr0 . For i = 1 to l, it computes: C 0 i,1 = C 1/cskctr i,1 , C 0 i,2 = C 1/cskctri,2 , C 0 i,3 = C 1/cskctr i,3 , C 0 i,4 = Ci,4/cskctr, C 0 i,5 = Ci,5/cskctr. The cloud outputs: CTctr = ((M, ρ), C0 0 , {C 0i,1 , C 0 i,2 , C0 i,3 , C0 i,4 , C0 i,5 }i∈[1,l]).

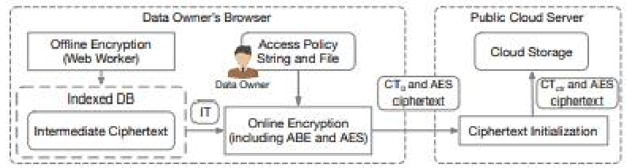


Fig. 4: Data Encryption Procedure of WebCloud

4) Data Decryption. As depicted in Fig. 5, file decryption is divided into two parts where the cloud server performsalmost all heavy computation. Outsourced decryption in cloud: On receiving a file down load request, the cloudserver checks that whether the user has been revoked. If revoked, the cloud rejects the request. Otherwise, itfinds the requested file from the cloud storage, parses CTctr from the file and transforms CTctr to TCT by callingthe algorithm Dec.Out(). The cloud packs TCT and AES ciphertext as a new file and returns the new file as aresponse to the request. User decryption in browser: In the browser side, TCT is parsed from the response. If theretrieval key RKu already exists in the browser memory, the key is used directly. Otherwise, the retrieval key RKuis obtained as in 2) Key Management. The algorithm Dec.User() is invoked to decrypt TCT to obtain encapsulatedkey Key. We derive the same AES key as in the encryption procedure. Finally, we decrypt the file with AES. Theabove-mentioned algorithms are as follows: Dec.Out(PK, CSKctr,TKu, CTctr). On input a public key PK, a cloudsecret key CSKctr = (cskctr, ctr), a conversion key TKu = (S, K0 0 , K0 1 , {K0 i,2 , K0 i,3 }i∈[1,k]) for an attribute set Sand a ciphertextCTctr for access structure (M, ρ), if S does not satisfy the access structure, the cloud serveroutputs ⊥. Otherwise, it calculates a set I = {i : ρ(i) ∈ S} P and computers the constants {ωi∈Zp}i∈I such that i∈Iωi· Mi = (1, 0, . . . , 0), where Mi is the i-th row of the matrix M. It then computes: B =e(w P i∈I C 0 i,4ωi , K0 1) · Y i∈Ie(C 0 i,1 , K0 1) ωi · Y i∈I (e(C 0 i,2 · u C 0 i,5 , K0 j,2) · e(C 0 i,3 , K0 j,3))ωi The cloud outputs TCT = (e(C 0 0 , K00 0)/B) cskctr = e(g, g) αs/τ . Dec.User(TCT, RKu). On input a TCT and a retrieval key RKu = τ , data consumer'sbrowser outputs the encapsulated key Key = TCTRKu = e(g, g) αs .

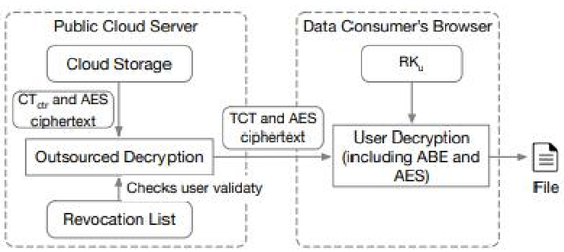


Fig. 5: Data Decryption Procedure of WebCloud

5) Cloud Secret Key Update. The cloud secret key CSK is of great importance to the revocation mechanism. If CSKleaks, the user revocation functionality is in vain. Therefore, we update CSK periodically or in emergencysituations (e.g., CSK is leaked or stolen). Cloud Secret Key Update: This procedure is depicted in Fig. 6. Assumecurrent cloud secret key is CSKctr. When CSKctr is required to be updated, the public cloud server invokes thealgorithm CSKUpdate() to generate a new key CSKctr0 and an increment Δctr0 . The cloud then updates all storedABE ciphertextsCTctr to CTctr0 by calling the algorithm CTUpdate(). Once all ciphertexts have been updated, thecloud sets current secret key to CSKctr0 , and deletes CSKctr and all old ABE ciphertextsCTctr from its storage.The above-mentioned algorithms are as follows: CSKUpdate(PK, CSKctr). On input a public key PK and a cloudsecret key CSKctr = (cskctr, ctr) where ctr∈ {1, 2, . . .}, the cloud server updates the counter ctr0 = ctr+1 and picksa random element βctr0 ∈ Z ∗p , computes cskctr0 = cskctr · βctr0 = Qctr0 i=1 βi . The cloud outputs the updatedcloud secret key CSKctr0 = (cskctr0 , ctr0) and the increment Δctr0 = βctr0 . CTUpdate(ctr0 , CTctr0−1, Δctr0). Oninput an updated counter ctr0 ∈ {2, 3, . . .}, a ciphertext CTctr0−1 = ((M, ρ), C0, {Ci,1, Ci,2, Ci,3, Ci,4, Ci,5}i∈[1,l])and an increment Δctr0 = βctr0 , the cloud server computes C 0 0 = C 1/βctr0 0 . For j = 1 to l, the cloud computesC 0 i,1 = C 1/βctr0 i,1 , C0 i,2 = C 1/βctr0 i,2 , C0 i,3 = C 1/βctr0 i,3 , C0 i,4 = Ci,4/βctr0 , C0 i,5 = Ci,5/βctr0 . Itoutputs: CTctr = ((M, ρ), C0 0 , {C 0 i,1 , C0 i,2 , C0 i,3 , C0 i,4 , C0 i,5 }i∈[1,l]). 6) Key and User Revocation.

WebCloud supports both key and user revocation, and does not support attribute revocation. To revoke a userkey, PKG runs KG(S, MSK) to generate a new secret key SK0 u and KG.Random(PK, SK0 u) to obtain RK0 u and TK0u . PKG replaces RKu with RK0 u and distributes TK0 u to the public cloud, who deletes previous transformationkey TKu directly. The key revocation is taken effect immediately after the cloud updates its transformation key. Torevoke a user from the system, PKG sends a revocation request to the public cloud server to revoke a dataconsumer, where the cloud inserts an entry to the revocation list L by calling the algorithm Revoke(). Therevocation is taken effect immediately after the insertion. On receiving a user's file download request, the cloudcompares the user identity u against the list L and rejects the request if a match is found. Without the help of thecloud, data

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

consumers cannot decrypt files individually. The size of the list L is the same as the number of revokeduser in the system. Many efficient algorithms exist for finding an element from a(n) (ordered) list e.g., binarysearch or hash table. The above-mentioned algorithms are as follows: Revoke(u, L). On input a user identity u,and a revocation list L = {(id)} where id is the user identity, the cloud server adds an entry (u) to the list L, i.e., L 0 =L ∪ {(u)}. 4.2 A Tailored CP-AB-KEM for WebCloudCiphertext-policy attribute-based key encapsulation mechanism(CP-AB-KEM) is an important component for WebCloud(Section 4.1). It simultaneously achieves offlineencryption and outsourced decryption, robust and immediate user revocation, while only a small number ofcomputations are left to the user. The proposed CP-AB-KEM derives from the offline encryption and outsourceddecryption techniques in [19], [20], [52], and combines the immediate user revocation mechanism in [53]. Forcompleteness, we give the syntax in Appendix A and elaborate the scheme in the supplementary material. Weemphasize that this CP-ABKEM is useful in many scenarios. Correctness: We require the standard correctnessproperty: for an attribute universe U, a user identity U and λ, N, N0 ∈ N, for all (PK, MSK, CSK1) ∈ Setup(λ, U), allSKu∈ KG(S, MSK), all (RKu,TKu) ∈KG.Random(SKu), all IT ∈Enc.Offline(PK, N0), all CT0 ∈Enc.Online(PK,(M, ρ), IT),all CTctr∈CTInit(CT0, CSKctr), all (CSKctr0 , Δctr0) ∈CSKUpdate(PK, CSKctr), all CTctr0 ∈CTUpdate(ctr0 , CTctr0−1,Δctr0), all TCT ∈Dec.Out(PK, CSKctr,TKu, CTctr), if S satisfies (M, ρ) and the user u was not revoked, Dec.User(TCT, RKu) outputs the encapsulated Key. Security: The security proof of the proposed CP-ABKEM scheme is givenin Appendices B and C.

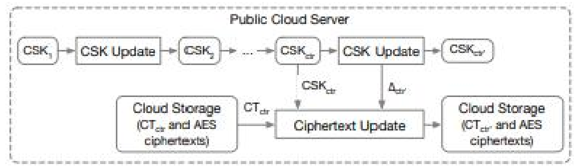


Fig. 6: Cloud Secret Key Update Procedure of WebCloud

In the WebCloud system, all users' secret key are derived from the master secret key MSK, which is stored in thetrusted PKG. In reality, a single point of failure, e.g., loss of MSK, will immediately lead to system failure. It is ofgreat importance to provide simple mechanisms to enhance the security of MSK and the system. An effective wayis secret sharing, i.e., splitting the MSK into multiple pieces. Without loss of generality, we consider a (t, n)threshold scheme. As shown in Fig. 7, there are 1 root PKG, n child PKGi(1 ≤ i ≤ n) and a combiner PKGc. PKG isresponsible for generating PK, MSK and n shares of α, and distributes the i-th share MSKi to PKGi . Whengenerating a user's secret key SKu, PKGi generates a partial secret key SKu,i using its share. The combiner PKGccombines any t partial keys to SKu, and invokes KG.Random to obtain TKu and RKu.

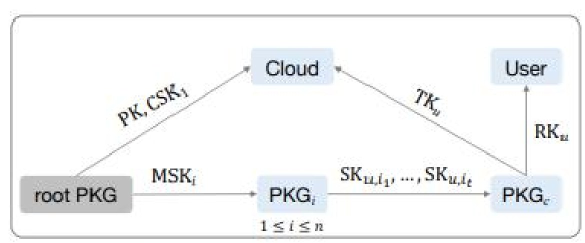


Fig. 7: Secret Sharing of Master Secret Key

6. IMPLEMENTATION

INTERNATIONAL JOURNAL OF TECHNO-ENGINEERING
IJTE

ISSN: 2057-5688

6.1 Data Owner

In this module, the data provider uploads their encrypted data in the Cloud server. For the securitypurpose the data owner encrypts the data file and then store in the server. The Data owner can havecapable of manipulating the encrypted data file and performs the following operations Register andLogin, Attackers, Upload File, View Files, Verify data(Verifiability), View and Delete Files, ViewAll Transactions.

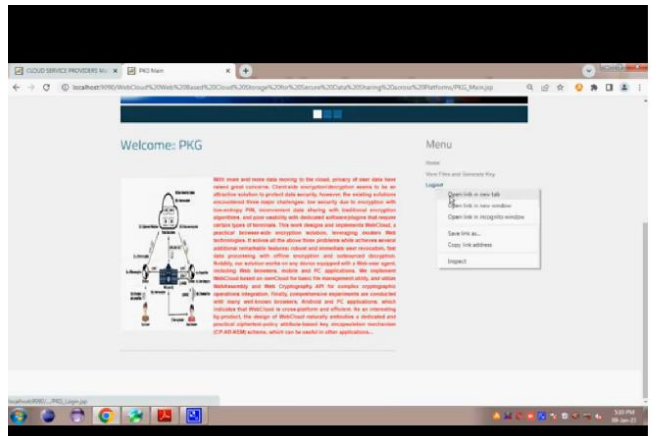
6.2 Cloud Service Provider

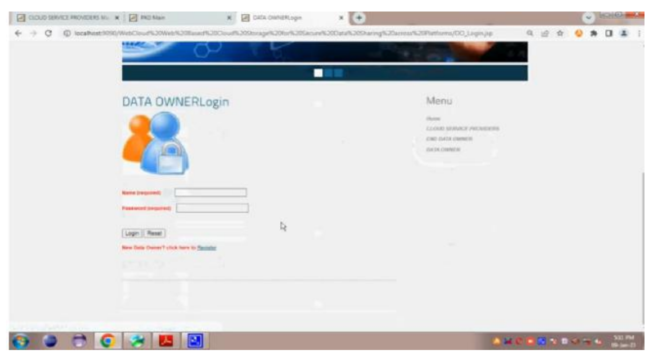
The Cloud server manages which is to provide data storage service for the Data Owners. Dataowners encrypt their data files and store them in the Server for sharing with data consumers. Toaccess the shared data files, data consumers download encrypted data files of their interest from theServer and then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

6.3 User

In this module, the user can only access the data file with the secret key. The user can search the filefor a specified keyword. The data which matches for a particular keyword will be indexed in thecloud server and then response to the end user and performing the following operations Registerand Login, My Profile, View Files, Search Files, Search Ratio, Top K Search, Req Search Control.

7. OUTPUT SCREENS





8. CONCLUSION

We propose Web Cloud, a practical client-side encryption solution for public cloud storage in theWeb setting, where users do cryptography with only browsers. We analyze the security of

WebCloud and implement Web Cloud based on own Cloud and conduct a comprehensive performanceevaluation. 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 inmany other applications.

9. 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: 02/icloud-data-breach-hacking-and-nude-celebrity –photos

[3] T. Hunt, "Hacked dropbox login dataof 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," inFourth International Conference on Network and System Security, NSS 2010, Melbourne, ictoria,Australia, September 1-3, 2010, Y. Xiang, P. Samarati, J. Hu, W. Zhou, and A. Sadeghi, Eds. IEEEComputer 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 ofterminals," in 4th IEEE International Conference on Cloud

ISSN: 2057-5688

Networking, CloudNet 2015, NiagaraFalls, 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 provablysecure 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 abeciphertexts." InUSENIX Security Symposium, vol. 2011, no. 3, 2011.

[20] S. Hohenberger and B. Waters, "Online/offline attributebased encryption," in InternationalWorkshop 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 outsourcedattribute-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 5 th 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/ope npgpjs

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