Web Cloud Web-Based Cloud Storage for Secure Data Sharing Across Platforms

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.

**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.

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 encryptionsolution for public cloud storage, whicheffectively combines modern Web techniques andcryptographic algorithms. WebCloud involves of akey management mechanism, a dedicated attribute basedencryption scheme and a high-speed implementation.More importantly, WebCloud is crossplatform(including major browsers, Android andPC) 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 theexisting ABE schemes suffer from high computationaloverhead, or some vital missing functionalities,e.g., inefficient data encryption, robust andimmediate user revocation, offline encryption andoutsourced decryption simultaneously. To solve thisproblem, we propose a dedicated ciphertext-policyattribute-based access control mechanism. The proposedscheme 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.

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL