Secret Sharing for Attorney-Client Data in a Multi-Provider Cloud Architecture

Joseph Zhang

Introduction

The accelerated employment of cloud computing among law firms arises from the **increased convenience in sharing legal information among business partners.** Specific benefits include the wide accessibility and inter-organizational sharing of client documents, which are all stored in one centralized place. Due to this ubiquitous access, improved and less costly legal services are created, opening new legal business models. However, this new cloud computing paradigm implies **severe security and privacy risks**, raising widespread concerns about the privacy of this vital, confidential information (Fig. 1). Law firms have access to important and confidential information, often documents and communications vital to businesses. Few of the current approaches in literature take into account cloud providers who could gain access to sensitive client data due to **improper encryption algorithm implementations or compromised keys**. This concern prompts **my novel architecture for sharing client documents with parallel, independent cloud providers.**

Case Study

From the 2019 Legal Technology Survey from the American Bar Association, I evaluated the current security of cloud computing use in the legal industry (see Figs. 1-3). I discovered that 26% of law firms have experienced a security compromise within the past year, an increase from previous years, confirming that data breaches are a significant threat. Furthermore, statistics from the survey show an **alarming neglect in standard precautionary measures** in securing cloud data given lawyers' high concern about confidentiality and control over information. For instance, only 35% of law firms use Secure Socket Layers for securing communications.

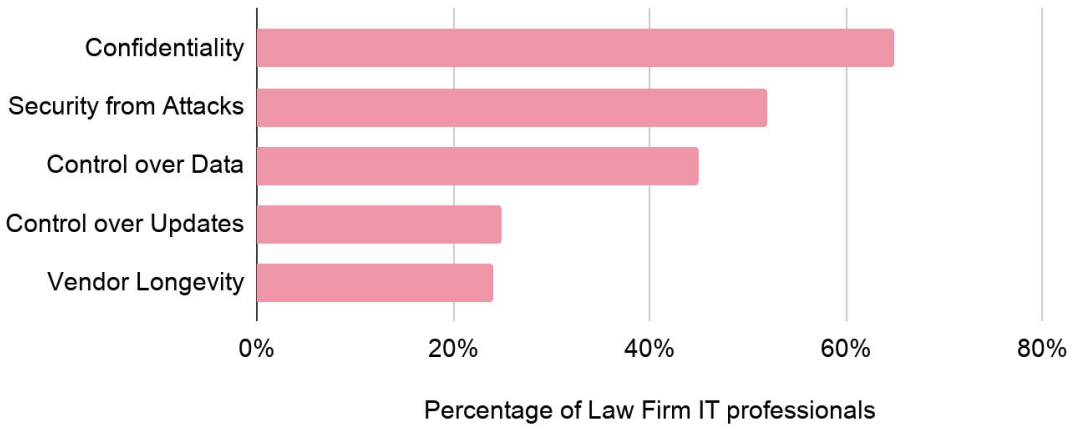

Fig. 1
Primary Concerns Among Lawyers Regarding Cloud Computing.

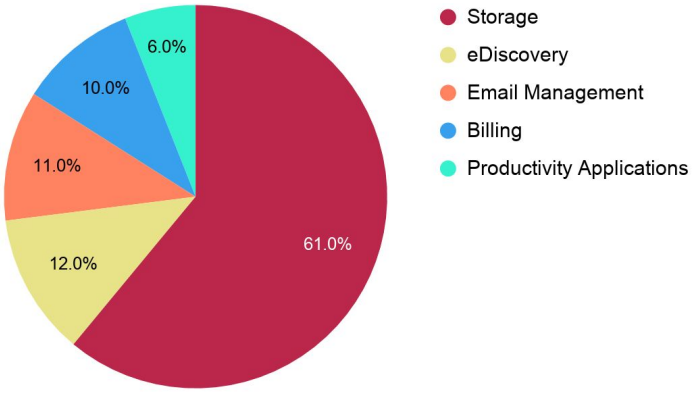

Fig. 2
Primary Cloud Usage within Law Firms.

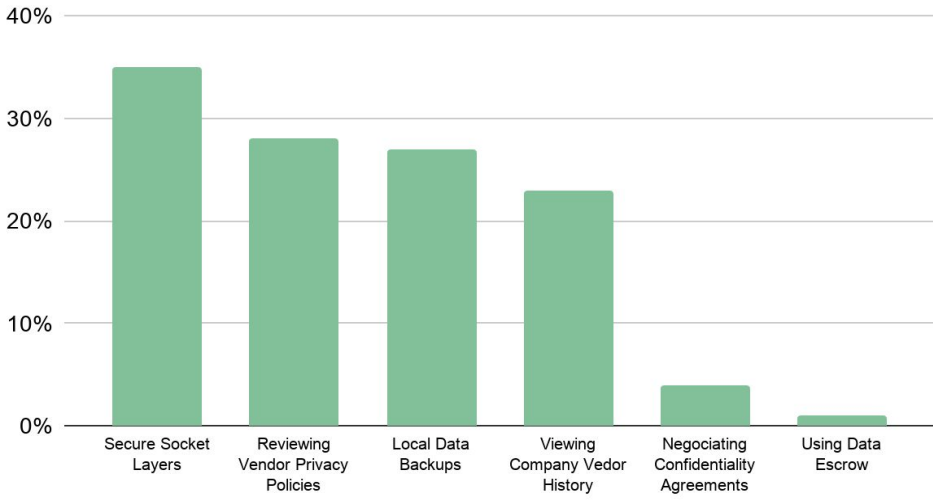

Fig. 3
Common Security Measures used in Law Firms.

Literature Review

The keyword search of the scope of cloud computing within the legal industry revealed 1440 publications, with only 10 regarding security and privacy issues (see Table I.). The current mechanisms in place concern novel encryption algorithms, traitor tracing, and improving access control. **None of the papers consider probing cloud providers who often reserve the right to monitor client information.** If decryption keys become compromised or the cloud provider uses improper encryption implementations, an attacker can gain access to the information as well. My scheme aims to address similar criteria found in the current literature as well as **further sophisticated privacy issues, such as simultaneous unlinkability between clients and attorneys and confidentiality of the existence of a document.**

Database	ProQuest	EBSCOhost	JSTOR	Science Direct	Total
Hits	532	60	466	382	1440
Relevant Hits	3	6	0	1	10

Table I. Results of the Keyword Search of Cloud Computing in the Legal Industry.

Engineering Goal

Confirmed as important by the case study and publications found in the literature review, the following security and privacy objectives of my secret sharing architecture for sharing legal information in the Cloud center around:

- Authenticity of client documents during storage.
- Availability of information in the Cloud.
- Confidentiality of the content of documents against external parties, including cloud providers.
- Unlinkability between documents and clients.
- Efficiency and usability.

Development Procedure

1. Concept Generation

After establishing the engineering goal, I formulated and compared multiple possible solutions, including novel access control policies and encryption algorithms. However, in order to satisfy the presented criteria, I decided to feature a combination of established as well as novel methods, notably including secret sharing which ensures availability and additional privacy in an efficient and elegant scheme.

2. Architecture Design

Throughout the design process, I focused on the two parties: law firms and cloud providers. My final architecture methods are shown in Figs. 4-5. Following the design criteria, I ensured that the whole storage and retrieval process is protected against unauthorized access and modification through various cryptographic encryption and signature operations. This bolsters security along with adding scalability since law firms can customize the system to their own conditional access policies. However, my approach boasts end-to-end encryption and privacy from probing cloud providers, assuming that classical network security protocols are administered, such as HTTPS for Internet data exchange and Virtual Private Networks between all party communications.

3. Implementation

I developed the document storage and retrieval process after understanding and choosing the specific secret sharing and encryption algorithm I will employ (see Table II.). Most of my redesigning and testing centered around finding efficient implementations for each component of the architecture. I wrote in C++ for maximum efficiency as well as the benefits of object oriented programming. Finally, I created a web application framework for the program to enhance its **usability**. I have posted a short demonstration of its highlights in the Photos/Videos folder. I worked on a **Mac OS notebook computer with a 1.4 GHz dual core processor and 4GB RAM on Visual Studio Code 1.42.1.** All my code is documented on Github:

Constructions: https://github.com/joezbub/Shamir-for-Cloud-Environments

Complete Framework: https://github.com/joezbub/LawCrypt

4. Testing and Rewriting

I took performance metrics of my software's runtime as shown in multiple charts (Figs. 8-10), evaluating this against my prior engineering design criteria. I executed all my tests on the consumer-grade hardware a lawyer has access to and made necessary revisions and retests upon

noticing errors or optimizations. I intend my final assessments to describe the whole architecture comprehensively, even comparing the secret sharing implementations against the popular AES algorithm (Figs. 8-9). This will ideally present a complete view of the feasibility of this system for future application in an authentic environment.

Multi-Cloud Secret Sharing Architecture

Storage Process

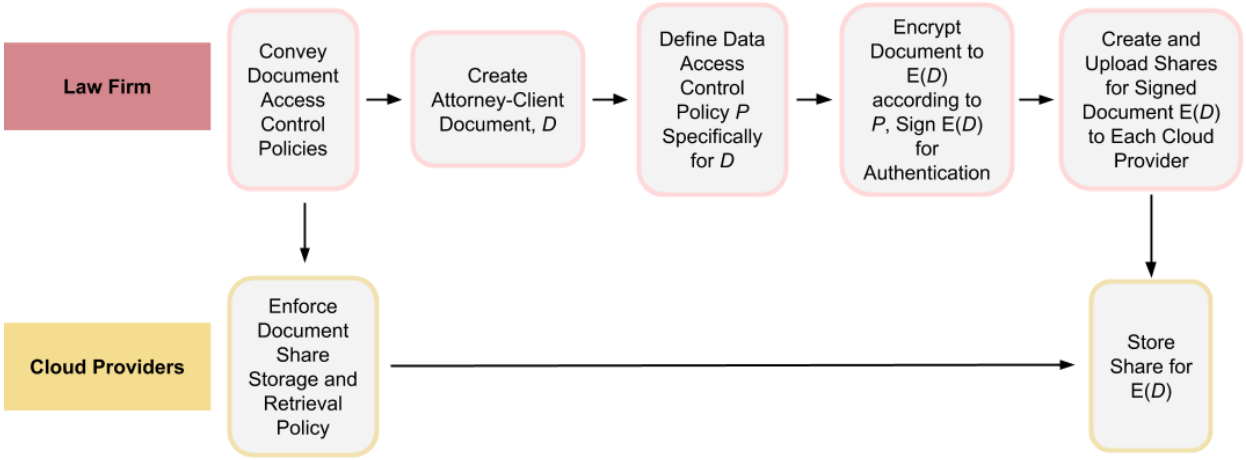

Fig. 4 Law Firm Storage Architecture

Retrieval Process

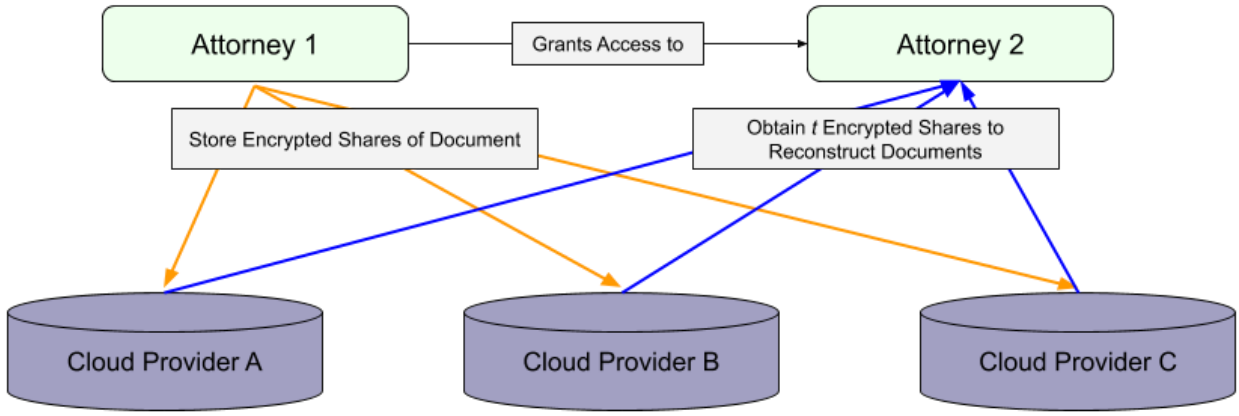

Fig. 5 Secret Sharing Retrieval Method

Encryption Construction Assessment

After assessing multiple secret-key encryption algorithms, I chose to compare two **authentication constructions** since both satisfy the criteria of **ensuring integrity and thus complete authenticity**. I evaluated AES256-GCM against ChaCha20-Poly1305, and I found that the latter is **significantly more efficient** (see Table II.) and sufficiently secure due to its use of a **256-bit key**. I suspect that this is because AES is usually performed with dedicated CPU instructions, which is not suitable on consumer hardware. Therefore, all my client document encryption and decryption is through **ChaCha20-Poly1305**.

Cipher Measured	40 bytes	576 bytes	1500 bytes	Internet Mix
AES256-GCM	43.47	34.96	35.02	35.57
ChaCha20-Poly1305	37.84	14.13	14.65	15.95

Table II. Evaluation of the two AEAD constructions; units in cycles per byte.

Secret Sharing Overview

Secret sharing is often defined as a threshold scheme, where **only t or more out of n participants can derive the original secret, where t ≤ n.** Each participant is given some partial information called a share. The shares are distributed so that no participant knows the share given to another participant.

In my case, the owner of the secret is the attorney or law firm, and the participants are the cloud providers. My implementation of a **(2, 3)-threshold scheme provides data redundancy for increased availability** since a share can be lost without preventing the reconstruction of the original secret. Therefore, secret sharing is a necessary preventative measure in the case of cloud provider key loss or simply adversaries revoking access to documents.

Evaluation of Secret Sharing Methods

The threshold scheme centers around **polynomial interpolation in a finite field.** We generate

the secret shares using the following equation:

$$f(x) = s + a_1 x + a_2 x^2 + \ldots + a_{t-1} x^{t-1} \bmod p$$

where p is a prime number, $a_i \in Z_p$, $i = [1..., 2]$, s is the secret, and $a_1,...,a_{t-1}$ and distinct $x_1,...,x_n$ are randomly chosen. The owner can reconstruct the shares by calculating the **Lagrange interpolating polynomial**:

$$f(x) = \sum_{j=1}^{t} y_j \prod_{k=1, k \neq j}^{t} \frac{x - x_k}{x_j - x_k} \pmod p$$

where x_j and y_j are the first equation's input and output respectively. The importance of *mod p* is shown in Fig. 7 since without it, an attacker can gain insights as points must lie along a smooth curve, reducing possible values of unknown points. Thus, **secret sharing has perfect privacy (confidentiality and unlinkability) and space efficiency.**

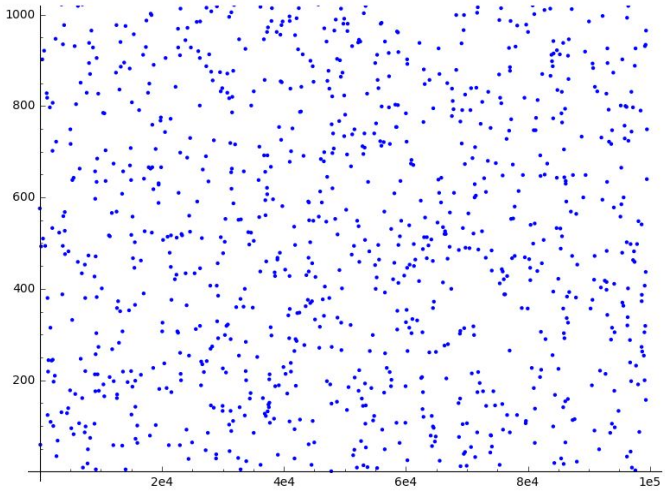

Fig. 7

Polynomial curve over a finite field is disjoint.

I measured the runtimes of secret sharing with a dataset of different file types and sizes (0-135 MB). In total, I tested **59 files in 29 different file types**, and I ran AES-256-CFB and secret sharing at the same time (Figs. 8-9). I repeated the experiments **10,000 times** and plotted the average to decrease sampling variability and the influence of outliers. Compared with AES, my secret sharing implementation is about **39% faster for splitting and 36% faster for combining**.

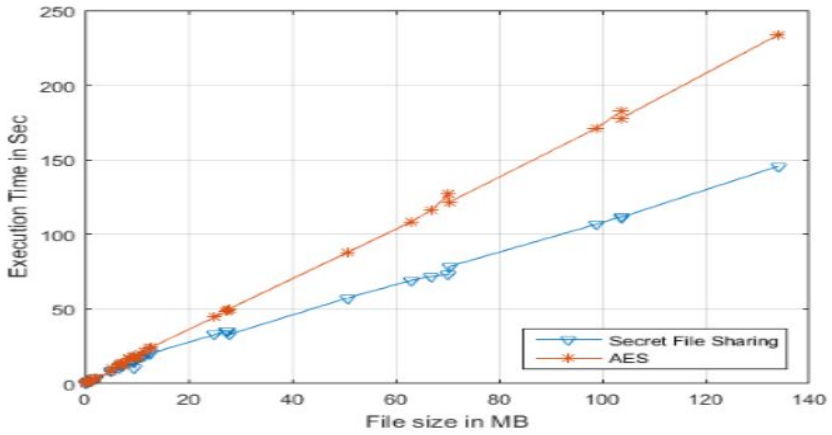

Fig. 8
Performance Comparison of Creating Shares with a (2, 3) Scheme.

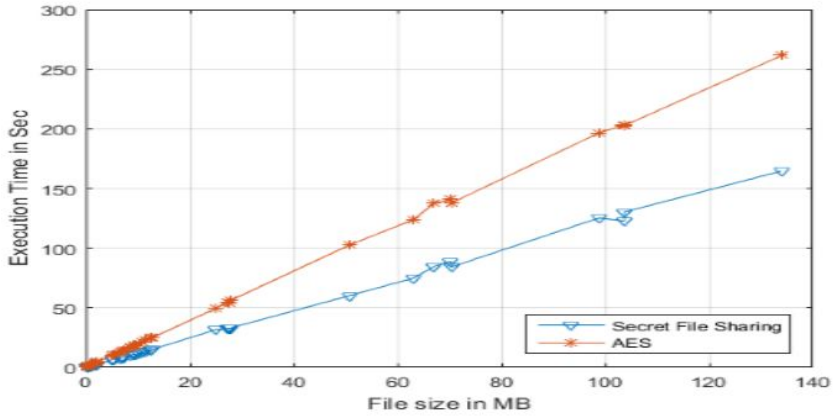

Fig. 9
Performance Comparison of Reconstructing Secret with a (2, 3) Scheme.

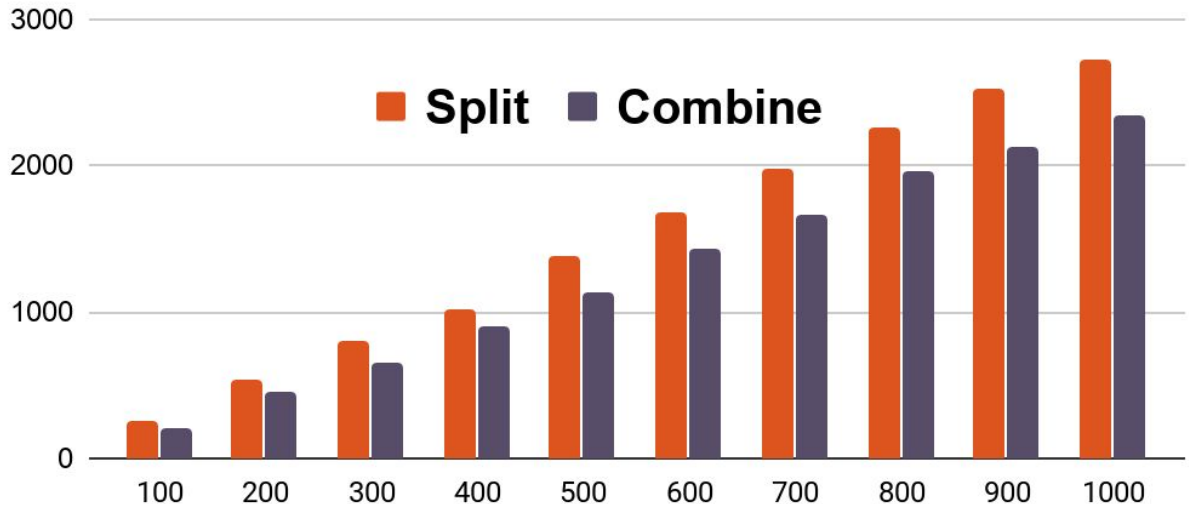

Fig 10.
Total Architecture Storage and Retrieval Runtimes for Different KB Document Sizes in Milliseconds.

Discussion

Based on Figs. 8-10, it is evident that my architecture is efficient even relative to other common encryption algorithms. The runtime lead of secret sharing increases with file size (see Figs. 8-9), ultimately giving it a substantial speed advantage over AES, widely used for encrypting data in the Cloud today. Due to numerous retests and revisions of the algorithms, my architecture adheres to the mathematics governing this scheme, guaranteeing perfect privacy and data availability. Furthermore, my website offers a usable, streamlined interface for the architecture.

Overall, my experimental results indicate the **low computational overhead** of adding my secret-sharing approach to a multi-cloud environment, **even on consumer grade hardware**. This is important for integrating law firms as active participants into my architecture in the future. **The efficient combination of algorithms satisfies the engineering goal as ChaCha20-Poly1305 warrants authenticity and privacy and secret sharing ensures availability and privacy while I preserve usability through the user-friendly website.** Compromising this system transpires only if multiple cloud providers collude, which is still unlikely given that the shares are additionally encrypted.

Conclusion

From the real-world case study and literature review, I established the need for secure law firm document sharing under the convenience and inexpensiveness of a cloud environment. I indicated current security and privacy challenges and goals of my multi-provider cloud architecture. In addition to classical encryption and other security measures, this architecture features secret sharing as an important measure to distribute attorney documents as fragments to different cloud providers, providing additional privacy protection in the probing cloud providers or broken encryption algorithm implementations.

I implemented the secret sharing scheme and the authentication construction, ChaCha20-Poly1305, on a usable website architecture interface, and performed code revisions and experiments on storing and retrieving client documents. These experiments indicate a low computational overhead while satisfying the engineering criteria, giving good indicators to my architecture's feasibility.

Future Work

My main goal in the future is to fully implement the secret sharing architecture specifically tailored to certain partner law firms where I will additionally evaluate corresponding

security assumptions and processes. This would give me a more realistic understanding of my work. I also aim to better involve the client into my architecture, such as through giving them partial control over shares of their documents. Particular challenges for my work include addressing the problem of ownership of information, reliably auditing and amending stored documents, and client consent for data access and access revocation.

Selected References

1. A. Shamir, “How to share a secret,” Commun. ACM, vol. 22, no. 11, pp. 612–613, 1979.
2. Aisha Abdallah, Mazleena Salleh, "Secret sharing scheme security and performance analysis", Computing Control Networking Electronics and Embedded Systems Engineering (ICCNEEE) 2015 International Conference on, pp. 173-180, 2015.
3. Philip J. Favro, Inviting Scrutiny: How Technologies Are Eroding the Attorney-Client Privilege, 20 Rich. J.L. & Tech 2 (2013), pp. 73-87.
4. “The Tide Has Turned and The Cloud Is Here.” Legal IT Professionals, 2012, www.legalitprofessionals.com/wpcs/cloudsurvey2012.pdf.
5. D. R. Stinson, An Explication of Secret Sharing Schemes. Des. Codes Cryptography 2 (4), 357-390, 1992.
6. W. Stallings, Cryptography and Network Security: Principles and Practice. Prentice Hall, 2010.