International Journal of Research

International Journal of Research (IJR)

e-ISSN: 2348-6848, p- ISSN: 2348-795X Volume 2, Issue 08, August 2015

Available at http://internationaljournalofresearch.org

Security Schemes for Sharing Secret Data and Key Distribution in Cloud Storage

A.Padmavathi[1]& Dr.B.M.G Prasad[2]

[1]Student,Dr.K.V.S.W College of Engineering for Women, Kurnool, Andhra Pradesh

[2]Professor,Dr.K.V.S.W College of Engineering for Women, Kurnool, Andhra Pradesh

Abstract:

Secret sharing *refers to methods for distributing a secret amongst a group of participants, each of whom is allocated a share of the secret. The secret can be reconstructed only when a sufficient number, of possibly different types, of shares are combined together; individual shares are of no use on their own. In one type of secret sharing scheme there is one dealer and n players. The dealer gives a share of the secret to the players, but only when specific conditions are fulfilled will the players be able to reconstruct the secret from their shares. The dealer accomplishes this by giving each player a share in such a way that any group of t (for threshold) or more players can together reconstruct the secret but no group of fewer than t players can. Such a system is called a (t, n)-threshold scheme (sometimes it is written as an (n, t)-threshold scheme). In this paper we are sharing various secret sharing methods.*

Keywords: Access control; authentication; attribute-based signatures; attribute-based encryption; cloud storage

1. INTRODUCTION

Cloud storage is a model of data storage where the digital data is stored in logical pools, the physical storage spans multiple servers (and often locations), and the physical environment is typically owned and managed by a hosting company. These cloud storage providers are responsible for keeping the data available and accessible, and the physical environment protected and running. People and organizations buy or lease storage capacity from the providers to store user, organization, or application data. Cloud storage services may be accessed through a co-located cloud computer service, a web service application programming interface (API) or by applications that utilize the API, such as cloud desktop storage, a cloud storage gateway or Web-based content management systems.

Cloud storage is based on highly virtualized infrastructure and is like broader cloud computing in terms of accessible interfaces, near-instant elasticity and scalability, multi-tenancy, and metered resources. Cloud storage services can be utilized from an off-premises service (Amazon S3) or deployed on-premises (ViON Capacity Services)

Cloud storage typically refers to a hosted object storage service, but the term has broadened to include other types of data storage that are now available as a service, like block storage.

Object storage services like Amazon S3 and Microsoft Azure Storage, object storage software like Openstack Swift, object storage systems like EMC Atmos and Hitachi Content Platform, and distributed storage research projects like OceanStore and VISION Cloud are all examples of storage that can be hosted and deployed with cloud storage characteristics.

Cloud storage is:

- Made up of many distributed resources, but still acts as one - often referred to as federated storage clouds
- Highly fault tolerant through redundancy and distribution of data
- Highly durable through the creation of versioned copies
- Typically eventually consistent with regard to data replicas

1.1 Attack surface area

Outsourcing data storage increases the attack surface area.

1.When data is distributed it is stored at more locations increasing the risk of unauthorised physical access to the data. For example, in cloud based architecture, data is replicated and moved frequently so the risk of unauthorised data recovery increases dramatically. (e.g. disposal of old equipment, reuse of drives, reallocation of storage space) The manner that data is replicated depends on the service level a customer chooses and on the service provided. Different cloud vendors offer different service levels. Risk of unauthorized access to data can be mitigated through the use of encryption, which can be applied to data as part of the storage service or by on-premises equipment that encrypts data prior to uploading it to the cloud.

2.The number of people with access to the data who could be compromised (i.e. bribed, or coerced) increases dramatically. A single company might have a small team of administrators, network engineers and technicians, but a cloud storage company will have many customers and thousands of servers and therefore a much larger team of technical staff with physical and electronic access to almost all of the data at the entire facility or perhaps the entire company.[17] Encryption keys that are kept by the service user, as opposed to the service provider limit the access to data by service provider employees.

3.It increases the number of networks over which the data travels. Instead of just a local area network (LAN) or storage area network (SAN), data stored on a cloud requires a WAN (wide area network) to connect them both.

4. By sharing storage and networks with many other users/customers it is possible for other customers to access your data. Sometimes because of erroneous actions, faulty equipment, a bug and sometimes because of criminal intent. This risk applies to all types of storage and not only cloud storage. The risk of having data read during transmission can be mitigated through encryption technology. Encryption in transit protects data as it is being transmitted to and from the cloud service. Encryption at rest protects data that is stored at the service provider. Encrypting data in an on-premises cloud service on-ramp system can provide both kinds of encryption protection.

1.2 Other concerns

- Security of stored data and data in transit may be a concern when storing sensitive data at a cloud storage provider[10]

- Users with specific records-keeping requirements, such as public agencies that must retain electronic records according to statute, may encounter complications with using cloud computing and storage. For instance, the U.S. Department of Defense designated the Defense Information Systems Agency (DISA) to maintain a list of records management products that meet all of the records retention, personally identifiable information (PII), and security (Information Assurance; IA) requirements
- Cloud storage is a rich resource for both hackers and national security agencies.

- Piracy and copyright infringement may be enabled by sites that permit filesharing. For example, the Codex Cloud ebook storage site has faced litigation from the owners of the intellectual property uploaded and shared there, as have the GrooveShark and YouTube sites it has been compared to.
- The legal aspect, from a regulatory compliance standpoint, is of concern when storing files domestically and especially internationally.

2. SECRET SHARING SCHEMES

Secret sharing schemes are ideal for storing information that is highly sensitive and highly important. Examples include: encryption keys, missile launch codes, and numbered bank accounts. Each of these pieces of information must be kept highly confidential, as their exposure could be disastrous, however, it is also critical that they should not be lost. Traditional methods for encryption are ill-suited for simultaneously achieving high levels of confidentiality and reliability. This is because when storing the encryption key, one must choose between keeping a single copy of the key in one location for maximum secrecy, or keeping multiple copies of the key in different locations for greater reliability. Increasing reliability of the key by storing multiple copies lowers confidentiality by creating additional attack vectors; there are more opportunities for a copy to fall into the wrong hands. Secret sharing schemes address this problem, and allow arbitrarily high levels of confidentiality and reliability to be achieved.

Secret sharing schemes are important in cloud computing environments. Thus a key can be distributed over many servers by a threshold secret sharing mechanism. The key is then reconstructed when needed. Secret sharing has also been suggested for sensor networks where the links are liable to be tapped by sending the data in shares which makes the task of the eavesdropper harder. The security in such environments can be made greater by continuous changing of the way the shares are constructed.

2.1 Shamir's scheme

In this scheme, any t out of n shares may be used to recover the secret. The system relies on the idea that you can fit a unique polynomial of degree $(t-1)$ to any set of t points that lie on the polynomial. It takes two points to define a straight line, three points to fully define a quadratic, four points to define a cubic curve, and so on. That is it takes t points to define a polynomial of degree $t-1$. The method is to create a polynomial of degree $t-1$ with the secret as the first coefficient and the remaining coefficients picked at random. Next find n points on the curve and give one to each of the players. When at least t out of the n players reveal their points, there is sufficient information to fit a $(t-1)$th degree polynomial to them, the first coefficient being the secret.

2.2 Blakley's scheme

Around the same time as the publishing of Shamir’s scheme, George Blakley published his own secret sharing scheme. Similar to Shamir’s scheme, Blakley’s scheme defined a threshold scheme based on hyperplane intersections, instead of polynomial interpolation. The hyperplanes used for this scheme are all in t dimensions, which allows t of the hyperplanes to intersect at a single point, inside of a finite field.

A. Share Distribution

Distribution of shares in Blakley’s scheme begins by generating a random t-dimensional point in a t-dimensional finite field, F. Regardless of how the random point x is distributed, the first coordinate of the point is then set to the secret being shared. Once the secret point is determined, t values of a are generated for each participant. Using the secret point and these values, the shares for each participant are generated by:

$y_i=a_{i1}x_1+a_{i2}+x_2+...+a_{it}x_t=y_i$

Once these shares are generated, each participant is given their share, which is one of the values of Y . Additionally, all values of a are made public, as these values are not sensitive to the security of the scheme

B. Key Reconstruction

For reconstruction of a secret in this scheme, the t participants who combine their shares each will have an equation with their corresponding a and y values, but with the x values unknown. Each equation is in the same form as (9), and with t participants, a matrix equation is generated by combining all of the shares:

$A_x=y$

The solution of the system of equations will give t values of x , which will match the coordinates of the secret point. The secret is then found simply by taking the first coordinate of the point, which is the same way the secret was hidden. Note the similarities to Shamir's scheme in how the reconstruction occurs. In Blakley's scheme, however, the participants are solving for the x values directly, instead of the a values in Shamir's. However, no simpler method for reconstruction is known for this scheme, aside from simpler methods of solving the system.

C. Security

Since each share is a value on the same range as the coordinates of the finite field, and the fact that one of the coordinates is the secret, this scheme has an information rate of 1, just like Shamir's scheme. However, this scheme is not ideal due to the fact that it is not a PSS. This is because as the number of shares combined increase, the number of possibilities for the secret point decrease. For example, each participant knows that the secret point exists on their hyperplane, which narrows down all possible points. For t-1 shares combined, the participants know of a line in which the point lies. This makes Blakley's Scheme not a perfect scheme, as security is decreased with t-1 shares .

2.3 Short Share Secret Sharing

A short share secret sharing scheme combines a computationally secure encryption scheme and IDA in order to generate shares of minimal size, while retaining the needed security. Distribution of a secret S in this scheme begins with encrypting the secret using the selected computationally secure encryption algorithm. Once this encryption is performed, the ciphertext is split into w fragments through the IDA, with t needed for successful reconstruction, where w and t are the same as the security parameters for the secret sharing scheme intended to be used. A perfect (t,w) secret sharing scheme is then used in order to generate the w shares desired. Each participant is then given a random share, and fragment of the encrypted file.

In this scheme, the size of each share given to a participant is approximately |F|/t+|K|, much lower than shares of size |F| that one would obtain from using Shamir's scheme alone.

For reconstruction of the key and file, as well as decrypting the secret, it is a straightforward process. First, the IDA is used in order to reconstruct the encrypted file, as any t fragments will be able to be combined for this reconstruction. A second reconstruction occurs using the standard key reconstruction method of the secret sharing scheme that is being used. Once the encrypted file and encryption key are reconstructed, decryption is the same as any other file encrypted using that algorithm.

2.4 Robust Secret Sharing

A variation of short share secret sharing exists in order to handle the chance that participants may return shares that are corrupted, possibly maliciously. A robust secret sharing scheme adds a third security parameter, m , to the secret sharing scheme, or the maximum number of corrupted/malicious shares given by participants that still allow the secret to be correctly reconstructed. For this to retain the security of the secret sharing scheme, so that a party of entirely malicious schemes can not reconstruct the secret, and to still require a majority of honest parties in reconstruction, the requirements on the security parameters are m<t and t≤w−m. The distribution and reconstruction of shares is exactly the same method as that of short share secret sharing, but with a public key signature system used to verify. After the encryption of the file, it is signed, as well as each of the shares of the key and fragments, prior to distribution. During reconstruction, the dealer will then verify the signatures of each share, and discard all fragments that cannot be verified. If the number of verified fragments exceeds the given security parameters, the secret is then revealed.

There are several downsides to this scheme, with regards to complexity and security needs. Because of the requirement for many signatures, and the overhead of the actual public key signature verification system, this scheme is much slower and computationally complex than other schemes. Additionally, in order to verify the signatures, the identity of the dealer must be known in order to adhere to the key verification. Some schemes may require anonymity of the participants for the security of the data. Regardless, it is proven that is can exist if needed, but is expensive to implement .

2.5 Verifiable secret sharing

If the players store their shares on insecure computer servers, an attacker could crack in and steal the shares. If it is not practical to change the secret, the uncompromised (Shamir-style) shares can be renewed. The dealer generates a new random polynomial with constant term zero and calculates for each remaining player a new ordered pair, where the x-coordinates of the old and new pairs are the same. Each player then adds the old and new y-coordinates to each other and keeps the result as the new y-coordinate of the secret.All of the non-updated shares the attacker accumulated become useless. An attacker can only recover the secret if he can find enough other non-updated shares to reach the threshold. This situation should not happen because the players deleted their old shares. Additionally, an attacker cannot recover any information about the original secret from the update files because they contain only random information. The dealer can change the threshold number while distributing updates, but must always remain vigilant of players keeping expired shares.

3. CONCLUSION

In this paper, we investigate the problem of data security in cloud data storage, which is essentially a distributed storage system. To achieve the assurances of cloud data integrity and availability and enforce the quality of dependable cloud storage service for users, we propose an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete, and append.

By utilizing the homomorphic token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., whenever data corruption has been detected during the storage correctness verification across the distributed servers, Through detailed security and extensive experiment results, we show that our scheme is highly efficient and resilient to Byzantine failure, malicious data modification attack, and even server colluding attacks.

4. REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring data storage security in cloud computing,” in *Proc. of IWQoS’09*, July 2009, pp.1–9.

[2] Amazon.com, “Amazon web services (aws),” Online at http://aws.amazon.com/, 2009.

[3] Sun Microsystems, Inc., "Building customer trust in cloud computing with transparent security," Online at https://www.sun.com/offers/details/sun transparency.xml, November 2009.

[4] M. Arrington, "Gmail disaster: Reports of mass email deletions, "Online at http://www.techcrunch.com/2006/12/ 28/gmail-disasterreports-of-mass-email-deletions/, December 2006.

[5] J. Kincaid, "MediaMax/TheLinkup Closes Its Doors,"Online at http://www.techcrunch.com/2008/07/10/ mediamaxthelinkup-closes-its-doors/, July 2008.

[6] Amazon.com, "Amazon s3 availability event: July 20, 2008," Online at http://status.aws.amazon.com/s3-20080720.html, July 2008.

[7] S. Wilson, "Appengine outage," Online at http://www.cio weblog.com/50226711/appengine outage.php, June 2008.

[8] B. Krebs, "Payment Processor Breach May Be Largest Ever," Online at Http://voices.washingtonpost.com/securityfix/2009/01/ payment processor breach may b.html, Jan. 2009.

[9] A. Juels and J. Burton S. Kaliski, "Pors: Proofs of retrievability for large files," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp.584–597.

[10] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable data possession at untrusted stores," in *Proc. of CCS'07*, Alexandria, VA, October 2007, pp. 598–609.