International Journal of Contemporary Research in Computer Science and Technology (IJCRCST) ISSN: 2395-5325
Volume 4, Special Issue 3 (March '2018)
Proceeding of - National Conference on Recent Trends in Information Technology and Communication,
Conference held at Velammal Engineering College,Chennai,India.

A SECURE ERASURE CODE-BASED CLOUD

S.Nivedhitha,
Student,
Department of Information Technology,
Velammal Engineering College
Chennai,Tamilnadu,India.

Ruchi V Singh,
Student,
Department of Information Technology,
Velammal Engineering College
Chennai,Tamilnadu,India.

K.Priyanka,
Student,
Department of Information Technology,
Velammal Engineering College
Chennai,Tamilnadu,India.

Abstract: Users find it difficult to fully trust cloud-based services because cloud-based data storage and protection methods are largely user transparent. General encryption schemes protect data confidentiality, but also limit the functionality of the storage system because a few operations are supported over encrypted data. Constructing a secure storage system that supports multiple functions is challenging when the storage system is distributed and has no central authority. In this paper, we propose a shared authority based privacy-preserving authentication protocol (SAPA) to address above privacy issue for cloud storage. In the SAPA, 1) shared access authority is achieved by anonymous access request matching mechanism with security and privacy considerations (e.g., authentication, data anonymity, user privacy, and forward security); 2) attribute based access control is adopted to realize that the user can only access its own data fields; 3) proxy re-encryption is applied by the cloud server to provide data sharing among the multiple users.

Keywords: Encrypting, Encoding, and forwarding, Storage Servers, Robustness.

I. INTRODUCTION

Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and business to use applications without installation and access their personal files at any computer with internet access. Users just use services without being concerned About how computation is done and storage is managed. In this paper, we focus on designing a cloud storage system for robustness, confidentiality, and functionality. Data robustness is a major requirement for storage systems. There have been many proposals of storing data over storage servers. One way to provide data robustness is to replicate a message such that teach storage server stores a copy of the message. It is very robust because the message can be retrieved as long as one storage server survives. Another way is to encode a message of k symbols into a codeword of n symbols by erasure coding. To store a message, each of its codeword symbols is stored in a different storage server. An example is introduced to identify the main motive- tin. In the cloud storage based supply chain management, there are various interest groups (e.g., supplier, carrier, and retailer) in the system. Each group owns its users which are permitted to access the authorized data fields, and different users own relatively independent access authorities.

It means that any two users from diverse groups should access different data fields of the same file. Thereunto, a sup-pliers may want to access a carrier's data fields, but it is not sure whether the carrier will allow its access request. If the carrier refuses its request, the supplier's access desire will be revealed along with nothing obtained towards the desired data fields. Actually, the supplier may not send the access request or withdraw the unaccepted request in advance if it firmly knows that its request will be refused by the carrier.

We consider the system model that consists of distributed storage servers and key servers. Since storing cryptographic keys in a single device is risky, a user distributes his cryptographic key to key servers that shall perform cryptographic functions on behalf of the user. These key servers are highly protected by security mechanic- isms. To well fit the distributed structure of systems, we require that servers independently perform all operations. With this consideration, we propose a new threshold proxy re-encryption scheme and integrate it with a secure decentralized code to form a secure distributed storage system. The encryption scheme supports encoding opera- tins over encrypted messages and forwarding operations over encrypted and encoded messages.

In this paper, we address the aforementioned privacy issue to propose a shared authority based privacy-preserving authentication protocol (SAPA) for the cloud data storage, which realizes authentication and authorization without compromising a user's private information. The main contra-bunions are as follows.

- Identify a new privacy challenge in cloud storage, and address a subtle privacy issue during a user challenging the cloud server for data sharing, in which the challenged request itself cannot reveal the user's privacy no matter whether or not it can obtain the access authority.
- Propose an authentication protocol to enhance a user's access request related privacy, and the shared access authority is achieved by anonymous access request matching mechanism.
- Apply cipher text-policy attribute based access con- troll to realize that a user can reliably access its own data fields, and adopt the proxy re-encryption to provide temp authorized data sharing among multi- plea users.

In the cloud environments, a reasonable security protocol should achieve the following requirements.

- Authentic- tin: a legal user can access its own data fields, only the authorized partial or entire data fields can be identified by the legal user, and any forged or tampered data fields can- not deceive the legal user.
- Data anonymity: any irrelevant entity cannot recognize the exchanged data and communication state even it intercepts the exchanged messages via an open channel.
- User privacy: any irrelevant entity can- not know or guess a user's access desire, which represents a user's interest in another user's authorized data fields.

II. RELATED WORK

We briefly review distributed storage systems, proxy re encryption schemes, and integrity checking mechanisms. Dunning and Kerman proposed an anonymous ID assignment based data sharing algorithm (AIDA) for multi-party oriented cloud and distributed computing systems. In the AIDA, an integer data sharing algorithm is designed on top of secure sum data mining operation, and adopts a variable and unbounded number of iterations for anonymous assignment. Specifically, Newton's identities and Sturm's theorem are used for the data mining, a distributed solution of certain polynomials over finite fields enhances the algorithm scalability, and Markov chain representations are used to determine statistics on the required number of iterations.

One way to reduce the expansion rate is to use erasure codes to encode messages. A message is encoded as a codeword, which is a vector of symbols, and each storage server stores a codeword symbol. A storage server failure is modeled as an erasure error of the stored codeword symbol. Random linear codes support distributed encoding, that is, each codeword symbol is independently computed. To store a message of k blocks, each storage server linearly combines the blocks with randomly chosen coefficients and stores the codeword symbol and coefficients.

To retrieve the message, a user queries k storage servers for the stored codeword symbols and coefficients and solves the linear system. Demakis et ill-considered the case that n ¼ AK for a fixed constant a. They showed that distributing each block of a message to v randomly chosen storage servers is enough to have a probability 1 k=p oð1Þ of a successful data retrieval, where v ¼ b in k, b > 5a, and p is the order of the used group. The sparsely parameter v ¼ b in k is the number of storage servers which a block is sent to. The larger v is, the communication cost is higher and the successful retrieval probability is higher.

The system has light data confidentiality because an attacker can compromise k storage servers to get the message. Nab eel et al. proposed a broadcast group key management (BGKM) to improve the weakness of symmetric key cryptosystem in public clouds, and the BGKM realizes that a user need not utilize public key cryptography, and can dynamically derive the symmetric keys during decryption. Accordingly, attribute based access control mechanism is designed to achieve that a user can decrypt the contents if and only if its identity attributes satisfy the con- tent provider's policies.

The fine-grained algorithm applies access control vector (ACV) for assigning secrets to users based on the identity attributes, and allowing the users to derive actual symmetric keys based on their secrets and other public information. The BGKM has an obvious advantage during adding/revoking users and updating access control policies. In the aforementioned works, various security issues are addressed. However, a user's subtle access request related privacy problem caused by data accessing and data sharing has not been studied yet in the literature.

Here, we identify a new privacy challenge, and propose a protocol not only focusing on authentication to realize the valid data accessing, but also considering authorization to provide the privacy preserving access authority sharing. The attribute based access control and proxy re encryption mechanisms are jointly applied for authentication and authorization.

Tang provide a better granularity on the granted right of a re-encryption key. A user can decide which type of messages and with whom he wants to share in this kind of proxy re-encryption schemes. Key-private proxy re encryption schemes is proposed by Attendees et al.. In a key-private proxy re encryption scheme, given a re-encryption key, a proxy server cannot determine the identity of the recipient. This kind of proxy re-encryption schemes provides higher privacy guarantee against proxy servers. Although most proxy re-encryption schemes use pairing operations, there exist proxy re-encryption schemes without pairing.

III. SCENARIO

An individual or group entity, which owns its data stored in the cloud for online data storage and computing. Different users may be affiliated with a common organization, and are assigned with index- pendent authorities on certain data fields. Cloud server. An entity, which is managed by a particular cloud service provider or cloud application operator to provide data storage and computing services. The cloud server is regarded as an entity with unrestricted

storage and computational resources. Trusted third party. An optional and neutral entity, which has advanced capabilities on behalf of the users, to perform data public auditing and dispute arbitration.

3.1 System Model

In the cloud storage, a user remotely stores its data via online infrastructures, platforms, or software for cloud services, which are operated in the distributed, parallel, and cooperative modes. During cloud data accessing, the user autonomously interacts with the cloud server without external interferences, and is assigned with the full and independent authority on its own data fields. It is necessary to guarantee that the users' outsourced data cannot be un authorized accessed by other users, and is of critical importance to ensure the private information during the users' data access challenges.

In some scenarios, there are multiple users in a system (e.g., supply chain management), and the users could have different affiliation attributes from different interest groups. One of the users may want to access other associated users' data fields to achieve bi-directional data

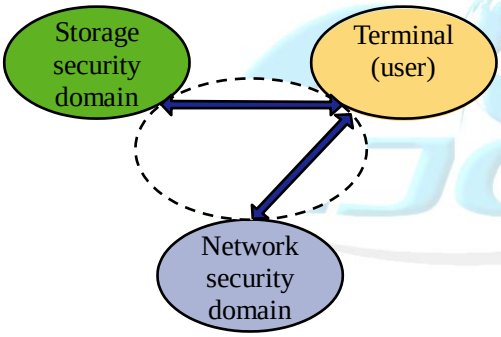


Figure1: System Model

sharing, but it cares about two aspects: whether the aimed user would like to share its data fields, and how to avoid exposing its access request if the aimed user declines or ignores its challenge. In the paper, we pay more attention on the process of data access control and access authority sharing other than the specific file oriented cloud data management.

3.2 Threat for system

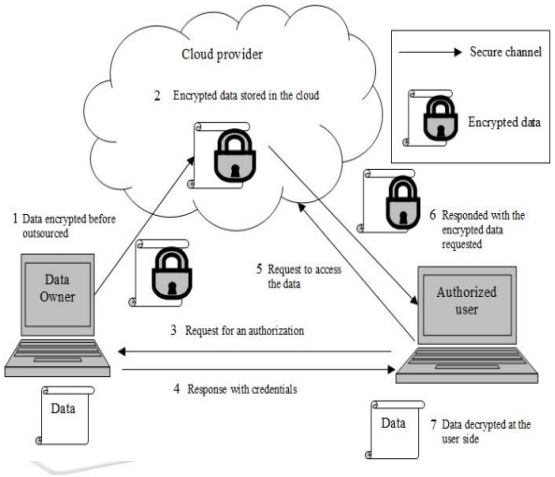
An attacker wants to break data confidentiality of a target user. To do so, the attacker collude switch all storage servers, non target users, and up toðt1Þkeyservers.The attacker analyzes stored messages in storage servers, the secret keys of non target users, and the shared keys stored in key servers. Note that the storage servers store all re encryption keys provided by users. The attacker may try to generate a new re encryption key from stored re encryption keys.

IV. CONSTRUCTION OF SECURE CLOUD STORAGE SYSTEMS

We consider that the message domain is the cyclic multiplicative group GG2 described above. An encoder generates a generate or matrix G¼½gi; for 1 ilk; 1 jinni as follows: for each row, the encoder and only selects an entry and randomly sets a value from ZZ to the entry. Then coder repeats this step v times with replacement for each row.

An entry of arrow can be selected multiple times but only set to one value. The values of the rest entries are set to0.Letthe message be ðm1; m2; mkÞ2GG2 our approach. We use a threshold proxy re-encryption Scheme with multiplicative homo orphic property. An encryption schemes multiplicative homomorphism .Thus, a multiplicative homomorphism encryption scheme supports the encoding operation over encrypted messages. We then convert a proxy re-encryption scheme with multiplicative homomorphism property into a threshold version. A secret key is shared to key servers with a threshold value t via the Shamir secret sharing scheme, where t _ k. In our system, to decrypt for a set of k message symbols, each key server independently queries 2 storage servers and partially decrypts two encrypted codeword symbols. As long as t key servers are available, k codeword symbols are obtained from the partially decrypted cipher texts.

.



Operation	Computation cost
Enc	k Pairing + k Exp_1 + k $Mult_2$
Encode (for each storage server)	k Exp_1 + k Exp_2 + $(k-1)$ $Mult_1$ + $(k-1)$ $Mult_2$
KeyRecover	$O(t^2)$ F_p
ReKeyGen	1 Exp_1
ReEnc (for each storage server)	1 Pairing +1 $Mult_2$
ShareDec (for t key servers)	t Exp_1
Combine	k Pairing + t $Mult_1$ + $(t-1)$ Exp_1+$O(t^2+k^3)$ F_p + k^2 Exp_2 + $(k+1)k$ $Mult_2$

Analysis

We analyze storage and computation complexities, correctness, and security of our cloud storage system in this section. Let the bit-length of an element in the group GG1 be l1 and GG2 be l2. Let coefficients gild be randomly chosen from f0; 1gl3. Storage cost. To store a message of k blocks, a

storage server Sis stores a codeword symbol be; bits, which is dominated by l3=l2 for a sufficiently large k. In practice, small coefficients, i.e., l3 l2, reduce the storage cost in each storage server. Computation cost. We measure the computation cost by the number of pairing operations, modular exponentiations in GG1 and GG2, modular multiplications in GG1 and GG2, and arithmetic operations over GFðpÞ. These operations are Denoted as Pairing, Exp1, Exp2, Mult1, Mult2, and Fop, respectively.

Mult2.The time of computing an Exp1 is1:5dlogpetimes as much as the time of Computing a Mult1, on average, (by using the square-and-multiply algorithm).Similarly, the time of computing Exp2 is1:5dlogpetimes as much as the time of computing Mult2, on average.

The probability of a successful retrival. A successful Retrieval event those a user successfully retrieves block of a message no matter whether the message is owned by him or forward him. The randomness comes from the random selection of storage servers in the data storage phase, the random coefficients chosen by storage servers, and the random selection of key servers in the data retrieval phase. The probability of a successful retrieval depend son (n; k; u; v) and all randomness. The methodology of analysis is similar to that in hand. However, we consider a different system model from the one in and a more flexible parameter setting forn¼akc than the settings. The difference between our system model and the one in that our system model has key servers.

It gives better flexibility for adjustment between the number of storage servers and robustness. This generalization is obtained by observing that Pr½E1is better bounded by choosing c1:5.The proofofTheorem1isgiven in Appendix A, which can be found on the Computer Society

V.CONCLUSION

In this paper, we consider a cloud storage system consists of storage servers and key servers. We integrate a newly proposed threshold proxy re-encryption scheme and erasure codes over exponents. The threshold proxy encryption scheme supports encoding, forwarding, and partial decryption operations in a distributed way. To decrypt a message of k blocks that are encrypted and encoded to n codeword symbols, each key server only has to partially decrypt two codeword symbols in our system. By using the threshold proxy re-encryption scheme.

We present a secure cloud storage system that provides secure data storage and secure data forwarding functionality in a decentralized structure. Moreover, each storage server independently performs encoding and re-encryption and each key server independently perform partial decryption. Our storage system and some newly proposed content addressable file systems and storage system are highly compatible. Our storage servers act as storage nodes in a content addressable storage system for storing content addressable blocks.

VI. REFERENCES

[1]. J. Kubiatowicz, D. Bindel, Y. Chen, P. Eaton, Doggerels, R.Gummadi, S.Rhea, H.Weatherspoon, W.Weimer, C.Wells,and B.Zhao, "Oceanstore: An Architecture for Global-Scale Persis- tent Storage". 201,2000.

[2]. P.Druscheland A.Rowstron,"PAST: A Large-Scale, Persistent Peer-to-Peer Storage Utility," Proc.Eighth Workshop Hot Topicsin Operating System (HotOSVIII), pp.75-80, 2001.

[3]. A.Adya, W.J.Bolosky, M.Castro, G.Cermak,R.Chaiken,J.R.Douceur,J.Howell,R.Lorch, M.Theimer,and R.Wattenhofer, "Farsite: Federated,Available, and ReliableStorage foran IncompletelyTrustedEnvironment,"Proc.FifthSymp.OperatingSystemDesignandImplementation(OSDI),pp.1-14,2002.

[4]. A. Haeberlen, A. Mislove, and P. Druschel, "Glacier: Highly Durable, DecentralizedStorage DespiteMassive Correlated Fail- ures," Proc.SecondSymp.NetworkedSystemsDesignandImplemen- tation (NSDI), pp.143-158, 2005.

[5]. Z.Wilcox-O' Hearnand B.Warner, "Tahoe: The Least-Authority filesystem,"Proc.FourthACMInt'lWorkshopStorageSecurityandSurvivability (StorageSS), pp.21-26, 2008.

[6]. Y.LinandW.-G.Tzeng "ASecure Decentralize dErasure Code forDistributedNetwork Storage," IEEETrans.Paralleland DistributedSystems,

[7]. D.R.Brownbridge, L.F.Marshall, and B.Randell, "TheNewcastle ConnectionorUnixesofthe World Unite!" Software Practiceand Experience,vol.12,no.12,pp. 1147-1162,1982.

[8]. R.Sandberg, D.Goldberg, S.Kleiman, Dowels, and Billion, "Design and Implementation of the Sun NetworkFilesystem, Proc.USENIXAssoc.Conf" 1985.

[9]. Kallahalla, E.Riedel, R.Swaminathan, Q.Wang, and K.Fu, "Plautus: ScalableSecure File Sharing on Untrusted Storage," Proc.SecondUSENIXConf.FileandStorageTechnologies (FAST), pp.29-42, 2003.

[10]. S.C.Rhea, P.R.Eaton, D.Geels, H.Weatherspoon, B.Y.Zhao, and J.Kubiatowicz, "Pond: The Oceanstore Prototype,"Proc.Second USENIX Conf. File and Storage Technologies(FAST),pp.1-14,2003

[11]. R.Bhagwan,K.Tati,Y.-C.Cheng, S.Savage, and G.M.Voelker, "TotalRecall:SystemSupport forAutomated Availability Management," Proc. First Symp. NetworkedSystems Design and Implementation (NSDI), pp. 337-350, 2004.

[12]. A.G. Dimakis, V. Prabhakaran,and K.Ramchandran,"Ubiqui-tous Access to Distributed Data in Large-Scale SensorNet-works through Decentralized Erasure Codes," Proc.Fourth Int' lSymp. Information Processing in Sensor Networks(IPSN), pp. 111-117,2005.