ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Panda: Public Auditing for Shared Data with Efficient User Revocation in the Cloud

Aravabumi Divya
Dept of Software Engineering,
SKR College of Engineering and Technology,
Nh-5, Kondurusatram, Manubolu, Spsr Nellore,Ap.

Nagala Venkatadri
Associate Professor,
Dept of CSE & IT,
SKR College of Engineering and Technology,
Nh-5, Kondurusatram, Manubolu, Spsr Nellore,Ap.

ABSTRACT:

With data storage and sharing services in the cloud, users can easily modify and share data as a group. To ensure shared data integrity can be verified publicly, users in the group need to compute signatures on all the blocks in shared data. Different blocks in shared data are generally signed by different users due to data modifications performed by different users. For security reasons, once a user is revoked from the group, the blocks which were previously signed by this revoked user must be re-signed by an existing user. The straightforward method, which allows an existing user to download the corresponding part of shared data and re-sign it during user revocation, is inefficient due to the large size of shared data in the cloud. In this paper, we propose a novel public auditing mechanism for the integrity of shared data with efficient user revocation in mind. By utilizing the idea of proxy re-signatures, we allow the cloud to resign blocks on behalf of existing users during user revocation, so that existing users do not need to download and re-sign blocks by themselves. In addition, a public verifier is always able to audit the integrity of shared data without retrieving the entire data from the cloud, even if some part of shared data has been re-signed by the cloud. Moreover, our mechanism is able to support batch auditing by verifying multiple auditing tasks simultaneously. Experimental results show that our mechanism can significantly improve the efficiency of user revocation.

EXISTING SYSTEM:

In existing mechanisms, a signature is attached to each block in data, and the integrity of data relies on the correctness of all the signatures.

One of the most significant and common features of these mechanisms is to allow a public verifier to efficiently check data integrity in the cloud without downloading the entire data, referred to as public auditing. This public verifier could be a client who would like to utilize cloud data for particular purposes or a thirdparty auditor (TPA) who is able to provide verification services on data integrity to users. With shared data, once a user modifies a block, she also needs to compute a new signature for the modified block. Due to the modifications from different users, different blocks are signed by different users.

For security reasons, when a user leaves the group or misbehaves, this user must be revoked from the group. As a result, this revoked user should no longer be able to access and modify shared data, and the signatures generated by this revoked user are no longer valid to the group. Therefore, although the content of shared data is not changed during user revocation, the blocks, which were previously signed by the revoked user, still need to be re-signed by an existing user in the group. As a result, the integrity of the entire data can still be verified with the public keys of existing users only.

DISADVANTAGES OF EXISTING SYSTEM:

1. Straightforward method may cost the existing user a huge amount of communication and computation resources.

2. The number of re-signed blocks is quite large or the membership of the group is frequently changing.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

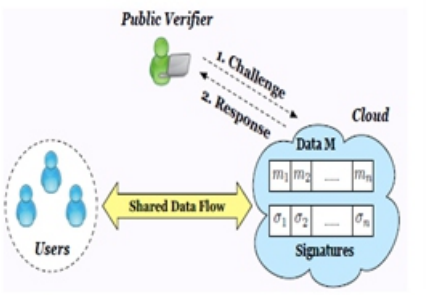
PROPOSED SYSTEM:

In this paper, we propose Panda, a novel public auditing mechanism for the integrity of shared data with efficient user revocation in the cloud. In our mechanism, by utilizing the idea of proxy re-signatures, once a user in the group is revoked, the cloud is able to resign the blocks, which were signed by the revoked user, with a re-signing key. As a result, the efficiency of user revocation can be significantly improved, and computation and communication resources of existing users can be easily saved. Meanwhile, the cloud, which is not in the same trusted domain with each user, is only able to convert a signature of the revoked user into a signature of an existing user on the same block, but it cannot sign arbitrary blocks on behalf of either the revoked user or an existing user. By designing a new proxy re-signature scheme with nice properties, which traditional proxy resignatures do not have, our mechanism is always able to check the integrity of shared data without retrieving the entire data from the cloud. Moreover, our proposed mechanism is scalable, which indicates it is not only able to efficiently support a large number of users to share data and but also able to handle multiple auditing tasks simultaneously with batch auditing. In addition, by taking advantages of Shamir Secret Sharing, we can also extend our mechanism into the multi-proxy model to minimize the chance of the misuse on re-signing keys in the cloud and improve the reliability of the entire mechanism.

ADVANTAGES OF PROPOSED SYSTEM:

1. It follows protocols and does not pollute data integrity actively as a malicious adversary.
2. Cloud data can be efficiently shared among a large number of users, and the public verifier is able to handle a large number of auditing tasks simultaneously and efficiently.

SYSTEM ARCHITECTURE:



MODULES

User Module:

» Registration
» File Upload
» Download
» Reupload
» Unblock module

Auditor Module:

» File Verification module
» View File

Admin Module:

» View Files
» Block user

MODULES DESCRIPTION:

User Module:

Registration:

In this module each user registers his user details for using files. Only registered user can able to login in cloud server.

File Upload:

In this module user upload a block of files in the cloud with encryption by using his secret key. This ensures the files to be protected from unauthorized user.

Download:

This module allows the user to download the file using his secret key to decrypt the downloaded data of blocked user and verify the data and reupload the block of file into cloud server with encryption .This ensure the files to be protected from unauthorized user.

Reupload:

This module allow the user to reupload the downloaded files of blocked user into cloud server with resign the files(i.e) the files is uploaded with new signature like new secret with encryption to protected the data from unauthorized user.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Unblock Module:

This module allows the user to unblock his user account by answering his security question regarding to answer that provided by his at the time of registration. Once the answer is matched to the answer of registration time answer then only account will be unlocked.

Auditor Module:
File Verification module:

The public verifier is able to correctly check the integrity of shared data. The public verifier can audit the integrity of shared data without retrieving the entire data from the cloud, even if some blocks in shared data have been re-signed by the cloud.

Files View:

In this module public auditor view the all details of upload, download, blocked user, reupload.

Admin Module:
View Files:

In this module public auditor view the all details of upload, download, blocked user, reupload.

Block User:

In this module admin block the misbehave user account to protect the integrity of shared data.

5 PANDA
5.1 Overview:

Based on the new proxy re-signature scheme and its properties in the previous section, we now present Panda — a public auditing mechanism for shared data with efficient user revocation. In our mechanism, the original user acts as the group manager, who is able to revoke users from the group when it is necessary.Meanwhile, we allow the cloud to perform as the semi-trusted proxy and translate signatures for users in the group with resigning keys. As emphasized in recent work [23], for security reasons, it is necessary for the cloud service providers to storage data and keys separately on different servers inside the cloud in practice.

Therefore, in our mechanism, we assume the cloud has a server to store shared data, and has another server to manage resigning keys. To ensure the privacy of cloud shared data at the same time, additional mechanisms, such as [24], can be utilized. The details of preserving data privacy are out of scope of this paper. The main focus of this paper is to audit the integrity of cloud shared data.

5.2 Support Dynamic Data:

To build the entire mechanism, another issue we need to consider is how to support dynamic data during public auditing. Because the computation of a signature includes the block identifier, conventional methods — which use the index of a block as the block identifier
(i.e., block mj is indexed with j) — are not efficient
for supporting dynamic data [8], [14]. Specifically, if a single block is inserted or deleted, the indices of blocks that after this modified block are all changed, and the change of those indices requires the user to re-compute signatures on those blocks, even though the content of those blocks are not changed. mi !i idi si
Block Signature Block Identifier Signer Identifier
Fig. 6. Each block is attached with a signature, a block identifier and a signer identifier. By leveraging index hash tables [8], [14], we allow a user to modify a single block efficiently without changing block identifiers of other blocks. The details of index hash tables are explained in Appendix A. Besides a block identifier and a signature, each block is also attached with a signer identifier (as shown in Fig. 6). A verifier can use a signer identifier to distinguish which key is required during verification, and the cloud can utilize it to determine which re-signing key is needed during user revocation.

5.3 Construction of Panda:

Panda includes six algorithms: KeyGen, ReKey, Sign, ReSign, ProofGen, ProofVerify. Details of Panda arepresented in Fig. 5. In KeyGen, every user in the group generates his/her public key and private key. In ReKey, the cloud computes a re-signing key for each pair of users in the group. As argued in previous section, we still assume that private channels exist between each pair of entities

6 EXTENSION OF PANDA:

In this section, we will utilize several different methods to extend our mechanism in terms of detection probability, scalability and reliability.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

6.1 Detection Probability of Panda:

As presented in our mechanism, a verifier selects a number of random blocks instead of choosing all the blocks in shared data, which can improve the efficiency of auditing. Previous work [3] has already proved that a verifier is able to detect the polluted blocks with a high probability by selecting a small number of random blocks, referred to as sample strategies [3]. More specifically, when shared data contains n = 1, 000, 000 blocks, if 1% of all the blocks are corrupted, a verifier can detect these polluted blocks with a probability greater than 99% or 95%, where the number of selected blocks c is 460 or 300, respectively. Further discussions and analyses about sample strategies can be found in [3]. To further reduce the number of the undetected polluted blocks in shared data and improve the detection probability, besides increasing the number of random selected blocks in one auditing task mentioned in the last paragraph, a verifier can also perform multiple auditing tasks on the same shared data. If the detection probability in a single auditing task is PS, then the total detection probability for a number of t multiple auditing tasks is

PM = 1− (1 − PS)t.

For instance, if the detection probability in a single auditing task is PS = 95%, then the total detection probability with two different auditing tasks on the same shared data is PM = 99.75%. Note that to achieve a higher detection probability, both of the two methods require a verifier to spend more communication and computation cost during auditing.

6.2 Scalability of Panda:

Now we discuss how to improve the scalability of our proposed mechanism by reducing the total number of re-signing keys in the cloud and enabling batch auditing for verifying multiple auditing tasks simultaneously.

Reduce the Number of Re-signing Keys. As described in Panda, the cloud needs to establish and maintain a re-signing key for each pair of two users in the group. Since the number of users in the group is denoted as d, the total number of re-signing keys for the group is d(d − 1)/2. Clearly, if the cloud data is shared by a very large number of users, e.g. d = 200, then the total number of re-signing keys that the cloud has to securely store and manage is 19, 900, which significantly increases the complexity of key management in cloud.

To reduce the total number of re-signing keys required in the cloud and improve the scalability of our mechanism, the original user, who performs as the group manager, can keep a short priority list (PL) with only a small subset of users instead of the entire PL with all the users in the group. More specifically, if the total number of users in the group is still d = 200 and the size of a short PL is d" = 5, which means the cloud is able to convert signatures of a revoked user only into one of these five users shown in the short PL, then the total number of re-signing keys required with the short PL of
5 users is 990. It is only 5% of the number of re-signing keys with the entire PL of all the 200 users.

Batch Auditing for Multiple Auditing Tasks. In many cases, the public verifier may need to handle multiple auditing tasks in a very short time period. Clearly, asking the public verifier to perform these auditing requests independently (one by one) may not be efficient. Therefore, to improve the scalability of our public auditing mechanism in such cases, we can further extend Panda to support batch auditing [7] by utilizing the properties of bilinear maps. With batch auditing, a public verifier can perform multiple auditing tasks simultaneously. Compared to the batch auditing in [7], where the verification metadata (i.e, signatures) in each auditing task are generated by a single user, our batch auditing method needs to perform on multiple auditing tasks where the verification metadata in each auditing task are generated by a group of users. Clearly, designing batch auditing for our mechanism is more complicated and challenging than the one in [7].
More concretely, if the total number of auditing tasks received in a short time is t, then the size of the group for each task is dj, for j # [1, t], each auditing message is represented as {(l, &j|l)}l%Lj, for j # [1, t], each auditing proof is described as {#j,%j , {idj|l}l%Lj },
where
#j = (#j|1, ...,#j|dj) and %j = (%j|1, ...,%j|dj), for j # [1, t],and all the existing users' public keys for each group are denoted as (pkj|1, ..., pkj|dj), for j # [1, t]. Based on the properties of bilinear maps, the public verifier can perform batch auditing as below

e(t$j=1dj$i=1%&jj|i,g)?=t$j=1dj$i=1e($l%Lj|iH(idj|l)!j|l •w#j|I , pkj|i)&j ,(3)

where)j # Z!p, for j # [1, t], is a random chosen by the public verifier. The correctness of the above equation is based on all the t auditing proofs are correct. The left hand side (LHS) of this equation can be expended as

LHS =t$j=1e(dj$i=1%j|i, g)&j=t$j=1dj$i=1e($l%Lj|I H(idj|l)!j|lw#j|i , pkj|i)&j .

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

According to the security analysis of batch verification in [25], the probability that the public verifier accepts an invalid auditing proof with batch auditing is 1/p (since randoms ()1, ...,)t) are elements of Zp), which is negligible. Therefore, if the above equation holds, then the public verifier believes all the t auditing proofs are correct. One of the most important advantages of batch auditing is that it is able to reduce the total number of pairing operations, which are the most time consuming operations during verification. According to Equation (3), batch auditing can reduce the total number of pairing operations for t auditing tasks to td+1, while verifying these t auditing tasks independently requires td + t pairing operations. Moreover, if all the t auditing tasks are all from the same group, where the size of the group is d and all the existing users' public keys for the group are (pk1, ..., pkd), then batch auditing on t auditing tasks can be further optimized as follows e(t $j=1d $i=1%&jj|i, g) ?=d$i=1e(t$j=1($l%Lj|iH(idj|l)!j|l •w#j|i)&j , pki).In this case, the total number of pairing operations during batch auditing can be significantly reduced to

ReKey!. Given private key skj = "j, user uj generates a random t − 1 degree polynomial fj(x) as
fj(x) = aj,t−1xt−1 + aj,t−2xt−2 + • • • + aj,1x + aj,0,
where (aj,t−1, ...,aj,1) R) Z! p and aj,0 = "j. Then, user uj computes s points (x1, yj,1), ..., (xs, yj,s) based on polynomial fj(x), where yj,l = fj(xl), for l # [1, s], is a piece of user uj 's private key and (x1, ...,xs) are public.
Proxy Pl generates a piece of a re-signing key as follows:
(1) proxy Pl generates a random r # Z!
p and sends it to user ui; (2) user ui computes and sends r/"i to user uj,
where
ski = "i; (3) user uj computes and sends ryj,l/"i to server Sl, where yj,l = fj(xl) is a piece of private key skj = "j; (4)
proxy Pl recovers rki"j,l = yj,l/"i # Zp, where rki"j,l is a piece of re-signing key rki"j .

ReSign!. Given public key pki, signature !k, block mk and
block identifier idk, the cloud first checks
e(!k, g) ?=e(H(idk)wmk , pki).
If the equation does not hold, the cloud outputs $; otherwise, each proxy converts this signature !k with its own piece of the corresponding re-signing key. Specifically, proxy Pl converts its part of signature !k as
!#k,l = !rki!j,lk = (H(idk)wmk)yj,l .
Finally, as long as t or more proxies are able to convert their
parts correctly, the cloud is able to recover signature !#
k based
on the t parts (!#
k,1, ...,!#k,t) as!#
k =t#l=1!#k,lFj,l(0), (4)
where !#
k,l is computed by proxies Pl, and Fj,l(x) is a
Lagrange basis polynomial of polynomial fj(x) and can be pre-computed as Fj,l(x) = #0<h&t,h'=lx − xhxl − xh, 1 * l * t.
Similar as in Algorithm ReSign in single proxy model, afterthe re-signing process in the cloud, the original user removesuser ui's id from the user list (UL) and signs the new UL.Fig. 7. Details of ReKey! and ReSign!.
only d + 1, which can further improve the efficiency ofbatch auditing. The correctness of the above equationcan be proved as

6.3 Reliability of Panda:

In our mechanism, it is very important for the cloud to securely store and manage the re-signing keys of the group, so that the cloud can correctly and successfully convert signatures from a revoked user to an existing user when it is necessary. However, due to the existence of internal attacks, simply storing these re-signing keys in the cloud with a single re-signing proxy may sometimes allow inside attackers to disclose these re-signingkeys and arbitrarily convert signatures on shared data, even no user is revoking from the group. Obviously, the arbitrary misuse of re-signing keys will change the ownership of corresponding blocks in shared data without users' permission, and affect the integrity of shared data in the cloud.

To prevent the arbitrary use of re-signing keys and enhance the reliability of our mechanism, we propose an extended version of our mechanism, denoted as Panda!, in the multi-proxy model. By leveraging an (s, t)-Shamir Secret Sharing (s ' 2t − 1) [18] and s multiple proxies, each re-signing key is divided into s pieces and each piece is distributed to one proxy. These multiple proxies belong to the same cloud, but store and manage each piece of a re-signing key independently (as described in Fig. 8). Since the cloud needs to store keys and data separately [23],

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

7 OVERHEAD ANALYSES:

Communication Overhead. According to the description in Section 5, during user revocation, our mechanism does not introduce communication overhead to existing users. The size of an auditing message {(l, yl)}l%L is c•(|n|+|q|) bits, where c is the number of selected blocks,|n| is the size of an element of set [1, n] and |q| is the size of an element of Zq. The size of an auditing proof {#,%, {idl, sl} l%L} is 2d • |p| + c • (|id|) bits, where d is the number of existing users in the group, |p| is the size of an element of G1 or Zp, |id| is the size of a block identifier. Therefore, the total communication overhead of an auditing task is 2d • |p| + c • (|id| + |n| + |q|) bits.Computation Overhead. As shown in ReSign of our mechanism, the cloud first verifies the correctness of the original signature on a block, and then computes a new signature on the same block with a re-signing key. The computation cost of re-signing a block in the cloud is 2ExpG1 + MulG1 + 2Pair + HashG1, where ExpG1 denotes one exponentiation in G1, MulG1 denotes one multiplication in G1, Pair denotes one pairing operation on e : G1 × G1 " G2, and HashG1 denotes one hashing operation in G1. Moreover, the cloud can further reduce the computation cost of the re-signing on a block to ExpG1 by directly re-signing it without verification, because the public auditing performed on shared data ensures that the re-signed blocks are correct. Based on Equation (2), the computation cost of an auditing task in our mechanism is (c + d)ExpG1 + (c+2d)MulG1 + (d + 1)Pair + dMulG2 + cHashG1 .

CONCLUSIONS:

In this paper, we proposed a new public auditing mechanism for shared data with efficient user revocation in the cloud. When a user in the group is revoked, we allow the semi-trusted cloud to re-sign blocks that were Experimental results show that the cloud can improve the efficiency of user revocation, and existing users in the group can save a significant amount of computation and communication resources during user revocation.

REFERENCES:

[1] B. Wang, B. Li, and H. Li, "Public Auditing for Shared Data with Efficient User Revoation in the Cloud," in the Proceedings of IEEE INFOCOM 2013, 2013, pp. 2904–2912.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "A View of Cloud Computing," Communications of the ACM, vol. 53, no. 4, pp. 50–58, Apirl 2010.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z. Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," in the Proceedings of ACM CCS 2007, 2007, pp. 598–610.

[4] H. Shacham and B. Waters, "Compact Proofs of Retrievability," in the Proceedings of ASIACRYPT 2008. Springer-Verlag,2008,pp. 90–107.

[5] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring Data Storage Security in Cloud Computing," in the Proceedings of ACM/IEEE IWQoS 2009, 2009, pp. 1–9.

[6] Q. Wang, C. Wang, J. Li, K. Ren, and W. Lou, "Enabling Public Verifiability and Data Dynamic for Storage Security in Cloud Computing," in the Proceedings of ESORICS 2009. Springer-Verlag, 2009, pp. 355–370.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in the Proceedings of IEEE INFOCOM 2010, 2010, pp. 525–533.

[8] Y. Zhu, H.Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic Audit Services for Integrity Verification of Outsourced Storage in Clouds," in the Proceedings of ACM SAC 2011, 2011, pp. 1550–1557.

[9] C. Wang, Q. Wang, K. Ren, and W. Lou, "Towards Secure and Dependable Storage Services in Cloud Computing," IEEE Transactions on Services Computing, vol. 5, no. 2, pp. 220–232, 2011.

[10] Y. Zhu, G.-J. Ahn, H. Hu, S. S. Yau, H. G. An, and S. Chen, "Dynamic Audit Services for Outsourced Storage in Clouds," IEEE Transactions on Services Computing, accepted.

[11] N. Cao, S. Yu, Z. Yang, W. Lou, and Y. T. Hou, "LT Codes-based Secure and Reliable Cloud Storage Service," i.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

About Authors:

1.Aravabhumi.Divya She Was Born In Budanam [V], Chillakuru [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India. She Received The B.Tech Degree In Information Technology From Jnt University, Anantapur In 2012 And Pursuing M.Tech Degree In Software Engineering From Jnt University, Anantapur. She Completed Her B.Tech Degree In Avs College Of Engineering & Technology, Venkatachalam [V] [M], S.P.S.R.Nellore [Dt], Andhra Pradesh And M.Tech Degree In Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R.Nellore [Dt], Andhra Pradesh, India.

2.Mr. Nagala Venkatadri He Was Born In Andhrapradesh, India. He Received The B.Tech Degree From Jnt University, Anantapur And M.Tech Degree From Jnt University, Anantapur. He Has 10 Years Experience In The Field Of Associate Professor. He Had Working As Associate Professor In Dept. Of Computer Science & Engineering And Software Engineering In Skr College Of Engineering & Technology, Konduru Satram [V], Manubolu [M], S.P.S.R Nellore [Dt], Andhra Pradesh, India. He Is Presently Pursuing His Doctorate In Dept. Of Computer Science & Engineering.