PRIVACY PRESERVING STORAGE AND RETRIEVAL MECHANISM OVER THE CLOUDS

P. Pushpa[1], Dr.G.N.K..Suresh Babu[2]
[1]Research Scholar, Research and Development Centre, Bharathiar University, Coimbatore
[2]Professor, Department of Computer Applications, Acharya Institute of Technology, Bangalore

Abstract

Cloud Computing is a growing exponentially, whereby there are now hundreds of cloud service providers (CSPs) of various sizes. This multi – cloud environment offers plenty of new opportunities and avenues to cloud environments. In this paper discussing about a privacy –preserving Storage and Retrieval (STRE) mechanism enables the cloud users to distribute and search their encrypted data. The STRE mechanism enables the cloud users to distribute and search their encrypted data in multiple cloud service providers (CSPs), and is robust even when a certain number of CSPs crash. Besides the reliability, STRE also offers the benefits of partially hidden search patterns.

Keywords: Cloud Service Providers, Searchable Encryption, Multi-clouds, Keyword Search

Introduction

Cloud consumers may enjoy cheaper data storage and powerful computation capabilities offered by multiple cloud,[1][2], consumers also face more complicated reliability issues and privacy preservation problems of their outsourced data. More specifically, as it is difficult to obtain clear guarantees on the trustworthiness [3] of each CSP cloud consumers are typically suggested to adopt searchable encryption techniques [4]to encrypt their outsourced data in a way that the encrypted data can be directly searched by the CSPs without decryption. Despite many efforts devoted to improving efficiency and security of the searchable encryption, there is little consideration on ensuring the reliability of the searchable encrypted data.

Existing reliability guarantees solely rely on each CSP's own backup solution, which however could be single -point of failure. For instance, the crash of Amazon's elastic computing services in 2011 took some popular social media sites off-line for a day and one energy department collaboration site unavailable for nearly two days. More seriously, this crash has permanently destroyed many customers data with serious consequences for some users.It is worth nothing that a comprehensive solution to simultaneously ensuring searchability, privacy, and reliability on data at multiple CSPs is the most straightforward method, we are not aware of any existing work that addresses the three requirements in a comprehensive manner.

STRE Mechanism

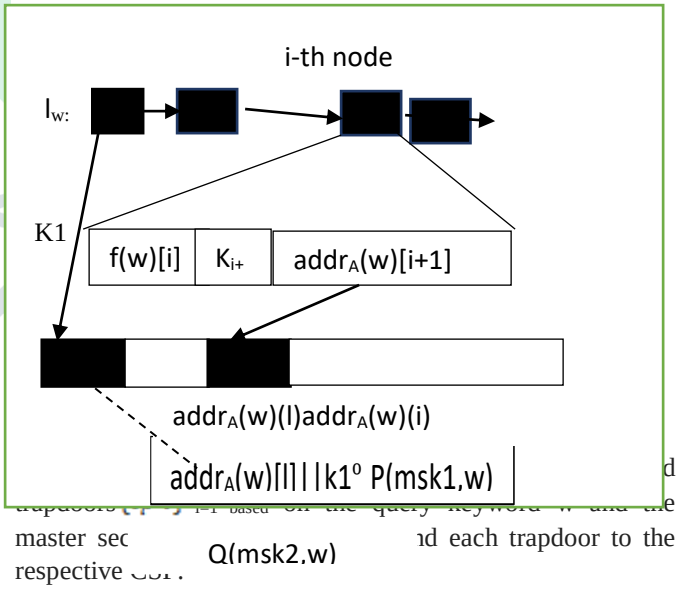
The STRE mechanism consists of two major phases: Storage Phase and Retrieval Phase.

Storage Phase: This phase consists of two main steps

Step 1: A master secret key *msk* is generated from a security parameter $1^{\lambda}$and given to the user. Note that the security parameter 1^{λ} which is assumed to be known to all the adversaries, specifying the input size of the problem. Both the resource requirements of the cryptographic algorithm or protocol and the adversary's probability of breaking security are also expressed in terms of the security parameter.

Step 2: Upon taking a collection of files f and master secret key *msk* as input, user generates and uploads encrypted file chunks and file index (c_i, I) to the ith CSP for i=1, 2…., n.

Fig 1: Compressed Index



master sec …d each trapdoor to the respective CSP.

Step 2: n CSPs collaborate together to search w, and ith CSP returns a collection of encrypted chunks y, back to the user for i=1, 2…...n. Note then if a certain CSP crashes, its response is y_i=□.

Step 3: The user uses his/her master secret key to obtain a collection of clear files x from at least t no-empty $y_i$in $\{y_i\}^{n}_{i=1.}$ The correctness of protocol requires that for any file f, f € x holds when and only when id(f) € f(w).

STRE Protocols

Storage Protocol

The Storage protocol is for users to encrypt and distribute their files to multiple CSPs. We present its details as follows.

Step 1: Given the security parameter1^{λ} , the following

(1) Initiate three pseudo-random functions: P ; $\{0,1\}^{\lambda}$ x $\{0,1\}^{\lambda}$x$\{0,1\}^{s}$ → [0,1…..,|Δ| , R :{ 0,1} $^{\lambda}$X{0,1}s+$\log_2(\max_{w\in\Delta}|f(w)|)$ → {0,1}$\log_2$r, where r is the total number of appearances of keywords in f and s is the bit size of each keyword .

(2) After computing msk1,msk2,msk3 €R $\{0,1\}^{\lambda}$and msk 4= SKE1. gen (1^{λ}), send the master secret key msk= (msk1, msk2, msk3, msk4) touser.

Step 2: User builds an index I similar (shown as fig 1) to [5]. This index includes a search array A and a look -up table T, which respectively contains r and |Δ| entries. We then describe how to construct this index for the files consisting of a keyword w €Δ.

Retrieval Protocol: In order to achieve privacy preserving keyword search over multiple clouds, we propose a novel retrieval protocol that consists of two steps: 1) query sharing stage; 2) and reconstructionstage. The query sharing stage generates a (t,n) secret sharing on the user's keyword query and distribute the shares to n CSPs.The reconstruction stage allows the user to obtain the query results when at least t (t ≤ n) CSPs are functioning.

Step 1: 1) query sharing stage; 2) and reconstruction stage. The query sharing stage generates a (t ,n) secret sharing on the user's keyword query and distribute the shares to n CSPs. The reconstruction stage allows the user to obtain the query results when at least t (t ≤ n) CSPs are functioning.

Step 1: Given the master key msk and query keyword w, the user first builds a secret matrix and its mirror matrix.
(1) Build (m X t) secret matrix S such that S [1][2]= Q(msk2,w)and random values are filled at the other entries. We can just set m=2t-2 to reduce communication overhead .

(2) Build a mirror S` as the same as S except S`[1][1] =S`[1][2]=0 .Then ,S` is publicated out for correctness check in future.

Secondly, the user performs the following computations to make a multiple secret sharing on the secret matrix S.

(3) After randomly picking a (m X t) matrix A of rank t, compute the projection matrix M=$A(A^TA)^{-1}$ A ^{T}mod p and publish the remainder matrix R=S-M mod p where p is a public big prime number.

(4) Randomly choose (t X 1) vectors x_i for i=1, 2…., n such that any t of $\{x\}^n_{i=1}$ are linearly independent, and compute each share v_i =Ax_i mod p.

Finally, user submits the share v_i to the ith CSP for i=1, 2…., n, to retrieve all the files containing the keyword w.

Step 2: For the ith CSP, i=1, 2…, n, upon receiving the share vi, it first submits and collects share through multi-party computation according to the following steps:

(1) Build a matrix $\overline{B_i}$ such that $\overline{B_i}[i]=v_i$ and the other entries of B_i are filled with 0.

(2) After making an (n,n) secret sharing on $\overline{B}_{i1}$… $\overline{B_{in}}$ such that $\sum_{j=1}^{n} Bij = Bi$, send $\overline{B_{ij}}$to the jth CSP for j=1,….,i-1,i+1,….,n Note that the matrix $\overline{B_{ii}}$ is kept at local by the ith CSP .

(3) Upon receiving $\overline{B_{ji}}$ from the other CSP, where j=1…, i-1. i+1…., n, send back (to the jth CSP) a response ack.Note that if the response is not received by the jth CSP, the jth CSP needs to set B_{ji} =B_{jj}+$B_{ji.}$

(4) Suppose $\overline{B1}$, ……, $\overline{B_{i-1}}$, $\overline{B_{i+1,}}$ …., B_{n}i have been successfully gathered and responded. Compute and broadcast $Bi = \sum_{j=1}^{n} Bij$

(5) After gathering $\overline{B1}$, ……, $\overline{B_{i-1}}$, $\overline{B_{i+1,}}$ …., $\overline{B_n}$ from the other CSPs and Bi from the local, ith CSP computes and obtains the share matrix B= $\sum_{j=1}^{n} Bj$

(6) Randomly collect any t column from B and construct the matrix B.

(7) Calculate projection matrix M=$(B(B^TB)^{-1}B^T)$ mod p and the secret matrix can be reconstructed as S''=M+R mod p.

(8) Verify the correctness of reconstruction by checking all the entries except S'' [1][1] and S'' [1][2] of S'' and S'. if not passed, return back to step 2 (6).

Step 3: Upon computing P=S'' [1][1] and Q=S'' [1][2]. The ith CSP proceeds to collect and return the code chunks:

(1) Compute T[Q]⊕P=tmp and parse tmp as loc and k.Then, loc is the location of the first node of lw in A and k is the symmetric key used for the encryption of this node .
(2) Compute info = SKE2.Dec(k, A[loc])
(3) After Parsing info as id, loc and k, fetch the code chunks (id, c_i) with (id, c_i)∈ci. Then, test loc||k; if loc||k≠O $\prec^{+\log_2 r}$, return back to Step 3 (2).

After gathering all the code chunks {(id, c_i)} id ∈r for i-1, 2……, n, Where Γ is an underlying set of file identifiers satisfying the search criterion as intrinsically indicated above, the ith CSP sends backs results (i, {(id, c_i)}id)∈ Γ)

Step 4: Upon receiving the results, the user continues to proceed as follows.

(1) Suppose $\tilde{\Omega}$ is a set of CSP identifiers i, the chunk of which have been successfully received. if $|\Omega|<t$ user reports "fail "and the protocol is aborted. Otherwise, she randomly selects i-element set $\Omega \subseteq \Omega$, d constructs a matrix E from thecorresponding t row vectors of E. Recall the E is the encodingmatrix of encryoted files maintained by user . The rank of E is t, which can be reconstructed by multiplying the inverse matrix of Ewith the corresponding code chunks . guarantees that E is invertible. Straightforwardly , for each file, its encrypted form

(2) Finally , User uses msk4 to decrypt the reconstructed encrypted files and obtains the search results is plain.

Conclusion

In this paper, we propose the STRE mechanism, to promote reliability of outsourced searchable encrypted data. In STRE, user's searchable encrypted data is strategically distributed to and stored at multiple CSPs, so as to achieve high crash tolerance. Besides reliability, the STRE mechanism also affords efficient and flexible storage properties and partially hidden search pattern

Reference:

1.Chen, H.C.H, Lee, P.P.C: Enabling data integrity protection in reengineering coding-based cloud storage. - Proceedings of the 31st IEEE International Symposium on Reliable Distributed System, pp 51-60 (2012)

2. Zhu, y., Hu, H, Ahn, G,J, YU, M: Cooperative provable data possession for integrity verification in multi cloud storage, IEEE transactions on parallel Distributed Systems 23(12) 2231-2244(2012)

3. Owens, D, Searching elasticity in the cloud, communications of the ACM 53,46-51 (2010)

4. Song, D.X. Wagner, D., Perrig, A Practical technique for searches on encrypted data, In IEEE Symposium on security and privacy, pp 44-55 (2000)

5. Curtmola .R, Garay J,A, Kamara ,S, Ostrovsky-Searchable Symmetric encryption :Improved definitions and efficient constructions – ACM Conference on Computer and Communication Security ,PP.79-88(2006).