www.ioirp.com
International Journal of Innovative Research in Computer Science and Engineering (IJIRCSE)
ISSN: 2394-6364, Volume – 3, Issue – 3. May 2018

MMB Cloud Tree: Authenticated Index for Verifiable Cloud Service Selection

Rathnadevi [1]
Department of CSE,
Greentech college of Engineering for women,
Attur, Salem.
rsrathnadevi@gmail.com

Ushanandhini.N.S[2]
Department of CSE,
Greentech college of Engineering for women,
Attur, Salem
ushadev20@gmail.com

Sanju.A.G [3]
Department of CSE,
Greentech college of Engineering for women,
Attur, Salem
Sanjuamutha@gmail.com

Punitha.S[4]
Department of CSE,
Greentech college of Engineering for women,
Attur, Salem
punithasekar@gmail.com

Abstract—Cloud brokers have been recently introduced as an additional computational layer to facilitate cloud selection and service management tasks for cloud consumers. However, existing brokerage schemes on cloud service selection typically assume that brokers are completely trusted, and do not provide any guarantee over the correctness of the service recommendations. It is then possible for a compromised or dishonest broker to easily take advantage of the limited capabilities of the clients and provide incorrect or incomplete responses. To address this problem, we propose an innovative Cloud Service Selection Verification (CSSV) scheme and index structures (MMB cloud-tree) to enable cloud clients to detect misbehavior of the cloud brokers during the service selection process.

Index Terms—Cloud service selection, brokerage system, Merkle hash tree, verification.

I INTRODUCTION

Cloud services offer a scalable variety of storage space and computing capabilities, which are widely employed by an increasing number of business owners. This has resulted in a large number of cloud service providers (CSPs), offering a wide range of resources. The availability of various, possibly complex options, however, makes it difficult for potential cloud clients to weigh and decide which options suit their requirements the best. The challenges are twofold: 1) It is hard for cloud clients to gather information about all the CSPs available for their selections; 2) It is also computationally expensive to choose a suitable CSP from a potentially large CSP pool. [3]They introducing an additional computing layer (referred to as cloud brokerage systems) on top of the base service provisioning to enable tasks such as discovery, mediation and monitoring. . That is, a broker provides clients with a list of recommended CSPs that meet the clients' needs. With the aid of cloud brokers, clients no longer need to collect, search or compare CSPs' services and capabilities. The underlying assumption in the existing cloud brokerage schemes is that brokers are completely trusted and thus will always provide unbiased best available options to clients [9]. Under this assumption, none of the existing works provides guarantees over the correctness or completeness of the service selection recommendations to the cloud clients. Without the ability to verify the correctness of the service recommendation, cloud clients could be easily cheated by malicious brokers. Therefore, it is important to equip the clients with verification capabilities of the obtained recommendations. In this work, we propose innovative authenticated index structures and verification protocols to allow clients to verify the completeness and authenticity of brokers' answers. In order to overcome the limitations of existing techniques, both in terms of efficiency and supported functionality, we propose a new authenticated index structure, called Multi-Merkle Bcloud-tree (MMBcloud-tree), which is a variant of the Merkle B+-tree and is speservice selections. In particular, we first design the Merkle B cloud-tree (MBcloud-tree) which is an authenticated index on the most common property (i.e., Price) of CSPs, and propose the corresponding verification protocol. Then, we extend the MBcloud-tree to the MMBcloud-tree by integrating a multi-dimensional indexing method (i.e., iDistance) with MBcloud-tree, to further improve the selection quality as well as reduce the verification burden at the client side. Our approaches are proved to ensure authenticity, satisfiability and completeness of the selected results., Our novel index structure is the core component of our Cloud Service Selection Verification (CSSV) scheme, which employs the idea of "separation of duties" to ensure strong security guarantees. Precisely, we introduce a trusted collector in the cloud brokerage system that separates the task of CSP information collection from the service selection. The collector does not directly interact with the cloud clients and is only in charge of gathering information from the CSPs, and hence it can be more devoted into adopting sophisticated defenses to filter out problematic data and building an authenticated database of CSPs' profiles. The collector is allowed to make

profit by selling the authenticated database to one or more cloud brokers. With the available authenticated databases, the cloud brokers focus on handling probably a large number of real-time service requests from clients.

II The Merkle Hash Tree:

The Merkle hash tree [31] has a binary tree as the base structure. The leaf nodes in the Merkle hash tree contain the hash values of the original data items. Each internal node contains the hash value of the concatenation of the hash values of its two children nodes. The hash value of the root of the tree is published for verification. If there is any change to the original data values, one would not be able to compute the same hash value of the root and hence detect such data tampering. Fig. 1 illustrates an example of the data verification using the Merkle hash tree. A verifier has the published hash value hroot of the root. He sends the request to the prover for data item x2. The prover returns the requested data item along with the auxiliary authentication information h1, h3||4 and h56||78, where || is the symbol of concatenation which will be used throughout this paper. Upon receiving the data item and the additional authentication information,

The verifier performs the following computation to check if the received x2 is authentic. First, the verifier computes h2 = Hash(x2), h1||2 = Hash(h1||h2), h12||34 = Hash(h1||2||h3||4), h1234||5678 = Hash(h12||34||h56||78), and then he checks if the calculated root hash (i.e., h1234||5678) is the same as the published one(i.e., h_{root}).If so ,the verifier will conclude that the data is authenticated.

Based on the similar idea of the Merkle hash tree, the Merkle B+-tree have been proposed to handle multiple entries per node.

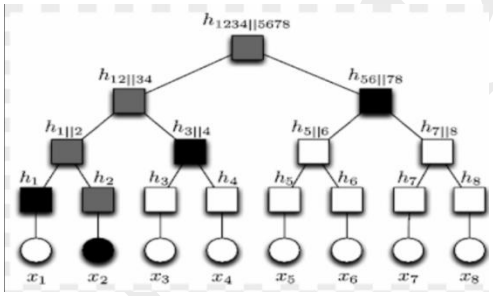

Fig. 1. Merkle Hash Tree .

III THE CSSV SCHEME:

To verify authenticity of a CSP's profile, a naive solution is to require the CSP to sign its profile and then let the client verify the signature. Satisfiability is also easy to verify since the client just needs to check if the profiles of the candidate CSPs in the query results actually satisfy the query. However, there is not a trivial way for the client to know if the query results returned by the broker contain all qualifying CSPs. In other words, the verification of completeness is the most challenging issue. To address this, we propose the Cloud Service Selection Verification (CSSV) scheme which is a comprehensive solution that is capable of guaranteeing all the three security requirements (i.e., authenticity, satisfiability and completeness). In the CSSV scheme, we introduce one more entity, the collector, besides CSPs, cloud clients and cloud brokers. The collector acts like a certificate authority and is assumed fully trusted, which is inline with the recent work . The collector is associated with a pair of public and private keys, and its public key is made available to CSPs, cloud brokers and cloud clients. Specifically, the CSSV scheme (as shown in Fig. 2) includes the following three phases:

1. Database construction by the collector.

2. Service selection by the broker.

3. Result verification by the clients.

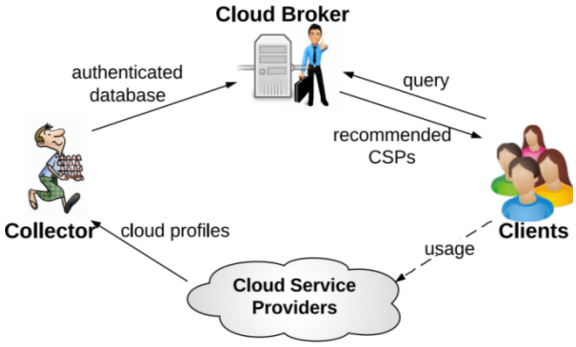

Fig.2.Architecture for CSSV system

IV MBcloud –TREE:

Our basic scheme follows the three Phases MMBcloud -TREE The basic approach using MBcloud-tree indexes only the Price property, and therefore has limited ability to deal with queries that do not include Price as one of the selection criteria, or with queries that have many other selection criteria besides Price. In either case, the basic approach may return many CSPs which satisfy only the Price criterion but not the whole query in the proof message for verification. Note that, the final query results are not affected since a refinement step is included to identify the actual qualifying CSPs. In order to balance the verification burden (i.e., reduce the number of false positives in filtering step), we propose an advanced scheme indexing multiple properties. The advanced scheme integrates the MBcloud-tree idea with a multi-dimensional index (i.e., iDistance) to build a novel authenticated index, MMBcloud-tree, on all the properties of the CSPs

CSSV system: the database construction, service selection and results verification. It is developed based on a variant of the Merkle hash tree

A) Database Construction:

In order to efficiently manage and retrieve CSPs' profiles while guaranteeing data integrity, we propose a MBcloud - tree to index CSPs' profiles on the Price property of the CSPs. The reason to choose Price as the indexing field is two-fold. First, given that most cloud providers employ a pay-per-use business model, Fig. 3 illustrates the structure of the MBcloud-tree, which is developed based on the B+-tree by adding one additional field to each entry to store a hash value for the subsequent authenticity check. The information stored in each tree component is described as follows: • Each CSP is represented by a data structure C consisted of its unique identity (ID), a list of property values (L) and a hash of its profile (Hash(prof)), computed using a secure collusion free hash function. • Each entry ei in the leaf node stores the information of CSPs offering the same price in the form of $\langle key_i , h_i , pt_i \rangle$, where key_i is the index key and contains the value of the price, pt_i points to a list of CSPs whose price equals to key_i , and hi is a concatenation of all the hash values of CSPs in the list. Suppose that CSP1, CSP2, ..., CSPk fall into the same entry and hash values of their representation structures are Hash(C1), Hash(C2). . . Hash(Ck), respectively. Then, the hash value of this entry will be hi = Hash(Hash(C1)||Hash(C2)|| . . . ||Hash(Ck)). The leaf nodes are chained with sibling leaf nodes via pointers. • Each internal node contains m index key values and m + 1 pointers to the child nodes which are defined in the same way as that in the B+-tree. Additionally, each pointer in the internal node is associated with one hash value that is computed by concatenating the hash values of entries in its child node. More specifically, let $_{hi}$,1, ... h_i,m denote the hash values in a child node pointed by pti , the hash value hi is computed as $h_i = Hash(h_i,1||h_i,2, ..., ||h_i,m)$. Notice that here we abuse notation by using the same symbol pt_i as the pointer in internal node pointing to a child node, and hi as hash value associated with pti . To construct the MB^{cloud}-tree, the information of each CSP is inserted in the same way as that in the B+-tree.

A hash value of the CSP is computed,and the hash value is then used to compute the hash values of its ancestor nodes all the way up to the root

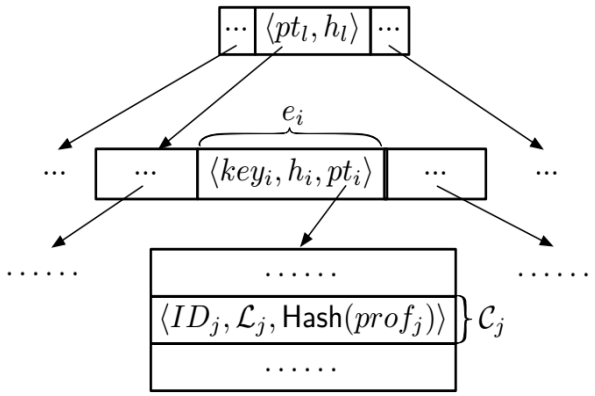

Fig.3. structure of merkle B[cloud-]tree

After the construction, the collector signs the hash value of the root node using its private key, and publishes the obtained signature σ. Finally, the collector sends the MB[cloud] - tree along with the profile database to the cloud brokers.

B) **Service Selection**:

The cloud broker takes requests from clients and executes the service selection queries on the Merkle Bcloud-tree obtained from the collector. The service selection query algorithm consists of two main steps: 1) filtering and 2) refinement. • Step 1 (Filtering): The cloud broker identifies candidate CSPs and the hash values to be returned for verification. Specifically, upon receiving a query Q = {(U1, q1),(U2, q2) , ...,(Uk, qk)}, the broker first checks if Q includes Price. If so, the broker will run a range query on the MBcloud-tree using the query range (denoted as qprice) of Price. In order for the client to check completeness later, the broker also records the left neighbor entry elow−1 of elow, and right neighbor entry ehigh+1 of ehigh (if elow or ehigh is the leftmost or rightmost entry of MBcloud-tree, then elow−1 or ehigh+1 would be set empty). These four entries together define the boundary of the query results. Note that if none of CSPs in the broker's database satisfies the query, elow−1 and ehigh+1 will be the rightmost and leftmost entries in the tree,

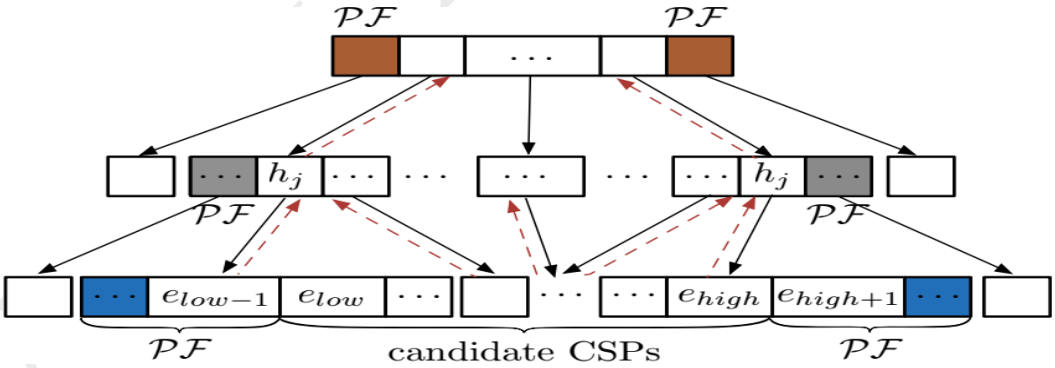

Fig.4.Proof Message in merkle B[cloud] -tree

Next, for each parent node of ei , the broker includes the hash values of non-parent entries (the shaded boxes in Fig. 4) in the proof to facilitate the client to reconstruct the hash value of the parent node at the verification phase. The same process is applied to all the ancestor nodes of ei all the wayup to the root. In summary, the proof message PF contains representation structures of candidate CSPs, boundary CSPs (i.e., elow−1 and ehigh+1), and a set of hash values of nonancestor entries in the ancestor nodes. • Step 2 (Refinement): The broker further exams the properties of candidate

CSPs to put those CSP profiles that satisfy all the requirements of the query in a query result set R. Then, the broker returns R and the proof message PF to the client.

C) Results Verification:

Upon receiving the results R and the proof message PF, the client will verify the correctness of the results in terms of satisfiability, authenticity, and completeness. The client does not need to verify the results every time but, it could choose to check the correctness of R at random time. To verify satisfiability of the result, the client simply double checks if the properties of CSPs in R meet his/her needs. Next, the client verifies the authenticity and completeness simultaneously by computing the hash value of the root using the information in the proof message .The client will detect the broker's misbehavior of any modification and deletion of qualifying results. Theorem 1. For any query Q ={(U1, q1), (U2, q2), ..., (Uk, qk)}, let R denote the query results returned by the approach described above. The correctness of R (in Definition 2) can be verified by the client.

V CONCLUSION:

An innovative Cloud Service Selection Verification (CSSV) system to achieve cheating-free cloud service selection under cloud brokerage architecture. The core of our system is an efficient authenticated index structure to ensure the authenticity, the satisfiability and the completeness of the service selection results. Our theoretical and experimental results demonstrate the effectiveness and efficiency of our schemes compared with the state-of-the-art. As part of our future work, we plan to consider a verifiable scheme for best service selection query whereby the broker returns only the best CSP instead of all candidate CSPs with respect to a client's request.

REFERENCE:

1.L. Xin and A. Datta, "On trust guided collaboration among cloud service providers," in 2010 6th International Conference on Collaborative Computing: Networking, Applications and Worksharing

2.S. Sundareswaran, A. Squicciarini, and D. Lin, "A brokerage-based approach for cloud service selection," in 2012 IEEE 5th International] . Conference on Cloud Computing (CLOUD). IEEE, Aug. 2012, pp. 558–565.

3. H. Pang, A. Jain, K. Ramamritham, and K.-L. Tan, "Verifying completeness of relational query results in data publishing," in Proceedings of the 2005 ACM SIGMOD International Conference on Management of Data, 2005, pp. 407–418.

4.H. V. Jagadish, B. C. Ooi, K.-L. Tan, C. Yu, and R. Zhang, "iDistance: an adaptive B+-tree based indexing method for nearest neighbor search," ACM Trans Database Systems (TODS), vol. 30, no. 2, pp. 364–397, Jun. 2005.

5.R. Zhang, Y. Zhang, and C. Zhang, "Secure top-K query processing via untrusted location-based service eprovider."in 2012 proceeding IEEE INFOCOM ,IEEE 2012,pp.1170-1178 .

[6] A. Lenk, M. Menzel, J. Lipsky, S. Tai, and P. Offermann, "What areyou paying for? performance benchmarking for Infrastructure-as-a-Service offerings," in *2011 IEEE. International Conference on CloudComputing (CLOUD)*, 2011, pp. 484–491

[7] L. Qu, Y. Wang, and M. A. Orgun, "Cloud service selection basedon the aggregation of user feedback and quantitative performanceassessment," in *2013 IEEE International Conference on Services Com-puting (SCC)*, 2013, pp. 152–159.

[8] Q. Zheng, S. Xu, and G. Ateniese, "Efficient query integrity foroutsourced dynamic databases," in *CCSW '12 Proceedings of the2012 ACM Workshop on Cloud computing security workshop*. ACM,2012, pp. 71–82.

[9] S. Bajaj and R. Sion, "CorrectDB: SQL engine with practical queryauthentication," *Proceedings of the VLDB Endowment*, vol. 6, no. 7,pp. 529–540, 2013.

[10] D. Papadopoulos, S. Papadopoulos, and N. Triandopoulos, "Tak-ing authenticated range queries to arbitrary dimensions," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM,2014, pp. 819–830.