International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 5 | ISSN : 2456-3307

A Data Ownership Privacy Provider Framework in Cloud Computing

[1]N. Ambika, [2] Dr. M. Sujaritha
[1]Research Scholar, Bharathiar University, Coimbatore, Tamil Nadu, India
[2]Associate Professor, Department of CSE, Sri Krishna College of Engineering & Technology, Coimbatore. Tamil Nadu, India

ABSTRACT

The complementary problem of secure storage of private cloud data has been studied extensively in the literature but cannot usually be applied while the data is in decrypted form for the duration of a computation. Secure multiparty computation and differential privacy are both powerful approaches to privacy preserving cloud computation on decrypted data, but are inapplicable to many real world cloud computations. In particular, jobs submitted to the cloud as arbitrary binary code are difficult to automatically reformulate as secure multiparty computations, and high differential privacy sometimes comes at the expense of highly imprecise, noisy results. In these cases, the level of privacy can sometimes be improved by concealing data ownership, provenance, and/or semantics from the participants in a computation in addition to or instead of anonymizing the data itself.

Keywords: Cloud Computing, Cloud Security, Data Centers, Anonymous Cloud.

I. INTRODUCTION

Anonymous Cloud system covers information provenance from cloud hubs that register over the information, and hides beneficiary characters as IP locations and possession marks. Anonymization is accomplished through the instantiation of a Tor anonymizing circuit inside the cloud, through which private information and occupations are namelessly provided by and come back to clients. Circuit length is a tunable parameter k, managing an adaptable exchange off between the level of secrecy and the computational overhead of the circuit. To keep up a compensation for every utilization plan of action, mists should definitely track possession data at some level for charging and examining purposes. AnonymousCloud in this manner executes an open key cryptography based mysterious validation that disassociates information proprietorship metadata from the private information it names. Hence, a different administrator hub that does not approach the private information can charge clients properly utilizing the possession metadata, while calculation hubs that approach the private information however not the metadata can safely complete the unknown employment. Supervisors are trusted not to intrigue with calculation hubs to damage security, however all different hubs including the ace hub are possibly vindictive. In this way, AnonymousCloud decentralizes the trust by decoupling charging data from the submitted occupations.

II. LITERATURE SURVEY

Data security concerns are for the most part seen as a significant deterrent to customer confide in dispersed registering [1]. Related troubles extend no under three arrangements of related work: secure remote stage affirmation (i.e., trusted enrolling), secure data storing, and information drove security [2]. Trusted figuring gives customers high certification that they are talking with a remote server containing known, put stock in gear and programming [3]. Secure limit regards the issue of safely securing private data in the cloud (customarily in mixed structure) between estimations that use it [4]. On the other hand, information driven procedures immerse data with self securing properties, for instance, by addressing it in a structure reasonable to coordinate count on figure compositions without interpreting [5]. Unknown Cloud's philosophy of decoupling private data from its provenance information can be viewed as an instance of the rest of these systems.

General data anonymization is a gigantic examination domain spreading over various decades; in any case, the most comprehensively used strategies for anonymization of data substance are starting at now differential security [6] and k-mystery for assurance sparing scaled down scale data release [28]. Such research benefits our work by giving an approach to customers to anonymize private data content before submitting it to the cloud. We along these lines acknowledge customers charmed by assurance submit data that reveals less favored bits of knowledge once it has been decoupled from provenance and semantic metadata, and that thus benefits by our anonymization tradition. Prior work has moreover explored decoupling chronicle content from setup and structure for more secure circulated stockpiling and getting ready [7]. For example, HTML files can be encoded in a setup that detaches their tree structures from the scholarly substance of segments and qualities. Since a larger piece of private data abides in the substance, these grants isolate treatment of helper based inquiries in the cloud without revealing the private data. To decouple and cover provenance metadata, AnonymousCloud uses onion coordinating in light of TOR [30]. Tor has transformed into the best open anonymity correspondence organization in the Internet, with an enormous number of customers general [8]. In Tor, initiators pick a path through framework and develop a circuit in which each center point or onion switch in the way knows only its successor and predecessor, however the same centers in the circuit. In light of the picked way or course, the initiator initially scrambles the data with one layer of encryption for each center point in the path, from the last center to the first. This is contrasted with the layers of an onion, with each bounce peeling one layer as the data is sent to its goal. The data must be examined in plaintext once it accomplishes the endpoint of the way and the entirety of what layers have been peeled. The Tor Cloud wander has realized a full scale Tor structure inside an era level cloud that continues running on the Amazon EC2 appropriated registering stage [9]. It gives a straightforward technique for passing on expansions to enable customers to get to an uncensored Internet. Tor Cloud shrouds customer pen names (IP numbers) from untrusted pariah organizations, yet does not do the trick to furtively get to data from an untouchable cloud [10], since fogs require a strategy for confirming customers with a particular true objective to control access to each customer's private data and charge them appropriately.

We along these lines widen cloud based onion controlling with an obscure capability system for affirmation [34]. Secretive confirmation gives zero data proof of identity, allowing data to be securely decoupled from provenance for updated security. More definite obscure accreditation structures for additional security properties, for instance, non-transferability, slow disavowal, and access movements. These are excessive for our system, yet rather could be substituted if such properties are appealing for various reasons. We donot consider the peril of end to end timing ambushes (except for that we arrange circuit lengths of no under 3 to hinder the minimum troublesome such strikes). Past works have exhibited that these ambushes are perhaps convincing against TOR and other onion guiding systems despite when the attacker controls only a few center points. The Tarzan structure secures against timing ambushes through period of produced spread development that shroud timing outlines in a sea of mimicry and confusion. Future work should consider the achievability of supplementing AnonymousCloud with equivalent securities. Other than realizing affirmations and traditions that particularly energize more critical insurance, frameworks that give more noticeable straightforwardness to internal cloud operations particularly course and organization of security fragile data is fundamental for bestowing more unmistakable trust in end customers. Future work should in this manner consider growing AnonymousCloud with parts that bear the cost of customers more noticeable control over data flow and arranging purposes of enthusiasm after Tor circuit improvement, and without surrendering anonymity.

III. CLOUD ARCHITECTURE

The system architecture of AnonymousCloud is given in Figure-1. It consists of a cloud provider CP and a separate manager M.

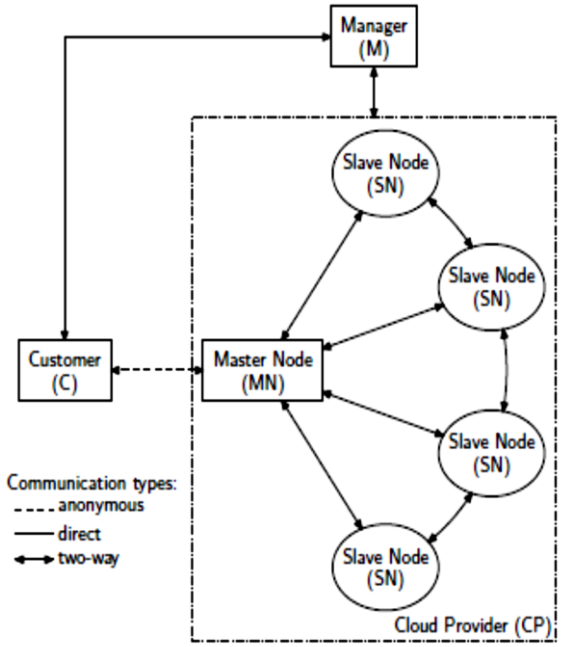

Figure 1: AnonymousCloud Architecture

3.2.1 Cloud Providers

CPs gives calculation administrations to clients C, who submit calculations as employments. Clients can get to these administrations in compensation as we go design, with installment oversaw by the different administrator. Distinctive CPs may change in the subtle elements of their inner designs. We accept just that occupations are submitted to the CP by means of a brought together ace hub MN, which parcels and calendars sub-calculations over a substantial accumulation (e.g., several thousands) of slave hubs SNs. All SNs are in this manner specifically associated with the MN and there is subjective availability between the SNs. AnonymousCloud changes Tor usefulness to the MN and SNs without adjusting the activity distribution and planning points of interest of the cloud in any capacity. All principals (M, C, MN, and SNs) are moreover outfitted with open private key sets from a settled endorsement expert CA. People in general keys fill in as the symmetric or common keys amid Tor circuit development.

3.2.2 Managers

Managers are separate from the CP's computing infrastructure, and facilitate only customer authentication and billing. They have four primary responsibilities related to our work:

- M provides central storage of public keys for MN and SNs and serves them to C on request.
- M maintains a graph of SN connectivity. This facilitates Tor circuit construction by encoding the universe of available circuit links for circuit initialization.
- M provides each C a unique access token t and credentials c (e.g., a password) via which Cs can authenticate them to M to obtain cloud services.
- M additionally generates a unique nonce n for each of C's transactions to protect the authentication system against replay attacks.

M likely has additional responsibilities related to authentication, such as key revocation, certificate update, auditing, customer billing, etc. These responsibilities are deployment specific, and are therefore beyond the scope of this work.

3.2.3 Authentication Protocol

The authentication and circuit construction protocol of AnonymousCloud is depicted in Figure - 2 and detailed in Authentication and circuit construction protocol Algorithm. C begins each service transaction by communicating its access token and credentials to M, and requesting an anonymizing circuit of length k. If at least k connected nodes are available, M returns such a list; otherwise it may offer a list shorter than k. The returned list includes the public keys K_{SN} of all the selected slave nodes, as well as the public key K_{MN} of the master node. M also generates a fresh nonce n for C and stores a local copy. To prevent replay attacks, the next service request from C will only be authenticated by M if it is labeled with n.

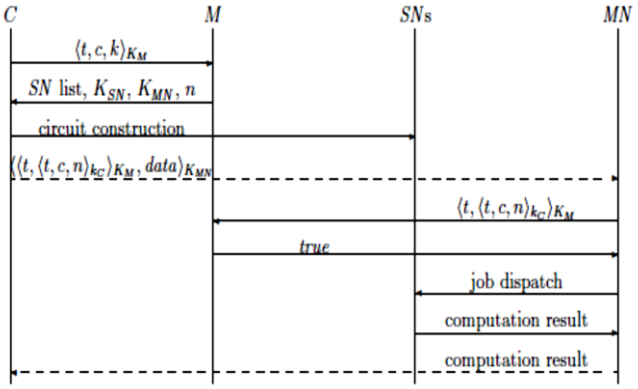

Figure-2 : Authentication and circuit construction message sequence. Solid lines denote direct communications, whereas dashed lines denote anonymous communication through the Tor circuit.

Authentication and Circuit Construction Protocol Algorithm

Step-1: C asks M to choose k available SN s based on SN connectivity

Step-2: if C has invalid token t or invalid credentials c then

M rejects the request from C

else

repeat

M selects k SN s (or the most available)

M provides C with public keys K_{SN} and K_{MN} and fresh nonce n

C validates keys K_{SN} and K_{MN} with the CA

if any key fails validation by the CA then

M revokes the invalid keys

end if

until all keys are valid

Step-3: C performs Tor circuit construction over the SNs using the Ks as symmetric keys

Step-4: C signs t, c, and n with private key k_C and encrypts it with public key K_M , yielding M = (t, (t, c), K_C, K_M)

Step-5: C sends (m, data)K_{MN} in layered encryption format over the circuit to MN

Step-6: MN anonymously receives and decrypts the message with private key K_{MN}

Step-7: MN forwards m to M for authentication

Step-8: M decrypts m using K_M and verifies signature K_C using K_C , yielding t, c, and n

Step-9: M verifies t, c, and n; and it verifies K_C with the CA

Step-10: if authentication fails then

M returns false to MN

MN discards the service request

else

M returns true to MN

MN dispatches the data computation

MN anonymously returns the result to C over the circuit

end if

end if

In step 8 of Authentication and circuit construction protocol Algorithm, C verifies the certificates with the certificate authority and stores them locally. To lessen the load, C may cache these results to avoid re-authenticating certificates that have not changed. C then transmits the requested computation and its data anonymously via the Tor circuit to MN in step 5, MN can read the data but not the encrypted ownership metadata M = (t, (t, c), K_C, K_M). It therefore forwards m to M for validation. M can read metadata M by decrypting it using its private key K_M however it has no access to the associated job's data. M verifies C's digital signature using public key K_C and validates K_C's certificate with the certificate authority (possibly caching the results to more efficiently service future requests). The access tokens t inside and outside the digital signature are additionally compared for equality, the credentials c are validated against t, and the nonce n is checked against the local copy. If all these steps succeed, M invalidates the nonce and returns true to MN; otherwise it returns false and the request is denied.

Upon effective validation, MN dispatches the asked for calculation as per the CP's inner engineering and conventions. In the event that client charging depends on computational asset utilization or other data that lone winds up plainly accessible as the calculation advances, MN can report such data to M without knowing the activity's proprietor by labeling it with encoded validation data m. M would then be able to ascribe the acquired costs to the right client. Once the calculation is finished, its outcomes are secretly conveyed to C through the Tor circuit. The Tor circuit is then destroyed and its assets recovered by the CP.

IV. RESULTS AND ANALYSIS

Each data point is the result of simulating 1000 customer service requests to a cloud consisting of 1 master node and N = 1000 slave nodes. A successful attack against our system is defined as the link ability of private data to its corresponding ownership metadata by one or more malicious principals. Principals include the manager, the master node, and all slave nodes. Ownership metadata includes customer pseudonyms (viz., access tokens and IP addresses) and authentication credentials. We assume that private data does not include pseudonyms or other information from which customer identities can be inferred; anonymizing the private data is the subject of related work.

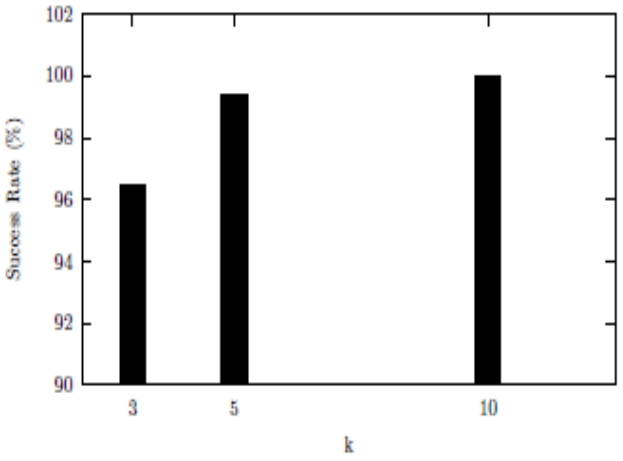

Figure-3: Privacy enforcement success as a function of Tor circuit length k in a cloud of p = 30% malicious slave nodes

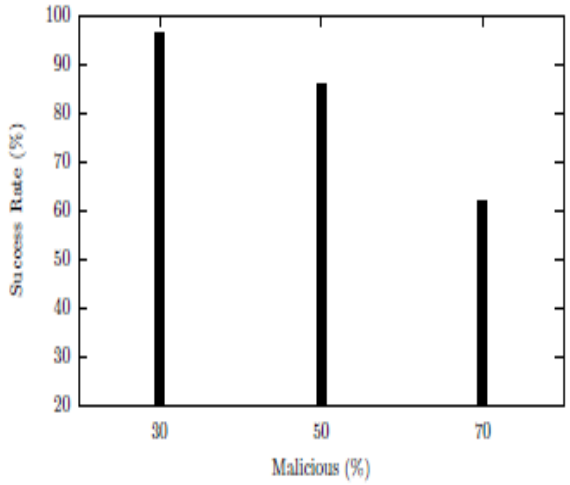

Figure-4: Privacy enforcement success for Tor circuits of length k = 3 as a function of percentage p of malicious slave nodes

All together for an assault against AnonymousCloud to succeed, the administrator or ace hub (or both) must be malevolent. Directors are the main principals that get decryptable access tokens or certifications, and every other correspondence including nom de plumes information are led through Tor circuits having the ace hub as the main untrusted endpoint. Directors are separate from CPs and have a considerably littler assault surface since they don't process client submitted calculations. We accordingly expect that directors are trusted, however that ace hubs are constantly malignant. Moreover, we expect that a rate p of slave hubs are likewise malignant and conspire with the vindictive ace hub with an end goal to abuse protection. Figure-4 plots the normal security requirement achievement rate for various Tor circuit lengths k in a cloud with a pernicious ace hub and 30% vindictive slave hubs. On the off chance that k = 0, AnonymousCloud does not give any secrecy; moreover, any length under 3 altogether expands the simplicity of effective end to end timing assaults. We in this way confine our regard for circuit lengths of no less than 3. At k = 3 we get an effectively high achievement rate of 96.5%. Expanding k to 5 additionally raises this 99.4%, and at k = 10 there were no protection disappointments by any stretch of the imagination. Figure-4 plots the achievement rate of a settled circuit length k = 3 in mists with differing rates p of pernicious slave hubs. The outcomes indicate how versatile our framework is against vindictive cooperatives. Notwithstanding when mists are half noxious, AnonymousCloud achieves a 85.8% protection safeguarding rate with just k = 3. At the point when 70% of the cloud is pernicious, the achievement rate drops to 62%, demonstrating that more drawn out circuits are required to oppose such unavoidable assaults. The outcomes announced in Figures - 3 and Figure - 4 can be summed up by watching that with high likelihood all k slave hubs in a Tor circuit must connive keeping in mind the end goal to trade off security.

V. CONCLUSION

AnonymousCloud appropriates trust by separating proprietorship data from the submitted employments to mists and enhances information protection in the cloud by decoupling private information content from metadata concerning its provenance and semantics. Our framework, AnonymousCloud, utilizes Tor onion steering inside cloud suppliers for clients to namelessly impart calculations and information to the framework. An unknown confirmation framework in light of open key cryptography encourages charging of mysterious clients without connecting their private information to their personalities. We show that AnonymousCloud gives unrivaled information proprietorship protection notwithstanding when a huge level of the cloud is malevolent.

VI. REFERENCES

[1]. Slamanig, D. (2011). More privacy for cloud users: Privacy-preserving resource usage in the cloud. In Selected Papers from the 4th Hot

Topics in Privacy Enhancing Technologies (HotPETs), pp. 15-27.

[2]. Jensen, M., S. Schage, and J. Schwenk (2010). Towards an anonymous access control and accountability scheme for cloud computing. In Proceedings of the IEEE 3rd International Conference on Cloud Computing (CLOUD), pp. 540-541.

[3]. Backes, M., J. Camenisch, and D. Sommer (2005). Anonymous yet accountable access control. In Proceedings of the ACM Workshop on Privacy in the Electronic Society (WPES), pp. 40-46.

[4]. Abbott, T., K. Lai, M. Lieberman, and E. Price (2007). Browser based attacks on Tor. In Proceedings of the 7th International Conference on Privacy Enhancing Technologies (PET), pp. 184-199.

[5]. Hopper, N., E. Y. Vasserman, and E. Chan-Tin (2010). How much anonymity does network latency leak? ACM Transactions on Information and System Security (TISSEC), 13 (2).

[6]. Freedman, M., E. Sit, J. Cates, and R. Morris (2002). Tarzan: A peer-to-peer anonymizing network layer. In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS), pp. 193-206.

[7]. Cornelli, F., E. Damiani, S. di Vimercati, S. Paraboschi, and P. Samarati (2002). Choosing reputable servants in a P2P network. In Proceedings of the 11th International World Wide Web Conference (WWW), pp. 376-386.

[8]. Gnutella (2010). http://www.gnutella.com.

[9]. Damiani, E., S. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante (2002). A reputation based approach for choosing reliable resources in peer to peer networks. In Proceedings of the 9th ACM Conference on Computer and Communications Security (CCS),pp. 207-216.

[10]. He, Q., J. Yan, H. Jin, and Y. Yang (2009). ServiceTrust: Supporting reputation-oriented service selection. In Proceedings of the 7th International Joint Conference on Service oriented Computing, pp. 269-284.

[11]. Bachrach, Y., A. Parnes, A. Procaccia, and J. Rosenschein (2009). Gossip based aggregation of trust in decentralized reputation systems. Journal of Autonomous Agents and Multiagent Systems (AAMAS) 19 (2), 153-172.

[12]. Stoica, I., R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan (2001). Chord: A scalable peer to peer lookup service for internet applications. In Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer and Communications (SIGCOMM), pp. 149-160.

[13]. Ratnasamy, S., P. Francis, M. Handley, R. Karp, and S. Schenker (2001). A scalable, content addressable network. In Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer and Communications (SIGCOMM), pp. 161-172.

[14]. Zhao, B., L. Huang, J. Stribling, S. Rhea, A. Joseph, and J. Kubiatowicz (2004). Tapestry: A resilient global scale overlay for service deployment. IEEE Journal on Selected Areas in Communications (JSAC) 22 (1), 41-53.