Security and Privacy for Group Data Sharing Mobile Computing Environment

[1]Sai Kumar, [2]Rakshita.N, [3]Sharon Rubanvathi.M,
[1,2,3]BCA, Student, Indian Academy Degree College Autonomous, Bangalore, India

Abstract: Time and Trend has its own significance to build the technology smarter, better and easier to the end user. To the Better stretch of the Information Technology, the Innovation and renovation has changed computing approach to the next level. In this paper, we try to give the glimpse of the contextual virtual cloud storage in the public data distribution in terms of the Database in the virtualized Data Storage. These days cloud storage become common, but having the constraint towards the technical advancement is the Security. If we consider behavioral aspect of the cloud storage, we will come across much aspect. Hence, In this one we have overcome the public protection in terms of the privacy towards the authorization of public audit, which we call it as the best to the trend of the acknowledgement based identification with the cryptographic model where ever the node to the parallel cloud distributed elastic stretchable environment with the high end cloud data center marinating the graphics of the flow triggering the security in the public Domain. Considering all the aspects ; the classical Database and these days database has its own advantage to make overcome the hindrance to make the scalability, performance , efficiency and at the end the security is the major which the no-sqlite like Cassandra can overcome.

Keywords: *Data Storage, Privacy Preserving, Cloud Database, Cassandra, Confidentiality, Encryption, Adaptively Cost Model*

I. INTRODUCTION

Cloud Cover trusts the cloud platform. Clouds can attain suitable trustworthiness through trust management, replication, virtualization, and a variety of other technologies not typically available to mobile devices and other, stand-alone, cloud-assisted machines. Privacy preservation of computation results is beyond our scope. For such protection, we refer the reader to numerous related works on that subject, including Anonymous Cloud; secure multiparty computation, and differential privacy. Proof validation through checkpoint chaining engenders a natural trade-o between assurance and computational expense through spot-checking. A spot-checking validated recomputed and checks each segment in the checkpoint chain with probability p. This reduces the total computation cost to a fraction p of the total, and detects erroneous computation results with probability p. This is necessary for the checkpoint equivalence. Cloud Cover therefore Extends Java Continuation class with an equal's method that compares two suspended program states for semantic equivalence. Thus, clients may tune parameter p in accordance with their desired level of assurance and the expense of cloud computing time. Although Java supports suspension and resumption of computations via continuations, it does not support continuation equivalence-checking.

II. RELATED

WORK Cloud Cover proofs have the advantageous quality that the task of verifying them can be parallelized almost arbitrarily even when the original computation is not parallelizable. Thus, they derive maximal benefit from massively parallel architectures, like clouds. To demonstrate, we implement Cloud Cover for Hadoop Map Reduce , and use it validate non-parallelizable Java computations for message digest generation using SHA1 (National Institute of Standards and Technology, 1995) and MD5 cryptographic hash functions. Experimental results indicate that Cloud Cover scales extremely well, with the only practical limit to parallelization stemming from the fixed overhead of dispatching new mappers and reducers. The checker is deployed on a HADOOP (Apache, 2013) cluster consisting of 6 Data Nodes and 1 Name Node. Node hardware is comprised of Intel Pentium IV 2.40, 3.00GHz processors with 2{4GB of memory each, running Unix operating systems.

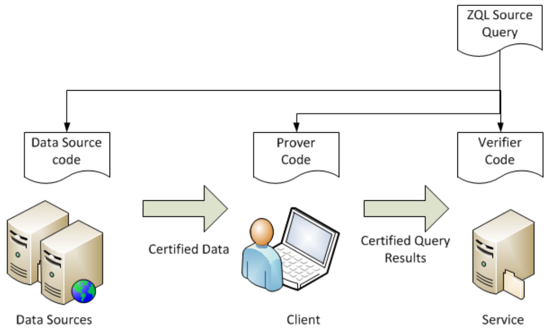

Figure 2. Data Center to Monitor the Data Flow

Java was installed and configured on each Data Node in the Hadoop distributed environment, making it available to distributed jobs. We implemented a mechanism for reading and writing checkpoints for map in Hadoop in an appropriate file format for equality-checking with Java. LZO compression was applied to all Hadoop file transfers to minimize transfer and storage costs. For trusted and trusted components of Cloud Cover, we use standard desktop computers with configurations similar to the individual cloud nodes above. For experiments, we select two non-parallelizable cryptographic.

III. METHODOLOGY

Technology has its own significance at the time when people having the extension for the more and more research and it's from Abacus to today's' cloud Computing. In the context of revolution of technology and its great advantage to its social, behavioral and other technical aspect where we come across the best of the cloud to province the virtual global village as the global world. In order for an attack against Anonymous Cloud to succeed, the manager or master node (or both) must be malicious. Managers are the only principals that receive decrypt able access tokens or credentials, and all other communications involving pseudonyms and data are conducted via Tor circuits having the master node as the only un-trusted endpoint. Managers are separate from CPs and have a much smaller attack surface because they do not process customer-submitted computations. Our experiments therefore assume that managers are trusted, but that master nodes are always malicious. In addition, we assume that a percentage p of slave nodes are also malicious and collude with the malicious master node in an effort to violate privacy. Aside from verifying checkpoint chain segments in parallel, we additionally parallelized the checkpoint equality checking procedure in our implementation. Continuations are stacks that can be partitioned arbitrarily into sub-stacks that can all be checked in parallel for equivalence. We implemented this for Java by introducing a continuation compare method. During comparison, instead of equality-checking each pair of objects inside the checkpoints, a map can redirect them to other map by submitting new jobs in HADOOP. The advantage is that if any individual checkpoint-pair is extremely large (e.g., very large stacks), then the checkpoint equality-checking job can be parallelized to compensate. In our experiments, the stacks are not that large, so this feature went unexercised.

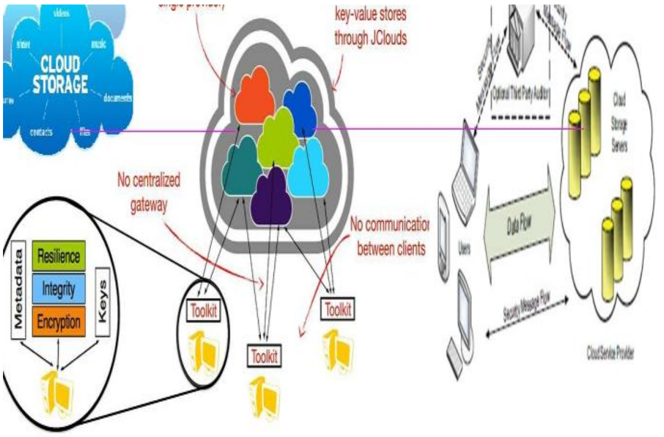

Figure 3. Architecture Design of the Secured Public Data in the Database of Cassandra

Therefore, using this approach not only the security and privacy concerns of cloud consumers can be addressed more effectively, but also the burden of managing endusers' identities and fine-granular access control will be reduced from cloud service providers. Unfortunately, all of these approaches require a significant redesign of most software. For example, typical Android apps are not easily modified to contain inextricable, secret computations or cryptographically verifiable compositions. Two states are equivalent if they consist of equal-length stacks whose corresponding slots contain equivalent values and objects. Deciding such semantic equivalence is non-trivial in general; for example, the states may contain objects with private fields to which the continuation object lacks access, or they may include fields whose values are semantically equivalent but no identical. Fortunately, all Java objects have their own equal's methods, which encode an object-specific notion of semantic equivalence.

A. Evaluation and Analysis

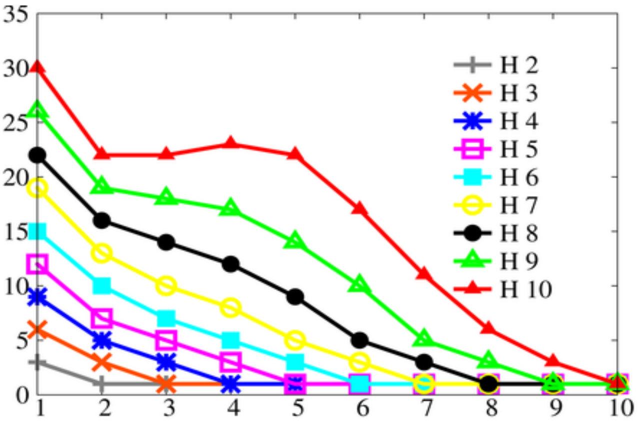

Figure 4. Node with the Peak Value

As a result, few mainstream mobile computing devices have adopted these technologies. Moreover, many of these solutions rely on software obfuscation, which does not provide rigorous guarantees, since clever attackers can potentially reverse the obfuscation the sharp increase in communications overhead potentially invites denial-of-service attacks by customers who request unreasonably long circuits. Recall that master nodes can report computational expense information associated with anonymous jobs to managers by labeling it with the encrypted ownership data they received during authentication.

We therefore recommend incentivizing reasonable values of k by charging customers proportionally to the communications overhead incurred by their demanded level of privacy. This allows the master node to report the expense without knowing the identity of the customer. Managers may also want to impose a mandatory upper limit on k during authentication to further control congestion.

CONCLUSION AND FUTURE WORK

Cloud computing is an emerging paradigm which its cost-effectiveness and edibility have given it a tremendous momentum. In this paper, we try to put forward the concept of the cloud in the aspect of the privacy preserving towards the public shared node data. It may lead to the extent of the cloud with the variant of the most suitable technological advancement of the recent solution. In the Public shared cloud computing where the data passed through the network which needs to be robust, secure and highly preserved in the sense no can call replicate the data while reaching to the next node of the cloud server? Hence, we can make the sense of the cloud for the further research oriented making the global world as the data can be se3cured in the cloud architectural designs model of the Data center. However, there are many security challenges that, if not addressed well, may impede its fast Adoption and growth. This dissertation primarily addresses the problem of sharing, managing and controlling access to sensitive resources and services in an integrated cloud environment. The primary conclusion of our research is that Adoption of user-centric security models and shifting certain parts of communication and computation to the client side allows us to provide the cloud consumers with more visibility and control over their resources.

References

[1] B. Wang, B. Li, and H. Li, "Oruta: Privacy Preserving Public Auditing for Shared Data in the Cloud," Proc. IEEE Fifth Int'l Conf.Cloud Computing, pp. 295-302, 2012.

[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," Comm. ACM, vol. 53, no. 4, pp. 50-58, Apr. 2010.

[3] K. Ren, C. Wang, and Q. Wang, "Security Challenges for the Public Cloud," IEEE Internet Computing, vol. 16, no. 1, pp. 69-73, 2012.

[4] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud Data Protection for the Masses," Computer, vol. 45, no. 1, pp. 39-45, 2012.

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

[6] B. Wang, M. Li, S.S. Chow, and H. Li, "Computing Encrypted Cloud Data Efficiently under Multiple Keys," Proc. IEEE Conf. Comm. and Network Security (CNS '13), pp. 90-99, 2013.