Scientific Journal Impact Factor (SJIF) : 1.711
International Journal of Modern Trends in Engineering and Research
ISSN (ONLINE) : 2349 - 9745 ISSN (PRINT) : 2393 - 8161

Enhanced Blind Signature Based Cloud Transactions

Mr.R.MOHANA SUNDARAM[1]
[1]*Department of MCA, SSM College of Engineering*

Abstract -A blind signature scheme allows a receiver to obtain a signature on a message such that both the message and the resulting signature remain unknown to the signer. We refer the readers for a formal definition of a blind signature scheme, which should bear the properties of verifiability, unlinkability and unforgeability. Blind signature scheme, where the restrictiveness property is incorporated into the blind signature scheme such that the message being signed must contain encoded information. As the name suggests, this property restricts the user in the blind signature scheme to embed some account-related secret information into what is being signed by the bank, such that this secret can be recovered by the bank to identify a user if and only if he double-spends. The restrictiveness property is essentially the guarantee for traceability in the restrictive blind signature systems. In order to maintain security of the network against attacks and the fairness among clients, the home server manager may control the access of each client by issuing tickets based on the misbehavior history of the client, which reflects the server manager's confidence about the client to act properly. Ticket issuance occurs when the client initially attempts to access the network or when all previously issued tickets are depleted. The client needs to reveal his real ID to the server manager in order to obtain a ticket since the server manager has to ensure the authenticity of this client.

Keywords-message; signature; bank; ticket; manager

I. INTRODUCTION

Cloud computing is the use of computing resources (hardware and software) that are delivered as a service over a network (typically the Internet). The name comes from the common use of a cloud-shaped symbol as an abstraction for the complex infrastructure it contains in system diagrams. Cloud computing entrusts remote services with a user's data, software and computation. Cloud computing consists of hardware and software resources made available on the Internet as managed third-party services. These services typically provide access to advanced software applications and high-end networks of server computers.

The goal of cloud computing is to apply traditional supercomputing, or high-performance computing power, normally used by military and research facilities, to perform tens of trillions of computations per second, in consumer-oriented applications such as financial portfolios, to deliver personalized information, to provide data storage or to power large, immersive computer games. The cloud computing uses network of large groups of servers typically running low-cost consumer PC technology with specialized connections to spread data-processing chores across them. This shared IT infrastructure contains large pools of systems that are linked together. Often, virtualization techniques are used to maximize the power of cloud computing.

Benefits of cloud computing

- **Achieve economies of scale** – increase volume output or productivity with fewer people. Your cost per unit, project or product plummets.

- **Reduce spending on technology infrastructure.** Maintain easy access to your information with minimal upfront spending. Pay as you go (weekly, quarterly or yearly), based on demand.
- **Globalize your workforce on the cheap.** People worldwide can access the cloud, provided they have an Internet connection.
- **Streamline processes.** Get more work done in less time with less people.
- **Reduce capital costs.** There's no need to spend big money on hardware, software or licensing fees.

II. RELATED WORK

A. Elastras: An Elastic Transactional Data Store in the Cloud

Over the last couple of years, "Cloud Computing" or "Elastic Computing" has emerged as a compelling and successful paradigm for internet scale computing. One of the major contributing factors to this success is the elasticity of resources. In spite of the elasticity provided by the infrastructure and the scalable design of the applications, the elephant (or the underlying database), which drives most of these web-based applications, is not very elastic and scalable, and hence limits scalability. In this paper, we propose ElasTraS which addresses this issue of scalability and elasticity of the data store in a cloud computing environment to leverage from the elastic nature of the underlying infrastructure, while providing scalable transactional data access.

B. Safety and Consistency in Policy-Based Authorization Systems

In trust negotiation and other distributed proving systems, networked entities cooperate to form proofs that are justified by collections of certified attributes. These attributes may be obtained through interactions with any number of external entities and are collected and validated over an extended period of time. Though these collections of credentials in some ways resemble partial system snapshots, these systems currently lack the notion of a consistent global state in which the satisfaction of authorization policies should be checked. In this paper, we argue that unlike the notions of consistency studied in other areas of distributed computing, the level of consistency required during policy evaluation is predicated solely upon the security requirements of the policy evaluator. As such, there is little incentive for entities to participate in complicated consistency preservation schemes like those used in distributed computing, distributed.

C. Automated Trust Negotiation Using Cryptographic Credentials

In automated trust negotiation (ATN), two parties exchange digitally signed credentials that contain attribute information to establish trust and make access control decisions. Because the information in question is often sensitive, credentials are protected according to access control policies. In traditional ATN, credentials are transmitted either in their entirety or not at all. This approach can at times fail unnecessarily, either because a cyclic dependency makes neither negotiator willing to reveal her credential before her opponent, because the opponent must be authorized for all attributes packaged together in a credential to receive any of them, or because it is necessary to fully disclose exact attribute values, rather than merely proving they satisfy some predicate.
Recently, several cryptographic credential schemes and associated protocols have been developed to address these and other problems. However, they can be used only as fragments of an ATN process. This paper introduces a framework for ATN in which the diverse credential schemes and protocols can be combined, integrated, and used as needed. A policy language is introduced that enables negotiators to specify authorization requirements that must be met by an opponent to receive various

amounts of information about certified attributes and the credentials that contain it. The language also supports the use of uncertified attributes, allowing them to be required as part of policy satisfaction, and to place their (automatic) disclosure under policy control.

D. An Efficient System for Non- Transferable Anonymous Credentials with Optional Anonymity Revocation

A credential system is a system in which users can obtain credentials from organizations and demonstrate possession of these credentials. Such a system is anonymous when transactions carried out by the same user cannot be linked. An anonymous credential system is of significant practical relevance because it is the best means of providing privacy for users. In this paper we propose a practical anonymous credential system that is based on the strong RSA assumption and the decisional Diffie-Hellman assumption modulo a safe prime product and is considerably superior to existing ones:

- We give the first practical solution that allows a user to unlinkably demonstrate possession of a credential as many times as necessary without involving the issuing organization.
- To prevent misuse of anonymity, our scheme is the first to offer optional anonymity revocation for particular transactions.
- Our scheme offers separability: all organizations can choose their cryptographic keys independently of each other.

Moreover, we suggest more effective means of preventing users from sharing their credentials, by introducing all-or-nothing sharing: a user, who allows a friend to use one of her credentials once, gives him the ability to use all of her credentials, i.e., taking over her identity. This is implemented by a new primitive, called circular encryption, which is of independent interest, and can be realized from any semantically secure cryptosystem in the random oracle model.

E. Recovery and Performance of Atomic Commit Processing in Distributed Database Systems

A transaction is traditionally defined so as to provide the properties of atomicity, consistency, integrity, and durability (ACID) for any operation it performs. In order to ensure the atomicity of distributed transactions, an atomic commit protocol needs to be followed by all sites participating in a transaction execution to agree on the final outcome, that is, commit or abort. A variety of commit protocols have been proposed that either enhance the performance of the classical two-phase commit protocol during normal processing or reduce the cost of recovery processing after a failure. In this chapter, we survey a number of two-phase commit variants and optimizations, including some recent ones, providing an insight in the performance trade-off between normal and recovery processing. We also analyze the performance of a representative set of commit protocols both analytically as well as empirically using simulation.

III IMPLEMENTATION

A. Server Model

In this paper, we design a cloud infrastructure consisting of a set of servers, where each server is responsible for hosting a subset of all data items belonging to a specific application domain.

B. Cloud User Module

In this cloud user module, users interact with the system by submitting queries or update requests encapsulated in ACID transactions, since transactions are executed over time, the state information of the credentials and the policies enforced by different servers are subject to changes at

any time instance, therefore it becomes important to introduce precise definitions for the different consistency levels that could be achieved within a transaction's lifetime. These consistency models strengthen the trusted transaction definition by defining the environment in which policy versions are consistent relative to the rest of the system. Before we do that, we define a transaction's view in terms of the different proofs of authorization evaluated during the lifetime of a particular transaction.

C. Transaction Manager

A transaction is submitted to a Transaction Manager(TM) that coordinates its execution. Multiple TMs could be invoked as the system workload increases for load balancing, but each transaction is handled by only one TM.A common characteristic of most of our proposed approaches to achieve trusted transactions is the need for policy consistency validation at the end of a transaction. That is, in order for a trusted transaction to commit, its TM has to enforce either view or global consistency among the servers participating in the transaction.

D. Certificate Authorities

We use the set of all credentials, which are issued by the Certificate Authorities (CAs) within the system. We assume that each CA offers an online method that allows any server to check the current status of credentials that it has issued. In this Certificate Authorities**,** we provide a Safe transaction. A safe transaction is a transaction that is both trusted (i.e., satisfies the correctness properties of proofs of authorization) and database correct (i.e., satisfies the data integrity constraints), also develop Two Phase Validation system. As the name implies, 2PV operates in two phases: collection and validation. During collection, the TM first sends a Prepare-to-Validate message to each participant server. In response to this message, each participant 1) evaluates the proofs for each query of the transaction using the latest policies it has available and 2) sends a reply back to the TM containing the truth value (TRUE/FALSE) of those proofs along with the version number and policy identifier for each policy used.

E. Evidence composition

Evidence is defined as information that is used to establish proof about the occurrence of an event or action, the time of occurrence, the parties involved in the event, and the outcome of the event. The purpose of Evidence is to resolve a dispute about the data resulted from data transmission.

F. Payment report composition/submission

A cloud report is updated by the Trusted Party. After the Trusted Party verification any transaction is updated. By using the Certificate, Public Key, Symmetric Key, the transaction is approved. If the transaction is faulty or fraud, then the Trusted Party verifies it and the transaction is cancelled.

IV CONCLUSION

In our system, the AC can process the payment reports to know the number of relayed/dropped messages by each node. In our future work, we will develop a trust system based on processing the data reports to maintain a trust value for each user. The users that relay messages more successfully will have higher trust values, such as the low-mobility and the large-hardware-resources nodes. Based on these trust values, we will propose a trust-based routing protocol to route messages through the highly trusted nodes to minimize the probability of dropping the messages, and thus improve the network performance in terms of throughput and packet delivery ratio. However, the trust system

should be secure against singular and collusive attacks, and the routing protocol should make smart decisions regarding node selection with low overhead.

REFERENCES

[1] M. Armbrust et al., "Above the Clouds: A Berkeley View of Cloud Computing," technical report, Univ. of California, Feb. 2009.
[2] S. Das, D. Agrawal, and A.E. Abbadi, "Elastras: An Elastic Transactional Data Store in the Cloud," Proc. Conf. Hot Topics in Cloud Computing (USENIX HotCloud '09), 2009.
[3] D.J. Abadi, "Data Management in the Cloud: Limitations and Opportunities," IEEE Data Eng. Bull., vol. 32, no. 1, pp. 3-12, Mar. 2009.
[4] A.J. Lee and M. Winslett, "Safety and Consistency in Policy-Based Authorization Systems," Proc. 13th ACM Conf. Computer and Comm. Security (CCS '06), 2006.
[5] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - Ocsp," RFC 2560, http://tools.ietf.org/html/rfc5280, June 1999.
[6] E. Rissanen, "Extensible Access Control Markup Language (Xacml) Version 3.0," http://docs.oasis-open.org/xacml/3.0/ xacml-3.0-core-spec-os-en.html, Jan. 2013.
[7] D. Cooper et al., "Internet x.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, http://tools.ietf.org/html/rfc5280, May 2008.
[8] J. Li, N. Li, and W.H. Winsborough, "Automated Trust Negotiation Using Cryptographic Credentials," Proc. 12th ACM Conf. Computer and Comm. Security (CCS '05), Nov. 2005.
[9] L. Bauer et al., "Distributed Proving in Access-Control Systems," Proc. IEEE Symp. Security and Privacy, May 2005.
[10] J. Li and N. Li, "OACerts: Oblivious Attribute Based Certificates," IEEE Trans. Dependable and Secure Computing, vol. 3, no. 4, pp. 340-352, Oct.-Dec. 2006.
[11] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-Transferable Anonymous Credentials with Optional Anonymity Revocation," Proc. Int'l Conf. Theory and Application of Cryptographic Techniques: Advances in Cryptology (EUROCRYPT '01), 2001.
[12] P.K. Chrysanthis, G. Samaras, and Y.J. Al-Houmaily, "Recovery and Performance of Atomic Commit Processing in Distributed Database Systems," Recovery Mechanisms in Database Systems, Prentice Hall PTR, 1998.
[13] M.K. Iskander, D.W. Wilkinson, A.J. Lee, and P.K. Chrysanthis, "Enforcing Policy and Data Consistency of Cloud Transactions," Proc. IEEE Second Int'l Workshop Security and Privacy in Cloud Computing (ICDCS-SPCCICDCS-SPCC), 2011.
[14] G. DeCandia et al., "Dynamo: Amazons Highly Available Key-Value Store," Proc. 21st ACM SIGOPS Symp. Operating Systems Principles (SOSP '07), 2007.
[15] F. Chang et al., "Bigtable: A Distributed Storage System for Structured Data," Proc. Seventh USENIX Symp. Operating System Design and Implementation (OSDI '06), 2006.
[16] A. Lakshman and P. Malik, "Cassandra- A Decentralized Structured Storage System," ACM SIGOPS Operating Systems Rev., vol. 44, pp. 35-40, Apr. 2010.
[17] B.F. Cooper et al., "PNUTS: Yahoo!'s Hosted Data Serving Platform," Proc. VLDB Endowment, vol. 1, pp. 1277-1288, Aug. 2008.
[18] W. Vogels, "Eventually Consistent," Comm. ACM, vol. 52, pp. 40-44, Jan. 2009.
[19] H. Guo, P.-A. Larson, R. Ramakrishnan, and J. Goldstein, "Relaxed Currency and Consistency: How to Say "Good Enough" in SQL," Proc. ACM Int'l Conf. Management of Data (SIGMOD '04), 2004.

