The International Journal Of Engineering And Science (IJES)
|| Volume || 5 || Issue || 3 || Pages || PP -01-04 || 2016 ||
ISSN (e): 2319 – 1813 ISSN (p): 2319 – 1805

Reconfiguration of systems implementing membership service to enhance reliability

[1]Vatsalya, [2]Dr. Kanhaiya Lal
[1,2]*Department of Computer Science, BIT Mesra, Patna Campus*

ABSTRACT
Byzantine-fault-tolerant replication enhances the availability and reliability of Internet services that store critical state and preserve it despite attacks or software errors. However, existing Byzantine-fault-tolerant storage systems either assume a static set of replicas, or have limitations in how they handle reconfigurations (e.g., in terms of the scalability of the solutions or the consistency levels they provide). This can be problematic in long-lived, large-scale systems where system membership is likely to change during the system lifetime. In this paper, we present a complete solution for dynamically changing system membership in a large-scale Byzantine-fault-tolerant system. We present a service that tracks system membership and periodically notifies other system nodes of membership changes. The membership service runs mostly automatically, to avoid human configuration errors; is itself Byzantine fault- tolerant andreconfigurable; and provides applications with a sequence of consistent views of the system membership. We demonstrate the utility of this membership service by using it in a novel distributed hash table called dBQS that provides atomic semantics even across changes in replica sets. dBQS is interesting in its own right because its storage algorithms extend existing Byzantine quorum protocols to handle changes in the replica set, and because it differs from previous DHTs by providing Byzantine fault tolerance and offering strong semantics. We implemented the membership service and dBQS. Our results show that the approach works well, in practice: the membership service is able to manage a large system and the cost to change the system membership is low.

Keywords*–Byzantine fault tolerance, reconfiguration, System membership, membership revocation*

Date of Submission: 07 March 2016 Date of Accepted: 21 March 2016

I. INTRODUCTION

Wireless Mesh Network (WMN) is a promising technology and is expected to be widespread due to its low investment feature and the wireless broadband services it supports, attractive to both service providers and users. However, security issues inherent in WMNs or any wireless networks need be considered before the deployment and proliferation of these networks, since it is unappealing to subscribers to obtain services without security and privacy guarantees. Wireless security has been the hot topic in the literature for various network technologies such as cellular networks, wireless local area networks (WLANs), wireless sensor networks, mobile ad hoc networks (MANETs), and vehicular ad hoc networks (VANETs). Anonymity and privacy issues have gained considerable research efforts in the literature, which have focused on investigating anonymity in different context or application scenarios. One requirement for anonymity is to unlink a user's identity to his or her specific activities, such as the anonymity fulfilled in the untraceable e-cash systems and the P2P payment systems, where the payments cannot be linked to the identity of a payer by the bank or broker. Anonymity is also required to hide the location information of a user to prevent movement tracing, as is important in mobile networks and VANETs. In wireless communication systems, it is easier for a global observer to mount traffic analysis attacks by following the packet forwarding path than in wired networks. Thus, routing anonymity is indispensable, which conceals the confidential communication relationship of two parties by building an anonymous path between them. Nevertheless, unconditional anonymity may incur insider attacks since misbehaving users are no longer traceable. Therefore, traceability is highly desirable such as in e-cash systems, where it is used for detecting and tracing double-spenders.

II. LITERATURE SURVEY

2.1Authentication in a Reconfigurable Byzantine Fault Tolerant System

Byzantine (i.e. arbitrary) faults occur as a result of software errors and malicious attacks; they are increasingly a problem as people come to depend more and more on online services. Systems that provide critical services must behave correctly in the face of Byzantine faults. Correct service in the presence of failures is achieved through replication: the service runs at a number of replica servers and as more than a third of the replicas are non-faulty, the group as a whole continues to behave correctly. We would like the service to be able to authenticate data. Authenticated data is data that more than a third of the service is willing to sign. If a long-lived replicated service can tolerate f failures, then we do not want the adversary to have the lifetime of the system to compromise more than f replicas. One way to limit the amount of time an adversary has to compromise more than f replicas is to reconfigure the system, moving the responsibility for the service from one group of servers to a new group of servers. Reconfiguration allows faulty servers to be removed from service and replaced with newly introduced correct servers. Reconfiguration is also desirable because the servers can become targets for malicious attacks, and moving the service thwarts such attacks.

2.2 Secure routing for structured peer-to-peer overlay networks

Structured peer-to-peer overlay networks provide a substrate for the construction of large-scale, decentralized applications, including distributed storage, group communication, and content distribution. These overlays are highly resilient; they can route messages correctly even when a large fraction of the nodes crash or the network partitions. But current overlays are not secure; even a small fraction of malicious nodes can prevent correct message delivery throughout the overlay. This problem is particularly serious in open peer-to-peer systems, where many diverse, autonomous parties without preexisting trust relationships wish to pool their resources. This paper studies attacks aimed at preventing correct message delivery in structured peer-to-peer overlays and presents defenses to these attacks. We describe and evaluate techniques that allow nodes to join the overlay, to maintain routing state, and to forward messages securely in the presence of malicious nodes.

2.3. Exploiting a Secure Log for Wide-Area Distributed Storage

Antiquity is a wide-area distributed storage system designed to provide a simple storage service for applications like file systems and back-up. The design assumes that all servers eventually fail and attempts to maintain data despite those failures. Antiquity uses a secure log to maintain data integrity, replicates each log on multiple servers for durability, and uses dynamic Byzantine fault tolerant quorum protocols to ensure consistency among replicas. We present Antiquity's design and an experimental evaluation with global and local test beds. Antiquity has been running for over two months on 400+ Planet Lab servers storing nearly 20,000 logs totaling more than 84 GB of data. Despite constant server churn, all logs remain durable.

2.4 Existing System

In Existing System, replication enhanced the reliability of internet services to store the data's. The preserved data to be secured from software errors. But, existing Byzantine-fault tolerant systems is a static set of replicas. It has no limitations. So, scalability is inconsistency. So, these data's are not came for long-lived systems. The existence of the following cryptographic techniques that an adversary cannot subvert: a collision resistant hash function, a public key cryptography scheme, and forward-secure signing key and the existence of a proactive threshold signature protocol.

2.5 Proposed System

In Proposed System, has two parts. The first is a membership service (MS) that tracks and responds to membership changes. The MS works mostly automatically, and requires only minimal human intervention; this way we can reduce manual configuration errors, which are a major cause of disruption in computer systems periodically, the MS publishes a new system membership; in this way it provides a globally consistent view of the set of available servers. The choice of strong consistency makes it easier to implement applications, since it allows clients and servers to make consistent local decisions about which servers are currently responsible for which parts of the service. The second part of our solution addresses the problem of how to reconfigure applications automatically as system membership changes. We present a storage system, dBQS that provides Byzantine-fault-tolerant replicated storage with strong consistency.

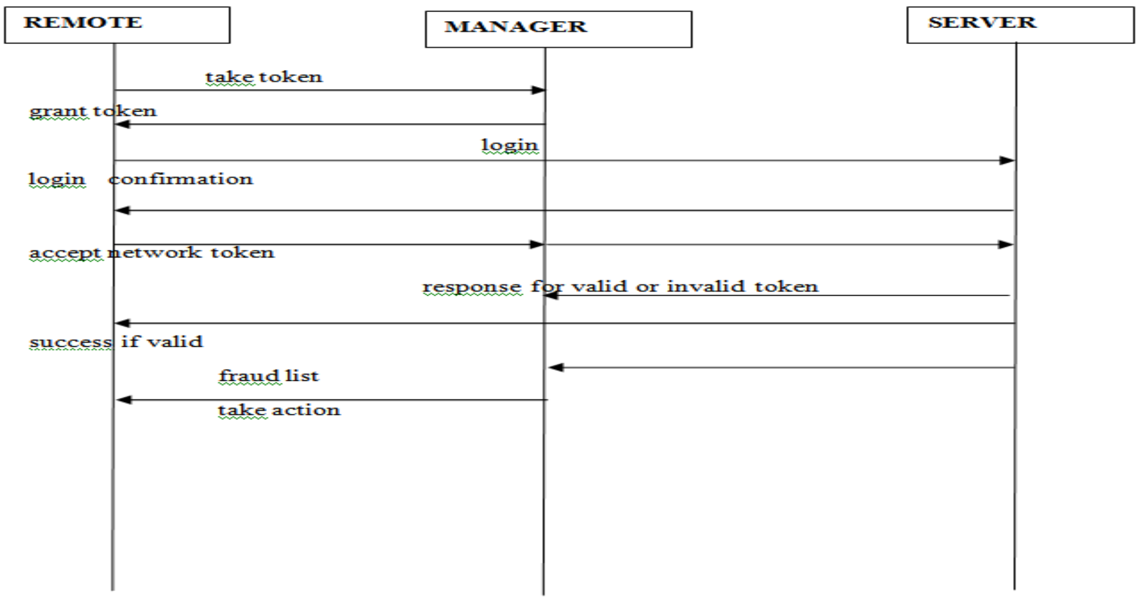
III. SYSTEM REQUIREMENT

This Chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of this dissertation as well as the functional and non-functional requirement of this dissertation.

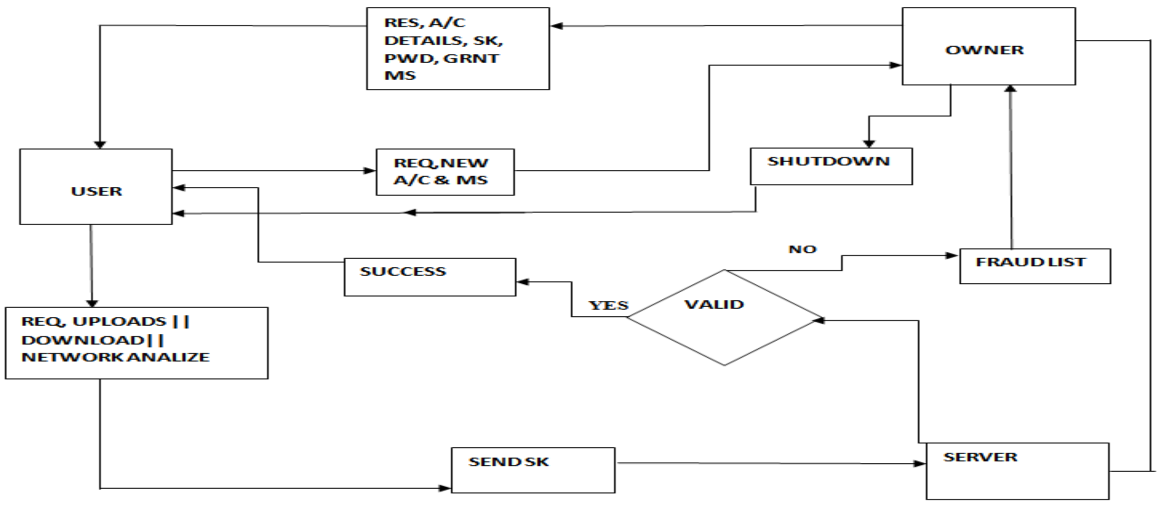
Functional	Control of Misbehaving Users at Gateway, Fraud Detections
Non- Functional	Server never controls the Manager
External interface	WiFi LAN , GateWay
Performance	Detecting Frauds based on the membership
Attributes	Membership Deposit, membershipRevocation, membership Issuance,Fraud Detection,Misbehaving Users

Table: 3.1 Summary of SRS

IV. SEQUENCE DIAGRAM



V. ACTIVITY DIAGRAM


RES, A/C DETAILS, SK, PWD, GRNT MS
OWNER
USER
REQ,NEW A/C & MS
SHUTDOWN
SUCCESS
NO
FRAUD LIST
YES
VALID
REQ, UPLOADS || DOWNLOAD|| NETWORK ANALIZE
SEND SK
SERVER

VI. CONCLUSION

This paper presents a complete solution for building large scale, long-lived systems that must preserve critical state in spite of malicious attacks and Byzantine failures. We present a storage service with these characteristics called dBQS, and a membership service that is part of the overall system design, but can be reused by any Byzantine-faulttolerant large-scale system.

The membership service tracks the current system membership in a way that is mostly automatic, to avoid human configuration errors. It is resilient to arbitrary faults of the nodes that implement it, and is reconfigurable, allowing us to change the set of nodes that implement theMSwhen oldnodes fail, or periodically to avoid a targeted attack. When membership changes happen, the replicated service has work to do: responsibility must shift to the new replica group, and state transfer must take place from old replicas to new ones, yet the system must still provide the same semantics as in a static system. We show how this is accomplished in dBQS.

Existing System

In Existing System, replication enhanced the reliability of internet services tostore the data's. The preserved data to be secured from software errors. But, existingByzantine-fault tolerant systems is a static set of replicas. It has no limitations. So,scalability is inconsistency. So, these data's are not came for long-lived systems.The existence of the following cryptographic techniques that an adversary cannotsubvert: a collision resistant hash function, a public key cryptography scheme, andforward-secure signing key and the existence of a proactive threshold signature protocol.

Proposed System

In Proposed System, has two parts. The first is a membership service (MS) thattracks and responds to membership changes. The MS works mostly automatically, andrequires only minimal human intervention; this way we can reduce manual configurationerrors, which are a major cause of disruption in computer systems periodically, the MSpublishes a new system membership; in this way it provides a globally consistent view ofthe set of available servers. The choice of strong consistency makes it easier toimplement applications, since it allows clients and servers to make consistent localdecisions about which servers are currently responsible for which parts of the service.The second part of our solution addresses the problem of how to reconfigureapplications automatically as system membership changes. We present a storage system,dBQS that provides Byzantine-fault-tolerant replicated storage with strong consistency

REFERENCES

[1] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A.Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and W.Vogels, "Dynamo: Amazon's Highly Available Key-Value Store,"Proc. 21st ACM Symp. Operating Systems Principles, pp. 205-220,2007.

[2] J. Dean,, "Designs, Lessons and Advice from Building Large Distributed Systems," Proc. Third ACM SIGOPS Int'l Workshop Large Scale Distributed Systems and Middleware (LADIS '09), Key-note talk, 2009.

[3] Amazon S3 Availability Event, http://status.aws.amazon.com/s3-20080720.html, July 2008.

[4] K. Birman and T. Joseph, "Exploiting Virtual Synchrony in Distributed Systems," Proc. 11th ACM Symp. Operating Systems Principles, pp. 123-138, Nov. 1987.

[5] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H.Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," Proc. ACM SIGCOMM, 2001.

[6] M. Reiter, "A Secure Group Membership Protocol," IEEE Trans.Software Eng., vol. 22, no. 1, pp. 31-42, Jan. 1996.

[7] H.D. Johansen, A. Allavena, and R. van Renesse, "Fireflies:Scalable Support for Intrusion-Tolerant Network Overlays," Proc.European Conf. Computer Systems (EuroSys '06) , pp. 3-13, 2006.