Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.49 – 55

RESEARCH ARTICLE

TRUSTED ATOMIC WEB SERVICE TRANSACTION IN CLOUD

K. LOHESWARAN, S.BAGYASHREE, K.KAVIYA

Information Technology & Tamilnadu College of Engineering, India

Information Technology & Tamilnadu College of Engineering, India

Information Technology & Tamilnadu College of Engineering, India

loheswaran.k@gmail.com, shreeeemh@gmail.com, kaviyakesavan@gmail.com

*Abstract— **— Cloud is the emerging technology used in recent trends and for all WS-BA.I n this paper we proposed by providing more security for atomic transaction in web service. Considering online net banking system, the user will enter their user id and password for accessing their account details. They can view all the accounts across all branches of Net Bank locations online and as well affect fund transfers on real time basis within the Bank network. The fund transfers are stored in Net bank database using some services and if some crash or data loss occurs in database the replica is created for efficient transaction by using BFT algorithm. The services includes, Activation service, Registration service, Completion service, Coordinator service***

In activation services, creates a Coordinator object and a transaction context for each transaction. Essentially, the Activation service behaves like a factory object that creates Coordinator objects. The Registration service allows the Participants and the Initiator to register their end point references. The Completion service allows the Initiator to signal the start of the distributed commit. The Coordinator service runs the 2PC protocol, which ensures atomic commitment of the distributed transaction.

Keywords: cloud, BFT, replica, fund transaction, services included

1. Introduction

Cloud computing is a computing in which virtualised and standard resources, software and data are provided as a service over the Internet. Consumers and businesses can use the cloud to store data and applications, and can interact with the Cloud using mobiles, desktop computers, laptops etc. via the Internet from anywhere and at any time. The technology of Cloud computing entails the convergence of Grid and cluster computing, virtualisation, Web services and Service Oriented Architecture (SOA) **-** it offers the potential to set IT free from the costs and complexity of its typical physical infrastructure, allowing concepts such as Utility Computing to become meaningful. Key players include: IBM, HP, Google, Microsoft, Amazon Web Services, Salesforce.com, Net Suite, VMware.

Distributed applications composed of collections of Web Services [1] may call for diverse levels of reliability in different parts of the system. One Web Service may have stringent availability requirements, while it may ac cess other Web Services that do not. This characteristic may be by design, or due to the different Web Services being owned or deployed by different organizations. Web Services that must be highly available can be built using replication. Byzantine fault tolerance (BFT) is a long studied high reliability, replication technique that is designed to protect against arbitrary problems ranging from crash faults to software bugs and security violations. The Web Services application model requires a more general, multi-tiered, distributed system architecture, in which the functionality of a single Web Service application is a function of the aggregate behaviour of a variety of individual Web Services that may communicate with each other and that Proceedings of may have diverse reliability requirements. Therefore, reliability techniques for Web Services must allow for being incorporated into selective parts of multi-tiered applications that have heterogeneous reliability requirements. Because of this, BFT systems in their current form do not directly support being used as general infrastructure to build Web Service applications.

To address the interoperability issue, Erven- et proposed an extension to the WS-B specification, referred to as the web services-business activity-initiator (WS-BA-I)protocol. The WS-BA-I protocol separates the coordination functionality and the business logic, and standardizes the interactions between the initiator and the co-ordinator. With the WS-BA-I extension to WS-BA, it becomes possible for a third party to offer coordination services to enterprises that want to conduct WS-BA with minimum modifications to their workflow engines. However, for such coordinator services to be widely ad- opted, they must be trustworthy for their users. Several groups of researchers have investigated the issue of trust in the Internet and the Web; useful summaries of that research are provided in [1]. Our use of the term trustworthiness in this paper is closest to that of Grandison and Sloman [1], who defines trust as “the firm belief in the competence of an entity to act dependably, securely, and reliably within a specified context.” In this paper, we are particularly concerned with high availability and high integrity of the WS-BA coordination services .In this paper, first we analyze the threats to the WS-BA coordination services, and explore strategies to mitigate such threats. Due to its high cost, we avoid the use of a traditional BFT algorithm designed for general-purpose**s** stateful server replication. Instead, we present a lightweight

BFT algorithm that exploits the state model defined in the WS-BA specification. Our lightweight BFT algorithm does not guarantee that messages are delivered in total order at non faulty replicas, as that is not needed in this case. The using of BFT algorithm is not that trust -worthy so for more trust and secure transmission of data we implement another algorithm such as AES(Advanced encryption standard),and Random key generation. The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data. AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware. Key generation is the process of generating keys for security purpose. In our system we use a random key for secure our uploading file.

2. Assumptions and System Model

We assume that Web Service applications may be composed of multiple Web Services, and that these services may have diverse reliability requirements. These services may be owned and operated by different organizations, and therefore may not be able to share a common infrastructure or fault model. For each Byzantine-fault-tolerant service, we consider a distributed asynchronous system, and an inherently unreliable transmission medium .and also here we proposed high security and fast transmission by adding AES algorithm and random key generation algorithm and light weight BFT algorithm.

3. WS-BA Specification

The WS-BA specification describes how to coordinate long running business activities where the atomic transaction model is not appropriate. It defines a participant-side service and a set of coordinator-side services. The coordinator-side services include activation, registration, and coordinator services, which run in the same address space of the coordinator. The initiator of a business activity requests the activation service to generate a coordination context for the business activity and to create a coordinator object for the business activity. The coordinator object provides the registration and coordinator services, within the scope of the business activity. When the business activity is propagated to it, a web service registers with the registration service by providing an endpoint reference to its participant service. The registration service replies with an endpoint reference to the coordinator service. The coordinator service interacts with the participants via the business-agreement-with coordinator completion (BAwCC) or business-agreement with participant completion (BAwPC) protocol and it interacts with the initiator via the WS-BA-I protocol . For convenience, we refer to these services collectively as the coordinator.

A participant registers one of the two coordination protocols (BAwPC or BAwCC) with the coordinator of the business activity. We briefly describe the two coordination protocols below. For detailed state transition diagrams of the coordination protocols, please see the WS-BA specification For the BAwPC protocol, when a participant has finished its work for a business activity, the participant informs the coordinator by sending a completed message. The coordinator replies with either a close message or a compensate message, depending on the circumstances. If the business activity has completed successfully, the participant receives a close message. Otherwise, the participant receives a compensate message, and it must undo the completed work and restore the data it recorded at the outset of the business activity.

A participant might encounter a problem or fail during the processing of the business activity. If an error occurred when the participant was trying to complete its normal activity, it generates a fail message and sends that message to the coordinator, possibly on recovery from a transient fault. Similarly, if an error occurred when the participant was trying to cancel or compensate an activity, it generates a cancel complete message and sends that message to the coordinator.

For the BAwCC protocol, the completion notification comes from the coordinator. The coordinator sends a complete message to the participants, informing them that they will not receive any new requests within the current business activity and that they should complete their processing. If a participant has successfully finished its work, it replies by sending a completed message. Other interactions between the coordinator and the participants are similar to those of the BAwPC protocol.

4. Algorithm

4.1 BFT: Lightweight BFT Algorithm

Based on the previous state model analysis, we have developed the lightweight BFT algorithm described below. The normal operation of the algorithm is for f ¼ 1.A client (i.e., the initiator or a participant) sends a request to all coordinator replicas. The request has the form <REQUEST; s; o; c>_c , where s is the sequence number of the request message, o is the operation to be invoked (including the co ordination context, if needed), c is the client identifier, and _c is the digital signature of the request message signed by the client .On receiving a request, a coordinator replica validates the signature, and checks the validity of the requested operation and whether the sequence number in the request matches its next expected sequence number. If the request passes this validation test, a coordinator replica multicasts a commit message to all of the other coordinator replicas. The commit message has the form <COMMIT; s; d; k>_k , is the sequence number of the request message, d is the digest of the request message, k is the replica identifier, and _k is the signature of the commit message signed by the replica k. and three theorems are used to prove light weight BFT .

4.1.1 Lightweight BFT Algorithm Working Flow

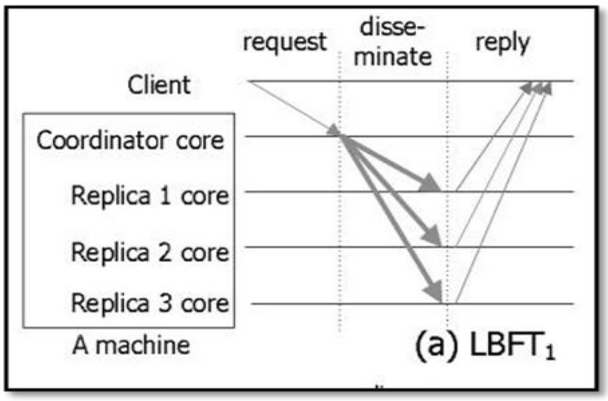

Fig: Lightweight algorithm work flow

Complex distributed applications combine the functionality of services from different providers to perform high level tasks. Mission-critical services shared by multiple applications must guarantee correct execution and availability in spite of failures. Fail-stop failures, such as host crashes, can be masked using redundant hosts. Achieving Byzantine Fault Tolerance (BFT) [1] requires a higher degree of replication1 since failures may be caused by malicious attacks band arbitrary software or hardware errors. Recent research has yielded practical algorithms [2–5] for Byzantine fault tolerant execution of passive2 services.

4.1.2 PROOFS

Theorem 1. If a Byzantine faulty sender sends conflicting requests (different requests with the same sequence number) to different non faulty coordinator replicas, at most one of those requests is delivered at all non faulty coordinator replicas. For proof refer [2]

Theorem 2. If a request is delivered at a non faulty coordinator replica, it is eventually delivered at all non faulty coordinator replicas in the sender's source order. For proof refer [1]
Theorem 3. The states of the non faulty coordinator replicas will eventually converge, and the initiator will eventually receive a response to a query regarding the state of the business activity, that is consistent with the converged state of the coordinator replicas.

4.1.3 Flowchart

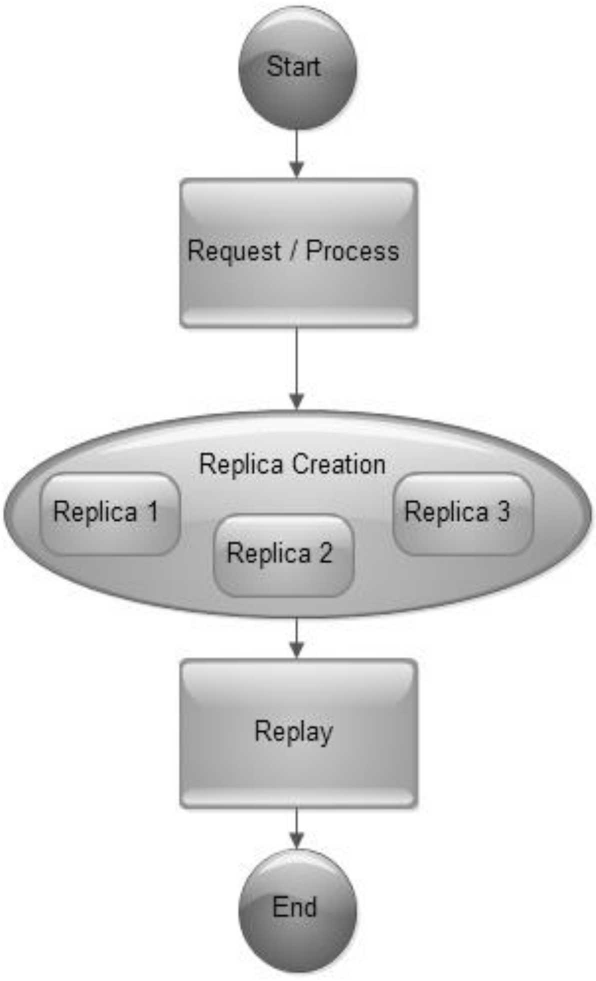

fig: flow chart of BFT algorithm

4.2 AES(Advanced Encryption Standard)

The Advanced Encryption Standard (AES) is a specification for the encryption of electronic data. AES is based on a design principle known as a substitution-permutation network, and is fast in both software and hardware.

AES is the successor of DES as standard symmetric encryption algorithm for US federal organizations (and as standard for pretty much everybody else, too). AES accepts keys of 128, 192 or 256 bits (128 bits is already very unbreakable), uses 128-bit blocks (so no issue there), and is efficient in both software and hardware. It was selected through an open competition involving hundreds of cryptographers during several years.

4.2.1 AES ALGORITHM WORK FLOW

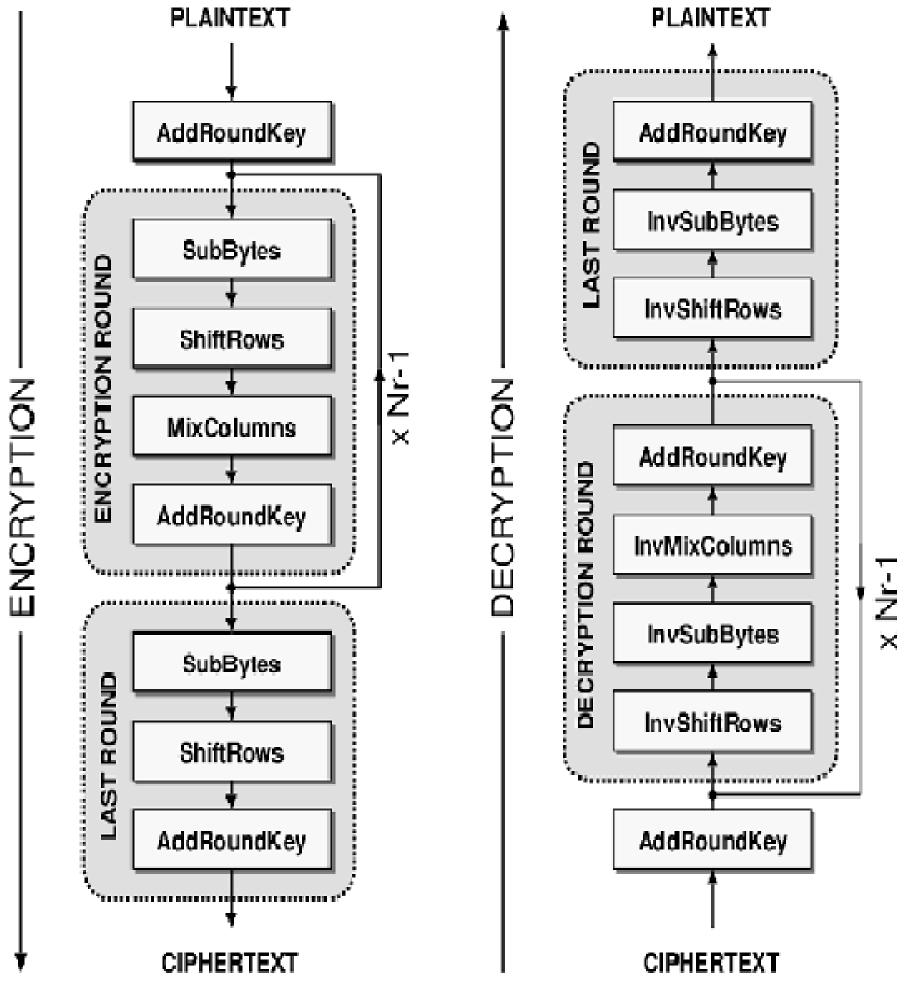

fig: AES algorithm work flow

4.3 Random Key Generation

Key generation is the process of generating keys for security purpose. In our system we using random key for secure our uploading file. We can share our keys to another user. The .net Framework provides RNG Crypto Service Provider class which Implements a cryptographic Random Number Generator (RNG) using the implementation provided by the cryptographic service provider (CSP). This class is usually used to generate random numbers. Although we can use this class to generate unique number in some sense but it is also not collision less. Moreover while generating key we can make key more complicated by making it as alpha numeric rather than numeric only for proof refer[2].

4.3.1 Design Principles

The objective is to produce a sequence of the required number of pseudo- random bits. The algorithm must comply with the standard. The method must be independent of hardware. No data is to be stored in permanent storage. The algorithm should make sure that subsequent calls in the same thread or two calls on parallel threads do not produce identical results. It should minimize the chances that two calls on different computers at the same time will produce an identical result. It should minimize the requirements to store persistent information between calls and be safe to use in a multi-threaded environment.

4.3.2 Security requirements

A minimum security requirement for a pseudorandom bit generator is that the length *k* of the random seed should be sufficiently large so that a search over $2^{\wedge k}$ elements (the total number of possible seeds) is infeasible for an adversary. Two general requirements are that the output sequences of a PRNG should be statistically indistinguishable from truly random sequences, and the output should be unpredictable to an adversary with limited computational resources.

5. Conclusion and Future Work

We have presented a comprehensive study of the threats to the coordination services of WS-BA. We have carefully analyzed the state model of the WS-BA coordination services, and have argued that it is unnecessary to perform total ordering of requests at the coordinator replicas, to achieve trustworthy coordination of WS-BA. Rather, it suffices to ensure that messages are delivered in source order, i.e., the order in which the sender sent them. This analysis has enabled us to develop a lightweight BFT algorithm that mitigates the threats and avoids the runtime overhead associated with traditional BFT algorithms. We have implemented the lightweight BFT algorithm and associated mechanisms, and have incorporated them into the open-source framework that implements the WS-BA specification and the WS-BA-I extension. The performance evaluation results obtained using the prototype implementation show moderate overhead, especially in a wide-area network, where WS-BA is typically deployed. The research that we have described in this paper fits in well with the recent trend of cloud computing. Many large companies, such as Amazon, Google, Microsoft, and Apple, are now offering cloud services. Some of them might provide coordination services for web services in the cloud. Such coordination services could enable smaller companies to offer composite web services to provide value-added services to their customers without having to invest heavily in computing infrastructures. The coordination service providers might find our lightweight BFT algorithm attractive because of its relatively low overhead. For the best fault isolation, it is desirable to deploy the coordinator replicas across geographically separated data centres, which are readily available for the large companies.

This can be enhanced by implementing the DES algorithm for efficient transaction. Data Encryption Standard (DES) is a widely-used method of data encryption using a private (secret) key. For each given message, the key is chosen at random from among this enormous number of keys. And other encryption techniques like triple DES algorithm can be also used. Some application where this techniques and methods can be implemented are Payment Gateways, Online Shopping ,Online Ticket Booking.

REFERENCES

[1] T. Grandison and M. Sloman, “A Survey of Trust in Internet Applications,” IEEE Comm. Survey Tutorials, vol. 4, no. 4, pp. 2-16, Oct.-Dec. 2000.

[2] Hua Chai, Honglei Zhang, Wenbing Zhao" Toward Trustworthy Coordination of Web Services Business Activities" VOL. 6, NO. 2, APRIL-JUNE 2013

[3] H. Erven, H. Hicker, C. Huemer, and M. Zapletal, “The Web Services-BusinessActivity-Initiator (WS-BA-I) Protocol: An Extension to the Web Services-BusinessActivity Specification,” Proc. IEEE Int’l Conf. Web Services, pp. 216-224, July 2007

[4] W. Zhao, “BFT-WS: A Byzantine Fault Tolerant Framework for Web Services,” Proc. Middleware for Web Services Workshop, Oct. 2007.

[5] W. Zhao and H. Zhang, “Byzantine Fault Tolerant Coordination for Web Services Business Activities,” Proc. IEEE Int’l Conf. Services Computing, pp. 407-414,July 2008

[6] A. Nadalin , M. Goodner, M. Gudgin, A. Barbir, and H. Grangvist, WS-Trust 1.4, OASIS Standard, Feb. 2009.

[7] S.L. Pallemulle, H.D. Thorvaldsson, and K.J. Goldman, “Byzantine Fault-Tolerant Web Services for N-Tier and Service Oriented Architectures,” Proc. IEEE 28th Int’l Conf. Distributed Computing Systems, pp. 260-268, June 2008.

[8] M. Reiter, “The Rampart Toolkit for Building High-Integrity Services,” Proc. Int’l Conf. Theory and Practice in Distributed Systems, pp. 99-110, 1995.
[9] W. Zhao, “Byzantine Fault Tolerance for Non-Deterministic Applications,” Proc. IEEE Int’l Symp. Dependable, Autonomous and Secure Computing, pp. 108-115, Sept. 2007.
[10] M. Merideth, A. Iyengar, T. Mikalsen, S. Tai, I. Rouvellou, and P. Narasimhan, “Thema: Byzantine-Fault-Tolerant Middleware for Web Services Applications,” Proc. IEEE 24th Symp. Reliable Distributed Systems, pp. 131-142, Oct. 2005.