Design of Secure Authenticated Key Management Protocol for Cloud Computing Environments

ABSTRACT

With the maturity of cloud computing technology in terms of reliability and efficiency, a large number of services have migrated to the cloud platform. To convenient access to the services and protect the privacy of communication in the public network, three-factor Mutual Authentication and Key Agreement (MAKA) protocols for multi-server architectures gain wide attention. However, most of the existing three-factor MAKA protocols don’t provide a formal security proof resulting in various attacks on the related protocols, or they have high computation and communication costs. And most of the three-factor MAKA protocols haven’t a dynamic revocation mechanism, which leads to malicious users can not be promptly revoked. To address these drawbacks, we propose a provable dynamic revocable three-factor MAKA protocol that achieves the user dynamic management using Schnorr signatures and provides a formal security proof in the random oracle. Security analysis shows that our protocol can meet various demands in the multi-server environments. Performance analysis demonstrates that the proposed scheme is well suited for computing resource constrained smart devices. The full version of the simulation implementation proves the feasibility of the protocol.

**EXISTING SYSTEM**

Earlier MAKA protocols are designed for single-server architecture. As Internet users grow exponentially, the number of cloud servers rendering different services has also grown significantly. For the single-server architecture, it is difficult for users to maintain a variety of passwords for each server. To improve user experience, many scholars propose more flexible MAKA protocols for multi-server environments. Combined with the unified management features of the cloud platform, such protocols can be conveniently applied. users and cloud servers only need to register in the registration center (RC) to mutual authentication and key agreement.

**Disadvantages**

* In the multi-server environments, the MAKA protocols can be further divided into two categories, two-factor MAKA protocols, namely identity, password, and three-factor MAKA protocols, namely identity, password, biometrics. The works in [11], [12] have shown that the password- based MAKA protocols suffer from several attacks such as guessing password attack.

**PROPOSED SYSTEM**

We propose a dynamic revocable three-factor mutual authentication and key agreement (3DRMAKA) protocol which has more comprehensive functions, reliable security and relatively higher execution efficiency. Our contribution can be summarized as follows:

* We design a three-factor MAKA protocol which implements three-factor security. And we show that the proposed protocol can meet the demands of multi-server architectures such as anonymity, nontraceability, resistance password guessing attack and smart card extraction attack, and so on.
* Our scheme achieves the user’s dynamic management. In our protocol, users can be dynamically revoked to promptly prevent attacks from malicious users. Without a dynamic revocation mechanism, RC can’t punish malicious users in a timely manner. This may result in such malicious users still active in the network to communicate with other servers.
* In the random oracle, we provide a formal proof of the proposed protocol based on BDH, CDH and Schnorr signatures unforgeability assumptions. We show that the proposed protocol is mutual authentication secure and authenticated key agreement secure. 4) Our protocol has a good execution efficiency. Especially on the client side, the computation cost of our scheme is the lowest in the related existing protocols. This shows that our protocol is more suitable for device mobiles with limited computing resource. And, to prove that the protocol is technically sound, we programmatically simulate the proposed protocol.

**Advantages**

Proposed a biometrics based MAKA protocol for multi-server environments. Unfortunately, after our analysis in the security comparisons and cryptanalysis subsection of this paper, their protocol is vulnerable the server impersonation attack and the man-in the-middle attack. On the other hand, the MAKA protocol is also widely used in other environments, such as Passive Internet of Things.

**Implementation**

**DATA OWNER:**

In this module, initially the data owner has to get register to the cloud server (CS1,CS2,CS3,CS4) . Data owner will login to the corresponding cloud server he got registered. Data owner encrypt will upload file to the cloud server (CS1, CS2, CS3, CS4) Data owner verifies the file he uploaded either it is safe or not. Data owner can view, how many file has been uploaded to the corresponding cloud servers(CS1,CS2,CS3,CS4) Data owner will send file to trust manager to store the data owner file t5o the corresponding cloud servers (CS1,CS2,CS3,CS4)

**CLOUD SERVER**

The cloud server manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with cloud consumer. To access the shared data files, data consumers download encrypted data files of their interest from the cloud and then decrypt them.

**TRUST MANAGER**

Trust manager provides login authorization for both data owner and the end user.

Trust manager can view all the cloud status .Trust manager can view the feed backs given by end user and lists all positive and negative feed backs. Trust manager lists no of users in cloud services(IAAS,PAAS,SAAS).Trust manager can view the attackers in cloud servers(CS1,CS2,CS3,CS4) and the no of time attacked.

**CLOUD CONSUMER**

Cloud consumer first has to register to the cloud server (CS1, CS2, CS3, CS4) which particular cloud he has to use. Cloud consumer has to login to the cloud he got registered. Cloud consumer feedback about the data (positive or negative feedback)

**ATTACKER**

Attacker will view registered users and cloud files

**1 Collusion Attacks - to** mislead feedbacks about the cloud

**2 Sybil Attacks -** When user uses more transaction per day (Exceeds the limit which is assigned by the Trust Manager)

**SYSTEM REQUIREMENTS**

➢ **H/W System Configuration:-**

➢ Processor - Pentium –IV

➢ RAM - 4 GB (min)

➢ Hard Disk - 20 GB

➢ Key Board - Standard Windows Keyboard

➢ Mouse - Two or Three Button Mouse

➢ Monitor - SVGA

**Software Requirements:**

* Operating System - Windows XP
* Coding Language - Java/J2EE(JSP,Servlet)
* Front End - J2EE
* Back End - MySQL