编号_____

南京航空航天大学

毕业设计附件材料

题　目　　校园易物系统的分析与设计

学生姓名　　尼玛扎西

学　　号　　091300422

材 料 目 录

序号	附件名称	数量	备注
1	翻译译文	1	
2	翻译原文	1	

二〇〇七年六月

译文:

基于票据的单点登录协议的设计

作者 LiFan

摘要

随着企业信息化的发展，企业信息应用系统的种类和数量越来越多。建立企业统一身份管理系统在应用系统之间提供单点登录是必然趋势。在本文中，提出了一种基于票据的单点登录协议和协议参考实现的设计。新协议改进了经典基于票据的单点登录协议（如 Kerberos）的局限性。对具有大量遗留帐户的应用系统实施单点登录更为简单和安全。

1. 简介

目前，随着企业信息化技术的发展，企业信息化应用系统的类型，数量越来越多，每个应用通常都有独立的认证机制，用户访问安全敏感的功能模块，应用系统必须进行用户认证,只有当用户认证状态合法，用户才能登入应用。一方面，这种认证机制是确保有效的应用安全手段，另一方面，每个应用认证功能都可能存在较大差异，如登录密码长度，复杂性和更新周期。同时，用户在不同应用系统中的认证信息一般不一样，用户需要访问不同应用在不同内存中的认证身份，建立统一的身份管理系统，实现单点登录成为信息技术发展的趋势。

当前主要单一登录实现机制，包括自动填充和基于纸张的访问。基于用户在不同应用系统中自动填写登录名和密码在文件或数据库中的单点登录形式创建映射，当用户首次访问应用系统时，根据用户在应用中的登录名和密码的映射，由程序管理模块自动生成给应用系统并提交登录表单登录信息。优点是自动填充机制原则上简单，小型应用系统集成的转换，低侵入性，但缺点是自动填写通常与逆向代理技术相关，系统需要的应用 必须逆向代理网关代理访问，容易导致性能瓶颈和单点故障。基于单点登录访问使用单个用户名和密码在整个系统中记录，应用记录对用户进行身份验证，作为系统用户登录凭据，

Kerberos 单点登录程序在通过访问记录方面更成熟，但是，用户登录的 Kerberos 应用程序使用不同的应用程序需要一个单一的用户身份，但如果用户在遗留系统中有多个应用程序存在不同的帐户，并且帐户留下了与业务数据相关的大量历史记录，会更棘手。

2. 协议设计

2.1 协议工作场景

目前，企业信息系统主要是基于 B / S 结构，信息系统一般包括企业信息门户（以下简称“门户”）和应用系统（以下简称“应用”），门户网站提供 综合应用代理工作，用户门户可以专注于不同的应用程序。

描述标签中使用的协议如表 1 所示。

表 1 显示了表协议标签

Mark	Explain
U	User
IAMS	Identity Management Server
P	Portal
A	Application
TKT_u	Notes the user's identity
$PRCP_x$	The user's identity credentials of X
ID_u, ID_x	the unique identifier of Users and X
N_u, P_u	User's global unified user name and password
N_x, P_x	Users in the application of X, the login user name and password
$E_{kkk}\{xxx\}$	Encrypted using the kkk for xxx
K_u, K_{u-1}	The user's public and private keys
K_{iams}, K_{iams-1}	Identity management server, public and private keys
K_p, K_{p-1}	Gateway to public and private keys
a, b, R_x	Random number
TSSO	the single sign-on protocol based on notes

2.2 初始化阶段

使用拟议的单一签署协议建立单一的登录系统，只需添加一个单一的身份管理服务器（Identity Access Management Server，IAMS）。 协议初始化过程

如下：

用户初始化

每个用户需要初始化身份验证用户身份管理服务器，完成初始化，身份管理服务器，给每个用户注释 TKTu 的身份，注意包含用户身份 IDu，作为有效开始时间 Ts，作为有效结束时间 Te ，身份管理服务器签名 DSiams 等信息。

TKTu={IDu,Ts,Te,DSiams}

初始化应用程序身份凭据

用户初始化时，每个用户可以管理服务器身份验证，以保护其在不同应用程序中的身份凭证 PRCPx（如登录名和密码 Nx Px）的自助功能，身份管理服务器将保存用户凭据在身份库中加密， 和用户身份 IDu，应用程序身份 IDx 键组合。

PRCPx={Nx,Px}（X 代表门户或应用）

2.3 协议工作流程

假设用户登录门户，门户代理工作列表在应用程序中发现的代理事项，事宜需要访问应用处理代理。 图 1 显示了通过 TSSO 实现的协议单工作过程可分为两个阶段：

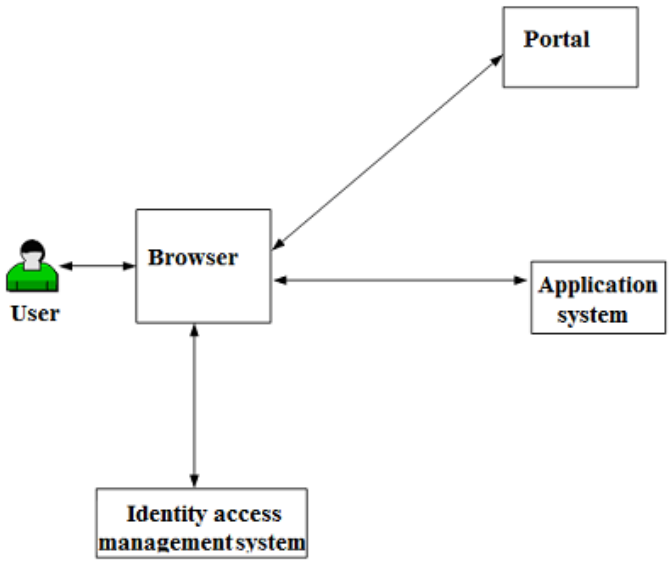

图 1 TSSO 单点登录协议工作流程

记录验证

用户身份管理服务器的公钥加密用户标识 IDu Kiams 并发送到随机身份管

理服务器。

Kiams{ IDu , a }　　(TSSO-1)

身份管理服务器使用私钥解密 Kiams-1 以获取用户身份 IDu 和随机数 a，选择一个随机数 b，用户的公钥 Ku 使用加密的随机数 b 和 Riams 返回给用户。

EKu{ b , Riams }　　(TSSO-2)

用户解密私钥 EKu-1 身份管理服务器返回随机数 b 和 Riams，使用会话密钥计算 a 和 b。

K=gab mod p ( )，K

加密使用用户名 Nu，密码 Pu，Riams 并发送到随机数 Ru 身份管理服务器。

EK{ Nu , Pu , Riams , Ru }　　(TSSO-3)

身份管理服务器使用会话密钥 K 解密用户名 Nu，密码 Pu，Riams 和随机数 Ru，如果 Riams 与 TSSO-2 在同一版本的 Riams 中，而在库中作为用户名，密码认证，如果验证来确认用户身份，用户身份接受验证或拒绝请求。IDu 作为 TKTu 用户身份，Ts 作为有效开始时间，Te 作为有效结束时间，认证等信息管理服务器签名 DSiams 组合。加密使用会话密钥 K 作为仪器 TKTu 和 Ru 返回给用户。

EK{ TKTu , Ru }　　(TSSO-4)

用户解密会话密钥 K 作为仪器 TKTu 和 Ru，Ru 和 TSSO-3 如果 Ru 的问题相同，那么识别仪器的合法性，或拒绝执行后续操作。

（1） 单点登录访问门户（应用程序）

当用户首次访问门户时，该门户使用门户网站用户的公钥加密应用程序 Ku 标志 IDp，随机数 Rpu 发送给用户。

EKu{ IDp , Rpu }　　(TSSO-5)

用户解密私钥 EKu-1 网关到网关应用程序发送一个随机数和识别 IDp Rpu。用户的公钥 Kp 使用门户用户身份 IDu 加密，Rpu，Rup 并使用随机数字获得认证过程作为会话密钥 K 加密笔记 TKTu，然后加密的消息到门户。

EKp{ IDu , Rpu , Rup , EK{TKTu}}　　(TSSO-6)

门户使用解密密钥 Kp-1 发送用户获取用户身份 IDu，Rpu，随机数 Rup，作为会话密钥 K 加密注释 EK {TKTu}，如果 Rpu 和 TSSO-6 发出 Rpu 相同，则确认用户的合法性，或拒绝遵循说明。

门户身份管理服务器使用公钥加密 Kiams 门户应用程序识别 IDp，用户身份 IDu，随机数 Rpi 和 Rup，EK {TKTu}，然后加密消息发送到身份管理服务器。

EKiams{ IDp , IDu , Rpi , Rup, EK{TKTu} } (TSSO-7)

身份管理服务器使用私钥进行解密以获取门户应用识别 IDp,用户身份 IDu，随机数 Rpi 和 Rup，EK {TKTu}，用于解密会话密钥 K EK {TKTu}，验证账单的合法性。 根据应用门户标识 IDp，用户识别 IDu 查询获取用户的门户访问凭据 PRCPp。 加密公钥 Kp 使用门户身份管理服务器标识 IDiams，用户的门户访问凭据 PRCPp，Rpi 并用会话密钥 K Rup 加密获取 EK {Rup}，然后加密消息后门。

EKiams{ IDiams , PRCPp , Rpi , EK{Rup} } (TSSO-8)

门户使用私钥解密 Kp-1 身份管理服务器发送 IDENS，用户的门户访问凭据 PRCPp，Rpi 和 EK {Rup}，如果 Rpi 和 TSSO-8 发出 Rpi 相同，则由账单验证。 门户网站作为用户，具有 PRCPp 访问门户的身份，使用门户和用户登录公钥加密门户应用程序标识 IDp Ku 和 EK {Rup}，消息将被加密返回用户。

EKu{ IDp , EK{Rup}} (TSSO-9)

获取使用私钥解密用户门户返回的 IDp 和 EK {Rup},用于解密会话密钥 K EK {Rup}，在登录过程中确认身份和验证网关管理服务器。

当用户直接从代理机构的门户访问应用程序或转向访问应用程序列表时，用户认证和单一登录过程以相同的进程访问门户，没有描述。

3. 协议安全分析

在本文中，双向认证协议 TSSO [6]，用户认证和身份管理服务器，用户和应用程序与门户，门户和身份管理和应用服务器完成通信对方身份的相互认证，比传统的 -way 认证提高了安全性。

协议的初始化阶段，用户身份管理服务器提供自助服务，在不同的应用凭证（登录名和密码）中管理他们的身份，同时身份凭证存储在库的加密身份中，既避免了用户的应用该管理员已知登录名和密码等，方便用户操作，提高安全性。

获取身份证书阶段的协议，使用基于公钥的相互认证和基于 Diffie-Hell -man 的密钥协商机制，确保攻击者获得的是会话密钥 K。

在 TSSO-1 协议过程中工作到 TSSO-3，即使成功攻击攻击者使用中介获取身

份管理服务器返回消息也无法解密消息，因为消息使用真实用户的公钥加密 Ku ，从而避免了中间人的攻击。 在协议处理步骤 TSSO-2 到 TSSO-8 中，使用随机数的认证机制，有效防止数据包重放攻击。

4. 实现

在本文中，TSSO 协议参考实现系统框架如图 2 所示，系统包括四个组件：身份管理服务器，数据库服务器作为客户端浏览器插件，服务器代理插件。

身份管理服务器是基于 Java 技术的核心功能组件,主要负责提供身份管理，会话管理，自助服务管理，用户身份管理，认证管理，应用管理，应用管理凭证，加密管理，连接管理，日志管理，缓存管理等功能。

客户端浏览器插件是面向用户的功能组件，基于 VC ++技术（目前专注于微软的 IE 6.0 或更高版本的浏览器），负责提供用户认证管理，会话管理，数据库管理安全账单，加密 管理，注销管理。

服务器代理插件面向应用的功能组件端，基于 JavaEE 技术（目前主要用于 JavaEE 应用系统），负责提供应用配置管理，认证管理，会话管理，TSSO 请求/响应处理，加密管理，事件处理取消，日志管理等功能。

数据库服务器是一个集中式身份存储用户信息，身份证件，应用程序凭据功能组件，基于 RDBMS 的关系数据库和 LDAP 目录技术，LDAP 目录，主存储用户信息，应用程序信息等数据，RDBMS 关系数据库存储为 主要仪器应用凭证等数据。

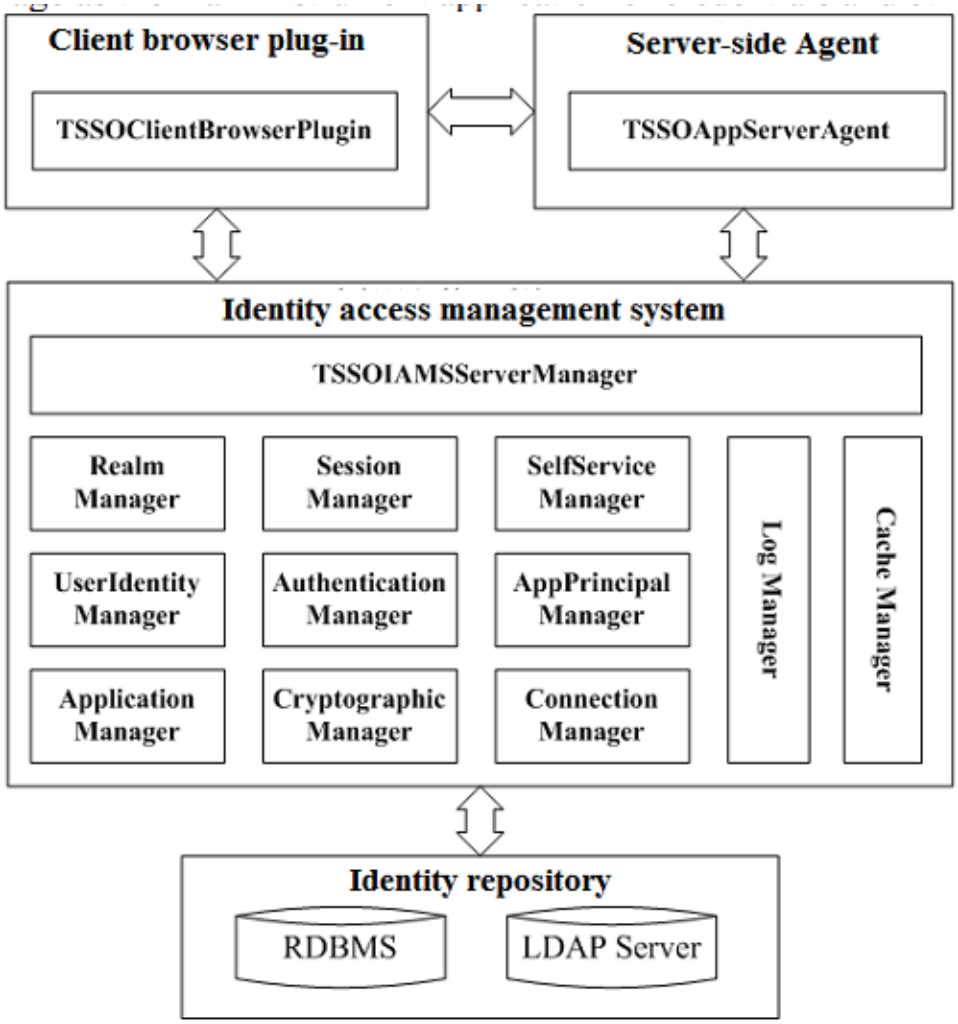

图 2 TSSO 参考实施系统框架协议

5. 结论

在本文中，TSSO 单点登录协议是基于单一登录访问纸质机制，结合公钥相互认证，Diffie-Hellman 密钥协商机制，完善传统的基于纸张的单点登录协议（如 Kerberos）要求使用用户身份通过系统测试全局统一的限制，基于 TSSO 协议参考实现系统，简单安全地实现了大量应用程序遗留的单点登录的历史账户的整合。

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Engineering 23 (2011) 537 – 542

Procedia Engineering

www.elsevier.com/locate/procedia

Design of A Ticket-Based Single Sign-On Protocol

LiFan[1,2]

1. School of Information Engineering, Wuhan University of Technology, Wuhan, 430070, China;
2. School of Computer, Chengdu University of Information Technology, Chengdu 610225, China

Abstract

With the development of enterprise information technology, the category and quantity of the enterprise information application system becomes more and more. It's a inevitable trend to establish a enterprise unified identity management system to provide single sign-on among the application systems. In this paper, a ticket-based single sign-on protocol and the design of a protocol reference implementation are proposed. The new protocol improves the limitation of the classical ticket-based single sign-on protocol such as Kerberos. It is easier and safer to implement single sign-on for application systems with a lot of legacy accounts.

Key words: Single Sign-On; Identity Authentication; Access Management

1. Introduction

At present, with the development of enterprise information technology, enterprise information application system type, quantity, more and more each application usually has a separate authentication mechanism, user access security-sensitive function module, the application system must user authentication, only those with legal status, users can log applications. On the one hand, this authentication mechanism is to ensure the effective application security means on the other hand, each application authentication function to achieve greater differences may exist, such as the login password length, complexity, and update cycles. Meanwhile, users in different application systems is generally not the same authentication information, users need to access different applications in different memory of the authentication identity, the establishment of a unified identity management system to achieve single sign-on become the development trend of information technology [1].

The main current single sign-on implementation mechanisms, including automatic filling and paper-based access. Based on automatically fill in a form of single sign-on by users in different application systems in the login name and password in the directory or database to create a mapping, when a user first access the application system, according to the mapping for the user in the application of the login name and passwords, automatically generated by the program management module to the application system to submit the login form login information. The advantage is the automatic filling mechanism is simple in principle, the transformation of the small application system integration, low invasive, but the drawback is automatically fill in a form usually associated with the reverse proxy [2] technologies, requires the

doi:10.1016/j.proeng.2011.11.2544

application of the system must reverse proxy gateway proxy access, easily lead to performance bottlenecks and single points of failure. Based on the single sign-on access to notes in the whole system uses a single user name and password, the user is authenticated to the application notes as the system user login credentials, access to notes on the more mature single sign program has Kerberos [3,5], However, users log in Kerberos applications used in different applications require a single user identity [4], but if the user has multiple applications in the legacy systems there are different accounts, and accounts have left a lot of history associated with business data, are more intractable.

2 . Protocol Design

2.1 Protocol work scene

At present, the enterprise information system is mainly based on B / S structure, information systems generally include enterprise information portal (later referred to as "portal") and an application system (later referred to as "application"), the portal provides an integrated application of the agency work, users portals can focus on different applications-dos.

Protocol used in the description tag shown in Table 1.

Table 1 shows the table-protocol label

Mark	Explain
U	User
IAMS	Identity Management Server
P	Portal
A	Application
TKT_u	Notes the user's identity
$PRCP_x$	The user's identity credentials of X
ID_u, ID_x	the unique identifier of Users and X
N_u, P_u	User's global unified user name and password
N_x, P_x	Users in the application of X, the login user name and password
$E_{kkk}\{xxx\}$	Encrypted using the kkk for xxx
K_u, K_{u-1}	The user's public and private keys
K_{iams}, K_{iams-1}	Identity management server, public and private keys
K_p, K_{p-1}	Gateway to public and private keys
a, b, R_x	Random number
TSSO	the single sign-on protocol based on notes

2.2 Initialization phase

Using the proposed single sign agreement to establish a single sign-on system, just add a single identity management server (Identity Access Management Server, IAMS). Protocol initialization process is as follows:

user initialization

Each user needs to initialize the authentication user identity management server, complete the initialization, the identity management server, an identity given to each user notes TKTu, note contains the user identity IDu, as the effective start time Ts, as an effective end time Te, the identity management server signing DSiams and other information.

TKTu={IDu , Ts , Te , DSiams}

initialize the application identity credentials

User initialization, each user can manage the server authentication to protect their self-service capabilities in different applications in identity credentials PRCPx (such as login name and password Nx Px), identity management server will save the user credentials are encrypted in the identity repository, and user identity IDu, application identity IDx combination of keys.

PRCPx={Nx , Px}（x represent Portal or application）

2.3 Protocol work process

Suppose the user login the portal, the portal agent working list found in the application of an agency matters, matters need to access the application processing agent. Figure 1 shows the agreement achieved by TSSO single sign of the work process can be divided into two phases:

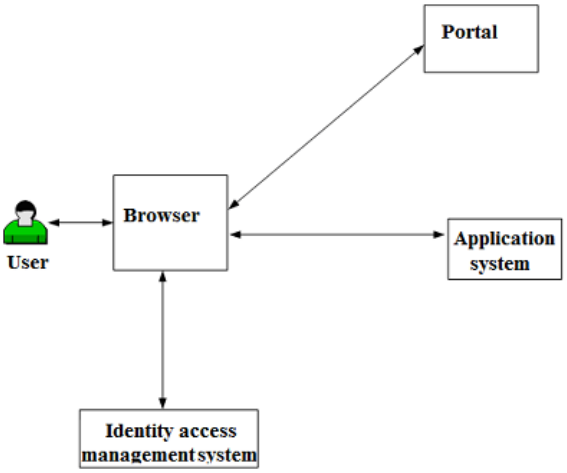

Figure 1 TSSO Single sign-on protocol work process

authentication for Notes

User identity management server's public key encrypted user identification IDu Kiams and sent to a random identity management server.

Kiams{ IDu , a } (TSSO-1)

Identity management server using the private key to decrypt Kiams-1 to get the user identity IDu and random numbers a, choose a random number b, the user's public key Ku using encrypted random number b and Riams returned to the user.

EKu{ b , Riams } (TSSO-2)

Users to decrypt the private key EKu-1 identity management server returns a random number b and Riams, a and b are calculated using the session key

K=gab mod p ( ), K encrypted using the user name Nu, password Pu, Riams and sent to the random number Ru identity management server.

EK{ Nu , Pu , Riams , Ru } (TSSO-3)

Identity management server using the session key K to decrypt the username Nu, password Pu, Riams and the random number Ru, if Riams with TSSO-2 in the same issue of Riams, while in the library as the user name, password authentication, if validated, to confirm user identity, user authentication or deny the request. Then, the identity management server for the user to generate a temporary identity paper TKTu, TKTu by the user identity IDu, as the effective start time Ts, as the effective end time Te, authentication and other information management server signature DSiams composition. Encrypted using the session key K as instruments TKTu and Ru returned to the user.

EK{ TKTu , Ru } (TSSO-4)

Users to decrypt the session key K as instruments TKTu and Ru, Ru and TSSO-3 if the issue of Ru in the same, then the legitimacy of identification instruments, or refused to perform follow-up operation.

(1) single sign-on access to the portal (Application)

When a user first accesses the portal, the portal using the portal user's public key encryption application Ku logo IDp, random number Rpu sent to the user.

EKu{ IDp , Rpu } (TSSO-5)

Users to decrypt the private key EKu-1 gateway to the gateway application to send a random number and identification IDp Rpu. User's public key Kp encrypted using the portal user identity IDu, Rpu, Rup and use random numbers to get certification process as a session key K encrypted notes TKTu, then the encrypted message to the portal.

EKp{ IDu , Rpu , Rup , EK{TKTu}} (TSSO-6)

Portal using the decryption key Kp-1 sends the user to get user identity IDu, Rpu, random number Rup, as the session key K encrypted notes EK {TKTu}, if Rpu and TSSO-6 issued Rpu the same, to confirm the user's legitimacy, or refuse to follow instructions.

Portal identity management server using the public key encryption Kiams portal application identification IDp, the user identity IDu, random number Rpi and Rup, EK {TKTu}, then the encrypted message to the identity management server.

EKiams{ IDp , IDu , Rpi , Rup, EK{TKTu} } (TSSO-7)

Identity management server using the private key to decrypt to get the portal application identification IDp, the user identity IDu, random number Rpi and Rup, EK {TKTu}, used to decrypt the session key K EK {TKTu}, verify the legality of bills. According to the application portal logo IDp, user identification IDu query to get the user's portal access credentials PRCPp. Encryption public key Kp using the portal identity management server identifies the IDiams, the user's portal access credentials PRCPp, Rpi and encrypted with the session key K Rup get EK {Rup}, then the encrypted message back door.

EKiams{ IDiams , PRCPp , Rpi , EK{Rup} } (TSSO-8)

Portal using the private key to decrypt Kp-1 identity management server to be sent IDiams, the user's portal access credentials PRCPp, Rpi and EK {Rup}, if Rpi and TSSO-8 issued Rpi same, verified by the bill. Portal as a user with PRCPp access to the portal in the identity, log in using the portal, and the user public key encryption portal application identification IDp Ku and EK {Rup}, the message will be encrypted to return the user.

EKu{ IDp , EK{Rup}} (TSSO-9)

Obtained using the private key to decrypt the user portal returned IDp and EK {Rup}, used to decrypt the session key K EK {Rup}, confirm the identity and authentication gateway management server in the login process.

When users access the application directly from the portal of the agency work or turn to access the application list, the user authentication and single sign-on process to access the portal with the same process, not described.

3. Protocol Security Analysis

In this paper, two-way authentication protocol TSSO [6], user authentication and identity management server, users and applications with the portal, portal and identity management and application server to complete the mutual authentication of the identity of communicating parties than the traditional one-way authentication improved security.

Initialization phase of the agreement, user identity management server to provide self-service to manage their identities in different applications credentials (login name and password), while the identity credential stored in the encrypted identity of the library, both to avoid the user application of the administrator login name and password, etc. being known, and easy user operation, to improve security.

Obtain the agreement of the identity paper stage, the use of public key-based mutual authentication, and Diffie-Hellman based key agreement mechanism to ensure the attacker to get the session key K.

Work in the agreement process step TSSO-1 to TSSO-3, even a successful attack the attacker to use an intermediary to obtain the identity management server returns the message can not decrypt the message because the message using the real user's public key to encrypt Ku , thus avoiding the middleman attacks. Work in the agreement process step TSSO-2 to TSSO-8, using a random number of authentication mechanisms, effectively preventing packet replay attacks.

4. Implementation

In this paper TSSO protocol reference implementation system framework shown in Figure 2, the system includes four components: identity management server, database server as the client browser plug-ins, server proxy plug-ins.

Identity management server is the core functional components, based on Java technology, the main domain responsible for providing identity management, session management, self-service management, user identity management, authentication management, application management, application management credentials, encryption management, connection management , log management, cache management and other functions.

Client browser plug-in is user-oriented side of the functional components, based on VC + + technology (currently focused on Microsoft's IE 6.0 or later browser), is responsible for providing user authentication management, session management, database management security bill, encryption management, write-off management.

Server proxy plug-in application-oriented side of the functional components, based on JavaEE technology (currently mainly for JavaEE application system), is responsible for providing application configuration management, authentication management, session management, TSSO request / response processing, encryption management, cancellation of event handling, logging management and other functions.

Database server is a centralized identity store user information, identity paper, the application credentials functional components, RDBMS-based relational database and LDAP directory technology,

LDAP directory, the main store user information, application information and other data, RDBMS relational database storage as the main instrument application of credentials and other data.

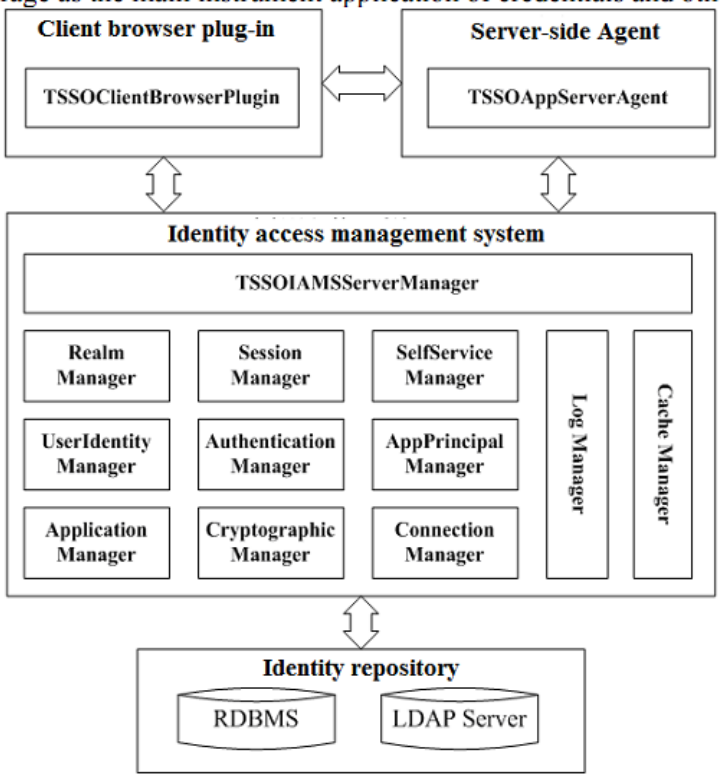

Figure 2 TSSO reference implementation system framework agreement

5 Conclusions

In this paper TSSO single sign-on protocol is based on the single sign-on access to paper-based mechanism, combined with public-key mutual authentication, Diffie-Hellman key agreement mechanisms, improve the traditional paper-based single sign-on protocol (such as Kerberos) require the use of the limitations of global unified user identity through the system tested, based on TSSO protocol reference implementation system to simply and safely achieve a large number of historical accounts of the application left the single sign-on integration.

References

[1] Gregg Kreizman. MarketScope for Enterprise Single Sign-On. Gartner RAS Core Research Note G00170568, September 2009

[2] Greg Barish, Katia Obraczka.World Wide Web caching: trends and techniques. Communications Magazine, IEEE, May 2000

[3] J. Kohl, C. Neuman. The Kerberos Network Authentication Service (V5). RFC1510, September 1993

[4] BELLOVIN S M,MERRITTM. Limitations of the Kerberos authenti-cation systems. ACM SIGCOMM Computer Communication Re-view, 1990, 20 (5):119-132 .

[5] Li Jiyong, Tao Ran. A single sign-on protocol design [J]. Computer Engineering, 2008, (14)

[6]Yang Zhi, Chen Xing Yuan, Zhang Bin, support dual authentication, single sign-on solution [J]. Computer Applications, 2007, (03).