Realizing A Secure Federation of Multi-Institutional Service Systems

Yu Enokibori and Nobuhiko Nishio

Department of Computer Science, Ritsumeikan University
1-1-1 Noji-Higashi, Kusatsu, Shiga 525-8577, Japan
Tel: +81 77 561 2741
vori@ubi.is.ritsumei.ac.jp nishio@cs.ritsumei.ac.jp

Abstract. This paper proposes an extended framework for service provision for ubiquitous spaces based on Kerberos. The framework allows groups of services and information about ordinary users that are managed on an organizational or personal level to be combined, handling service systems with different management bases as units of *Space*, while defining the security relations between different spaces. A combination of spaces merges two spaces on equal terms allowing the spaces to use each others services, while a fusion of spaces represents a power-relationship where one space are allowed to use resources in another space.

1 Introduction

Today, many organizations and individuals are creating a large variety of services. However, even if such services were connected through the network, there are still few examples where such services are interconnected in a mutual way. We regard one of the reasons for this to be because the governing bodies/individuals managing users, are different between organizations, and it is necessary to offer mutual security between the different parties. As an example of this, in the case that a user only is authorized in one service system, if such a user is to take advantage of services in another service system, how should the capabilities of the user be handled, and if the user is visiting another organization, how should the user be recognized when the user is to connect to the visited system with their own terminal?

The amount of processing, the complexity, and the difficulty in negotiation, is obstructing the interconnection between separate management bodies, and is a large hurdle for the realization of the ubiquitous society.

For distributed file systems, Andrew File System[1](AFS) is an example of a solution to this problem. AFS allows for mutual secure interconnections between distributed file systems managed by different organizations, and is successful in creating a global route with this facility covering the all of the globe. With this paper, we are solving the same problem for the ubiquitous service world, proposing a framework that allows the services and terminals of each individual to be connected freely and mutually in secure and plugable way in the course of our United Spaces Project.

In chapter 2 we discuss technology that is already available, in chapter 3 presents the algorithms and frameworks that we used in order to realize the secure system, discussing the unit of *Space*, and the use of this definition. In chapter 4 the user management, in chapter 5 the service management, while chapter 6 we discuss the management of each space as a whole, and the inter-space operations of *combination* and *fusion*. In chapter 7 we look at applications, and in chapter 8 we report progress before the conclusion in chapter 9.

2 Related Work

There are a number of systems that realize ubiquitous spaces, the GAIA project[2] at University of Illinois at Urbana-Champaign, the Smart Space Laboratory[3] (SSLab) at the Tokuda-lab at Keio University and the STONE room[4] at the Aoyama/Morikawa Laboratory at Tokyo University.

Research efforts on collaboration between ubiquitous spaces have previously been conducted between SSLab and STONE. However, this collaboration did not implement the necessary security features. We assume that such ubiquitous systems will be installed at separately managed institutions and collaboration needs to be set up temporarily in a practical way. In this paper, our aim is to provide an easy way to secure collaboration for such multi-institutional ubiquitous service systems.

3 System Overview

We regard the set of service systems that is managed by a singular organization or user, as a *Space* unit, and we enable mutual secure interconnectivity between these spaces by achieving safety over the communication paths, and by defining the user authentication between each space. Space is a logical unit, and is not limited to physical spaces such as real rooms or buildings. Service systems located in different rooms can be grouped, and organized as one space. One the other hand, mobile terminals that are carried around by users and their laptop computers can also be regarded as one space, and the secure connection between the mobile terminal and the service system is realized through the connection between the small space and the larger space in this situation.

As a space indicates the range of a service system managed by a singular subject, user information management and service information management become necessary.

Furthermore, the spaces should be adaptable to diverted information from existing sources. However, there are few cases where information can be adapted as-is, and we use Adapter Interfaces (*User Directory Adapter (UDA) and Service Directory Adapter (SDA)*) in order to bridge the gap between existing user/service information directories and the space management. Also, as it is conceivable that one organization maintains several directories, multiple Adapter Interfaces might be necessary, and in order to group several Adapters, we use an

Integrated Directory for User and Service respectively (*Integrated User Directory (IUD), Integrated Service Directory (ISD)*). Finally, a framework managing the User and Service, handling the two Integrated Directories, as well as offering control of the services to the user, is offered in order to manage the relationship between the different spaces (*Space Server (SS)*). An example of the system with a singular management subject is showed in Figure 1.

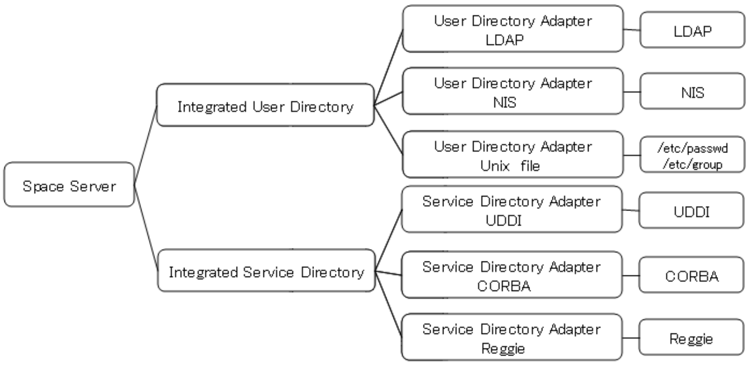
Space Server
Integrated User Directory
Integrated Service Directory
User Directory Adapter LDAP
User Directory Adapter NIS
User Directory Adapter Unix file
Service Directory Adapter UDDI
Service Directory Adapter CORBA
Service Directory Adapter Reggie
LDAP
NIS
/etc/passwd
/etc/group
UDDI
CORBA
Reggie

Fig. 1. An Example of Single Space Management System

3.1 Usage of This System

For actually using the resources inside a space there are two different procedures. One procedure is that "the client operates resources inside the space" and the other is that "a service manipulates (uses) another service". For the latter in particular, there are two patterns of usage; using a service that is within the authority of a client that is already using another basic service, and a service that uses another service within its own authority. The basic flow when the client uses a service inside the space, and when a service uses a service is outlined below.

- The client uses resources inside a space.
  1. An authentication request is transferred to the Space Server, where it is processed.
  2. When the authentication is finished, the client received the session key used by Kerberos[5, 6]. At this time the authority that can be used for this session is fixed.
  3. A service list request is transmitted to the Space Server and the Space Server inquiries the ISD before it returns the answer.
  4. The client specifies which services to use, sends the service connection requests to the Space Server, establishing a Kerberos session, and establishes the service session.
- The service uses another service.
  1. In the case the service is used by a client (the service is not operating autonomously), authority transfer processing is requested to the client.

2. If transfer processing is carried out, the transferred authority is used, and if not, the service uses its own authority, and transmits a connection request for the target service to the Space Server.
3. Upon completing the Kerberos session setup, a session towards the service is established.

The spaces that are to be combined creates a space user for the partner space in its user information management module. As the space management server logs in to the target space using this space user, it is confirmed that the space is allowed for space combination in advance, and encryption of the communication links is established.

Furthermore, a space combination is accomplished when an inter-space combination is established with each space on an equal term, while on the other hand a space fusion occurs when one space is absorbed into another space; from the outside of the absorbed space it looks as if one space has disappeared.

After the security of the inter-space communication pathways have been established, the flow is different between the space combination, and the space fusion, as described in later sections of this paper.

4 User Directory Management

User directory management is done by an IUD (Integrated User Directory) which integrates multiple UDA 's (User Directory Adapter) that interpret conventional user management systems to accommodate them into our space management.

Role

Service access control in a space is done by using *Roles*. When a user is successful y authenticated by a UDA, UDA determines which role should be assigned to the user.

Key

The authentication information required by UDA includes conventional username/passphrase pairs, biometrics features, RFID and so forth depending on UDA-implemented facilities. Therefore, when user tries to get authenticated, he/she has to submit specific information on authentication method, e.g. specific UDA implementation. *Key* is used for this specification including the following two items: a unique name for UDA which this key should be applied, and authentication information such as username/passphrase, finger print, etc.

Key-Ring

Users who have registered to multiple user information directories will require several roles simultaneously, while other role might not be used in ordinary cases. For example, the administrator role should be used as little as possible. *Key-Ring* is introduced for cases where users want to use an appropriate set of keys occasionally. This set of keys is passed to the IUD, and the IUD tries to authenticate this user via all the keys passed to all the UDA 's. Afterward IUD preserves all the collected roles for this newly generated session.

5 Service Directory Management

In order to accommodate conventional service directory systems, we has prepared SDA (Service Directory Adapter) interface. For integration of multiple service directory systems, ISD (Integrated Service Directory) is introduced for merging multiple SDA 's and providing an integrated service usage interface. The SDA interface hides protocol differences among multiple service lookup systems like UDDI, Jini 's lookup server or etc. As for services that do not which don 't preserve session information like ones using HTTP, SDA offers them a session management so that they can utilize the encryption feature of Kerberos. Each SDA maintains the service-wise ACL information between service and required role for its access.

6 Space Management

A space consists of a pair of IUD and ISD and the space is maintained by the Space Server.

6.1 Space Server

Space Server functions as a KDC [1] of Kerberos, and it also manages user authentication, session initiation between components, and space combination/fusion control.

Kerberos specification does not care about malicious hosts that pretends a KDC. It does not offer secure processing before establishing login. Therefore, Space Server and users utilize PKI and digital signature issued by the third party to justify the server and obtain secure login processing. Both the digital signature and the PKI are also applied for login among spaces.

6.2 Space Combination

To establish a combination between spaces, one space should log into another space. At this moment, any user/resource which belongs to the former space can utilize resources which belong to the latter space in using the authority of the role assigned. When for example Space A has logged into Space B and has received a role as Bob, all users logging into Space A are capable of utilizing resources in Space B using the role as Bob. Since this is done without any authentication for users in Space A, they do not even have to be registered in Space B. Administrators of Space B only have to take care of the role of space A which represents all the users logging into Space A. Figure 2 illustrates role handling for space combination.

Meanwhile, the case shown above is relatively rare because it needs agreement among administrators beforehand. Suppose a certain space which resides in a

[1] Key Distribution Center It issues a secret key for authentication and encrypted session.

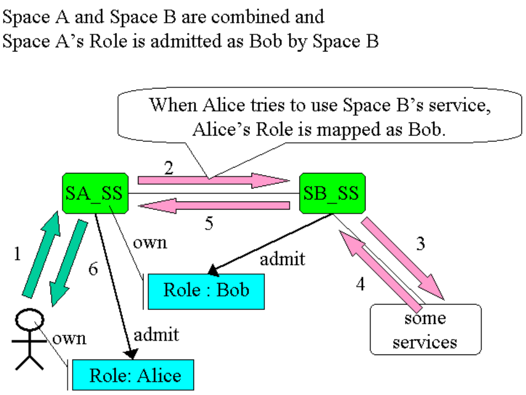
Space A and Space B are combined and
Space A's Role is admitted as Bob by Space B
When Alice tries to use Space B's service,
Alice's Role is mapped as Bob.
2
SA_SS
SB_SS
5
1
6
own
admit
3
4
Role : Bob
some services
own
admit
Role: Alice

Fig. 2. Using Service Sequence in case of Space Combination

Laptop has mobility. When this space visit somewhere, it will try to cooperate with spaces that have no beforehand knowledge of the laptop. In such cases, a guest role will be assigned, although this depends on the IUD 's administrative policy. Some spaces might prohibit such guest role assignments.

6.3 Space Fusion

Whereas a combination makes a agreement on equal terms possible, there is a power relation in a fusion. The space that wishes to go into a fusion, logs in to the other space in order to prove that it has the necessary permissions in order to enter a fusion. Upon a successful login, the ISD and IUD of the space to be fused, is absorbed in the SDA and UDA of the space it is fused into. This framework is realized in the form of a service-to-service relationship, where the absorbing space IUD uses the IUD of the absorbed space. The flow is shown in Figure 3.

The biggest difference between combination and fusion is that whereas the authorities upon establishing a combination is mapped from the authorities of the space user, while for a fusion the authorities for each individual user can be used across spaces. Due to this, when a service is established across spaces, leveling of authorities does not occur. Also, a feature of the system is that for users that are not familiar with the absorbed space, the space looks reinforced, and they are not even aware that the absorbed space existed (exists) as a separate entity. Because of this, there is no particular need to be conscious about the absorbed space in itself, and the resources of the absorbed space can be used. At this time clients or other spaces in combination, even if they are using services separated by several spaces, only need to carry out identification between client and space, or between several spaces at one time.

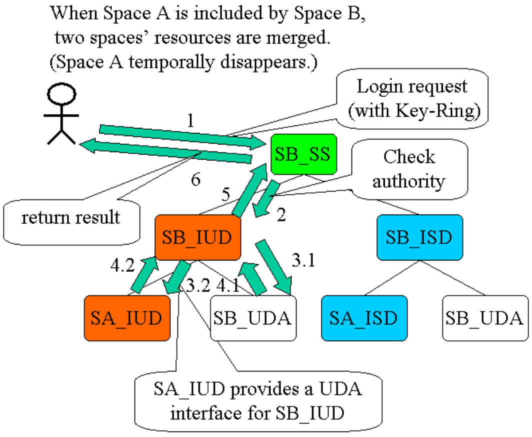

Fig. 3. User Login Sequence in case of Space Fusion

7 Application Examples

In the case that ones own mobile terminal constructs its own space, this space is fused with the service system at a visited location, and files inside the mobile host can use presentation services in the visited space. Alternatively, the screen shown by the presentation service can be transferred to ones own screen by a presentation viewers service. By using screen viewer, sound input, and sound system services for service systems at separate locations, secure video conferencing systems can also be constructed.

Also, a broker service could be started based on our space management technology in order to securly integrate multi-institutional ubiquitous service providers. This kind of broker management would work as Verisign which provides PKI and SSL facilities in the sense that it is the third party for clients and only offers authoriy of security which is completely seperated from any service provision.

8 Current Status and Future Work

Currently we are implement authentication and encrypted communication based on Kerberos on top of JINI. Although Sun's JINI implementation has Kerberos-based authentication, because it utilizes the Java Authentication and Authorization Service[9](JAAS) framework for authority transfer processing and session key handling, we independently inject it into the socket layer. Therefore, upper layer protocols such as JERI, HTTP and SOAP running upon Kerberos are kept operational.

To prevent Role spoofing, the role is embedded into the SessionKey which establishes the secure communication link. This could be done by one of the Kerberos features. As for directory integration, we support JINI for services and LDAP for user administration.

After building authentication, encrypted communication link and Role assignment for sessions within a single space, we are proceeding to handle integration of multiple directories within a single space and cordination of multiple spaces.

9 Conclusion

In this paper, we have proposed a framework to allow separate management entities to be combined in a secure fashion. Allowing plural ubiquitous environments to be combined, vastly increases the possibilities for existing systems, and we are currently involved in several experimental and research activities towards this goal. Among these, a large number of ubiquitous spaces with differing management subjects are being realized, and we have great expectations towards the improvement of security and practicality for such interconnected systems.

10 Acknowledgments

We would like to thank the Advanced Solution R&D Center at Uchida Yoko, for providing the necessary experimental facilities and personnel in order to realize and verify the system in this paper. We would also like to thank the members at the Ubiquitous Computing and Networking Laboratory at Ritsumeikan University.

References

[1] Satyanarayanan, M.: "Scalable, Secure, and Highly Available Distributed File Access" IEEE Computer May 1990, Vol. 23, No. 5.
[2] GAIA Project Official Page:http://choices.cs.uiuc.edu/gaia/
[3] Smart Space Laboratory Project
Official Page: http://www.ht.sfc.keio.ac.jp/SSLab/
[4] Tokyo University Aoyama Morikawa Laboratory
Official Page:http://www.mlab.t.u-tokyo.ac.jp/
[5] MIT Kerberos
Official Page:http://web.mit.edu/kerberos/www/
[6] KTH Heimdal
Official Page:http://www.pdc.kth.se/heimdal/
[7] Charlie Kaufman, Radia Perlman, Mike Speciner: Network Security, Prentice Hall, 2002/04/15.
[8] Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, and Randy H. Katz: "An Architecture for a Secure Service Discovery Service" Mobicom ‘ 99 Seattle Washington USA. 1999.
[9] Java Authentication and Authorization Service
Official Page:http://java.sun.com/products/jaas/index.jsp