유비쿼터스 도시를 위한 통합 인증 서비스

한 혁* 김 신 규** 염 헌 영***

목 차

1. Introduction
2. Background and Related Work
3. Architecture
4. Encoding Mechanism
5. Conclusion

1. Introduction

It is very important to control accesses of distributed resources. An authorization service needs to manage users' permissions in an integrated manner because users access resources with different groups or different permissions, and resources should embed a mechanism that they check users' permission locally again.

The objective of U-City project is to build and deploy six grid-based and ubiquitous services. Different sites manage different resources and they provide resources in the form of grid-based services. An authorization service in the U-City project should manage users' permissions of each service in a fine-grained manner because each user accesses grid services [1, 2, 3] with his identity. An authorization mechanism should enforce minimal modification of service codes. It should also be flexible to add new resources. The points listed above require an integrated and flexible authorization service.

We present design considerations of an authorization service (UAuthService) for the U-City project. Our key design issue is to minimize modification of service codes and existing libraries. Thus, we propose the skeleton-level authorization mechanism and the non-critical extension of grid certificates. The skeleton-level authorization mechanism allows codes of grid services and UAuthService to be well modularized. The mechanism exploits the message context in the skeleton of the Remote Method Invocation (RMI) over Web Service Resource Framework (WSRF). The message context includes much information such as user identity, user's certificate, and the desired method. Our local permission-check modules grant access to the resource based on the information.

The non-critical extension of grid certificates provides the capability-based authorization without modifications of Grid Security Infrastructure (GSI). User's capabilities are encoded in the non-critical section of user's certificate and the user with his certificate accesses grid services. The encoding mechanism is based on the well-known cryptographic algorithms, which requires relatively small computation cost.

In this paper we describe the UAuthService, which has the capabilities listed above. The rest of this paper is organized as follows. Section 2 describes background material. Section 3 explains the architecture of the

* 서울대학교 전기컴퓨터공학부 박사과정
** 서울대학교 전기컴퓨터공학부 석사과정
*** 서울대학교 컴퓨터공학부 교수

UAuthSerivce. Section 4 proposes the encoding/decoding mechanism. Section 5 concludes the paper.

2. Background and Related Work

2.1 Grid and Web Service

A Grid can be defined as a layer of networked services that allow users single sign-on access to a distributed collection of computing, data, and application resources. The Grid services allow the entire collection to be seen as a seamless information processing system that the user can access from any location. However, the heterogeneous nature of the underlying resources remains a significant barrier. Many applications often require extensive collections of libraries that are installed in different ways on different platforms.

Web service is an important emerging distributed computing paradigm that focuses on simple, Internet-based standards to address heterogeneous distributed computing. Web services define a technique for describing software components to be accessed, methods for accessing these components, and discovery methods that enable the identification of relevant service providers.

2.2 Related Work

Akenti[4] was developed at the Lawrence Berkeley National Laboratory to facilitate setting access policies by independent organizations, and to provide a virtual, organization-wide user identity. Akenti implements the pull model of authorization. There is no need for the user to request a specific credential from an external authority. The authority is contacted independently by the service once the user has successfully authenticated.

The Community Authorization Service (CAS) is another authorization system, developed by the Globus Project. CAS[5] uses the hybrid model, which allows a resource site to grant a community access to resources and the authorization server for that community to grant access to the community members. This is implemented by having the user contact the CAS server to get a delegated proxy certificate, which includes a rights restriction extension that limits what resources can be accessed. The resource gatekeeper interprets the restricted rights extension and verifies that the community has such rights to the resource.

The Virtual Organization Membership Service (VOMS)[6] is an ad hoc solution to authentication in a GSI-enabled Grid. It is similar to the CAS model, but the VOMS server is run by a virtual organization and supplies authorization information about its own members. VOMS can operate in either a push or a pull mode. This service is used in the European Data Grid.

The Resource Oriented Authorization Manager (ROAM)[7] was created to maintain a coherent authorization scheme over the distributed resources of FusionGrid. ROAM is based on a pull model of authorization. The heart of ROAM is its information model, which is designed to represent FusionGrid authorization in a coherent way. It consists of a framework of resources, permissions, users, and authorizations. The architecture of ROAM consists of two tiers. The bottom tier is a relational database containing all the authorizations for the various FusionGrid resources. The top tier is a Web interface which accepts HTTPS connections from users and resource providers.

3. Architecture

This section of the paper describes the architecture of the UAuthService. Figure 1 shows the architecture of the UAuthService. The sequence of the capability-based access is as follows:

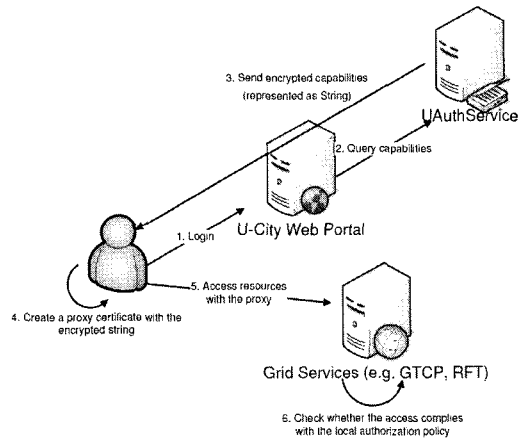

Fig. 1. UAuthService Architecture

- A user login via the U-City web portal
- The web portal queries user's capabilities to the UAuthService
- The UAuthService returns encoded capabilities (represented as String)
- The user creates a proxy certificate with the encrypted string
- The user accesses resources with the proxy certificate
- The local permission-check modules check whether the access complies with the local authorization policy

The local permission-check modules decode the encoded capabilities and grant access to the resource based on the decoded capabilities. The local check procedure needs to minimize modification of grid services or to modularize grid services well. The procedure is located at the skeleton of the RMI over WSRF for the minimization and modularization. Each RMI call has a message context and the message context includes much information such as user identity, user's certificate, and the desired method. The procedure exploits the information to check whether the access complies with the local authorization policy. Notice that encoding/decoding mechanism is explained in the Section 4.

The local check procedure is located at the skeleton of the RMI over WSRF. In the case of Globus Toolkit[8], the procedure can be implemented at the Policy Decision Point (PDP) and Policy Information Point (PIP)[9]. If PDPs and PIPs are configured internally, our procedure can evaluate incoming requests.

Services which U-City provides assumes that permissions of each user be based on scenarios. Scenarios are different from methods or operations that grid services export and scenarios consist of several methods. For example, a user with the "controlling the gateway" permission means that the user can access any methods or operations of related grid services. Thus, our local check procedure needs the mappings between scenario-based permissions and operation-based permissions. We store mapping information to permanent databases and the information is managed by grid-service administrators.

In the emergent case, the encoded capabilities in the proxy certificate are not synchronized with the UAuthService. For example, after a user with the "controlling the gateway" permission creates a proxy certificate, administrators of the UAuthService turn off the permission due to the gateway problem. In this case, the local check procedure will grant requests from the user. Thus, synchronization mechanism between the UAuthService and the local check procedure is necessary and we adopt publish/subscribe message services[10, 11, 12] for broadcasts of any changes of the UAuthService as shown in the Figure 2.

4. Encoding Mechanism

When the U-City web portal queries user's permissions, the UAuthService returns user's capabilities. If the UAuthService sends user's capabilities in the readable form, any malicious users can forge their own capabilities. In order to prevent forged capabilities,

capabilities should be encrypted. The local check procedure should decrypt the encrypted capabilities to evaluate incoming requests. Any kind of encryption and decryption methods can be used, but we adopt block cipher algorithms such as the Advanced Encryption Standard (AES)[13] due to the relatively small computation cost. Notice that capabilities are encrypted and decrypted in hosts which execute UAuthService and the local check procedure respectively. Keys in the algorithms should be updated periodically because capabilities can be forged if keys are known. We use secure publish/subscribe message services[14] to update keys as shown in the Figure 2 and administrators of the UAuthService decide on the period of the key update.

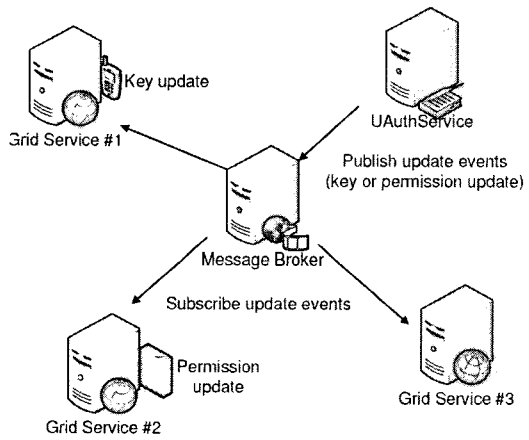

Fig. 2. Update Event Propagation

When a user creates a proxy certificate with the encrypted capabilities, they are embedded into the section of Common Names (CN) which is non-critical. In other words, another CN section is created to embed the encrypted capabilities. Adding the capabilities to a non-critical section does not require modification or upgrade of the core GSI library, and proxy certificates in U-City services can be used even in "UAuthService-unaware" grid environments. This point allows our mechanism to be flexible and scalable.

5. Conclusion

This article has proposed the UAuthService. In order to modularize implementation, we present the skeleton-level authorization mechanism and the non-critical extension of grid certificates. The skeleton-level authorization mechanism modularizes grid-service and authorization codes, and the non-critical extension minimizes the modification of the existing GSI library. Our encoding/decoding scheme for capabilities is efficient and scalable. We are convinced that our system will be a core security infrastructure in U-City grid services.

References

[1] Tuecke, S., Czajkowksi, K., Foster, I., Frey, J., Graham, S., Kesselman, C., Maguire, T., Sandholm, T., Snelling, D., Vanderbilt, P.: Open grid services infrastructure (ogsi) version 1.0. Technical Report GFD-R.015, Global Grid Forum (2003)

[2] Foster, I., Kesselman, C., Nick, J., Tuecke, S.: The Physiology of the Grid:An Open Grid Services Architecture for Distributed Systems Integration. In: Open Grid Service Infrastructure WG,Global Grid Forum. (2002)

[3] Foster, I., Kesselman, C., Tuecke, S.: The anatomy of the grid: Enabling scalable virtual organizations. International Journal of SuperComputer Applications 15(3) (2001) Detailed description of the grid architecture.

[4] Thompson, M.R., Essiari, A., Mudumbai, S.: Certicate-based authorization policy in a pki environment. ACM Trans. Inf. Syst. Secur. 6(4) (2003) 566-588

[5] Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: A community authorization service for group collaboration. In: POLICY '02: Proceedings of the 3rd International Workshop on Policies for Distributed Systems and Networks (POLICY'02),

Washington, DC, USA, IEEE Computer Society (2002) 50

[6] Aleri, R., Cecchini, R., Ciaschini, V., dell'Agnello, L., Frohner, A., LLorentey, K., Spataro, F.: From gridmap-le to voms: managing authorization in a grid environment. Future Gener. Comput. Syst. 21(4) (2005) 549-558

[7] Burruss, J.R., Fredian, T.W., Thompson, M.R.: Roam: An authorization manager for grids. Journal of Grid Computing 4(4) (2006) 413-423

[8] Globus Alliance: Globus Toolkit 4.0 Release Manuals. (2005) http://www.globus.org/toolkit/docs/4.0/.

[9] Freeman, T., Ananthakrishnan, R.: Authorization processing for Globus Toolkit Java Web services. (2005) http://www-128.ibm.com/developerworks/grid/library/gr-gt4auth.

[10] Fox, G., Pallickara, S., Rao, X.: A Scaleable Event Infrastructure for Peer to Peer Grids. In: Proceedings of the ACM Java Grande ISCOPE Conference. (2002)

[11] Uyar, A., Pallickara, S., Fox, G.: Towards an Architecture for Audio Video Conferencing in Distributed Brokering Systems. In: Proceedings of the 2003 International Conference on Communications in Computing. (2003)

[12] Creare Inc.: RBNB Data Turbine (2005) http://www.creare.com/rbnb.

[13] Stinson, D.R. In: CRYPTOGRAPHY Theory and Practice. Chapman & Hall/CRC (2006)

[14] Pallickara, S., Pierce, M., Gadgil, H., Fox, G., Yan, Y., Huang, Y.: A Framework for Secure End-to-End Delivery of Messages in Publish/Subscribe Systems. In: Proceedings of the 7th IEEE/ACM International Conference on Grid Computing (GRID 2006). (2006)

◐ 저 자 소 개 ◑

한 혁
2003년 서울대학교 컴퓨터공학부 졸업 (공학사)
2006년 서울대학교 전기컴퓨터공학부 졸업 (공학석사)
2006년~ 서울대학교 전기컴퓨터공학부 박사과정
관심분야: 분산시스템, 데이터베이스

김 신 규
2006년 서울대학교 컴퓨터공학부 졸업 (공학사)
2006년~ 서울대학교 전기컴퓨터공학부 석사과정
관심분야: 분산시스템, 네트워크

염 헌 영
1984년 서울대학교 계산통계학과 졸업 (이학사)
1986년 Texas A&M University 전산과학 졸업 (공학석사)
1992년 Texas A&M University 전산과학 졸업 (공학박사)
1992년~1993년 삼성데이타시스템 연구원
1993년~1998년 서울대학교 컴퓨터공학부 조교수
1998년~2004년 서울대학교 컴퓨터공학부 부교수
2004년~ 서울대학교 컴퓨터공학부 교수
관심분야: 분산시스템, 멀티미디어 시스템, Transaction processing