Virtual community based secure service discovery and access for 3D video steaming applications

Virtual Community Based Secure Service Discovery and Access for 3D Video Steaming Applications

Shudong Chen, Igor Radovanovic, Johan Lukkien, Richard Verhoeven, Melissa Tjiong, and Remi Bosman

Department of Mathematics and Computer Science
Eindhoven University of Technology, the Netherlands
{shudong.chen, i.radovanovic, j.j.lukkien, p.h.f.m.verhoeven, m.tjiong, r.p.bosman}@tue.nl

Abstract. The Freeband I-Share project aims to define the mechanisms for trust, willingness, resource discovery and sharing mechanisms in virtual communities. To improve the secure and performance of a 3D video streaming application, which is a research vehicle of the I-Share project, we propose a virtual community based access control approach for secure service discovery and access (VICSDA) which groups services in virtual communities and only grants authenticated community members to discover and access these community services. There are two main contributions associated with this approach. First, different from most of the other access control approaches it adopts a dual access control mechanism which allows community services to define their local access control policy besides following the community membership policy. Second, behavior of these community services is monitored in order to guarantee a better QoS provision. Using this approach, the 3D video streaming application can be guaranteed with authentication and message confidentiality through the dual secure service discovery and access mechanism. Better application performance can also be achieved through the community member behavior audit.

Keywords: 3D video streaming, virtual community, access control.

1 Introduction

The Freeband I-Share project is a joint research project of various Dutch public universities and companies. I-Share aims to define the mechanisms for trust, willingness, resource discovery and sharing mechanisms in virtual communities, which are dynamic groups of devices and services that are willing to collaborate for the better of the whole [1]. Emphasis is on protocols for resource/service discovery and for efficient data/file/video distribution over heterogeneous networks and devices. In particular, mechanisms for distributed and layered content processing of the multi-view 3D video utilizing resources shared in virtual communities are considered as a research vehicle. The aim for this paper is to extend an existing application [2] with techniques and methods that enable establishment of confidence in secure operation of a distributed system built out of cooperating services provided by multiple parties.

N. Sebe, Y. Liu, and Y. Zhuang (Eds.): MCAM 2007, LNCS 4577, pp. 391–397, 2007.

Paper [2] presents a layered framework for 3D-TV transmission, combining multi-view and depth-based approaches in a scalable fashion. Besides texture and depth information, specific layers are added for coded occlusion data and edge-mask information to allow high-quality 3D rendering of key objects in the scene. By relying on the concept of resource sharing for the creation and streaming of virtual viewpoints in a network overlay, the range of view-points selectable by the user (FTV) is extended. An experimental 3D video transmission application running on this layered framework was implemented as a standalone program, running on a powerful platform. Both the sender and the receiver were implemented on a desktop PC running Linux OS. A problem with this is a lack of flexibility and security. It's hard to adapt the system when there are some configuration changes, for example, another type of receiver arrives. And also because there is no access control involved, the system can easily be attacked, for example, by a malicious user pretends to be a legal receiver to steal the content of the 3D video.

The rapidly increasing technology in Service Oriented Architecture (SOA) [3-6] enables the flexibility of software. In SOA, software components are wrapped into network-exposed services with explicitly described interfaces to provide functionalities; applications can be built by interconnecting services, leaving the binding until runtime. Relying on standards SOA provides a highly flexible and adaptable implementation for services and applications; eventually, it becomes possible to switch from a particular service to a different one without adaptations. Thus, to achieve high flexibility of the 3D video streaming application, it is necessary to borrow the concept of SOA and wrap the existing components into services. In this way it is much easier to develop and adapt the components independently.

However some issues still remain in current SOA systems. There are three activities in a SOA system: publish, discover and invoke. A service provider defines a service description and publishes it to a service registry which acts as an intermediary between providers and user where services are stored. A service user uses the service registry's search capability to discover service descriptions and their respective providers. Later the user will invoke service interfaces for required functionalities. Services published in a service registry [7] are supposed to be discoverable by all service users. This may lead to a privacy problem when a service provider only wants to reveal its services to trusted service users. In addition, service providers and service users should have some guarantee that the agreements in the contract, for example, promised functionalities and QoS in the service description, are supported. That is difficult when a SOA lacks monitoring and enforcement authority.

In order to achieve a flexible and secure 3D video streaming application, we propose a virtual community based access control approach for service discovery and access namely VICSDA. In this approach, we extend SOA with the concept of virtual communities to help protect the privacy of service providers and include incentive for high QoS provision [1]. Services are grouped into virtual communities. Only authenticated community members can be allowed to discover and access these services. There is a large volume of publications on access control and some of them are on secure service discovery and access [8, 9]. Compared with other approaches, our proposed approach contains two main contributions: (i) a dual access control mechanism which is that community services are autonomous to define their local access control policy; (ii) an audit functionality of the virtual community maintenance

mechanism which is designed to guarantee better QoS provision to applications through monitoring services' behavior.

The remainder of this paper is organized as follows. In section 2, we elaborate the virtual community based secure service discovery and access approach. Then we present the 3D video streaming application as an example to validate the feasibility of the proposed approach. Conclusion is drawn in Section 4.

2 Virtual Community Based Secure Service Discovery and Access

A virtual community is a dynamic contract-based aggregation whose members have commonalities and interact via shared services by means of a digital network for example the Internet. A virtual community has rules that each member has to follow. It provides services to members and has the potential to develop different applications through service cooperation and external orchestration.

The advantage of sharing functionalities and resources in type of services with other community members is that limitations of individual devices can be overcome by collaboration with others. The sharing not only benefits the individual participating devices, but also the system as a whole since functionalities, resources and communication of software can be optimally distributed. We are interested in investigating the SOA approach for composing distributed applications from cooperating services. In this process, service users do not necessarily take part in this service cooperation; it is referred to an external orchestration as an orchestrator.

Services are grouped in virtual communities through adding additional functionalities to them, for example can only be exposed to authenticated community members. Then a virtual community is an overlay network for the existing services. It provides at least the following functionalities to its members: (i) register a service provider or a service user as a member and deregistering a member; (ii) publish/discover/invoke required community services; (iii) protect the privacy of service providers and provide secure service access; (iv) maintain the member and service list. Rich functionalities of a virtual community can be making trustable recommendations for a service user during the service selection stage, and more.

Based on this virtual community overlay, we designed the dual access control approach. First, we use the member boundary of a virtual community to control the service discovery scope and filter the un-trusted or malicious access requests to services. Second, in our opinion services are autonomous entities and it is natural to allow services to define their local access control policy and to deny access requests coming from their un-trusted users. However, services' autonomy may cause the entire application fails because required service providers may stop providing functionalities. To prevent this, we grant each community service a credit value to represent its behavior. A service user can evaluate a service's QoS. This evaluation affects a service's credit value. This value may increase because of a positive evaluation and may decrease when a negative one arrives. When it is below a threshold it will be deregistered from the community. Using this maintenance mechanism, service autonomy can be restricted and then guarantee a better QoS provision to applications. Using the UML notation [10], Figure 1 depicts the essential ingredients that together compose a virtual community.

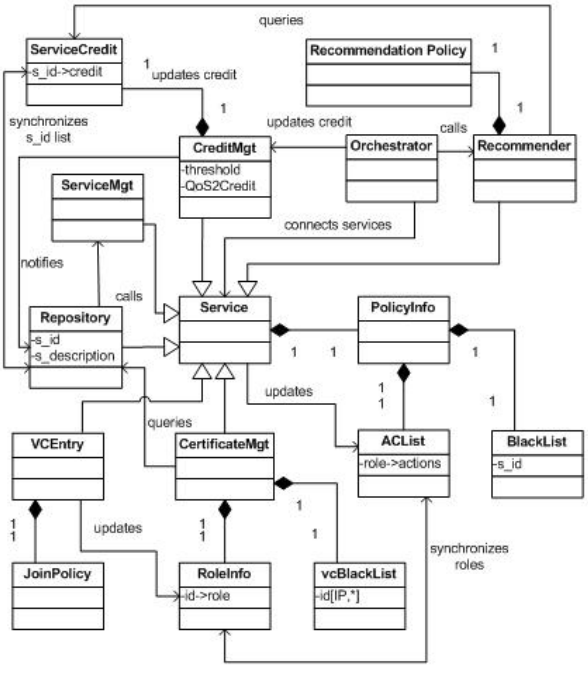

Fig. 1. Ingredients of a virtual community

VCEntry and *CertificateMgt* are accessible to any parties including users on the network and internal community members. While others, for example, the *Repository* and the *CreditMgt* are only available for community members. *VCEntry* is a service to deal with members' actions inside a community including member registration and deregistration. When one meets the community joining policy in *JoinPolicy*, it will be authorized to be a community member. Then specific roles are assigned to it by the *VCEntry*. It could be a service user to discover and access services, or a service provider to publish services, or a combination of available roles in *RoleInfo*. When a member publishes services at a virtual community *Repository,* the *ServiceMgt* will be called by the *Repository*. *ServiceMgt* can wrap a service as a community service by adding virtual community properties to it including *ACList*, *BlackList* and *Credit.* A service can specify its local access control policy by assigning capabilities to different roles in its *ACList* and by adding its un-trusted users to the *BlackList*.

During the execution of an application, an orchestrator will first discover required services and then orchestrates these services cooperate with each other. The virtual community based dual access control approach will react on each transaction. An example is when the *Orchestrator* submits the service discovery request to the *Repository*, a valid ticket which is distributed by the *CertificateMgt* will be checked by the *Repository*. This ticket is an authentication that the *Orchestrator* is a community member. The *Repository* will first verify the authenticity of the signature and the validity of the ticket. If approved, the *Repository* will execute a second round of access control based on its local policy. In this stage, the *Repository* will query its

BlackList to check whether the *Orchestrator* should be trusted. If it trusts the *Orchestrator*, the *Repository* looks up its *ACList* to grant the *Orchestrator* with corresponding authorization according to the roles indicated in the ticket. For example, the *Orchestrator* can invoke the service discovery interface but not the service list update. Only after the *Orchestrator* passes the dual access control, its service discovery request can be executed by the *Repository*.

CreditMgt is designed to facilitate the member list, community services list maintenance and the better QoS guarantee for service cooperation. It periodically compares a service's credit value with a threshold. When a credit is smaller than the threshold, the *CreditMgt* will inform the *Repository* to deregister that service.

3 Service-Based 3D Video Streaming Application

In order to validate the effect of improving the feasibility and security of the 3D-TV transmission framework through adopting this virtual community based access control approach, in this section, we describe an experimental application. This application demonstrates a 3D video generated by two cameras shown on a terminal from different viewpoints in an interactive fashion. A user either selects an arbitrary viewpoint and a viewing direction, or the user's movements are continuously tracked and the displayed content is automatically adjusted to the view points.

In particular, we integrate the proposed dual access control approach with an early version of this application and pervious software components are wrapped as community services. In the new application, the two cameras generate two video streams of an object from different viewpoints and stream them to the 3D video generation service. Then the 3D video generation service calculates the depth map files of these two videos that will be used to generate a 3D video. At the mean time, the 3D video generation service is bound to a mouse service. The mouse's event indicates a user's viewpoints switch requirements. A user can control that mouse service to express her expected viewpoints of the 3D video. These viewpoints are received by the 3D video generation service. During the video streaming the 3D video generation service continuously tracks a user's viewpoints and correspondingly adjusts the generated 3D video's content. The mouse service may work on another device for example a PDA. The working principle of the 3D video generation service and the video receiver is shown in Figure 2 [2].

All needed software components are visible as services on the network. The early 3D video generation program which is written in C++. We have modified it and wrapped it into a web service [11]. We have also developed two mouse services. Both of them are running as web services with one running on the PC and the other running on the PDA. During the 3D video streaming, particularly, the coding of the displayed content is automatically adapted to the control information coming from an external service: the mouse of the PC or the touch-screen of the PDA.

The architecture of the 3D video streaming application is depicted in Figure 3. We wrap all these services into virtual community services. So their availability is restricted to users with community credentials. We use an orchestrator to execute the 3D video streaming application. It first discovers all required services from a virtual

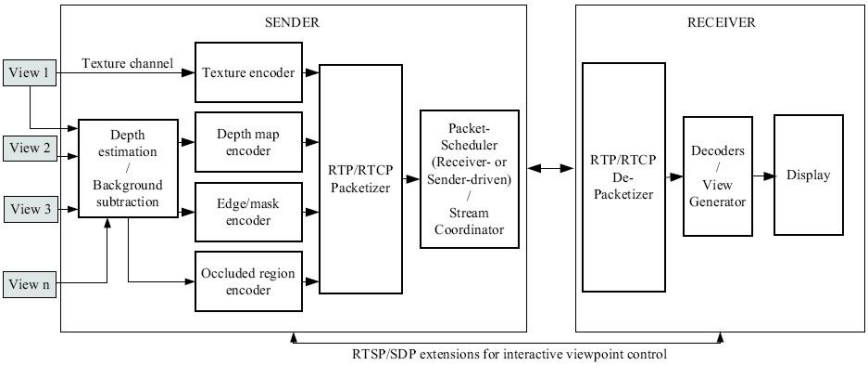

Fig. 2. Constituents of the 3D video generation service and the video receiver service

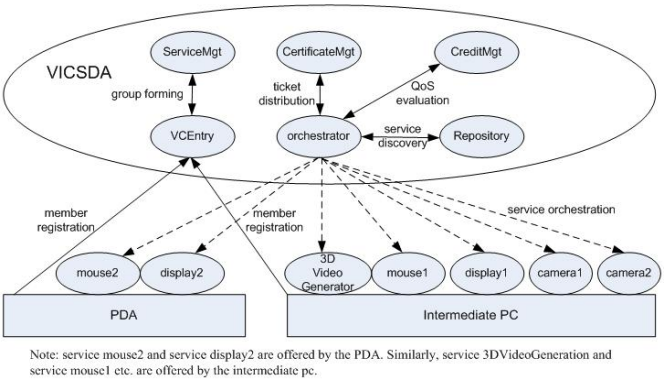

Fig. 3. Architecture of the 3D video streaming application

community repository. Then it configures and binds these services together to achieve the application. For example, it binds the 3DVideoGenerator service with one mouse service and one display service. During the course of service discovery, binding and access, the dual access control approach, which includes the ticket validation and the capability granting, is applied between every two communication parties. For example, the 3DVideoGenerator service and the mouse service have to negotiate before they are bound together.

4 Conclusions

This paper describes the virtual community based access control approach for secure service discovery and access, which is aimed to solve the flexibility and security problems of the 3D video streaming application by wrapping existing software into community services. This proposed approach contains two novel features: (i) a dual access control mechanism which allows autonomous services to define their local

access control policy besides following the community membership policy; (ii) a member behavior audit functionality which aims at better QoS provision to applications. This approach can also be applied to other service-based ubiquitous computing systems. There are immediate challenges facing us: (i) how to reduce the authentication overhead to improve the performance of the system? (ii) how to recover from the overload or fail of the *CertificateMgt* service without affecting the system's performance? Our ongoing research is aimed to solve those problems.

Acknowledgments. We thank Goran Petrovic and Peter H. N. de With for their technical feedback at the implementation stage of this work. We also thank our anonymous reviewers for their invaluable feedback which helped in improving the quality of the paper.

References

1. Freeband I-Share D1.5. Project Deliverable. http://www.win.tue.nl/san/publications/
2. Petrovic, G., de With, P.H.N.: Framework for Layered 3D Video Streaming. In: 27th Symposium on Information Theory. Noordwijk, The Netherlands (2006)
3. SOA. http://www.service-architecture.com/index.html
4. Kreger, H.: Web Services Conceptual Architecture (WSCA 1.0) http://www-306.ibm.com/software/solutions/webservices/pdf/WSCA.pdf
5. UPnP Device Architecture. Version 1.0. http://www.upnp.org/
6. Sun Microsystems. Jini technology architectural overview. White Paper http://www.sun.com/jini/whitepapers/architecture.html
7. UDDI Version 3.0. Published Specification. http://www.uddi.org/
8. Czerwinski, S., Zhao, B.Y., Hodes, T., Joseph, A., Katz, R.: An Architecture for Secure Service Discovery Service. In: 5th International Conference on Mobile Computing and Networks (MobiCom'99), Seattle, WA, pp. 24–35 (1999)
9. Pearlman, L., Welch, V., Foster, I., Kesselman, C., Tuecke, S.: A Community Authorization Service for Group Collaboration. In: 3rd International Workshop on Policies for Distributed Systems and Networks, Monterey, CA, USA (2002)
10. Fowler, M.: UML Distilled: A Brief Guide to the Standard Object Modeling Language, Version 3.0. Addison-Wesley (ISBN 0-321-19368-7)
11. Cerami, E.: Web Services Essentials: Distributed Applications with XML-RPC, SOAP, UDDI & WSDL, 5th edn. p. 304. Publisher: O'Reilly (February 2002) ISBN: 0-596-00224-6