Distributed Credential Chain Discovery in Trust-Management with Parameterized Roles

Xian Zhu, Shaobin Wang, Fan Hong, and Junguo Liao

College of Computer, Huazhong University of Science & Technology, Wuhan 430074,China

Abstract. Trust-management subjects face the problem of discovering credential chain. In this paper, the distributed credential chain discovery algorithms in trust-management with parameterized roles are proposed. The algorithms extend the RT_0's and are goal-oriented also. Based on the concept of parameterized roles in RT_1, they search the credential graph via the constant matching and variable solving mechanisms. The algorithms can perform chain discovery in most trust-management systems and can support the protection of access control policies during automated trust negotiation. Soundness and completeness of the algorithms are given.

1 Introduction

It is a hotspot how to solve the access control problem effectively in decentralized collaborative environments. In these systems, resources and the subjects requesting them belongs to different security domains controlled by different authorities. So traditional access control mechanisms cannot be used in these environments. In [1], Blaze et al introduced the trust-management (TM) systems to deal with the problem. Then, some famous TM systems, such as KeyNote [2], SPKI/SDSI [3,4] and RT [5,6,7] were proposed. Delegation is a core concept in these systems, which is the ability of an entity A to give another entity B the authority to act on A's behalf. To make the access control decisions, a credential chain from the source of authority to the requester must be discovered. We call this the credential chain discovery problem. With the tenet of TM systems—decentralized control, the credentials are typical issued and stored in a distributed manner. So the problem evolved to the distributed credential chain discovery problem.

Now there are some algorithms to solve the problem [8,9,10]. But most of them assume the credentials are stored with its issuer, which will make some bottleneck. In [6], Li Ninghui et al present some goal-oriented algorithms. The credentials can be stored with its issuer or subject. The query can be answered by doing backward, forward or bi-direction searching, which makes the searching efficiency improved greatly. However, RT_0 is a basic language in RT family. RT_1 is the most important extension to RT_0, which extends RT_0 to allow parameterized roles. In RBAC96 [11], a role name is an atomic string. But parameterized role name is constructed by applying a role identifier to a tuple of data terms (the parameter). The parameterized roles can be used as follows [5,12,13]:

Y.G. Desmedt et al. (Eds.): CANS 2005, LNCS 3810, pp. 334–348, 2005.

. With the same role identifier plus some different parameters, it can be used to aggregate some roles with few differences between them. The numbers of roles will be decreased greatly, thus simplifying the management of roles.

. It can represent relationships between entities, thus supporting more fine-grained access control. For example, we can use Alpha.managerOf(employee) to name the manager of an employee.

. It can represent attributes that have fields. For example, a diploma typically contains school, degree, year, *etc.*

. It can also represent access permissions that take parameters identifying resources and access modes.

The credential in KeyNote and SPKI/SDSI can be expressed in RT_1 with the parameterized roles [5,7,14], so the algorithms are more general when it's based on RT_1. In addition, automatic trust negotiation (ATN) based on RT_0 has the problem of supporting the protection of access control policies [15,16]. However, if we design ATN based on RT_1, the problem may be solved. For example, when Alice requests a service in a server, the server responds with the target: IBM.employee$\xleftarrow{?}$Alice. It will reveal that the server have a business relationship with IBM, which may be a secret. Based on RT_1, the target can be transformed to CoalitionA.employee(?company)$\xleftarrow{?}$Alice.

To make the algorithms in RT_0 more applicable, we must extend it to be based on RT_1 and to support parameterized roles. Although there are some distributed credential chain discovery algorithms supported parameterized roles proposed, but they are imperfectness[1]. In this paper, we give some algorithms based on RT_1. They extend the RT_0's and are goal-oriented also. The credential chain can be discovered by searching from the issuer side, subject side or both when the credentials are stored distributed. We first proposed the centralized algorithms. And the distributed algorithms are designed by incorporating the type systems of RT_0 with them. The algorithms are based on a graphical representation of RT_1 credentials. They construct the credential graph via the constant matching and variable solving mechanisms to find the path connecting the source of authority and the requester. We present the time and space complexity. And the soundness and completeness theorems with respect to the RT_1's logic program semantics are proved.

The rest of this paper is organized as follows. Section 2 introduces RT_1's syntax and semantics. In section 3, the new algorithms based on RT_1 are proposed. Soundness and completeness of the algorithms are given in section 4. Some related work is discussed in section 5. We conclude the paper in section 6.

2 RT_1's Syntax and Semantics

RT is a newly proposed trust management system with many advanced features. RT combines the strengths of RBAC and TM systems. An entity in RT is a uniquely identified individual or process. Entities are also called principals in the

[1] It will be discussed in section 5.