Chapter 5

Optimistic Access Control for Collaborative Applications

Asma Cherif
Umm Al-Qura University, Saudi Arabia

Abdessamad Imine
Lorraine University, France & Inria Nancy Grand-Est, France

ABSTRACT

Collaborative applications are important applications, allowing users to cooperate in order to perform a given task. Their importance has grown significantly over the recent years since they are required in many fields. However, they still lack of an appropriate access control mechanism which limits their full potential. It is hard to conceive an access control model for collaborative applications since they need to change dynamically access rights while maintaining high local responsiveness. This chapter presents a decentralized access control model based on replicating the shared document and its access control policy at each collaborating site. The interaction between document updates and authorizations updates is carefully studied to maintain the convergence of the shared data. Our model relies on an optimistic approach to enforce the access control, i.e. users may temporarily violate the access control policy if their rights were revoked concurrently. Illegal operations are undone selectively to eliminate their effects and converge to the same final state of the shared object.

DOI: 10.4018/978-1-5225-0448-1.ch005
INTRODUCTION

Most usually, collaborative applications aim at sharing and updating graphical or textual documents (e.g. wikis or version control systems). The most famous collaborative editing framework is Google docs. It provides both synchronous and asynchronous editing functionalities based on revision control. Also, Wikipedia represents a large scale collaborative editing application. Collaborative applications, mainly aim at providing better and more complex projects than those produced by individuals which is very useful for academic as well as commercial communities. They also improve learning experiences as well as the process of reviewing scientific documents.

There are two kinds of collaborative editors (see Figure 1): real-time collaborative editors (RCE) and non-real-time. RCE allow several users to edit simultaneously shared documents while non-RCE do not allow editing of the same document at the same time, therefore being similar to revision control systems. RCE are based on two kinds of collaborative editing algorithms: centralized and decentralized algorithms. RCE based on centralized algorithms require the presence of a central server to coordinate between updates performed on the shared document. For instance, Google Docs, SOCT4 algorithm proposed by Vidot et al. (2000), GOT proposed by Chengzheng and Clarence (1998) as well as COT proposed by David and Chengzheng (2009), rely on client-server architecture in order to get a global and unique order of execution, hence do not scale well. The second approach based on decentralized algorithms allows concurrent requests to be executed in any order, but does not allow necessarily for an arbitrary number of users. For example in adopted proposed by Matthias, Doris, and Gunzenhauser (1996), SOCT2 proposed by Suleiman, Cart and Ferrié (1998), GOTO proposed by Chengzheng and Clarence (1998), and SDT proposed by D. Li and R. Li (2004), the use of state vectors is necessary to detect causality relationship between different updates. Consequently, these solutions do not scale well and it is difficult to adapt them for a P2P context. The algorithm OPTIC proposed by Imine (2009) is a scalable solution since it allows for an arbitrary number of users thanks to the use of the semantic dependency notion used to coordinate concurrent updates.

This chapter focuses on RCE regardless of the algorithm used to coordinate different updates. To improve the availability of data in such applications, each user has a local copy of the shared documents. In general, the collaboration is performed as follows: each user updates are locally executed in a non-blocking manner and then are propagated to other sites to be executed on remote copies.

Although being distributed applications, RCE are specific in the sense that they must consider human factors (Imine, 2009). So, they are characterized by the following requirements:
An Integrated Security Governance Framework for Effective PCI DSS Implementation
www.igi-global.com/article/integrated-security-governance-framework-effective/58982?camid=4v1a

Wild-Inspired Intrusion Detection System Framework for High Speed Networks (f|p) IDS Framework
www.igi-global.com/chapter/wild-inspired-intrusion-detection-system/72749?camid=4v1a