

CITADEL: A Content Protection Architecture for Decentralized Peer-to-Peer File Sharing Systems

Paul Judge and Mostafa Ammar

College of Computing, Georgia Institute of Technology, Atlanta, GA 30332

{judge,ammar}@cc.gatech.edu

Abstract—There is an increased interest, by content creators and owners, in content protection systems that provide the ability to control or restrict the content that can be shared on peer-to-peer file sharing systems. Some content protection systems have been proposed for centralized peer-to-peer systems (such as Napster) where a central authority controls all indexing and querying. These systems cannot be applied to decentralized peer-to-peer systems since they rely on a central server. Also, such systems limit the ability of end-users to effectively share content and can make the peer-to-peer distribution model resemble a client-server model in many respects. In this paper, we propose CITADEL, a novel content protection architecture designed to operate in decentralized peer-to-peer systems (such as Gnutella). CITADEL enforces a range of protection policies while maintaining an open peer-to-peer distribution model. CITADEL builds a protected file sharing environment over a normal peer-to-peer network using secured content objects and file sharing software enhanced to perform protection operations. A flexible content importation system that is part of CITADEL allows all users to insert new content as well as additional copies of protected content.

I. INTRODUCTION

Peer-to-peer file sharing systems allow content to be shared between distributed end-systems or peers. Files are stored at the peers rather than at a central server and are transferred directly between peers rather than via the server. The last few years have seen the popularity of such systems grow tremendously. One such system, Kazaa [1] was one of the fastest growing application on the Internet boasting nearly 200 million users downloading its software. Users are drawn to peer-to-peer systems due to the ability they provide to locate and obtain a wide variety of content.

The large number of users freely exchanging content has sparked increased interest, by content creators and owners, in the protection of the content that can be shared on these systems. They, along with the legal system, are forcing peer-to-peer system operators to control the exchange of content on their systems. In this context we use the term *content protection* to refer to the ability, within a peer-to-peer file sharing environment, to control or restrict the exchange of content.

We emphasize here the central importance that effective content protection will play in the future success of many peer-to-peer systems. Napster was essentially shut down because of its lack of content protection, and will re-emerge only after implementing stringent content protection functions [2]. Other existing *decentralized* peer-to-peer systems such as Gnutella [3] and KaZaA[1] con-

tinue to operate without content protection, but some are constantly under legal pressure to implement content protection [4]. It is a matter of time before many peer-to-peer file sharing systems will have a need for content protection features. While on one hand content providers are actively trying to stop these peer-to-peer systems from allowing uncontrolled content distribution, at the same time, content providers are actively seeking content protection technology that will allow them to effectively leverage the popularity of peer-to-peer distribution.

Some people are opposed to content protection systems because they feel that such systems unfairly restrict the users ability to access content. We aim to provide a system that provides the level of protection needed by content owners while maintaining the flexibility that end-users desire. This is ultimately beneficial to end-users for a number of reasons: 1) adequate protection will allow content owners to offer higher quality and a greater variety of content; 2) individual artists will be able to take advantage of the peer-to-peer distribution channel while also protecting their content; 3) a peer-to-peer system that is supported by providers would be more reliable and secure. By providing a system that meets the needs of both parties, this will open the door for access to a greater variety of content and flexible business models that users are accustomed to from other distribution models.

Some content protection systems have been implemented or proposed for centralized peer-to-peer systems [5] but rely on the central authority that maintains the indexing and location functions to provide content protection and, therefore, cannot be applied to decentralized peer-to-peer systems. In this paper we present CITADEL, a novel content protection system designed specifically for use within a decentralized peer-to-peer system¹. CITADEL builds a protected file sharing environment over a normal peer-to-peer network using secured content objects and file sharing software enhanced to perform protection operations. Access to objects is protected and users must obtain access tokens in order to access the retrieve content in the system. This allows content providers to utilize the peer-to-peer distribution system while at the same time establishing and enforcing a policy specifying

¹The system can also be viewed as an alternative to current proposals for content protection in a centralized system. We focus, in this paper, on its use in decentralized systems.

what users have the rights to access their content. The open peer-to-peer sharing experience is maintained with the use of techniques such as a flexible content importation system that allows all users to insert new content as well as additional copies of protected content.

This paper is organized as follows. We begin by discussing the objectives of CITADEL in Section II. Section III explains the design issues in a content protection architecture for peer-to-peer file sharing systems. Section IV presents the CITADEL architecture. We present a threat analysis in Section V and performance evaluation in Section VI. In Section VII, we describe our implementation of CITADEL. We discuss other work in content protection in Section VIII before concluding in Section IX.

II. CITADEL OBJECTIVES

We outline the design objectives of a content protection system for decentralized peer-to-peer file sharing systems.

- **Content Protection:** In the past, the goal of content protection in peer-to-peer systems has been to restrict certain content from being exchanged within the system. Thus, these systems only provided all-or-none access; if content was allowed in the system, then anyone in the system could access it. In CITADEL, the fundamental content protection goal is to be able to control access to content on a per user basis. Controlling access refers to providing *protected distribution* by controlling the user's ability to retrieve the content and providing *protected storage* by controlling the user's ability to access the plaintext content within a local copy of the content. CITADEL does not aim to provide *output protection*—protection for the analog or digital output after access has been granted to an authorized user.

- **Maintain an open peer-to-peer sharing experience:** We define an open peer-to-peer environment as one in which, even in the face of content protection, all peers are equally able to insert content into the system including independent content and copies of protected content including variations such as different formats or compression rates. This requires the content protection system to be able to appropriately secure all content that is introduced into the system without regard to the peer inserting the content. Without this ability, systems have struggled to find the correct balance of openness and security [6]. Without the openness, the peer-to-peer system loses many of its attractive features and resembles a client-server model in many respects.

- **Avoid dependency on trusted client software:** Providing content protection in a decentralized peer-to-peer system requires modifications or additions to the file-sharing software. We assume the presence of malicious users that wish to circumvent the content protection system. The system should be robust against attacks by users

with full access to the software and the operating system on their computer. Additionally, the system should not rely on the file-sharing software being tamper-proof or trusted software.

- **Maintain privacy:** The content protection system should at least maintain the level of privacy that exists in the normal file-sharing environment. It should be possible to allow a user to obtain access rights and to be authorized without providing identifying information. Additionally, the system should be able to interoperate with the work in anonymous systems such as anonymous communication [7], anonymous authorization and payment [8], and anonymous peer-to-peer file sharing [9], [10].

- **Provide the flexibility to support common content distribution business models:** There are a number of common content distribution business models that peer-to-peer systems have been unable to support because they lack enhanced protection functionality [11]. While content control has been criticized in some circles as spelling the end of true peer-to-peer file sharing, we suggest that it may actually be beneficial in that it has the potential to enable many different and desirable service models.

III. BACKGROUND

A. Peer-to-Peer Systems

Peer-to-peer file sharing systems have two parts, the file location process and the file retrieval process. In most peer-to-peer systems, the file retrieval process is decentralized. That is, files are transferred directly between peers rather than through a client-server model. Peer-to-peer systems differ in the file location process. There are centralized and decentralized file location systems.

Even though these systems vary significantly in the way that files are located, the file retrieval processes of the systems are nearly identical. In CITADEL, therefore we create a content protection system that focuses on the file retrieval process. Because of its independence from the file location process, CITADEL can be used in any peer-to-peer file sharing system including centralized, decentralized, or some hybrid systems

B. Content Protection Design Space

Due to the nature of decentralized peer-to-peer systems, there is no central authority in the file sharing process, so policy enforcement must be done at the peers. This, of course, implies that peers must know the access control policy. The question is how do the peers securely and efficiently access the global content rights list in order to enforce it. The content rights list is a form of an access control list(ACL), so we also refer to it as the ACL. One approach to peers accessing the ACL, the *distributed ACL approach*, is to distribute the content rights list to all peers. However, there is significant overhead due to distributing the entire content rights list to all peers. An al-

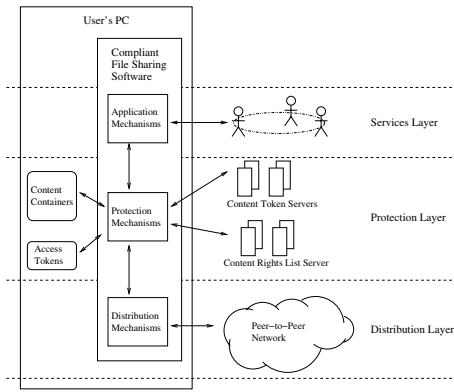


Fig. 1. The CITADEL protected file sharing environment.

ternate approach, the *queried ACL approach*, is to have the peers access the list by querying the content rights list server as necessary for each access to a content object. However, there is significant overhead due to repeatedly querying the content rights list server.

To avoid such overhead, our system takes a different approach in which the access control policy for each content object is stored with the content object. Thus, every access control policy that a peer has to enforce is available and is accessed locally. Compared to the queried ACL approach, this approach behaves like a cache of the relevant access control policy information.

IV. CITADEL ARCHITECTURE

With CITADEL, we introduce the concept of a protected file-sharing environment that creates a protected environment over a normal peer-to-peer network. Only protected objects can be exchanged within this peer-to-peer system. Thus, all content objects in the system are protected and all access to these objects is controlled. Each content object specifies the rights that are necessary to access it. Users must obtain the appropriate access tokens in order to retrieve content in the peer-to-peer system. Access tokens can specify the right to all content in the system, some subset of content, or a single file. Thus, obtaining access tokens is an infrequent event and does not interfere with the normal peer-to-peer interaction. In CITADEL, the file sharing software transparently handles tasks such as presenting and verifying tokens upon file request as necessary.

Figure 1 shows the protected file-sharing environment. The system uses the peer-to-peer network strictly as a means of file location and distribution. CITADEL exists as the protection layer built upon this distribution layer. The service layer is, in turn, built on top of the protection layer. Thus, service providers and application developers can introduce new services and applications based on a peer-to-peer distribution model by building the services on top of the CITADEL protection layer.

The remainder of this section introduces the components of the CITADEL architecture include the the file sharing software, secured content objects, access tokens, and the content importation system.

A. File-sharing software

File sharing software that is enhanced with content protection software modules is referred to as *compliant file sharing software*(CFSS). Within figure 1, there is a high-level diagram of the compliant file sharing software. The CFSS provides three types of operations: *distribution operations*, *protection operations*, and *application operations*. Distribution operations involve normal duties of peer-to-peer file sharing software such as interacting with the file location system and the file retrieval system. Protection operations involve interacting with the secured content objects to control access to the content. Protection operations also include periodic interactions with the content rights list server and the content token server as necessary. Application operations involve supporting the application and the service model interacting with the protection layer to provide access to the content files in the secured content objects. In Section V, we summarize our risk analysis that shows that even without any software protection or tamper resistance, the design of the system makes the CFSS robust against any such attacks.

B. Content Containers

Access to content is protected by the use of secured content objects or *content containers*, cryptographically secured objects consisting of a protection label and an encrypted content file. The protection label contains a content label that provides content identification and ownership information and the content's ACL. The ACL only requires a small number of entries due to the use of role-based access control (RBAC) [12]. The protection label is digitally signed by the appropriate authority to ensure integrity. The content file is encrypted with a random key, K_R . K_R is encrypted with K_{-RTS_ID} , a unique decryption key associated with the token needed to access the content. This encrypted form of K_R is stored in the content container. Additionally, the encrypted K_R and the protection label are encrypted with K_{CFSS} , the decryption key that is built-in the compliant file sharing software. Thus, only the CFSS can access the content container and even the CFSS can not access the content file in the content container without K_{-RTS_ID} which is provided to authorized users along with the content access token.

C. Access Tokens

Users receive rights to some set of content by obtaining access tokens from a content token server. There are numerous proposals for electronic payment systems that can be leveraged for user authorization for token distribution; see for example [13], [14]. Here, we explain

the basic functionality necessary for CITADEL. For example, a user would contact a content seller and obtain a subscription to access all content on the network or perhaps all content in a particular category. For now, the point is that this process should be thought of as occurring infrequently and outside of the normal file-sharing experience. It is similar to how a cable subscription is set up for certain channels, but a user watches television regularly without dealing with obtaining new rights. The system can involve different content token services for different content providers. For each content provider, the content token service can be a single server or a group of distributed servers.

Upon obtaining rights, the host receives an access token and the accompanying content decryption key, K_{-RTS_ID} . The access token includes the RTS_ID, the public-key for RTS_ID (K_{+RTS_ID}), and the expiration time, T_{exp} . K_{-RTS_ID} is used for the content decryption key and for authentication. Recall that the content decryption key is required because it is used by the CFSS to access the encrypted content file in the content container.

To provide an efficient solution that reduces the size of the content ACL and reduces the number of tokens needed by each user, we use RBAC. The basic notion of RBAC is that permissions are assigned to roles, users are assigned to appropriate roles, and users obtain permissions by being members of roles.

D. Content Importation System

The only objects that can be shared in this protected environment are objects that have been imported into the system in the form of a content container with the appropriate access rights. The CITADEL architecture includes a *content importation system*(CIS) that controls the insertion of objects into the system. It functions as the secure gateway to import any objects into the protected file-sharing environment. A key design goal of the CIS is that it allows content providers to easily protect content and insert it into the distribution network. The CIS enforces that content is identified correctly, labeled with the correct policy, and encapsulated in a content container. The CIS helps maintain an open peer-to-peer sharing experience by allowing all peers to insert new content or existing content. The system examines the file to extract identifying properties and then determines the actual identity of the content by comparing the properties with a database of all known content files. This can be achieved using a content-based identification algorithm. Content-based identification algorithms analyze the perceptual qualities of the content to derive a fingerprint of the content [15].

V. THREAT ANALYSIS

In this section, we discuss how the system maintains the level of assurance relative to its protection goals by being robust against compromise. As in any protection

Goal	Single User Attack	Collusion Attack
Protected Distribution	Forge Access Token Replay Access Token	Bypass Authorization Checks Re-distribute Tokens Alter ACL
Protected Storage	Determine K_R Determine K_{CFSS} Alter ACL	Re-distribute Tokens
Output Protection	Obtain K_{RTS_ID} Obtain Plaintext Content	Re-distribute Content

TABLE I

SUMMARY OF ATTACKS ON PROTECTION GOALS

system, we assume the presence of malicious users that aim to gain unauthorized access to content.

There are different levels of threats to content protection ranging from the casual user to the hobbyist/hacker to the professional pirate [16]. The goal of commercial content protection is to “stop unauthorized, casual copying of commercial entertainment content” [16]. These are somewhat modest goals compared to the protection goals in many military and financial applications.

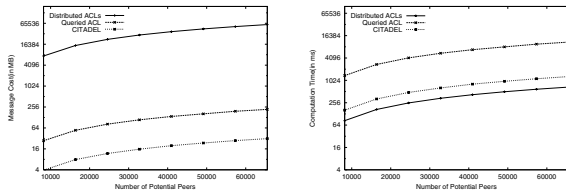
Table I shows the protection goals of the system and possible attacks on these goals. We look at attacks by an individual user as well as attacks by a collusion. Recall that CITADEL aims to provide protected distribution and protected storage. We briefly mention extensions to CITADEL that can provide output protection. Some of the attacks listed in the table provide no benefit to an attacker and some of the attacks are sufficiently complex due to fundamental properties of cryptography. For example, without an appropriate access token, determining K_R can only be done by brute force attack. For these reasons and due to space constraints, we do not detail each of the attacks in this section.

We assume a hobbyist/hacker end-user that has appropriate knowledge and resources. This user has full access to and control of the CFSS and the operating system. Since users have access to the CFSS on their PC, it will be a target for attackers seeking to circumvent the protection of the system. Our analysis shows that even without software protection or tamper resistance, attackers can not defeat the CFSS to obtain unauthorized access to content. We assumed an attacker reverse engineers the CFSS or otherwise fully compromises the CFSS. We analyzed the risk associated with each attack and show that the damage is limited and does not defeat our protection goals.

VI. PERFORMANCE EVALUATION

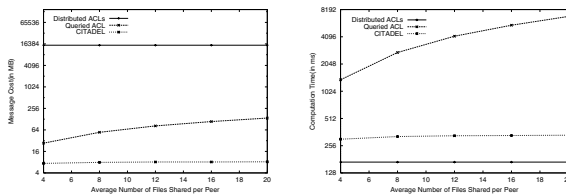
We present simulation results to show the performance of CITADEL relative to the performance of the distributed ACL and queried ACL approaches to decentralized content protection that we discussed in Section IV. We use a simulated peer-to-peer file sharing system largely based on the results of a measurement study of two large peer-to-peer file sharing systems [17].

Figure 2 shows the message overhead and computation overhead at the ACS as a function of the number of poten-



(a) Message Overhead (b) Computation Overhead

Fig. 2. Overhead as a function of number of potential peers: Message and Computation Overhead at the ACS



(a) Message Overhead (b) Computation Overhead

Fig. 3. Overhead as a function of number of files shared by peers: Message and Computation Overhead at the ACS

tial peers. Figure 3 shows the message and computation overhead at the ACS as a function of the average number of files shared per peer.

VII. PRACTICAL IMPLEMENTATION

We have implemented a CITADEL prototype using open source components including the Gnutella network as the distribution layer [3] and the LimeWire graphical user interface-based Gnutella client [18] as the file-sharing software. One of the key goals of the implementation efforts was to show that the CITADEL architecture can be deployed as part of any popular file-sharing network. Our implementation does authorization based on X.509 public-key certificates. Content containers are created using Secure/Multi-purpose Internet Mail Extension (S/MIME).

VIII. RELATED WORK

There are numerous works in content protection. Content protection covers many environments, distribution methods, content types, and protection goals. Three phases of content protection that are common across different distribution methods and content formats are *protected distribution*, *protected storage* and *output protection*. Protected distribution deals with providing conditional access or enforcing an access policy in the distribution model. This essentially controls access to protected objects. Protected storage deals with controlling access to the actual content in a protected object. This essentially

controls playback of protected objects. Output protection deals with protecting content after an authorized user is accessing the content. This focuses on restricting access to the content as it is played by the user.

IX. CONCLUDING REMARKS

Predicting how peer-to-peer systems will evolve is difficult at best. What has become clear, however, is that content protection will play an increasingly important role in the success of such systems. The challenge has been how to design a system that provides adequate content protection and yet maintains the openness of the peer-to-peer model. In this paper we present and evaluate the design of CITADEL, a content protection system designed to operate in a decentralized peer-to-peer system. The system can be viewed as providing a content protection layer in between the peer-to-peer distribution infrastructure and the layer providing services to users. One of our central premises is that content protection should be viewed as an opportunity to enable new service functionality and not a nuisance that designers have to deal with. This layered approach to content protection and the focus on providing a flexible framework allows CITADEL to perform the role that we envision for it.

REFERENCES

- [1] KaZaA. <http://www.kazaa.com>.
- [2] P. Cohen, "Napster remains closed following court order." <http://maccentral.macworld.com/news/0107- /12.napster.shtml>, July 2001.
- [3] Gnutella. <http://gnutella.wego.com>.
- [4] S. Bonisteel, "RIAA sues Napster clones: Kazaa, Morpheus and Grokster." <http://www.newsbytes.com/news/01/170798.html>, Oct. 2001.
- [5] B. King, "Napster to start filtering songs." <http://www.wired.com/news/politics/0,1283,42140,00.html>, Mar. 2001.
- [6] "Report: Napster users lose that sharing feeling." <http://www.cnn.com/2001/TECH/internet/06/28- /napster.usage/>, June 2001.
- [7] C. Shields and B. Levine, "A protocol for anonymous communication over the internet," in *ACM Conference on Computer and Communication Security*, Nov. 2000.
- [8] D. Chaum, "Blind signatures for untraceable payments," in *Proceedings of CRYPTO '82*, pp. 199–203, 1982.
- [9] V. Scarlata, B. Levine, and C. Shields, "Responder anonymity and anonymous peer-to-peer file sharing," in *ICNP*, Nov. 2001.
- [10] Freenet. <http://freenet.sourceforge.net>.
- [11] P. Q. Judge and M. H. Ammar, "The benefits and challenges of providing content protection in peer-to-peer systems." Submitted for publication.
- [12] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 20, no. 2, pp. 38–47, 1996.
- [13] G. Medvinsky and B. C. Neuman, "Netcash: A design for practical electronic currency on the internet," in *Proceedings of the First ACM Conference on Computer and Communications Security*, vol. 1993, pp. 102–106, 1993.
- [14] D. Chaum, A. Fiat, and M. Naor, "Untraceable electronic cash," *Lecture Notes in Computer Science*, pp. 319–327, 1988.
- [15] E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio," in *IEEE Multimedia*, 1996.
- [16] B. Pearson, "Digital transmission content protection." http://www.dtcp.com/data/dtcp_tut.pdf, June 1999.
- [17] S. Saroiu, P. K. Gummadi, and S. Gribble, "A measurement study of peer-to-peer file sharing systems," in *Multimedia Computing and Networking (MMCN)*, 2002.
- [18] LimeWire. <http://www.limewire.org>.