A File Protection Method for Peer-to-Peer Systems

Antonio Liotta[1], Rossana Motta[2] and Ling Lin[1]

[1] Department of Electronic System Engineering
University of Essex, Colchester, CO4 3SQ, UK
{aliotta, llini}@essex.ac.uk
[2] Florida State University, Tallahassee, FL 32306, USA
motta@cs.fsu.edu

Abstract. The benefits of peer-to-peer (P2P) systems as vehicles for information dissemination are overwhelming. They beat the conventional client-server (C-S) approach in terms of scalability, availability, and network efficiency to mention a few. Nevertheless, the problem of protecting copyrighted material in P2P systems is far from solved, whilst several solutions have been proposed and are in use in a C-S environment. In this article, we propose a framework which allows 'legal' P2P file exchange, making it possible for copyright holders to employ P2P as a distribution vehicle. We illustrate a scenario where all parties involved, including the end user, benefit from our approach. Lessons learned and conclusions are drawn from a prototype implementation.

1 Introduction

The Internet as we know it today has become the major vehicle for distributing multimedia digital material directly and conveniently to the end user. But what is the ultimate technology for content provisioning?

The main mechanisms for reaching the user terminal are known as client-server (or C-S) and peer-to-peer (or P2P). In C-S, the content is made available via computer servers (or server farms). The user can search for relevant information (via any of the popular search engines) and finally retrieves the content through the browser (i.e. the client application). Clearly this approach is limited in terms of scalability, availability, and efficiency, since the server (or server cluster) constitutes a computational and network bottleneck. We have all experienced occasions when a web page (or server) is not accessible or when we cannot connect to a real-time streaming service.

Most limitations of C-S have been overcome in P2P systems. In this case, the user terminals themselves form a sort of self-organized content distribution network. Any of the terminals joining the P2P network can in fact play not only the role of passive user (or client) but also, and seamlessly, the role of content provider (or server) - because of this, all terminals are peers. Assuming that suitable publish and search mechanisms are in place, content can be downloaded from any terminal (holding a copy of it) to any other terminal. In the most common P2P systems, whenever the demand and popularity of a file increases, more and more copies of that file will be

Advances in Computer Science and Engineering.
Research in Computing Science 23, 2006, pp. 151-160

Received 13/07/06
Accepted 04/10/06
Final version 10/10/06

spread in the P2P network. In fact, many peers will have that file among their resources and other peers will have the chance to download it from several sources, feeding a positive circle chain. Thus, we can easily see why this approach is considerably more powerful than its C-S counterpart (there is no single server, no single point of failure, and no bottleneck).

Despite its superiority as a mechanism for moving content around, P2P presents a major problem when it comes to protecting copyrighted material. In fact, we can confidently say that none of the largest media production companies is currently conceiving P2P as a vehicle for distributing copyrighted material. P2P systems have in fact become popular in the realm of free (or illegal) content distribution, while most of the non-free content reaches the user via C-S transactional systems for which suitable security means are available.

In this paper, we propose a system which allows content providers to use P2P as a vehicle for distributing copyrighted material, addressing the three key legal challenges of the copyright law: copy infringement; distribution infringement, and vicarious and contributory liability (a thorough explanation of these issues is given in [1]).

In our approach, a trusted entity (for instance the network operator) handles user authentication, registration, authorization and transactions, acting as an intermediary between content provider and users. This portion of the system works in C-S mode, whereas copyrighted content circulates among users in a P2P fashion. We explain how existing encryption and key management technologies can be used in a new way in order to protect files in P2P systems. Our approach is combined with an incentive-based scheme where all parties involved benefit from the 'legal' circulation of information.

Conclusions are drawn from a prototype implementation which has been developed by an academic institution in collaboration with a major network operator. Our prototype runs on thin mobile terminals (personal digital assistants), in addition to ordinary personal computers, and is able to operate not only in wired and wireless networks but also over UMTS.

2 Related Work

2.1 Existing Solutions

While there is a huge variety of different applications and protocols when considering P2P and Digital Right Management (DRM) contexts separately, it appears that no effective solutions are actually available, when trying to merge such contexts.

The great most of P2P systems simply do not have any features at all that allow DRM policy to be integrated. To our knowledge, the only P2P system that claims to distribute DRM-protected content is Kazaa [4]. The latter, however, exploits software from Altnet company [5], which has been recognized by any antivirus as responsible of spywares and other malwares (*i.e.*, "software installed without the user's informed consent, is difficult or impossible to uninstall, and transmits information about the user's activities without notice or consent" [3]). It is even not completely clear if

those files are downloaded in a C-S manner, as some sources report (for instance [8]), or via P2P, as Kazaa claims. During trials that we have performed ourselves after installing Kazaa, we have not been able to find any DRM-protected file residing on peers, whilst all of them were on central servers. Thus we are not able to confirm that Altnet system embedded in Kazaa is actually P2P. In any case, the installation of spywares as an integral part of DRM policy is clearly unacceptable and not viable, if seeking a business approach.

A number of different protocols exist for DRM itself. Actually one of the biggest problems that DRM is facing is the lack of interoperability [9], so that users are tied to very specific applications every time they want to render their files. This clearly feeds a large-scale mistrust toward DRM systems in general. The situation has been made even worse after the clamorous case of one of the most popular worldwide record companies, which has grossly exploited spyware techniques, in the intention of achieving efficient copyright respect [3].

There is actually a growing interest from several major companies dealing with DRM for the so called OMA-DRM project [7]. This represents an attempt toward a large-scale standardization of DRM and at the same time it includes several features that would make it portable in a P2P environment. OMA-DRM is actually still at a development stage and further studies on its implementation in large-scale real systems would be needed, in order to draw more precise conclusions [10].

2.2 Requirements for DRM-enabled P2P

The P2P realm is evolving at an unprecedented pace, which makes it very difficult to capture a complete snapshot of P2P systems and technologies. To the best of our knowledge, none of the existing P2P systems satisfies the following requirements, which are fundamental to effective DRM for commercially-viable P2P:

- *Content provider protection*: prevents the user from illegally duplicating and distributing digital material. Current encryption schemes used in C-S systems cannot be easily exported to P2P - users can currently easily inject content (obtained either legally or illegally) into the P2P network. Some P2P systems such as *Freenet* [2] guarantee user's anonymity, further exacerbating the illegal file distribution problem.
- *P2P provider protection*: the P2P platform developer incurs in vicarious liability if it can be proved that they can exert some form of control over direct infringers' behavior [1]. This is one of the reasons why *Napsters* lost a lawsuit and was shut down.
- *User protection*: some P2P systems make use of *spyware* to monitor user actions and achieve some sort of DRM. As mentioned, this is the case of *Kazaa*.

Our solution, as proposed below, satisfies these three requirements, while also having the following advantages:

- It makes use of well-established encryption technologies;
- It is generic, and may be used with different devises and file formats other than the ones used to develop our prototype;

- It is based on user incentives (it allows the implementation of different economic and rewarding schemes), and ensures that all parties involved (user, content provider, P2P service provider) benefit from joining the service;
- It does not require an unbreakable DRM system (at present it would be unrealistic to assume that such a system can be realized) since the user (i.e. the potential hacker) would stop receiving rewords in that case (this will become clear after the following section). Our system is sufficiently robust to prevent involuntary copyright infringement and would require substantial expertise and facilities (beyond the abilities of the typical expert Internet user) to be broken.
- It is based on a robust authentication framework handled by the network operator, which instills a sense of trust and encourages user participation to the P2P playground.

3 DRM-enabled P2P system

3.1 System entities

In a real-case scenario there will be several entities that will compete for a slice of the media selling market and others who will collaborate to enhance the user accessibility to content. However, for the sake of simplicity we assume that the key players of our P2P content distribution systems are:

- *A Content Provider (CP)*. This will be a media production or broadcasting company producing music, videos, films, *etc.* and holding the copyrights.
- *A Trusted Entity (TE)*. This will be a network operator, service provider or any other company that manages a large amount of subscribers and has gained their trust. The TE has the means to handle user authentication, authorization, charging and billing (via a transactional system).
- *The users (or Peers)*. These will be ordinary users with a terminal connected to the Internet (subscribing to a TE) and who want to purchase copyright protected material. Peers will seamlessly act as content distributors in the P2P network.

In general there will be several content providers advertising their products (music files, videos, etc) via brokers (yellow or white pages). Users will register with different (competing) trusted entities (for instance their own mobile network operator) that will also have to federate in order to allow user and service portability (similarly to what mobile network operators do to support the roaming user). Our approach to DRM-enabled P2P has been designed to work across network operator boundaries (inter-domain) and independently from access network technologies and user terminal type. The CP may also act as TE or *vice versa*.

3.2 Bootstrapping

The distribution of digital content from producer to the end user will be 'initially' mediated by the TE who, for each file, will produce the encrypting/decrypting key and encrypt the file itself. One of the state-of-the-art symmetric encryption techniques will be used. We'll see how, once the P2P network gets populated with copies of the original individual files, the TE will not have to act as a content distribution hub anymore, since content will be distributed directly by the peers. The TE will still hold a master copy of all files injected in the P2P system.

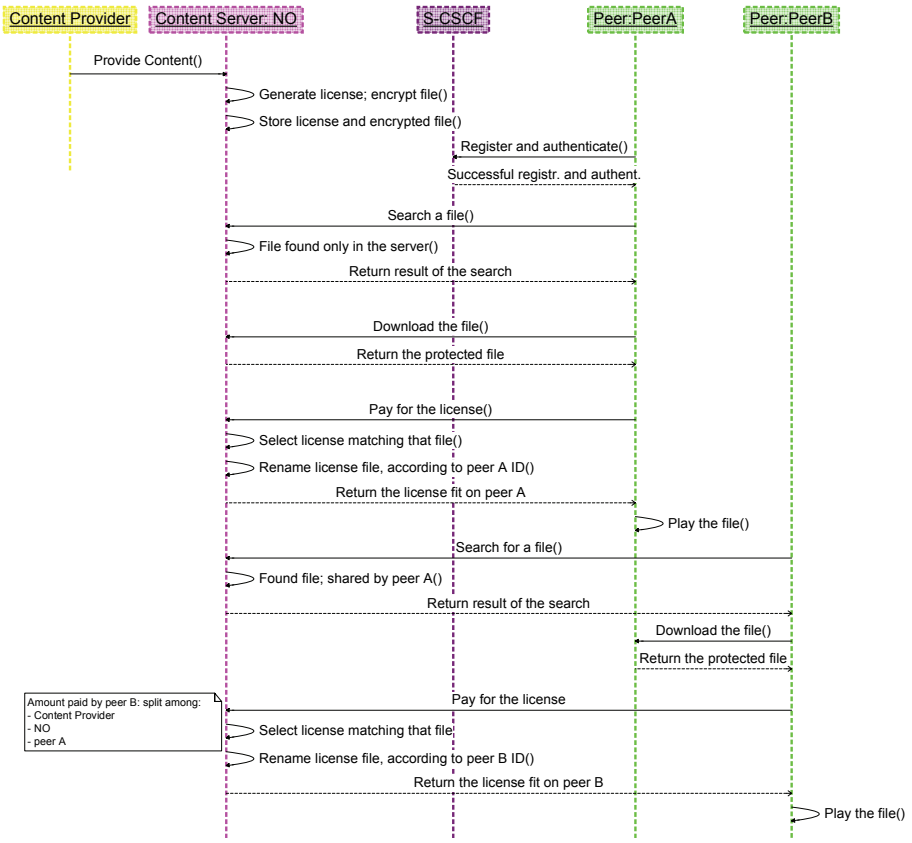

Fig. 1. Sequence of diagram describing main entities and interactions**.**

Figure 1 illustrates the bootstrapping of the content distribution process, starting from the scenario where there are no peers and all content is stored in the content server. We assume that this is managed by the Network Operator (NO) who is also in charge of user and transaction management.

The first operation any peer will have to perform before getting access to the services/content available in the P2P network is registration/authentication. `PeerA`

will then join the P2P network and have access to the search facilities which will provide the same sort of functionality of ordinary Internet search engines.

3.3 File protection hurdles

If the retrieved file is copyright protected, `PeerA` will have to be debited the relevant cost before obtaining the license key that will decrypt the file. This process would be straightforward in a C-S system but is extremely detrimental in the context of P2P. Once `PeerA` gets hold of the key, it could easily share the license key of the file with any other peer. This is in fact the crucial problem with current P2P systems which we want to counter since this is what generates the copy and distribution infringement issue.

The technical problem we have to address is how to allow the same encrypted file to circulate in the P2P network, allowing authorized users to read (or play) it freely but preventing illicit swapping of licenses. We tackle this problem from different fronts. First, we protect the key (Sect. 3.4). Then, since it is quite hard to create an unbreakable DRM system (especially in the context of P2P), we couple the P2P system with a strong user authentication and authorization system (provided by the network operator in our prototype). Finally, we encourage lawful peer participation via an incentive mechanism, which, on the other hand, demotivates hacking actions (Sect. 3.5).

3.4 P2P file protection

The application running on the peer (*e.g.*, the player if the file is a music or video file) expects the key to have a specific filename. The filename is generated by the server using a secret function, *F,* that gets as input the tuple `{peer ID; ID of the encrypted file}`. *F* can be specified starting from state-of-the-art cryptographic algorithms and will incorporate a set of operations consisting of bits manipulations, shifts, bitwise mathematical operations etc. Every bit of the input tuple is considered by *F*, so any variation of the input would result in a different output.

Going back to Figure 1, upon receiving the payment for the relevant license, the server renames the license file univocally to `PeerA` and forwards it to the peer. `PeerA` will now use *F* to compute the location (filename) where the license key will have to be in. It will, then, feed the actual key and source file to the player. A further advantage of our approach is that the only proprietary module is the one running *F*, while it will work with any type of file and application. For instance, if we are using the system to distribute music files, our system will operate with any (locally installed) player application.

Let us see what happens when a second peer, `PeerB` is searching the same file that `PeerA` has just obtained (for simplicity Figure 1 does not illustrate `PeerB`'s registration phase). Now there are two copies of the same encrypted file, one in the server and the other in `PeerA`'s shared resources. Clearly, this time `PeerB` will download the file from `PeerA`, so the server will be relieved from this task. Given that the file is protected, `PeerB` will have to pay for the license, expecting the key to

be stored in a file whose filename is uniquely tailored to `PeerB`. This is something that `PeerA` cannot produce, unless it successfully reverse-engineers *F*. Then, even if `PeerA` were to pass its own license key to `PeerB`, this would not be in the correct filename and path and would be unusable by the decryption routine.

As more peers join the P2P system and express interest in certain files, the latter would gradually populate the system. The licenses, though, will always have to come from the server, which is how we obtain a DRM-enabled P2P. In addition, the more peers are interested in a given file, the more copies of that file will be circulating into the system, leading to increasing levels of scalability and availability.

3.5 Trust, policing, and incentives

In order to further increase the level of security, the file protection mechanism described above is carried out in conjunction with a strong user authentication and authorization component. Our prototype, as specified in Section 4, is integrated in the context of the IP Multimedia Subsystem (IMS), which is the service provisioning framework adopted by the major network operators [6]. Because of that, the IMS comes with a strong emphasis on user management and with specific service components for transaction management, charging and billing. The operator will, therefore, be in a good position to act as trusted entity, monitor the user activity, and handle the necessary key management process.

Effectively, what the operator's platform gives us is the ability to know that the peers in our P2P system are who they say they are. Peers will have also accepted appropriate terms and conditions at registration time. In this way it will be possible to prosecute copyright infringers.

Experience has shown that law enforcement *per se* is never sufficient to stop the compulsive hackers. Economic incentives, instead, appear to be more successful at deterring illegal activities. In our system, a P2P file exchange is handled as an atomic transaction which is concluded when the license has been paid for. Referring to Figure 1, once `PeerB` pays, the operator charging system will credit a portion to itself, a portion to the copyright holder and another portion to `PeerA`, as a reward for acting as file distributor.

Therefore, peers have concrete advantages from 'legal' file circulation. Even if they managed to hack the system, obtain an unencrypted copy of the original file, and get away with the network operator's policing system, they would damage themselves by injecting the unencrypted file in the P2P system – i.e., they would not receive any economic incentive in doing so.

4 The Prototype

What we have actually prototyped is a P2P system built on the IMS (Figure 2). We believe to be first to have accomplished a concrete example of an operator-mediated P2P system, where P2P operations are supervised, traced and charged by the network operator (*i.e.*, the trusted entity).

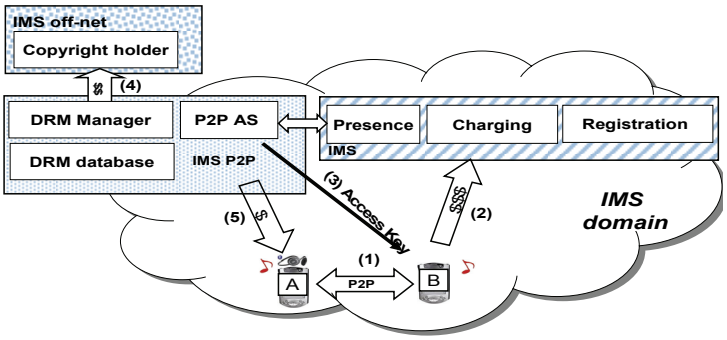

Fig. 2. Logical view of prototype implementation**.**

Figure 2 presents a simplified view of the system which is, in fact, considerably more complicated, since the IMS is a comprehensive service provisioning platform. The *presence*, *charging* and *registration* components have been implemented according to standards specifications [6]. We have built our DRM-P2P system as an extension to the IMS. The content provider does not belong, in general, to the operator's (IMS) domain, although this is not a requirement.

Our IMS P2P set of functionalities has been implemented in Java 1.5.06 (December 2005 release) and has been tested on Linux and Windows XP. We have also prototyped a peer-side application for mobile terminals – specifically the Hewlett-Packard iPAQ h5555 Pocket PC - based on J2ME and WindowsMobile. In this way we have proved that the proposed approach works on relatively thin terminals in addition to ordinary computers.

For encryption and key management we have adopted the AES (advanced encryption standard) symmetric block cipher algorithm - CBC (cipher block chaining) cipher mode and PKCS7 padding mode. CBC needs not only a key but also an Initialization Vector (IV). Hence, in our prototype, the license includes two strings (128 bits each), which are pseudo-randomly generated by the server at encryption time.

For the input parameters of the encryption function, *F* of Sect. 3.4 we have adopted the SIP identifier (to uniquely identify the peer ID) and as file ID the name of the encrypted file. A more robust solution would be to use hash functions (*e.g.* based on MD5). But this operation would have been too heavyweight (especially as file size grows), considering that our peers maybe thin mobile clients. *F* can be defined starting from ordinary cryptographic algorithms, performing an original set of operations, manipulations and shifts on the input parameters.

Just to get an idea of the order of magnitude of the decryption overheads incurred on a terminal with a 2.4 GHz CPU, 1 Gbyte RAM, we obtained the following figures (similar on Linux and Windows XP):

File size (Mbytes)	Overheads (seconds)
3.7	1
6	2
15	4
50	14

Finally, the user rewarding scheme has been implemented according to the functionalities specified by OMA DRM v2 [7].

The aforementioned technological choices are merely an indication of how our system may be readily realized. However, other similar (or future) technologies can be chosen.

5 Conclusions

The community interested in media and information dissemination is split between those who advocate and those who oppose the copyright law. The widespread appearance of P2P systems has re-fuelled this controversy, since the P2P approach is significantly more efficient than its C-S counterpart. P2P is, however, being largely used for distributing free content while also sparking the illegal distribution of copyrighted material.

The perspective presented in this article is that the copyright infringement issue would naturally disappear if all parties involved (including the user) could benefit from the legal distribution of copyrighted material. We have presented a framework which exploits P2P as a content distribution platform (giving the user the role of distribution hub) while also addressing the DRM issue.

Our system can be used to experiment with different types of economic (incentive-based) models. An interesting exercise would be to see what happens when different incentive schemes are deployed in a very large scale. We plan to run simulations to unveil the effects that different incentives have on system scalability.

We do expect to see an increase in user participation to the P2P distribution role. Current P2P systems suffer from the so-called *free-riding* issue, whereby peers obtain content from the network but do not act as distributors. They do that in order to save the consumption of their own computer and network resources. But, in doing so, they fail the very purpose of P2P and lead to a dramatic reduction in performance – the performance of P2P systems degenerates dramatically when free-riders dominate the scene. By providing economic incentives to user participation, our approach will combat the free-riding issue.

Having integrated our prototype with a state-of-the-art (standardized) service provisioning platform (*i.e.*, the IMS), we are effectively showing an evolutionary path towards commercially-viable P2P content distribution. We have also demonstrated that such approach can work across network boundaries (we have tested it on UMTS, as well as on wired and wireless access networks) and for different terminals (PDAs and personal computers).

Hence, our findings indicate that P2P can, indeed, become the ultimate technology for content provisioning. A lot of work lays ahead in order to fully address a number of legal, standardization and interoperability issues.

Acknowledgements. This work has been carried out in the context of the PeerMob project, which is funded by Vodafone Group Plc. We are particularly grateful to Patrick Brick, Max Gasparroni, and Marco Ballette (all from Vodafone) for their insightful comments.

References

1. S.T. Logan, "Peer-to-Peer Technology and the Copyright Crossroads". In "Peer-to-Peer Computing", Idea Group Publishing, 2005, pp.166-193.
2. I. Clarke et al., "Protecting Free Expression Online with Freenet", *IEEE Internet Computing*, IEEE, Jan-Feb 2002, pp. 40-49.
3. E.W. Felten, J. A. Halderman, "Digital Rights Management, Spyware, and Security", *IEEE Security and Privacy*, IEEE, Jan-Feb 2006, pp. 18-23.
4. The Kazaa system is available at www.kazaa.com
5. The Altnet software is available at www.altnet.com
6. G. Camarillo, M.A Garcia-Martin "The 3G IP Multimedia Subsystem (IMS): Merging the Internet and the Cellular Worlds". John Wiley and Sons Ltd, December 2005.
7. The Open Mobile Alliance Digital Right Management specifications are accessible at www.openmobilealliance.org/release_program/drm_v2_0.html
8. http://en.wikipedia.org/wiki/Altnet
9. Heileman, G. L. and Jamkhedkar, P. A. 2005. DRM interoperability analysis from the perspective of a layered framework. In Proceedings of the 5th ACM Workshop on Digital Rights Management (Alexandria, VA, USA, November 07 - 07, 2005). DRM '05. ACM Press, New York, NY, 17-26.
10. Thull, D. and Sannino, R. 2005. Performance Considerations for an Embedded Implementation of OMA DRM 2. In *Proceedings of the Conference on Design, Automation and Test in Europe - Volume 3* (March 07 - 11, 2005). Design, Automation, and Test in Europe. IEEE Computer Society, Washington, DC, 46-51.