A Method of Fair Use in Digital Rights Management*

Yong Zhong[1,2], Zhu Zhen[2], Dong-mei Lin[2], and Xiao-lin Qin[3]

[1] Department of Computer Science and Technology,
Nanjing University of Science and Technology,
210094 Nanjing, China
[2] Information and Education Technology Center, Foshan University,
528000 Foshan, China
[3] Institute of Information Security,
Nanjing University of Aeronautics and Astronautics,
210016 Nanjing, China
zhongyong@fosu.edu.cn

Abstract. Fair use is a difficult problem to implement in DRM systems due to its vagueness and uncertainty. We propose a fair use mechanism based on *rights assertion* without limitation, *audit logging* and *misuses trigger*, which brings a fair use mechanism nearer to offline world than that of existing DRM systems.

Keywords: Intellectual property rights, fair use, digital rights management.

1 Introduction

Fair use is a difficult problem to implement in Digital Rights Management (DRM). Mulligan et al.[1] show that current Rights Expression Languages (RELs) could not express or even approximate most of the limitations posed in copyright law. Cheun et al.[2] propose a two-part approach to approximate fair use rights in LicenseScript by *Rights assertion* and *Audit logging*. But in their method, content owners still retain strong control over the rights asserted by users. We propose a fair use mechanism based on *rights assertion* without limitation, *audit logging* and *misuses trigger*, which is the first method to permit users to assert any new fair use rights without restrictions but still can prevent users abusing their rights, which is closest to the offline world.

2 Fair Use in LucScript

LucScript[3] is a new Rights Expression Languages developed by us, which is based on Active-U-Datalog[4] language that is a logical update language integrating active rules. An Active-U-Datalog program $P=IDB \cup EDB \cup AR$ consists of an extensional database *EDB*, of an intensional database *IDB* and of a set of active rules *AR*. A license of LucScript is a 5-tuple (*D*, *IDB*, *AR*, *EDB*, *BV*) where *D* is a unique identifier

---

representing the data that the license refers to, *IDB* is the deductive rules set of the license, *AR* is the active rules set of the license, *EDB* is the set of ground extensional atoms of the license, *BV* is a set of bindings, i.e., a set containing elements of the form *name≡value* such as {*expires≡*'06/12/31'}. Predicate *binding*(*name*, *value*) is used to manage binding set. We call Δ=*IDB* $\cup$ *EDB* $\cup$ *AR* $\cup$ *BV* the program of the license and show a license as *lic*(*D*, Δ).

In our fair use mechanism showed as Fig.1, the content owner can't limit the fair use rights assertion of consumers directly, but they can set some trigger rules in *assertion license* or *audit license* to alert the third party authority if the consumers have some misuses matching the rules. Consumers have three methods to execute their fair use rights: (1) a range of preauthorized fair uses modeled on existing fair uses in offline world are encoded into the license, (2) a consumer who wants to engage in a non-preauthorized use could submit a request to the third party authority to acquire an authorization, and (3) if the third party authority denies the request, or if the consumer judges the burden of applying to the authority to be too great, the consumer can simply assert the right and executes it by his own way, and the tradeoff is that the assertion and some of information of his will be recorded in an *audit license*.
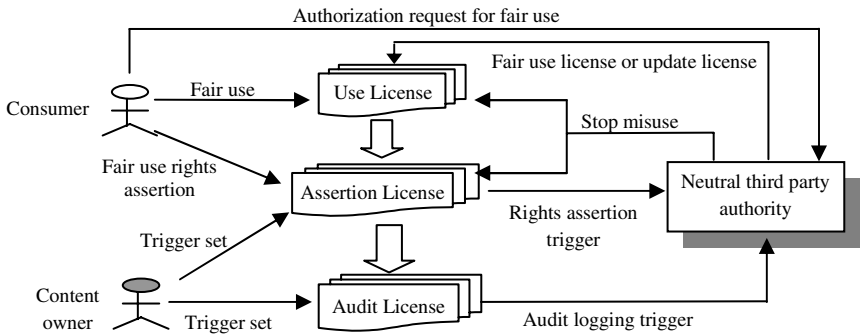

Fig. 1. Mechanism of fair use in LucScript

Suppose Alice wants to use the book *e-book_stars*. Firstly, She must download the encrypted book from the publisher. Then she must purchase a *use license* in order to read the book. Assume the license purchased by Alice is as follows. The license permits the read privilege to Alice before expire time (line 2), where the predicate *display*(*D*, *n*, *m*) means the consumer can read the book *m* pages from page *n*.

(1) *License (e-book_stars, //D*

(2)   { *display*(*D*, *n*, *m*)←*current_date*(*d*), *binding*('*expire*', *d*'), *d*≤*d*' },   *//IDB*

(3)   { }, *//AR*

(4)   { }, *// EDB*

(5)   { *owner≡*'Alice'; *id≡*'12346'; *version≡*'11.6.1'; *expire≡*'07/12/31' ; *type≡*'use' }, *//BV*

(6)   )

Alice thinks she should have the privilege to print the book, but there unexpectedly is no print function in the license. So she decides to request the right by sending the *request license* as follows, whose *IDB* rule (line 2) is the right that Alice is requesting.

(1) *License (*   *e-book_stars,* *//D*
(2)   { *print(D, n, m)←current_date(d), binding('expire', d'), d≤d' },*   *//IDB*
(3)   { },   *//AR*
(4)   { },   *// EDB*
(5)   { *owner='Alice'; id='12347'; expire='07/12/31' ; type='request' },* *//BV*
(6))

After receiving the request, the third party authority thinks the request is reasonable, so an *update license* as follows is sent back to the consumer. The update rule of the license (line 2) will insert an *EDB* rule to the *use license*.

(1) *License (*   *e-book_stars,* *//D*
(2)   { *update()←+EDB("print(D, n, m)←current_date(d),binding ('expire', d'), d≤d' ") },* *//IDB*
(3)   { },   *//AR*
(4)   { },   *// EDB*
(5)   { *owner='Alice'; id='12348'; expire='07/12/31' ; type='update' },* *//BV*
(6))

After the *update license* comes, the update will be added to Alice's *use license* and the license will become the following license, which grants the print privilege to Alice.

(1) *License (*   *e-book_stars,* *//D*
(2)   { *display(D, n, m)←current_date(d), binding('expire', d'), d≤d' ;*
(3)   *print(D, n, m)←current_date(d), binding('expire', d'), d≤d'*   },   *//IDB*
(4)   { },   *//AR*
(5)   { },   *// EDB*
(6)   { *owner='Alice'; id='12346'; version='11.6.1'; expire='07/12/31' ; type='use' },* *//BV*
(7))

For excerpt privilege, Alice decides to assert the rights simply. However, the content owner thinks there are no any reasons that a consumer can excerpt his books more than a half, so he decides to set a trigger to alert third party authority if the consumer excerpts his books more than a half. After Alice assert her rights by submitting the transaction ?-(+*IDB*("*modified(e-book_stars, n, m)←.*"), the *assertion license* is as follows, where *modified(e-book_stars, n, m)* means the consumer can excerpt the books *m* pages from page *n* and since the assertion has no limit on *n* and *m*, which grants Alice excerpts any page. Extensible predicate *modified_pages* (*p*) keeps the pages that have been modified by consumers, where *p* is the type of set.

(1) *License (*   *e-book_stars,* *//D*
(2)   { **modify(e-book_stars, n, m)←. ;** *// the asserted right*
(3)   *alert(D, p)←create(lic(D, Δ)), insert(lic(D, Δ), p), binding('agentIP', I), send(I, lic(D, Δ));*
(4)   *insert(lic(D, Δ), p)←Δ?+(+BV(b)), BV_name(b, name), name<>'type';*
(5)   *insert(lic(D, Δ), p)←Δ?+(+binding('type', 'alert'));*
(6)   *insert(lic(D, Δ), p)←Δ?+(+ modified_pages(p))*

(7)   },   //IDB
(8)   { -modified_pages(p), +modified_pages(p')← +modify_a(D, n, m), modified_pages(p), append(p,
                    n, m, p');
(9)   +()←+modify_a(D, n, m), modified_pages(p), binding('max_pages', max), sizeof(p, k), plus(k,
                    max, max'), max'>max, append(p, n, m, p'), alert(D, p')
(10)   },   //AR
(11)   { modified_pages ({}) },   // EDB
(12)   { owner≡'Alice'; id≡'12360'; type≡'assert'; max_pages≡'100', agentIP≡'202.192.168.38'}, //BV
(13))

The rule of line (3) explains how to create the *alert license*, where *create*(*lic*(*D*, Δ)) create a license in memory and *send*(*I*, *lic*(*D*, Δ)) sends the license to the third party authority. The rule of line (4) inserts the *BV* data of *assertion license* except *type* binding to the *alert license*. The rule of line (5) sets the *type* of the license. The rule of line (6) set the pages that have been modified by consumers. The *AR* rule of line (8) appends the page modified by consumers to *modified_pages*(*p*) when consumers modify a page. The *AR* rule of line (9) activates the creation and send of *alert license* when the pages modified by the consumers are more than half of the book, where +() is a no update operation for rule syntax. The *alert license* will be as follows.

(1) License (e-book_stars, //D
(2)   { },   //IDB
(3)   { },   //AR
(4)   { modified_pages ({1,3,7,…}) },   // EDB
(5)   { owner≡'Alice'; id≡'12360'; type≡'alert'; max_pages≡'100', agentIP≡'202.192.168.38'}, //BV
(6))

At last, when Alice assert her rights by submitting the transaction ?-(+*IDB*("*modified*(*e-book_stars, n, m*)←."), a transaction -(+*log*(*User*, *Action*, *time*, *purpose*)) will also be submitted to the *audit license*, where *User* is the identifier of the consumer, *action* is the asserted right, *purpose* is the purpose of asserting the right. The *audit license* is as follows after above right assertion.

(1) License (e-book_stars, //D
(2)   { },   //IDB
(3)   { },   //AR
(4)   { log('Alice', 'modify(e-book_stars, n, m)←.', '1/7/2007 13:30:25', 'comment') },   // EDB
(5)   { owner≡'Alice'; type≡'audit'; max_pages≡'100', agentIP≡'202.192.168.38'}, //BV
(6))

3   Conclusions

In this paper we propose a fair use mechanism based on LucSciprt Language, which we show is a unique solution to fair use in DRM and is nearer to offline world than that of existing DRM systems.

References

1. Erickson, J.S.: Fair Use. DRM, and Trusted Computing. Communications of the ACM 46(4), 34–39 (2003)
2. Chong, C.N., Etalle, D.S., et al.: Approximating Fair Use in LicenseScript. In: 6th Int. Conf. of Asian Digital Libraries, Kuala Lumpur, Malaysia, pp. 432–443 (2003)
3. Zhong, Y., Lin, D.M.: Report of LucScript Rights Expression Languages. Technical report, Information and Education Technology Center, Foshan University, China (2007), http://www.fosu.edu.cn/zhongyong/LucScript.pdf
4. Bertino, E., Catania, B., Gori, R.: Active-U-Datalog: Integrating Active Rules in a Logical Update Language. In: Kifer, M., Voronkov, A., Freitag, B., Decker, H. (eds.) Dagstuhl Seminar 1997, DYNAMICS 1997, and ILPS-WS 1997. LNCS, vol. 1472, pp. 107–133. Springer, Heidelberg (1998)