Extending an OMA-based DRM Framework with Non-Repudiation Services

Jose A. Onieva[1], Javier Lopez[1], Jianying Zhou[2],and Rodrigo Roman[1]
[1] Computer Science Department, Univ. of Malaga, 29071 - Malaga, Spain
{onieva,jlm,roman}@lcc.uma.es
[2] Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
jyzhou@i2r.a-star.edu.sg

Abstract - *Digital Rights Management (DRM) is an umbrella term for any of several arrangements which allows a vendor of content in electronic form to control the material and restrict its usage in various ways that can be specified by the vendor. These arrangements are provided through security techniques, mainly encryption, and the distribution, in a detached manner, of content and rights. This allows free access to the content by the consumers, but only those carrying the proper Right Object (RO) will be able to process such content. As a security service considered in different layers of the security framework defined by ITU X.805, almost all applications need to consider non-repudiation in the very beginning of their design. Unfortunately this has not been done so far in DRM specifications due to practical issues and the type of content distributed. We analyze this service for the a DRM framework and provide a solution which allows the right objects acquisition to be undeniable.*

Keywords - *digital rights management, non-repudiation, secure electronic commerce, mobile applications.*

I. Introduction

The traditional industry for multimedia contents has used classical technologies for distribution and consumption. Nevertheless, with the introduction of digitalized multimedia and the use of telecommunication networks, content production and distribution has become easier and faster than ever before. These contents demand more protection from theft and prying eyes. This increasing need of content protection is driven by two trends. The first is mass piracy and theft of intellectual property and proprietary information. The second is that more "sensitive information" such as financial statement, medical records, and contracts are available in digital form and must be securely stored, shared, or distributed within and between organizations.

This is precisely the niche in which DRM comes out to offer us a solution. Technically, DRM is defined as a set of technologies and systems that can collectively support the entire life cycle of contents (creation, manipulation, distribution and consumption) by preventing illegal copying, imposing fees, processing payments, tracking contents, and protecting each principal's right and profit.

In these systems, content and rights are distributed in a detached manner. This technique simplifies the download of content and its management. No protection of the content is needed, such that any user can download it. But, of course, in order to consume it, a user needs to access (purchase) the corresponding *digital right object*. Here, two possible approaches for rights management exist:

Centralized: A user needs to access the corresponding right from a central manager each time it wants to consume content. It is very effective against malicious users, but not so against malicious rights managers. Additionally, this approach suffers from scalability problems.

Distributed: A user maintains its rights and just makes use of them when needed. It overcomes the existing drawbacks of centralized systems, but nevertheless, in order to avoid illegal use of the rights, a tamper-resistant hardware or *Trusted Personal Device* (TPD) is needed (that locally manages the rights in a certified and tamper-proof way).

With the advent of cellular networks, the distributed approach allows the convergence of user and industry needs. Combining DRM solutions with mobile networks, users can access the digital rights by using their mobile handset as a TPD. Telecom operators can drive the users for accessing or purchasing digital rights as well as certifying the secure management of digital rights in the handset (see Figure 1).

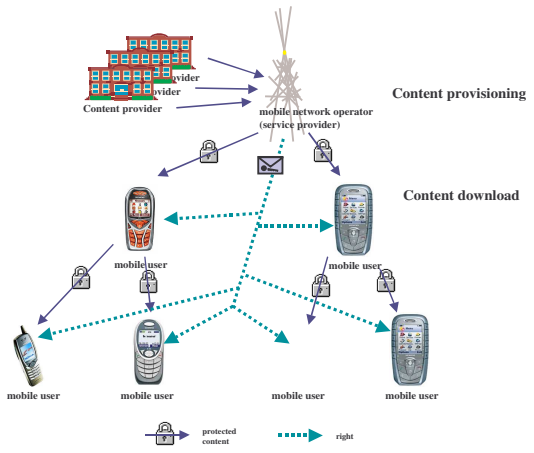

Fig. 1. Content Distribution

We modified a platform based on the OMA DRM specification 2.0 [10] (which has become an approved standard from the *Open Mobile Alliance*) for the distributed rights manage-

ment. The modified scheme proposed in the European project UbiSEC[1] will enable a more secure framework for charging on the digital rights acquisition by the consumer, taking into account important issues as anonymity and efficiency (see Figure 2).

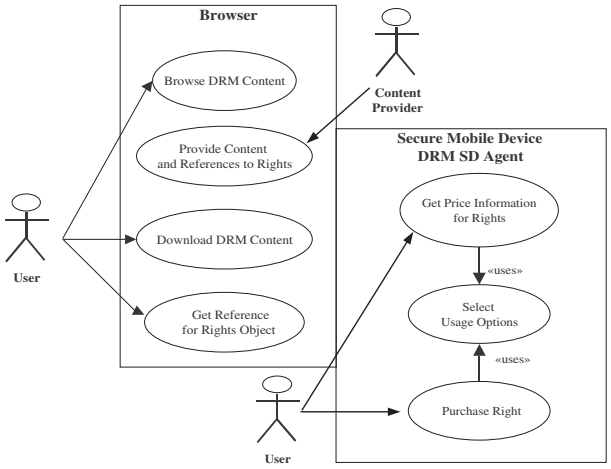

Fig. 2. DRM

The distribution of the RO to the user through a *Mobile Network Operator* (MNO) comes out as a final important step on the fair distribution of digital content (see Figure 3). Anonymous purchase of rights is supported, as the Content Provider and the *Rights Issuer* (RI) do not require privacy details of consumers. Consumer billing is performed through the MNO to whom the consumer is subscribed. Evidence will be generated, such that, if any dispute arises among the parties, they will be able to demonstrate their participation in the DRM scenario. Even though this solution strongly relies on trusted third parties (MNO and RI), non-repudiation issues on content distribution have to be considered, without having an impact on all the above mentioned properties.

Considering the user as the customer which receives content and rights in order to be able to consume such content, non-repudiation could be a valuable service for the customer in the last phase when it has to access the Right Issuer (through the Mobile Network Operator) to get the RO in exchange for the payment. (The MNO charges the user for the RO value in its monthly bill.)

Even though the MNO and the RI are considered trusted entities, there can be several difficulties in the process (e.g., a network failure or loss of data) which can end in disputes among the parties. Such possible disputes could be as follows.

- The MNO charges the user for the RO it did not purchase or receive. (It could also occur that the amount of money charged does not coincide with the one expected by the user.)
- The user receives a corrupted RO while already having paid for it.
- The user denies having sent a request (RORequest) for purchasing the RO.
- The MNO denies having received a request from the user.
- Similar disputes between the MNO and the RI.

From this list, and according to the definition of non-repudiation services given by the ITU, the non-repudiation of origin and non-repudiation of receipt services have to be provided between the user and the MNO and between the MNO and the RI, thus establishing a logical non-repudiation channel between the user and the RI.

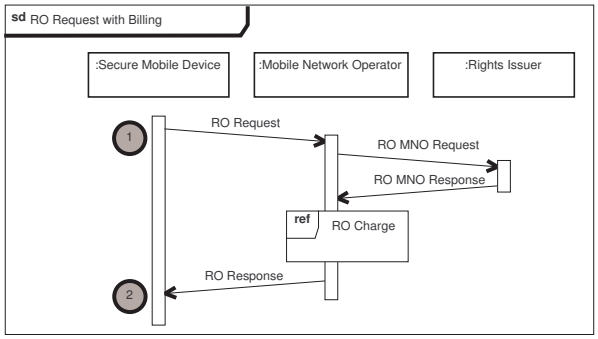

Fig. 3. Right Object Acquisition

The rest of this paper is organized as follows. In Section II we review the related work. In Section III we describe the specification and operation of the non-repudiation protocol, as well as its dispute resolution process. In Section IV we sketch some properties of the design phase, and discuss some implementation issues. We conclude the paper in Section V.

II. Related Work

Non-repudiation is an important requirement in electronic transactions [11]. In our case, it must not be possible for a Right Issuer to claim that he sent the RO when he did not. In the same way, it must not be possible for a user to falsely deny having received the RO. Evidence should be collected to resolve these disputes arisen between participating entities in a DRM scenario. Digital signature serves as a major type of cryptographic evidence, which links a message with its originator and also maintains the integrity of the message.

Fairness is also a desirable requirement in electronic transactions. A number of solutions to fair non-repudiation have been developed [7]. Some of them use a *Trusted Third Party* (TTP) that plays the role of a trusted intermediary between the participating entities. The major disadvantage of this approach is the communication bottleneck created at the TTP. Nevertheless, Zhou and Gollmann presented a protocol [12] where the TTP intervenes during each execution as a "low weight notary" rather than as an intermediary. Other solutions use an off-line TTP, assuming that participating entities have no malicious intentions and the TTP need not to be involved unless there is an error in the protocol execution. This is called the optimistic approach. There are also solutions that eliminate the TTP's involvement, but based on a strong requirement:

[1]Ubiquitous Networks with a Secure Provision of Services, Access, and Content Delivery (FP6-2002-IST-1-506926)

all participating parties must have the same computational power. Therefore, in typical non-repudiation protocols three types of entities can be found: originators (O), recipients (R), and TTPs.

Several initiatives with respect to multi-party non-repudiation protocols [5], [8], [6], [9] coexist. All of them are theoretical studies. Using those basic construction elements, we have designed a protocol that is integrated into our DRM framework. It uses an intermediary and allows fair exchange of evidence in the RO acquisition phase [2].

III. The Protocol

Collecting, verifying and storing evidence about an electronic interaction is required, but might be operationally undesirable for final entities . Hence, *intermediary* entities are useful in such scenarios to help final entities to carry out their protocol exchanges. In addition, these entities can act as "hubs", increasing the market and opportunities not only for customers but also for merchants. It is clear that this philosophy matches the Mobile DRM approach in which the Mobile Network Operator serves as an intermediary entity, and users have direct access to the MNO and implicitly place certain degree of trust on it.

The MNO plays a critical role in this scenario, so it is important to analyze its behavior. As the MNO has interest (billing) in a transaction, it will be willing to reach a successful transaction. But occasionally, the MNO may collude with another (external or internal) entity and, for instance, hide some evidence. Therefore, we assume the MNO is not fully trusted. We presume that the MNO is not going to hide the initial message RORequest from the consumer to the RI. As the MNO communicates directly with the RI, it could help the customer in the non-repudiation protocol itself.

A. Protocol Description

The general notation used in the protocol can be found in Table I:

$A \rightarrow B : X$	entity A sends message X to entity B
$A \leftrightarrow B : X$	A fetches message X from B
X, Y	concatenation of messages X and Y
$S_P(X)$	digital signature of user P over message X
$h(X)$	one-way hash function with input X

TABLE I
General notation

More detailed notation for the protocol is as follows:

- $l = h(U, RI, MNO, TTP, t, RORequest)$: label of message $RORequest$
- t: a timeout chosen by the user U, before which the TTP has to publish some information
- $EOO = S_U(MNO, RI, TTP, l, t, PriceInfo, RORequest)$: evidence of origin of having sent $RORequest$, generated by U
- $EOO_{MNO} = S_{MNO}(RI, TTP, l, t, ROMNORequest)$: evidence of origin of $RORequest$ issued by the MNO for the RI
- $EOR = S_{RI}(MNO, l, t, ROMNOResponse)$: evidence of receipt of $ROMNORequest$ generated by the RI
- $EOR_{MNO} = S_{MNO}(U, RI, TTP, l, t, PriceInfo, ROResponse)$: evidence of receipt of $RORequest$ issued by the MNO for U and evidence of origin of $ROResponse$ at the same time
- $Con = S_{TTP}(MNO, RI, l, t, PriceInfo, ROResponse)$: evidence of confirmation issued by the TTP

The protocol is as follows. It is assumed that a flag is included in each signature to indicate the purpose of the message to be signed.

1) $U \rightarrow MNO$: $MNO, RI, TTP, l, t, PriceInfo, RORequest, EOO$
2) $MNO \rightarrow RI$: $RI, TTP, l, t, ROMNORequest, EOO_{MNO}$
3) $RI \rightarrow MNO$: $MNO, l, ROMNOResponse, EOR$
4) $MNO \rightarrow U, TTP$: $U, RI, l, t, RORequest, PriceInfo, ROResponse, EOR_{MNO}$
5) $All \leftrightarrow TTP$: $MNO, RI, l, PriceInfo, ROResponse, Con$

The protocol works in the following way:

1) U sends the MNO evidence of origin corresponding to the $RORequest$ message and $PriceInfo$ as obtained after browsing for rights. There is no breach of fairness if the protocol stops.
2) The MNO distributes U's information (maybe after a negotiation or agreement with the RI and after having prepared $ROMNORequest$ from user's $RORequest$) and sends to the RI evidence of involvement in the transaction. Again, fairness is maintained if the protocol is halted.
3) The RI replies with evidence of receipt of $RORequest$ together with the $ROMNOResponse$. It is assumed that a secure channel exists between the MNO and the RI. The protocol still remains fair if it stops, since none entity obtains what they expected. (U needs $ROResponse$ while the RI and the MNO need final evidence of the transaction performed). Note that $RORequest$ is uniquely identified in label l.
4) The MNO sends to U the Digital Rights Object ($ROResponse$) together with evidence of having received $RORequest$ and sends a copy to the TTP. U and the TTP will check all evidence carefully before proceeding to the next step. For U, this is the only evidence it will collect from the MNO and will be used in case of disputes to prove the MNO's responsibility of the exchange. The MNO will store the RI's evidence

[2]Although the requests and responses are XML signed in the DRM specification, this does not ensure fair exchange of items and thus it does not provide a complete non-repudiation service.

of receipt in its evidence database and U can retrieve it later if needed. The MNO cannot claim that it did not store this evidence since EOR_{MNO} demonstrates it did if a dispute arises. U and the TTP check:

- $l = h(U, RI, MNO, TTP, t, RORequest)$
- the info received is signed by the MNO in EOR_{MNO}
- $actual\_time < t$

If $ROResponse$ is the expected object (together with its associated price information), U does not really need to continue the protocol (as it got what it needed). Otherwise, i.e., if $ROResponse$ or the price information is not obtained or it is corrupted, it goes to the next step. The following step undertaken with an extra entity represents an addition with respect to the steps explained so far in the DRM scenario.

5) The TTP releases the confirmation message. U fetches $ROResponse$, $PriceInfo$ (if not satisfied in previous step) and Con as evidence of the digital right purchased. The MNO fetches Con as evidence that U received (or could fetch from the TTP) EOR_{MNO} and RO (and the corresponding charge) offered by the RI. The RI fetches Con as evidence to prove its origin. Note that if the MNO proceeds with the step 4 with $actual\_time > t$, it will gain no advantage. Furthermore, U could get RO without having to pay for it, as the TTP will not generate Con.

   On the other hand, if the MNO tries to cheat the TTP by changing the deadline, then it will obtain evidence Con which does not match with the rest of evidence collected. Thus, all entities are safe after the deadline time t.

At the end of the protocol, each party will hold the corresponding evidence.

- The user collects EOR_{MNO} and/or Con as evidence from the MNO.
- The MNO collects EOO, EOR, and Con as evidence of origin and evidence of receipt, respectively, which allows the MNO to demonstrate its good behaviour during the protocol.
- The RI collects EOO_{MNO} as evidence of origin of $RORequest$ issued by the MNO. Con must also be collected to complete the evidence.

This protocol takes only five steps and anonymity could be preserved, that is, unless the consumer is willing to communicate with a pre-selected Right Issuer, neither the consumer nor the Right Issuer needs any knowledge (i.e., digital certificates) about each other in order to reach a successful protocol end. This feature, preserves the anonymity property of our DRM framework, and can be used if the MNO is allowed to select different RIs (e.g., depending on trust deposited or price information).

B. Dispute Resolution

In our model, common disputes which might arise are depicted below. If the evidence has an expiry date, the disputes should be settled with the help of an arbitrator prior to that date. Entities (including the TTP) only store evidence during its lifetime, which usually will not exceed a month period (if bills are paid in a monthly manner).

Disputes between User and MNO

If the user receives a corrupted Right Object while already having paid for it but the MNO denies the fact, the user has to provide $ROResponse$, $PriceInfo$, EOR_{MNO} and/or Con to the arbitrator. The arbitrator will check the validity of label l, and also check that $(l, PriceInfo, ROResponse)$ is signed by the MNO in EOR_{MNO} or by the TTP in Con. If successful, the arbitrator determines that the MNO did not provide a valid Rights Object to the user.

If the MNO charges the user for a Right Object (embedded in $ROResponse$) but the user denies purchasing or receiving it, the MNO has to present EOO and Con to the arbitrator. The arbitrator will check U's signature on EOO (demonstrating its request) and the TTP's signature on Con. If successful, the arbitrator settles that U got $ROResponse$ (or could fetch from the TTP), and thus, the Right Object from the MNO.

Disputes between RI and MNO

If the MNO denies delivering message $RORequest$ (reformatted as $ROMNORequest$ from the MNO to the RI) to the RI, the RI presents evidence EOO_{MNO} and the arbitrator checks the MNO's signature on it. If successful, the arbitrator settles that $RORequest$, originated from U, is delivered by the MNO to the RI. If the RI denies having received message $RORequest$, the MNO presents EOR and the arbitrator checks the RI's signature on it. If successful, the arbitrator settles that the MNO delivered $RORequest$ to the RI.

The RI fetches Con to demonstrate the transaction was finished with the user. This is useful in case the RI charges the MNO depending on the number of successful Rights Object distributions.

IV. Design and Implementation

We briefly sketch the design and implementation of the system (see Figure 4). Firstly, we identify the different operations (either as processes or part of an API) to be implemented and describe in detail its functionality. Due to space restrictions, only the major operations are shown:

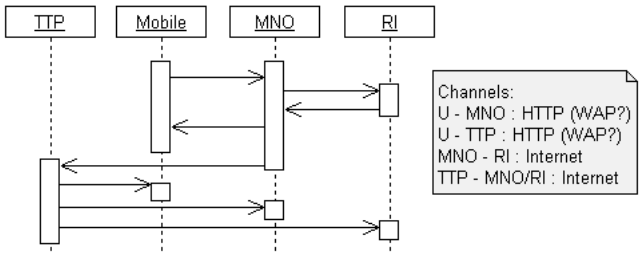

Fig. 4. HTTP Communication Flow

U - Mobile Phone User: The user manages the mobile phone, obtaining DRM services. The operations inside the mobile phone are:

- *(API)* ***ObtainROResponse***. Enter: $ROResponse$. Exit: $[ROResponse|Error]$.
  Internal Operation: Mobile phone negotiates with the MNO (sends *EOO* to the MNO and receives EOR_{MNO}) and with the TTP (fetches *Con* from the TTP), obtaining the rights inside $ROResponse$ together with the communication evidence.
  Side Effects: U **must** test and store EOR_{MNO} and/or Con as evidence of receipt. *Notes*: U contacts the TTP if EOR_{MNO} is corrupted or lost.

MNO - Mobile Network Operator: It provides service for rights acquisition, by contacting a TSP (Third-party Service Provider) that acts as a RI. The operations are:

- *(Process)* ***ManageRORequestFromU***. Triggered by: *EOO*. Halt: on $Error$.
  Internal Operation: The MNO receives *EOO* from U. It creates and sends EOO_{MNO} to the RI, receives *EOR* from the RI, and creates and sends EOR_{MNO} to U and the TTP.
  Side Effects: The MNO **must** test and store EOO and EOR. *Notes:* This process must have an interface to access global resources from the mobile network operator infrastructure, such as billing databases and evidence databases.

RI - Rights Issuer: It listens to $RORequest$ messages from other entities, and accesses the DRM Objects for obtaining an adequate $ROResponse$.

- *(Process)* ***ManageRORequestFromMNO***. Triggered by: EOO_{MNO}. Halt: on $Error$.
  Internal Operation: The RI receives EOO_{MNO} from the MNO. It calls the DRM $ROResponse$ Object with the $RORequest$ parameter. It sends *EOR* to the MNO.
  Side Effects: It **must** test and store EOO_{MNO}.

TTP - Trusted Third Party: It receives keys from mobile phone networks, and distributes them alongside with other evidence information.

- *(Process)* ***ReceiveKeyFromMNO***. Triggered by: EOR_{MNO}. Halt: on $Error$.
  Internal Operation: The TTP receives EOR_{MNO} from the MNO. After testing that the message has been received before the deadline t, it creates *Con* and stores it for later use.
  Side Effects: It **must** store message *Con* alongside with label l. Later, U, the MNO and the RI will fetch the message by using that label l.

Although the TTP is a separated entity from the MNO, a GPRS connection at network layer for contacting the TTP is possible as long as the http connection supports SSL. Nevertheless, in our first implementation we are considering a 802.11 connection (IP) with the TTP server, thus avoiding the flow of the protocol traffic through the MNO.

As we have already mentioned in Section II, digital signatures are the main tool for managing evidence. Nowadays, generating digital signatures with limited devices is not a restricting operation. For example, in our first tests, the mobile phone (model Siemens SX1) calculated all the cryptographic operations in 6 seconds.

For the implementation of the Mobile Phone system we are using J2ME-MIDP 1.0 [3] whereas for the rest of components (RI, MNO, TTP) we are using J2SE Java Programming and J2EE-Servlets in the server-side. Crypto operations are done (in both J2ME and J2SE/J2EE environments) with the Bouncy Castle Crypto Lightweight Library [4]. (There is an on-going standard for MIDP, JSR 219 [1], but it is not available yet.) Finally, for XML-processing in constrained environments, kXML (Lightweight XML library for mobile phones) is being used [2].

The protocol and implementation will be validated as part of the validation process of the UBISEC project. The validation perspectives are on a per-stakeholder basis and a weighted criteria will be used depending on the role. The validation criteria concerns the fulfilment of the requirements (omitted in this paper) and according to test cases previously defined. The evaluation results will be recorded by the Technical Team and according to the Evaluation Plan (D4.4, not released yet) and based on statical calculations for each validation criteria. Tables and statistical calculations have been defined.

V. Conclusions

As the technology evolves, content downloading will be an inexpensive operation. In order to protect Intellectual Property Rights, distributed DRM appears as a very good approach. Furthermore, DRM frameworks will be enriched by the implementation of security services from the very beginning. Non-repudiation is one of them.

We have designed a non-repudiation protocol for a DRM platform that takes into account all participants in the acquisition of rights, namely, the user, the Mobile Network Operator and the Rights Issuer, thus providing all of them with sufficient evidence to be used in case a dispute arises.

The implementation of the protocol is briefly sketched. It is designed such as to integrate with the Mobile DRM framework we are modifying from the OMA DRM standard. We are still in a test phase, and the necessary API has not been deployed yet. This is the main field in which we plan to continue our work.

Acknowledgement

The work described here is partially funded by the FP6-2002-IST-1 project UBISEC, contract number 506926. The first author has been funded by the Consejeria de Innovacion, Ciencia y Empresa (Junta de Andalucia) under the III Andalusian Research Plan, and the fourth author has been funded by the Ministry of Education and Science of Spain under the Programa Nacional de Formacion de Profesorado Universitario.

References

[1] JSR 219: Foundation Profile 1.1. http://jcp.org/en/jsr/detail?id=219.
[2] kXML. http://kxml.sourceforge.net/index.orig.shtml.
[3] Mobile Information Device Profile. http://java.sun.com/products/midp/.
[4] The Legion of the Bouncy Castle. http://www.bouncycastle.org/.

[5] S. Kremer and O. Markowitch. A multi-party non-repudiation protocol. In *Proceedings of 2000 International Conference on Information Security*, pages 271–280, Beijing, China, August 2000.

[6] S. Kremer and O. Markowitch. Fair multi-party non-repudiation protocols. *International Journal of Information Security*, 1(4):223 – 235, July 2003.

[7] S. Kremer, O. Markowitch, and J. Zhou. An intensive survey of fair non-repudiation protocols. *Computer Communications*, 25(17):1606–1621, November 2002.

[8] O. Markowitch and S. Kremer. A multi-party optimistic non-repudiation protocol. In *Proceedings of 2000 International Conference on Information Security and Cryptology*, LNCS 2015, pages 109–122, December 2000.

[9] J. A. Onieva, J. Zhou, M. Carbonell, and J. Lopez. A multi-party non-repudiation protocol for exchange of different messages. In *Proceedings of 2003 International Conference on Information Security*, pages 37–48, Athens, Greece, May 2003.

[10] Open Mobile Alliance. *DRM Specification*, 2nd edition, 2004.

[11] J. Zhou. *Non-repudiation in electronic commerce*. Computer Security Series, Artech House, 2001.

[12] J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proceedings of 1996 IEEE Symposium on Security and Privacy*, pages 55–61, Oakland, USA, May 1996.