Telecommun Syst (2007) 35: 161–176
DOI 10.1007/s11235-007-9050-4

Integration of non-repudiation services in mobile DRM scenarios

Jose A. Onieva · Javier Lopez · Rodrigo Roman · Jianying Zhou · Stefanos Gritzalis

Published online: 22 September 2007

Abstract In any kind of electronic transaction, it is extremely important to assure that any of the parties involved can not deny their participation in the information exchange. This security property, which is called non-repudiation, becomes more important in Digital Rights Management (DRM) scenarios, where a consumer can freely access to certain contents but needs to obtain the proper Right Object (RO) from a vendor in order to process it. Any breach in this process could result on financial loss for any peer, thus it is necessary to provide a service that allows the creation of trusted evidence. Unfortunately, non-repudiation services has not been included so far in DRM specifications due to practical issues and the type of content distributed. In this paper we analyze how to allow the integration of non-repudiation services to a DRM framework, providing a set of protocols that allows the right objects acquisition to be undeniable, alongside with a proof-of-concept implementation and a validation process.

Keywords Digital rights management · Non-repudiation · Secure electronic commerce · Mobile applications

J.A. Onieva (✉) · J. Lopez · R. Roman
Computer Science Department, University of Malaga, 29071 Malaga, Spain
e-mail: onieva@lcc.uma.es

J. Lopez
e-mail: jlm@lcc.uma.es

R. Roman
e-mail: roman@lcc.uma.es

J. Zhou
Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613, Singapore
e-mail: jyzhou@i2r.a-star.edu.sg

A. Gritzalis
Information and Communication Systems Security Laboratory, Department of Information and Communication Systems Engineering, University of the Aegean, Samos, Greece
e-mail: sgritz@aegean.gr

1 Introduction

One of the most influential aspects of the actual world is the capability to access to any kind of information from anywhere around the globe. Such exchange of knowledge is allowed by the existence of digitalized multimedia and the use of telecommunication networks. Still, not all the information generated and distributed through these networks has to be easily accessible by everyone. First, it is necessary to protect certain content, such as medical data, from unauthorized peers. Also, some content, like copyright-protected music or books, must be only consumed by users who paid for it. It is in this context where the Digital Rights Management (DRM) technologies can be able to offer a solution.

The basic services that these DRM technologies can offer are the protection of the digital information through encryption techniques, which can effectively hide the multimedia contents, the use of digital watermarks for copy detection, assuring that the original source of some data can be known, and the management of right objects (RO), which can provide access to the multimedia contents for those who have paid for it. It is not enough for these services to behave correctly, though: they must adapt themselves to behave correctly in all kinds of circumstances. For example, one of the most active fields in content distribution is downloading digital contents, such as ring tones and music, using a

mobile device. Due to the constraints associated to these devices and to the distribution networks, Mobile DRM has to be practical in terms of scalability and efficiency.

There are other important requirements in Mobile DRM scenarios, and without doubt non-repudiation services are at the top of the list. Non-repudiation is in charge of ensuring that no party can deny having participated in a part or the whole transaction. Without this service, there will be no evidence of malicious activities by any of the peers. Examples of these kind of activities are a service provider charging more money for a certain service, or a user stating that he/she did not bought a multimedia content. Unfortunately, this service has not been included so far in DRM specifications due to practical issues and the type of content distributed.

The purpose of this paper is to show how it is possible to create a simple protocol that can be integrated into an existing DRM architecture (in our case, a platform based on the OMA DRM specification 2.0 proposed in the European project UbiSEC), and illustrate how this protocol can be used for resolving any kind of dispute between all participants. This protocol allows the secure exchange of right objects (RO) between the consumer and the rights issuer through a mobile network operator, aided by a trusted third party. The usability of this protocol is proved by a proof-of-concept implementation using Java ME and Java EE technologies, and its correctness is assured by a validation process focused on the fulfilment of the requirements.

The contents of this paper are as follows: the DRM technologies are introduced in Sect. 2, alongside with the Mobile DRM requirements and standard initiatives. In Sect. 3, non-repudiation is introduced as a security service, describing the properties that non-repudiation protocols should provide and what the actual state of the art for DRM architectures. The protocol is then introduced in Sect. 4 presenting both a basic version and a extended version for extremely constrained devices. The reference implementation is presented in Sect. 5, and the validation process and a more complete resolution policy is shown in Sect. 6. Finally, conclusions are presented in Sect. 7.

2 Digital rights management

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. Summarizing, Digital Rights Management systems are the technological measures built into the hardware or software of any device for managing the relationships between users and protected expression [19].

The WIPO Copyright Treaty [9], recognized by at least 60 nations, refers to DRM as "technological measures" used to exercise rights and restrict unauthorized acts, and as the "copyright management information" needed to identify authors, rights holders and the terms of authorized use. So, DRM systems take three approaches to securing content. The first approach is "containment", the content is encrypted so that it can only be accessed by authorized users. The second is "marking", that consists on placing a watermark on content as a signal that the media is copy controlled. The third is "Separate Delivery", achieved by delivering the media and usage rights via separate channels, allowing a device to forward the content, but not the usage rights.

It is generally agreed that DRM involves different aspects: protection, such as copy protection or watermarking, information representation, e.g., metadata and rules, and the negotiation of the rights and agreements.

In order to improve the management of Rights in the Digital environment (Digital Rights), there is a need for a common language for DR representation. This kind of language is aimed to help building reliable networks where intellectual property rights can be managed in an open, global and adaptable form, so people can share, sell, buy, etc. content subject to DR, depending on their needs. A semantic approach seems a more flexible and efficient way of achieving these activities than a syntactic one.

Using metadata for referencing multimedia material is becoming more and more usual. This allows better ways of discovering and locating this material published in any kind of communication network. Several initiatives for establishing standards for metadata models are being carried out at the moment.

Currently, digital media commerce requires the integration of rights management systems with proprietary, often incompatible, back-end systems such as e-commerce management, customer relationship management, and asset management. In order to create interoperable digital commerce, including cross-system rights management, rights

holders and retailers need a set of standard business rules to define the parameters of media usage—for example, establishing that a piece of content be viewed a certain number of times per payment. Rights expression languages (RELs) are a means of expressing the rights of a party to certain assets and serve as standardized exchange formats for rights expressions. There are many initiatives around the standardization of DRM. Examples are ODRL, XMCL, XrML, and DPRL.

DRM concerns many stakeholders such as authors and publishers, consumers, libraries, schools and educational institutes, infrastructure providers, hardware and software manufacturers, government or standard bodies. Therefore, any DRM related research must take into account both, the complexity and the various stakeholders. Moreover, it is necessary to find the balance between the appropriate security and the protection of consumer privacy.

Different techniques are used in DRM systems. There are techniques to identify original content such as hash codes in digital files, watermarks in images and hidden sound codes in music files, and encryption to secure communication and distribution. For instance, *Copy protection* schemes attempt to find ways, which limit the access to copyrighted material and/or inhibit the copy process itself. Examples of copy protection include encrypted digital TV broadcast, access controls to copyrighted software through the use of license servers and technical copy protection mechanisms on the media. On the other hand *Copyright protection* inserts copyright information into the digital object without a loss of quality. Whenever the copyright of a digital object is in question, this information is extracted to identify the rightful owner. It is also possible to encode the identity of the original buyer together with the identity of the copyright holder, which allows tracing of any unauthorized copies. The most prominent way of embedding information in multimedia data is the use of digital watermarking [20].

2.1 Mobile DRM

Mobile DRM (MDRM) is a set of actions, procedures, policies, product properties, and tools that can be used to manage rights in digital contents according to requirements over mobile networks. A MDRM System tries to establish a trusted computing environment and trusted infrastructure. This infrastructure supports the secure preparation and transmission of protected digital contents. Additionally it prevents the misuse of the protected digital contents. Therefore a MDRM System must prevent illegal acts on the protected content, but also on the associated rights. But it also has to be practical in terms of scalability, simplicity, implementation / operation cost and efficiency. This is sometimes a challenge that has to be met.

The hardness of the challenge of course depends on the type of contents. Depending on the content MDRM can be classified into different groups [25]:

- *Rich* MDRM: The content managed by the MDRM system is rich media, such as video, e-books, which can only be consumed by high-end mobile devices. Both cryptographic and watermarking technologies are needed for protecting the contents and controlling the usage.
- *Light* MDRM: The content managed by the MDRM system is light media, such as ring tones, images, music, which can be consumed by medium-end or low-end mobile devices, like older mobile phones, whose platform is close. Cryptographic protection may not be necessary. Watermarking can be used instead. The device handles enforced usage.
- *Minimal* MDRM: No digital contents are attached. The digital-right itself claims the holder's rights to be served. The typical examples are e-Tickets and e-Coupon. The digital rights just have to be saved in a secure mobile wallet.

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:

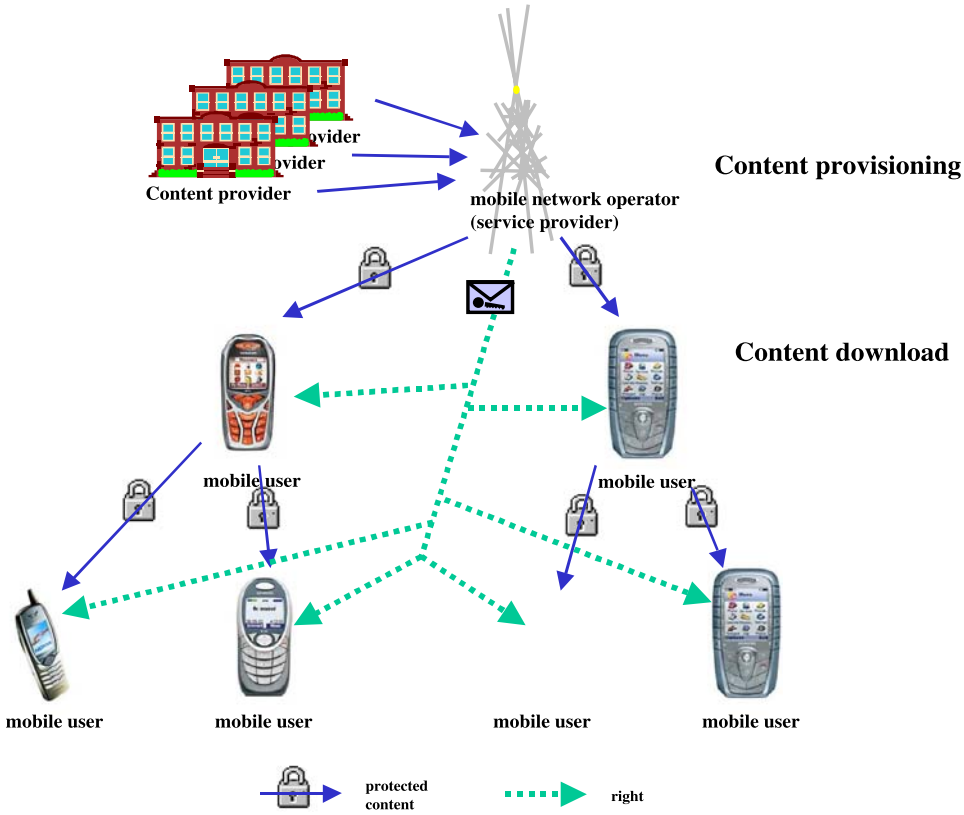
Centralized: A user needs to access the corresponding right from a central manager each time it wants to consume content. It is very effective against malicious users, but not so against malicious rights managers. Additionally, this approach suffers from scalability problems.

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).

One of the main DRM services today is downloading digital contents from a service provider. Protected contents, like films, music, ring tones, e-books, games, etc., are downloaded from the service provider to the mobile devices for consuming. The service provider obtains these contents from one or more content providers. In order to open the protected contents, the user needs to purchase a digital right from the service provider via mobile payment.

The right will be stored securely in his mobile device. With a correct digital right, the user can open DRM protected contents and consume the contents only with the help of the above said mobile device. The user can superdistribute the protected contents to other user's devices, what

Fig. 1 Content distribution



means peer-to-peer distribution among friends and communities. But similarly to the distributing user, these users will have to order digital rights for consuming the contents. The contents are DRM protected using either cryptographic methods and/or digital watermarking, no matter how they are distributed.

With respect to the DR management approach, the selected approach should allow users to access content when no connection to a central server is possible and, at the same time, it should allow industry to introduce a minimum number of changes to the existing business platform for distributing multimedia content in a secure (and right-protected) manner. 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 phone 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 Fig. 1).

Different standardization organizations and initiatives coexist for MDRM. The *third Generation Partnership Project* [6] is a collaboration agreement between a number of standardization organizations. It was established in December 1998. The main goal is to provide globally accepted and applicable technical specifications for third generation mobile communications (3G). 3GPP first planned to introduce a MDRM specification in their set of standards. A document manly containing requirements for enabling DRM was completed [21]. But in September 2002 the responsibility of 3GPP's MDRM standardization word was transferred to the Open Mobile Alliance.

The *Open Mobile Alliance* [8] was founded in June 2002 by the Open Mobile Architecture Initiative and the WAP Forum. The main goal of OMA is to introduce open standards and specifications based upon market and consumer requirements for the mobile industry. One of its specifications for the mobile industry is on MDRM. The OMA DRM 2.0 specification [18] introduces different methods for administering digital rights. One of then (and the most important for us) is *separate delivery*.

With the separate delivery method the content and the rights are delivered via separate channels to the mobile device. The content must be encrypted and converted into a special format, the DRM Content Format (DCF). A DCF object can only be accessed with the correct Content Encryption Key (CEK). This key is contained within the separately delivered right. With separate delivery the mobile device is allowed to forward the protected content, namely the DCF object, to other mobile devices. The rights containing the CEK can not be forwarded to other devices. To access the content the receiving device of a DCF object must request a new right containing the needed CEK. With this feature separate delivery enables the super-distribution of content.

Table 1 Existing MDRM systems

System	Architecture	RDL	DRM techniques	Comment	More info
NDS	Mobile/distributed	ODRL	Symmetric encryption	OMA DRM 1.0 compliant	www.nds.com
InterTrust	Hybrid/distributed		AES, DSA, SHA-1	supports MPEG-4	www.intertrust.com
Content guard	Centralized/hybrid	XrML	Digital signature, Hash, Watermarking	no superdistribution	www.contentguard.com
Coremedia	Distributed/mobile	ODRL		OMA DRM 2.0 compliant	www.coremedia.com

Other initiatives are IPMP (Intellectual Property Management and Protection), from the Moving Picture Experts Group [7], integrated in standards MPEG-4, MPEG 7, and MPEG-21. The IPMP extension do not actually standardize complete DRM systems. They just standardize the DRM interface which can be used by other DRM applications. Table 1 compares some of the existing mobile or hybrid DRM systems [19, 25].

After reviewing the existing products and initiatives, the UBISEC[1] consortium analyzed the common requirements and identified shared weaknesses to be overcome. Mobility is considered in the way that the client device for managing digital rights is a secure mobile device, which could in particular be a smart card. The secure mobile device is keeping the rights to execute protected content and connects (via an appropriate connection) to a MNO which in turn obtains the desired rights from a Rights Issuer and forwards them to the secure mobile device. The right may not be forwarded to other devices (as opposed to other proposals, which allows under certain circumstances to transfer rights to other devices). In contrast, protected content can be distributed without any restrictions, as no one is able to consume the protected content without the correct decryption key, anyway.

Anonymous purchase of rights is supported, as the Content Provider and Rights Issuer do not require privacy details of consumers. Consumer billing is performed through MNO to whom the consumer is subscribed. Contracts between network operators, rights issuers, and content providers have to regulate payments for content usage, but this is not in the scope of our specification.

Taking all this into account, we modified a platform based on the OMA DRM specification 2.0 for the distributed rights management. The modified scheme proposed in the European project UbiSEC enables a more secure framework for charging on the digital rights acquisition by the consumer, taking into account important issues as anonymity and efficiency (see Fig. 2).

In this architecture the user browses and downloads the desired content. The Content Provider supplies reference to the corresponding Right Object. Using this reference, the consumer will make use of his TPD for accessing the Right Object once he gets price and usage information. This basic use case can be seen in Fig. 3.

In our scheme, 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 Fig. 4). The MNO participation in this process is one of main changes introduced to the OMA specification. Detaching the user and the RI in the right acquisition process, we do not only instantiate the billing service provider but also introduce anonymity and push forward a required property (and often ignored): **non-repudiation**.

3 Non-repudiation in DRM architectures

As a security service considered in different layers of the security framework defined by ITU X.805 [11], 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. In this section, the analysis of this service for a DRM framework allows us to provide a solution which enables the right objects acquisition to be undeniable.

3.1 Non-repudiation: A security service

Repudiation is one of the fundamental security threats existing in paper-based and electronic environments. Dispute of transactions is a common issue in the business world. Transacting parties want to seek a fair settlement of disputes, which brings the need of non-repudiation services in their transactions. The motivation for non-repudiation services is not just the possibility that communicating parties may try to cheat each other. It is also the fact that no system is perfect, and that different and unexpected circumstances can arise in which two parties end up with different views of something that happened. Network failures during the protocol run is a representative example.

We define a *basic transaction* as the transferring of a message M (e.g. electronic goods, electronic cash or electronic contracts) from user A to user B, and represent this event with the following flow: $A \rightarrow B : M$. Thus, typical disputes that may arise in a basic transaction with a deadline T could be

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

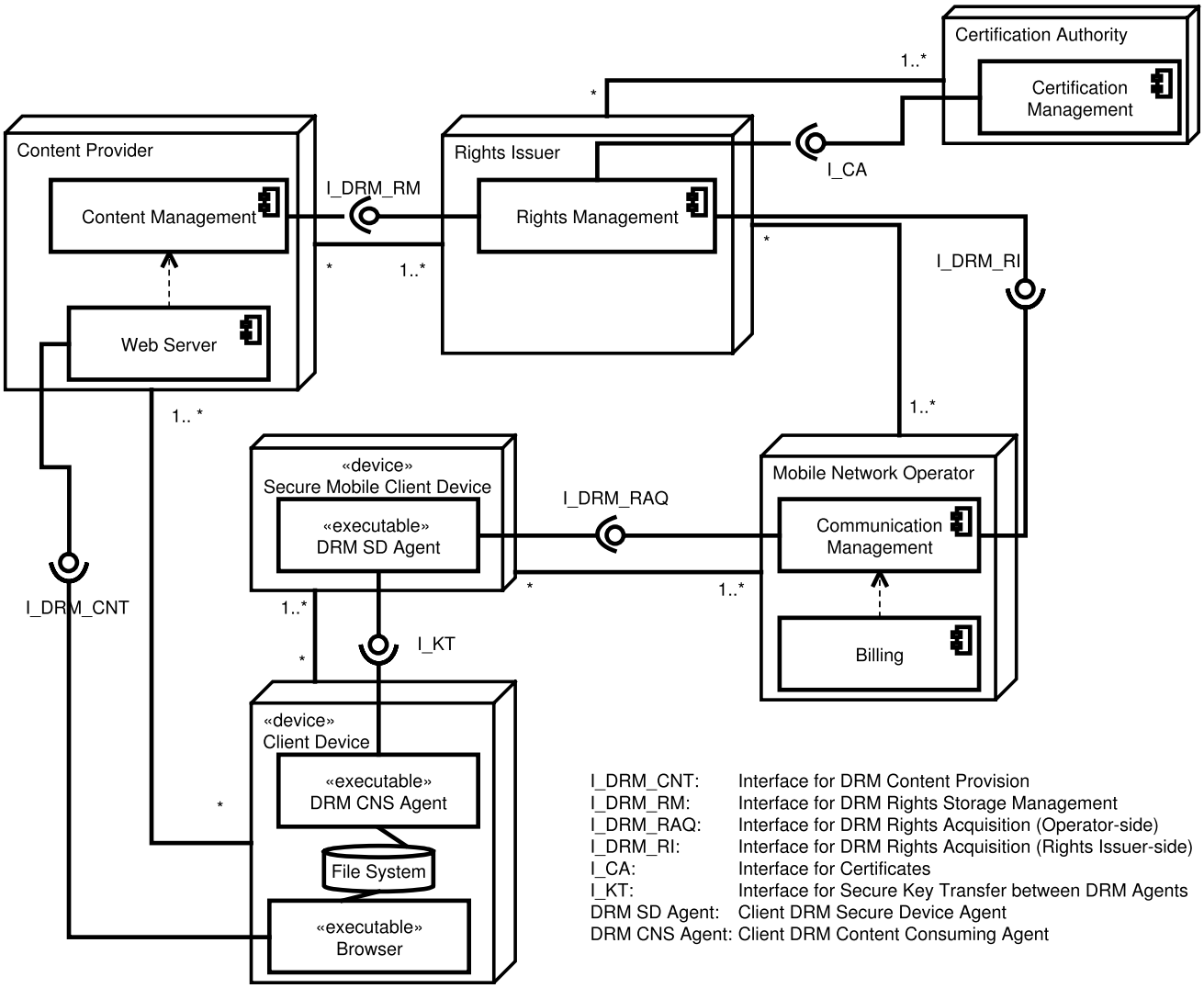


Fig. 2 UBISEC DRM architecture

- A claims that it has sent M to B while B denies having received it;
- B claims that it received M from A while A denies sending it;
- A claims that it sent M before T while B denies receiving it before T.

Non-repudiation must ensure that no party involved in a protocol can deny having participated in a part or the whole of the protocol. Therefore, a non-repudiation protocol must generate cryptographic evidence to support dispute resolution. In a typical non-repudiation protocol, a *trusted third party* (TTP) helps entities to accomplish their goals. Non-repudiation is especially important in electronic commerce to protect customers and merchants. It must not be possible for the merchant to claim that he sent the electronic goods when he did not. In the same way, it must not be possible for the customer to deny having received the goods.

Non-repudiation can be considered as an extended *fair exchange* problem in which non-repudiability is made an integral requirement of the exchange (in general it is not required). Exchange of one data item for another between mutually distrusted parties is usually the difficult part of an electronic transaction. We can find various instances of the general exchange problem in different types of commercial activities: a purchase, contract signing, certified mail or, more generally, in any barter conducted by means of digital networks. An exchange is said to be *fair* if at the end of the exchange, either each player receives the item it expects or neither player receives any additional information about the other's item. For instance, in payment protocols, fair exchange can ensure that a customer receives a digital good from a vendor if and only if the vendor receives payment from the customer.

Fig. 3 DRM

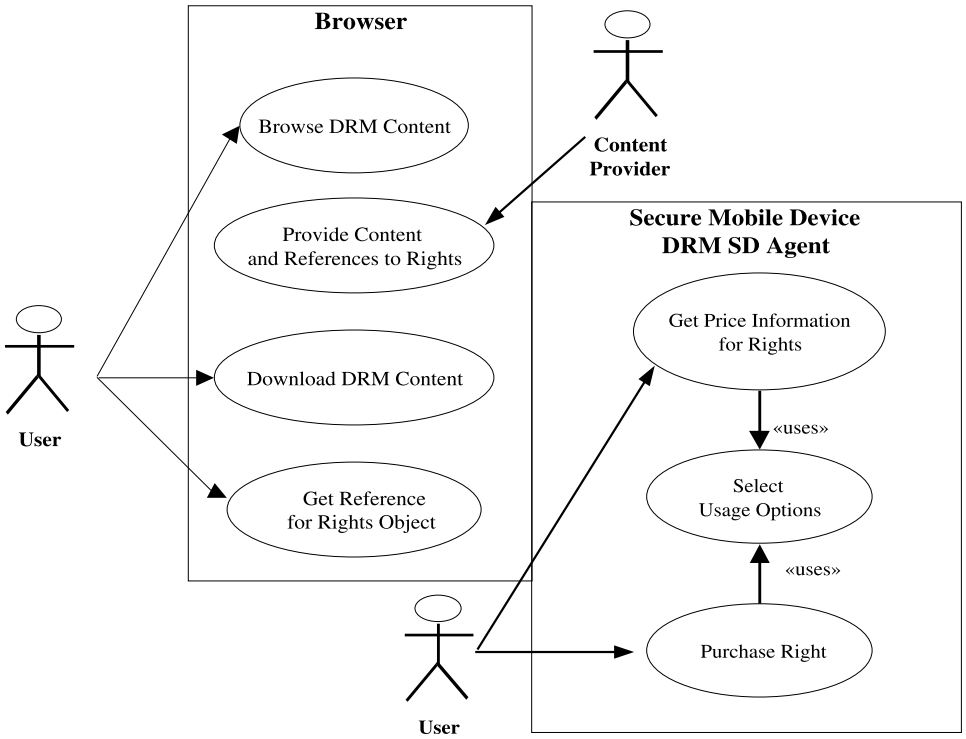
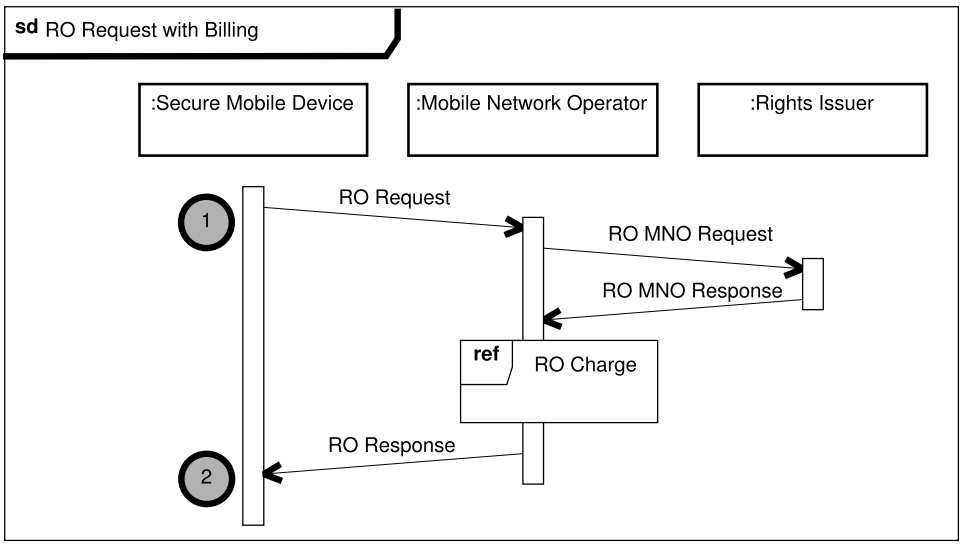


Fig. 4 Right object acquisition



For any non-repudiation service, evidence processed is a crucial object. There are different activities at each phase of processing. The non-repudiation policy defines the behavior of these activities. Finally, the eventual success of non-repudiation depends upon technical and legal supports. In order to achieve a non-repudiation service, some common phases have to appear in the protocol:

Service request. One or more parties involved must somehow agree, prior to its origination and delivery, to utilize non-repudiation services and to generate the necessary evidence for a non-repudiation service.

Evidence generation. Depending on the non-repudiation service being provided and the non-repudiation protocol being used, evidence could be generated by the originator, the recipient, or the trusted third party. The elements of non-repudiation evidence and the algorithms used for evidence generation are determined by the non-repudiation policy in effect and service request phase. Namely, evidence can be generated using secure envelopes or digital

signatures. The latter is more widely employed. A digital signature basically links a message with its originator, and also maintains the integrity of the message.

Evidence transfer. The evidence generator must transfer the evidence to the party who may ultimately need to use it. The principal participants may utilize trusted third parties to receive evidence.

Evidence verification and storage. Newly received evidence should be verified to gain confidence that the supplied evidence will indeed be adequate in the event of a dispute arising. The verification procedure is closely related to the mechanism of evidence generation. As the loss of evidence could result in the loss of future possible dispute resolution, the verified evidence needs to be stored safely. The duration of storage will be defined in the non-repudiation policy in effect.

Dispute resolution. This phase will not be activated unless disputes related to a transaction arise. When a dispute arises, an adjudicator will be invoked to settle the dispute according to the non-repudiation evidence provided by the disputing parties. The evidence required for dispute resolution and the means which the adjudicator will use to resolve a dispute are determined by the non-repudiation policy in effect.

A non-repudiation protocol generates at least the following important evidence for the participating entities:

Evidence of origin. This evidence is generated by the originator (perhaps with the assistance of a TTP) for a particular message and intended to the recipient, such that the originator cannot deny having sent that message.

Evidence of receipt. This evidence is generated by the recipient (perhaps with the assistance of a TTP) for a received message and intended to the originator, such that the recipient cannot deny having received that particular message from the originator.

In a typical two-party non-repudiation service, we identify several requirements, some of which could be optional, depending on the application the non-repudiation service is running over:

Fairness. A non-repudiation protocol provides fairness if neither party can gain an advantage by quitting prematurely or otherwise misbehaving during a protocol. At the end of the protocol either the sender gets evidence of receipt and the recipient receives a message as well as evidence of origin for that message or none of them gets any valuable item.

Timeliness. A non-repudiation protocol provides timeliness if any of the participating entities has the ability to reach the end of the protocol in a finite amount of time without loss of fairness.

Confidentiality. A non-repudiation protocol provides confidentiality if none but the intended parties can get access to the (plaintext) message sent during the non-repudiation protocol.

Several solutions to fair non-repudiation have been developed [14]. Some of them use a TTP which plays the role of a delivery agent 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 [26] where the TTP intervenes during each execution as a "low weight notary" rather than as a delivery agent. Other solutions use an off-line TTP, assuming that participating entities have no malicious intentions and the TTP does not need to be involved unless there is an error in the protocol execution. This is called an *optimistic approach*. There are also solutions that completely eliminate the TTP's involvement. However, they need a strong requirement: all involved parties must have the same computational power in *gradual exchange* protocols, or fairness depends on the number of protocol rounds in *probabilistic* protocols.

Previous work on non-repudiation in the literature was mostly focused on the two-party scenario. There has been some work with participation of several entities in related topics like *fair exchange*, where multiple entities exchange items among themselves without loss of fairness [3–5, 12]. Markowitch and Kremer extended the two-party non-repudiation scenario to allow one originator to send the same message to multiple recipients in a single protocol run [13, 16], whereas Onieva et al. extended this scenario for sending different messages to multiple recipients. The work done in this paper is based in [17], which presents a semi-trusted intermediary for multi-party non-repudiation, which helps final entities to collect, verify, and store evidence in electronic transactions. 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]

3.2 Non-repudiation in the UBISEC DRM architecture

Since the rights acquisition process means an exchange of money (or other valuable item) for rights via a mobile payment, evidence of the exchange needs to 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 the proposed architecture 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.

[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.

Considering the user as the customer which receives content and rights in order to be able to consume such content, non-repudiation is a valuable service for the customer in the last phase when it has to access the Rights 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 up in disputes among the parties. Such possible disputes could be as follows:

1. 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.)
2. The user receives a corrupted RO (ROResponse) while already having paid for it.
3. The user denies having sent a request (RORequest) for purchasing the RO.
4. The MNO denies having received a request from the user.
5. Similar disputes between the MNO and the RI.

From this list, 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.

Nevertheless, collecting, verifying and storing evidence about the digital right purchased might be operationally undesirable for mobile users. On the other hand, *intermediary* entities are useful in such scenarios to help final entities to carry out their protocol exchanges. It is thus clear that this philosophy matches our MDRM 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.

4 A non-repudiation protocol for mobile DRM frameworks

The context of the scenario where non-repudiation should be available was presented in Fig. 4, and it can be summarized as follows: the user (U) access a web server to know what Right Object (RO) should purchase in order to use a protected content. Later, that user contacts the Rights Issuer (RI) through the Mobile Network Operator (MNO), sending a request for rights (RORequest). The MNO is the entity that really interacts with the RI, obtaining the RO, sending it to the user (ROResponse), and billing him/her in the process.

All the interests of the stakeholders are fulfilled by the previous scenario. The user can easily obtain the adequate RO without providing the RI with any information, preserving his/her anonymity. The RI can obtain a benefit by billing the user through the infrastructure provided by the MNO. And the MNO can also benefit from this process by microcharging the user and the RI for every transaction. However, it is possible that one of the entities may collude with another for its own benefit. As a result, it is essential to create a protocol that can provide the previous functionality with non-repudiation capabilities.

Table 2 General notation

$A \rightarrow B : X$	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

The basic non-repudiation properties that such protocol should fulfill are fairness (no party has an advantage in the protocol by early quitting the process) and timeliness (the end of the protocol can be achieved in any moment without losing fairness). The protocol has also to provide enough evidence of origin and receipt to resolve any dispute between the participants. This evidence will be generated by all participants and by a Trusted Third Party (TTP).

In this section, we present two versions of a non-repudiation protocol for mobile DRM frameworks that fulfills all the requirements imposed by the scenario. The main difference between both versions are the use of asymmetric cryptography: As the user part of the protocol is executed inside a mobile phone, it is then necessary to be able to reduce the number of computational-intensive operations like public key signatures.

4.1 Basic protocol description

This protocol uses asymmetric cryptography on the client's side for the generation of evidence, although the protocol has been optimized to sign and verify as less as possible (only one signature and one verification). The general notation used in the protocol can be found in Table 2.

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 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).

4.2 Extended protocol description

The main critic on the basic protocol could come on the practical efficiency of having the user U producing digital

evidences with its mobile phone. We will see in next section that, at present, technology can handle the use of asymmetric cryptography in handheld devices. However, if very limited mobile devices are to be used, the concept proposed by Asokan in [1, Chap. 4, Sect. 2] of Server-Supported Signatures (S^3) can be integrated in our protocol using the MNO as a signature server. In such a way, users do not need to be able to produce digital signatures using asymmetric cryptography, but only need to be able to verify digital signatures which can be much easier (if, for instance, RSA is used with a small public exponent). At the same time, the user will need to generate hash chains which is computationally less costly than public key cryptography.

In this approach, the mobile user generates a secret key K_U, randomly chosen from the range field of its hash function. Based on it, U computes the hash chain $K_U^0, K_U^1, \ldots, K_U^n$ where

$$K_U^0 = K_U, \qquad K_U^i = h^i(K_U) = h(K_U^{i-1})$$

with h^i meaning applying i times hash function h. $V_U = K_U^n$ constitutes *U's root verification key* and will enable U to authenticate n messages. U submits this key to a CA for certification. A certificate for U's root verification key is of the form: $Cert_U = S_{CA}(U, n, V_U, MNO)$. Each MNO can easily access or acquire this type of user's certificate.

With initial $i = n$; i is decreased during each run and the steps of the protocol are slightly modified as follows:

– $EOO_{MNO} = S_{MNO}(feoo, RI, TTP, l, t, ROMNORequest, \mathbf{i}, \mathbf{K_U^i})$: evidence of origin of *RORequest* issued by the MNO for the RI

1. $U \rightarrow MNO : feoo, MNO, RI, TTP, l, t, PriceInfo, RORequest, i, K_U^i$
2. $MNO \rightarrow U : EOO_{MNO}$
3. $U \rightarrow MNO : K_U^{i-1}$
4. $MNO \rightarrow RI : feoo, RI, TTP, l, t, ROMNORequest, PriceInfo, EOO_{MNO}, K_U^i, K_U^{i-1}$
5. $RI \rightarrow MNO : feor, MNO, l, ROMNOResponse, EOR$
6. $MNO \rightarrow U, TTP : feor, U, RI, l, t, RORequest, PriceInfo, ROResponse, EOR_{MNO}$
7. $All \leftrightarrow TTP : fcon, MNO, RI, l, ROResponse, Con$

In the second step, when receiving the message from the user U, MNO verifies K_U^i based on U's root verification key and U's certificate obtained from CA, by checking $h^{n-i}(K_U^i) = V_U$. MNO has to ensure that only one evidence of origin is generated on behalf of user for a given (U, i, K_U^i). In this case, MNO records K_U^i as consumed and sends signature back to U. This *candidate* non-repudiation token is needed by U for demonstrating a possible MNO's misbehavior.

In the third step, U verifies the received signature and stores it. It also records K_U^i as consumed by replacing i by $i-1$. It also reveals K_U^{i-1} for providing definitive evidence. Note that U must consume each element in the hash chain in sequence and must not skip any of them. In particular, U must not ask for a signature using K_U^{i-1} unless it has received MNO's signature under K_U^i. Otherwise, MNO could use that to create a fake non-repudiation evidence, which U cannot repudiate during a later dispute.

Dispute resolution process, as explained in next subsection, is not modified but in case RI needs to demonstrate validity of EOO_{MNO} signature; i.e., MNO's signature on behalf of U, it provides it together with K_U^{i-1}. The arbiter will do the following (of course, checks are done by RI before sending step 5 as well):

- extract the root verification key from U's certificate
- as before, verify MNO's signature on EOO_{MNO}
- verify that K_U^i is in fact a hash of the alleged pre-image K_U^{i-1}
- verify that the root verification key V_U can be derived by repeated hashing $h^{n-i}(K_U^i) = V_U$.

If these checks are successful, U can still repudiate the evidence showing that MNO is cheating by showing a different signature EOO_{MNO} corresponding to the same K_U^i.

Although Asokan also proposed this approach for evidence of receipt, we do not consider it because generally, the RI will not be a resource-limited entity. At the same time, several other considerations show up, as for instance, how to avoid the need for the user of storing all evidences generated on its behalf by the MNO, which is an important consideration in limited-devices. This and other issues will not be considered by us (cf. Asokan's thesis REF for further reference).

4.3 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).

4.3.1 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.

4.3.2 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.

5 Design and implementation in a mobile environment

A proof-of-concept of the non-repudiation protocol was developed at the University of Malaga as part of the validation process of the UBISEC project. This design proved that the protocol proposed in Sect. 4.1 could be easily integrated inside the UBISEC DRM architecture without adding any major overhead. The foundation of the design is very simple: the communications between all entities are primarily managed by the protocol objects, which are in charge on creating the evidence and interacting with the functional components of the system where they belong, such as evidence databases, billing subsystems, and right acquisition subsystems.

In the design, all protocol steps are implemented as objects with a single method that invokes their functionality. The protocol messages can be received by the communications subsystem of every entity, and passed directly to these objects. All what these objects have to know about their environment is where to locate and how to invoke the other functional components through well-defined interfaces. A sketch of the overall message flow is presented in Fig. 5.

In the following, we describe in detail the internal functionality of the objects in the system. For interoperability purposes, all objects were implemented using the Java language.

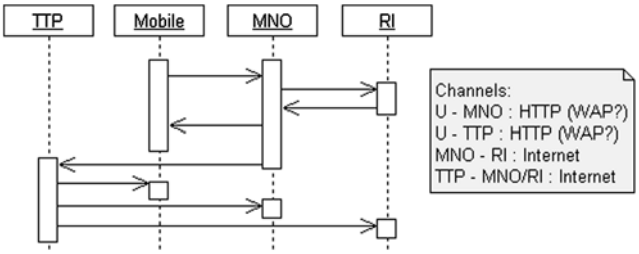


Fig. 5 HTTP communication flow

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

- *(Object)* ***ObtainROResponse***. Enter: *RORequest*. 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 the 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}. *Notes:* This process must have an interface to access the global resources from the rights issuer, such as content databases and RO creation subsystems.

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.

All operations in mobile phones used the J2ME MIDP 1.0 profile over the CLDC 1.0 configuration. The purpose of using this profile was to target a broader range of mobile phones, showing that the application could be used even by older phones in the market at the time of the validation process, such as the Siemens SX1. A side effect of the use of the MIDP profile was to demonstrate the viability of using the application in other mobile platforms, such as Pocket PCs powered by Windows Mobile 2003 Phone Edition using the J9 IBM Java virtual machine.

On the server side, all entities (RI, MNO, TTP) used the Java SE APIs and the Java EE Servlet technology for implementing the webservices. In our first implementation, we considered an IP connection (802.11) with all involved entities. Unsecured HTTP connections were used initially because confidentiality was not one of the primary objectives of the implementation (cf. Sect. 4.1), and the protocol was still protected against message replay attacks (due to the RORequest structure) and integrity attacks (due to the evidence attached in every message). Nevertheless, when the client contacts the TTP through a GPRS connection, it must then use SSL for avoiding "Man-In-The-Middle" attacks performed by the MNO.

The mobile phone had also to be able of using computational-intensive and memory-intensive operations such as XML management and cryptographic primitives. For XML-processing in constrained environments, kXML (Lightweight XML library for mobile phones) was used [15]. On the other hand, all cryptographic operations were done (in all environments, mobile and server) with the Bouncy Castle Crypto Lightweight Library [22]. Generating and verifying the digital signatures with limited devices was not a restricting operation: our testbed (mobile phone model Siemens SX1) calculated all the cryptographic operations required by the protocol in 6 seconds.

6 Protocol validation

Both the protocol and the implementation were validated as part of the validation process of the UBISEC project. The validation perspectives were on a per-stakeholder basis, and concerned the fulfilment of the requirements according to the test cases defined in the project. The results of this process proved that the protocol and its associated proof-of-concept implementation correctly fulfilled the fairness and timeliness conditions, and also provided a more comprehensive list of policy rules for evidence management and dispute resolution. These conclusions are presented in more detail on the following paragraphs.

- *Strong Fairness*: As it has been depicted along the description of the protocol, each party is in possession of proper evidence and no party is in an advantageous position during a transaction even if it aborts it. As the TTP acts in a lightweight online manner, the communication channel from participants to the TTP has to be resilient. Note that, even if MNO and RI collude, fairness in the protocol is preserved, since they will need to present to a digital arbiter the possession of non-repudiation of origin signed by the user.
- *Confidentiality*: This requirement is not needed in this protocol. Still, at the implementation level, it was possible to use protected channels such as SSL for allowing the secure exchange of messages.
- *Efficiency*: The protocol is not optimistic, but the TTP acts in a light-weight manner (i.e., receiving information, processing it and storing digital evidence in a network accessible directory with read-only permissions). Regarding the implementation, all performance-critical operations such as digital signatures were successfully handled by the mobile phone.
- *Timeliness*: The protocol fulfills synchronous timeliness by the use of a deadline.
- *Policy*: Again, policy about which arbiter to use in case of disputes, which cryptographic algorithms, etc., needs to be defined targeted to a mobile scenario. The minimum elements it needs to contain are those explicitly mentioned through the description of the protocol as, for instance, the guideline to use when entering a dispute resolution process. Nevertheless, we define here a more detailed policy considering the limited capabilities of user's device. For this task, we make use of the verbs "SHALL" "MUST" and "MAY" which are to be interpreted as described in [2].

  (a) Rules for evidence generation and verification: Cryptographic algorithms to be used. E.g. RSA with a public exponent 3. This will help limited-resource devices to easily verify digital signatures without reducing the security of the RSA algorithm if MNO is used as a digital signature server. If not, keys of 1024 bits are to be selected. Keys format and extensions needed if X.509v3 digital certificates are to be used for verification. The certificates (from MNO and TTP) will

be stored on user's mobile memory. Nowadays almost all phones are sold with preloaded digital certificates which can be on Wireless Transport Layer Security (WTLS) or X.509 format depending on the operating system implemented in the mobile phone. For our implementation we used X.509 formatted certificates, although for more limited devices, WTLS certificates can be used as well. WTLS defines a compressed certificate format which broadly follows the X.509v3 certificate structure, but uses smaller data structures.

The policy MUST define the language (XML) for representing them, and the hash functions to be used (SHA-1 which is still an alternative though its end is approaching [23, 24]). Of course, all legal restrictions coming from use of cryptography MUST be inherited.

Rules related to evidence generation:

– What evidence should be generated in the non-repudiation service?
  User *U* generates evidence of origin of the Rights Object request message. The Mobile Network Operator *MNO* generates evidence of U's origin of this request to the Rights Issuer *RI* and evidence of RI's receipt of the request (which serves as evidence of origin of the Right Object Response as well). RI creates evidence of receipt for the MNO.
– Which TTP should be involved in evidence generation?
  Explicitly stated.
– What elements should be included in the evidence?
  Explicitly stated.
– Which type of evidence should be generated?
  Explicitly stated.
– Which are the parties involved in the generation process?
  Each participant generates its own digital evidence (mobile user MAY obtain the support of the MNO for the generation of digital signatures in a verifiable way as explained in Sect. 4.2).

Rules related to evidence transfer:

– Which non-repudiation protocol will be used?
  Explicitly defined.
– Which are the channel assumptions?
  Since the TTP acts in an on-line manner, the channel between all entities and the TTP MUST be resilient for ensuring fairness.

(b) Rules for evidence storage: The database storage SHALL make at least an hourly backup. Only the TTP SHALL have write permissions and every input, query and operation in general MAY be logged (the format will be established by the TTP itself). Timestamps signed by the TTP for final evidence validity and logging purposes MAY be issued as well.

Regarding the mobile user, it MUST store all evidence received (EOR_{MNO} and/or *Con*) for solving future disputes. Nevertheless, because nowadays mobile technology supports the use of bluetooth connections and compacted memory cards, evidence can be off-line downloaded in other device with less storage constraints. If this is not possible, Asokan also suggested in his thesis an approach for transferring the necessity of storing digital evidence to the digital signature server (MNO) as well as an approach for using this entity as a time stamping server. In case of using this last modification to our protocol, the mobile user MUST store the hash chain corresponding to its root verification key certificate as well.

– What mechanism will be used for maintaining the validity of evidence?
  Expiry date.
– How long should the evidence be stored?
  Two months for expiration, since the mobile user will have an extra month (assuming monthly billing) for repudiating any Digital Right Object it has been charged for.
– Does the evidence need to be confidential?
  No.
– Which are the access control rules for accessing the final evidence (*Con*)?
  Write permissions for the TTP. Read permissions for users. The access control procedure MAY be established by the TTP. E.g. password-based. Authentication is not really important because evidence is not confidential and only valuable to intended users.

(c) Evidence use: Explicitly stated.

(d) Dispute resolution process: It has been already defined. Rules related to dispute resolution:

– Which entity will play the role of adjudicator?
  Explicitly stated in the protocol policy. If it is a digital arbitrator, X.500 DN (Distinguished Name) [10] can be used for uniquely referring to a specific entity. The user *U* MAY be involved in a dispute resolution process using other device different from its handheld device (for which an evidence transfer feature MUST be available).
– Which parties should be involved in dispute resolution? And which TTP?
  Only participants; i.e., U, MNO and RI.
– Which law should be referenced to enforce the arbitration?
  Depends on the country. E.g. LSSI in Spain.

– *Verifiability of TTP*: The TTP needs to be trusted, so it is not verifiable. On the other hand, MNO's behavior is verifiable and can be disputed by the originator and RI with the help of an arbiter.

– *Transparency of TTP*: As the TTP is lightly involved in every protocol execution, there is no transparency property.

7 Conclusions

Non-repudiation services are essential for the completeness of Digital Rights Management frameworks. Nevertheless, such services have been usually ignored, and only a few protocols have been developed till now. In the area of mobile DRM, this necessity is even more significant, but the solutions are more scarce. This has been the task of this paper: the creation of a non-repudiation protocol for mobile DRM frameworks that could be easily integrated into an existent architecture, and that takes into account the special requirements of the constrained architectures used in this kind of context.

The service provided by the non-repudiation protocol is effective, and fulfills the timeliness and strong fairness requirements needed by the applications. Such protocol 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. Moreover, the validation process include a detailed report on the policies to apply in case a dispute arises.

Finally, a proof-of-concept implementation is described. It is designed such as to integrate with the Mobile DRM framework modified from the OMA DRM standard in the course of the UBISEC project. The implementation was done in the Java language for interoperability purposes, allowing the program to be tested in mobile phones and other personal devices such as Pocket PCs. We may then believe that these non-repudiation services can be included as an additional service in actual devices, further improving the e-commerce and the satisfaction of customers.

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 third author has been funded by the Ministry of Education and Science of Spain under the Programa Nacional de Formacion de Profesorado Universitario.

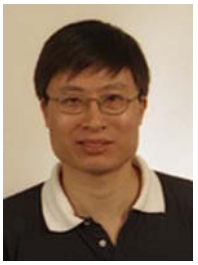
References

1. Asokan, N. (1998). *Fairness in electronic commerce*. PhD thesis, University of Waterloo, Computer Science.
2. Bradner, S. (1997). *RFC 2119. Key words for use in RFCs to indicate requirement levels*. IETF.
3. Franklin, M., & Tsudik, G. (1998). Secure group barter: Multi-party fair exchange with semi-trusted neutral parties. In *Lecture notes in computer science: Vol. 1465. Proceedings of financial cryptography 1998* (pp. 90–102). Springer.
4. González-Deleito, N., & Markowitch, O. (2001). An optimistic multi-party fair exchange protocol with reduced trust requirements. In *Lecture notes in computer science: Vol. 2288. Proceedings of the 4th international conference on information security and cryptology* (pp. 258–267). Springer.
5. González-Deleito, N., & Markowitch, O. (2002). Exclusion-freeness in multi-party exchange protocols. In *Lecture notes in computer sciences. 5th International conference on information security (ISC 2002)* (pp. 200–209). Springer.
6. http://www.3gpp.org/ (n.d.).
7. http://www.chiariglione.org/mpeg/ (n.d.).
8. http://www.openmobilealliance.org (n.d.).
9. http://www.wipo.int/treaties/en/ip/wct/index.html (n.d.).
10. ITU. (1997). *Information technology—open systems interconnection—the directory: Overview of concepts, models and services*.
11. ITU. (2003). *Security architecture for systems providing end to end communications*.
12. Khill, I., Kim, J., Han, I., & Ryou, J. (2001). Multi-party fair exchange protocol using ring architecture model. *Computers & Security*, *20*(5), 422–439.
13. Kremer, S., & Markowitch, O. (2000). A multi-party non-repudiation protocol. In *Proceedings of SEC 2000: 15th international conference on information security*. IFIP World Computer Congress (pp. 271–280).
14. Kremer, S., Markowitch, O., & Zhou, J. (2002). An intensive survey of fair non-repudiation protocols. *Computer Communications*, *25*(17), 1606–1621.
15. kXML (n.d.). http://kxml.sourceforge.net/index.orig.shtml.
16. Markowitch, O., & Kremer, S. (2000). A multi-party optimistic non-repudiation protocol. In *Lecture notes in computer science: Vol. 2015. Proceedings of 3rd international conference on information security and cryptology* (pp. 109–122). Springer.
17. Onieva, J. A., Zhou, J., Carbonell, M., & Lopez, J. (2003). Intermediary non-repudiation protocols. In *Proceedings of 2003 IEEE fifth conference on electronic commerce* (pp. 207–214).
18. Ope. (2006). *DRM specification* (2 ed.).
19. Plaza, P., Gonzalez, J. L., Lacoste, M., Stern, D., Bormann, F., Zoth, C., Tacken, J., Lopez, J., Onieva, J., Soriano, M., Forne, J., Marin, A., Almenarez, F., Görlich, J., Eikerling, H.-J., Müller, W., & Schäfer, R. (2004). *Mobile security: Requirements and state of the art analysis*. Technical Report D2.1, UBISEC Consortium.
20. Seitz, J. (2005). *Digital watermarking for digital media*. Hershey: Information Science.
21. Services, T. S. G., & Aspects, S. (2001). *3gpp s1-01 1197. ts 22.242*. Technical report, 3rd generation partnership project. V6.2.0.
22. *The legion of the bouncy castle*. (n.d.). http://www.bouncycastle.org.
23. Wang, X., Lai, X., Feng, D., Chen, H., & Yu, X. (2005). Cryptanalysis of the hash functions MD4 and RIPEMD. In R. Cramer (Ed.), *Lecture notes in computer science: Vol. 3494. Advances in cryptology* (pp. 1–18). EUROCRYPT, Springer.
24. Wang, X., & Yu, H. (2005). How to break MD5 and other hash functions. In R. Cramer (Ed.), *Lecture notes in computer science: Vol. 3494. Advances in cryptology* (pp. 19–35). EUROCRYPT, Springer.
25. Yan, Z. (2001). Mobile digital rights management. In L. Staffans & T. Virtanen (Eds.), *T-110.501 seminar on network security*. Helsinki: Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory.
26. Zhou, J., & Gollmann, D. (1996). A fair non-repudiation protocol. In *Proceedings of IEEE symposium on security and privacy* (pp. 55–61). IEEE Computer Society Press.

Jose A. Onieva received his M.S. in Computer Science in 2002 in the University of Malaga, Spain, actively collaborating with the Computer Science Department in a PKI-related funded project. Afterwards, he stayed as a "Research fellow" in Infocomm Research Institute (I2R), Singapore, during a year attachment, period in which initiated a research in the areas of non-repudation, mobile agents and P2P. Funded by the Junta de Andalucia government, he joined again the Security Group of the Computer Science department at UMA where he received his PhD degree in 2006. Among other activities he has been involved in the IST European project from the VI Programme Framework—UBISEC (Ubiquitious Networks with a Secure Provision of Services, Access and Content Delivery) and has actively collaborated in security-related National funded projects. He has published several international journal and papers in the field of Security for Information Technologies and serves as a PC member of several international conference committees. Currently, he is an Assistant Professor at the University of Malaga.

Javier Lopez received his M.S. and Ph.D. in Computer Science in 1992 and 2000, respectively, from University of Malaga. From 1991 to 1994 he worked as a systems analyst in the private sector, and in 1994 he joined the Computer Science Department at the University of Malaga, where he currently is an Associate Professor. His research activities are mainly focused on network security and critical information infrastructures, leading some national and international research projects in those areas, and being the Technical Coordinator of EU FP5 project CASENET as well as UMA's principal investigator of FP6 projects UBISEC and SERENITY. Dr. Lopez is the Co-Editor in Chief of Springer's International Journal of Information Security (IJIS), member of the Editorial Boards of Information Management and Computer Security Journal (IMCS), International Journal of Internet Technology and Secured Transactions (IJITST), International Journal of Computational Science (IJCS), member of the Advisory Board of International Journal of Smart Home (IJSH), and Spanish editor of the European Critical Information Infrastructure Protection Newsletter (CIIP). Additionally, Dr. Lopez is the Spanish representative in the IFIP Technical Committee 11 on Security and Protection in Information Systems, a member of the Steering Committee of ERCIM's Working Group on Security and Trust Management, a member of the Spanish Mirror Committee JTC1 of ISO, and Chair of the IFIP Working Group on Trust Management.

Rodrigo Roman is a doctoral student working in the Department of Computer Science at University of Malaga. He obtained its Master in Computer Science in 2003, and stayed afterwards as a "Research fellow" in the Institute of Infocomm Research (I2R) in Singapore for a year and a half. He focused his research on Sensor Network Security, which is nowadays his primary research area. At present, he is also working on Security for Ubiquitous and Pervasive Computing and Security for small devices.



Jianying Zhou is a senior scientist at the Institute for Infocomm Research (I2R), and heads the Network Security Group. He is also an adjunct senior scientist in University of Malaga, an adjunct professor in University of Science and Technology of China and in Shanghai Jiaotong University. He received PhD in Information Security from University of London in 1997. He worked in China (CAS), Singapore (NUS and KRDL) and USA (Oracle) before joining I2R. He is actively involved in the academic community, having served over 80 times in international conference committees as general chair, program chair, and PC member, and published over 100 referred papers at international conferences and journals. He is a leading researcher on non-repudiation, and authored the first book on this topic "Non-repudiation in Electronic Commerce" which was published by Artech House in 2001. He is a co-founder and steering committee member of International Conference on Applied Cryptography and Network Security (ACNS) and a co-founder and coordinating editor of Cryptology and Information Security Series (CIS) published by IOS Press.

Stefanos Gritzalis holds a BSc in Physics, an MSc in Electronic Automation, and a PhD in Informatics all from the University of Athens, Greece. Currently he is an Associate Professor, the Head of the Department of Information and Communication Systems Engineering, University of the Aegean, Greece and the Director of the Laboratory of Information and Communication Systems Security (Info-Sec-Lab). He has been involved in several national and EU funded R&D projects in the areas of Information and Communication Systems Security. His published scientific work includes several books on Information and Communication Technologies topics, and more than 140 journal and national and international conference papers. The focus of these publications is on Information and Communication Systems Security. He has served on program and organizing committees of national and international conferences on Informatics and is an editorial advisory board member and reviewer for several scientific journals. He was a Member of the Board (Secretary General, Treasurer) of the Greek Computer Society. He is a member of the ACM and the IEEE. Since 2006 he is a member of the "IEEE Communications and Information Security Technical Committee" of the IEEE Communications Society.