Identity Based DRM: Personal Entertainment Domain

Paul Koster, Frank Kamperman, Peter Lenoir, and Koen Vrielink

Information and System Security Department, Philips Research,
Professor Holstlaan 4 (WY71), 5656AA Eindhoven, The Netherlands
{R.P.Koster, Frank.Kamperman, Peter.Lenoir,
Koen.Vrielink}@Philips.com

Abstract. Digital Rights Management (DRM) enforces the rights of copyright holders and enforces their business models. This imposes restrictions on the way users handle content. These restrictions apply specifically in networked environments. Authorized Domain (AD) DRM concepts remove, or at least reduce, several of these restrictions to a large extent, while at the same time taking into account the content providers' need to limit the proliferation of content. In this paper we describe the design and operation of an Authorized Domain system, which we call the Personal Entertainment Domain (PED).

Keywords: Digital Rights Management, DRM, Authorized Domain.

1 Introduction

The concept of Authorized Domain (AD) Digital Rights Management (DRM) [1-3] aims to fulfill the requirements of both the content owners and the users (see section 0), which often appear to be conflicting. For an AD the general idea is that content can flow freely between the devices that belong to the domain, while content transactions between ADs are restricted.

Companies [4-9] and standardization bodies such as DVB (Digital Video Broadcasting) [1] and OMA (Open Mobile Alliance) are investigating and developing the concept of Authorized Domains. Up until now people have taken a device-oriented approach [2], where an AD groups a set of devices that belong to a certain household.

We conducted a study on alternative approaches to device-based AD concepts; these provide better solutions to enable the user to access content anywhere, at any time and on any device. The outcome of that study was that the Personal Entertainment Domain (PED) AD concept was the most promising candidate and this paper therefore presents a realization for such a PED-DRM system. PED-DRM does not have many of the disadvantages of device-based AD and it also represents a feasible solution for the near future.

PED-DRM is characterized by its structure, i.e. the relationship between various entities such as content, devices and persons, and by its policy, i.e. the rules that govern content access and proliferation. The PED-DRM structure is characterized by the fact that one single person is the member/owner of the domain, that content is bound to that person and that a number of devices are bound to the user (see fig. 1).

J. Dittmann, S. Katzenbeisser, and A. Uhl (Eds.): CMS 2005, LNCS 3677, pp. 42 – 54, 2005.

The PED-DRM policy is characterized by the fact that domain content can be accessed on a set of permanent domain devices without user authentication, allowing convenient content usage at home, including the sharing of content among family members. The only thing people must do is to register their device once to their domain. On all other compliant devices content can be accessed temporarily after user authentication, enabling people to access their content anywhere and at any time. Devices may be a member of multiple domains, both permanent and temporary. This paper presents the architecture and design of a PED-DRM realization together with a trade-off of alternatives and an overview of the requirements and threats for DRM domain concepts in general.

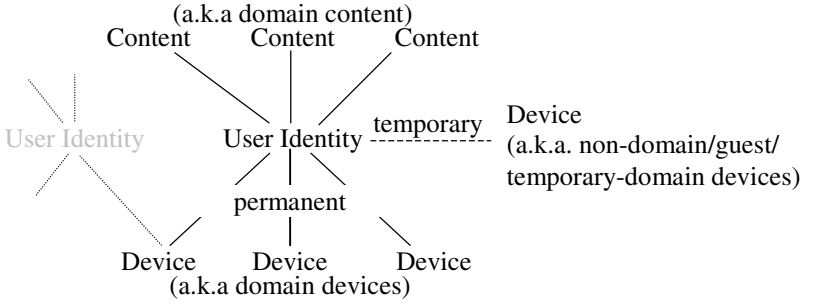

Fig. 1. PED-DRM concept

The outline of this paper is as follows. Section 2 introduces some typical scenarios and requirements for PED-DRM. Section 3 discusses the threat model and attacks. In section 4 we describe the functional design of the PED-DRM system. Section 5 elaborates on the main PED-DRM operations. The paper ends with an overview of related work and conclusions.

2 Scenarios and Requirements

The following scenarios demonstrate some of the typical PED-DRM functionality and the expected user experience and interaction with the system. The upcoming sections elaborate on the technical realization of these scenarios. In the scenarios that follow we assume that a user has a user identity device, such as a smartcard or mobile phone, with which he can authenticate – preferably wirelessly – to other devices. *Access family content at home:* A user operates his media center (a.k.a. Personal Video Recorder) connected to his TV in the living room. He selects a movie his wife bought using the remote control and presses play. The content starts to render (see fig.1: relation between permanent device, user identity and content). *Content access at remote location (guest access):* A user arrives at his hotel and decides he wants to render some content stored on his media center at home. He authenticates to the hotel TV and the TV lists the available content. He selects some content he bought some time ago and renders the content, which is streamed from his home (see fig.1: relation between user identity, temporary device and content). *Device registration to domain:*

A user buys a new TV for his bedroom. He holds his user identity device close to the TV and a menu pops up, asking him if he wants to register the TV to his domain. He confirms the registration using the TV remote (see fig.1: relation between user identity and permanent device). *UI (User Interaction) limited device registration to domain:* A user buys a cheap new music player with flash memory, a USB and preferably a wireless near field communication interface, but no screen. He holds the music player and personal authentication device close to his TV, which then displays the option to register the music player to his domain. He confirms the operation using the TV remote. *Device deregistration from domain:* A user decides to replace his media center with a better one he has just bought and gives his old media center away to his friend. Before giving it away he holds his user identity device close to his old media center and deregisters it using the media center remote and the menu shown on the connected TV. The application on the TV suggests he moves his content and licenses to another device so that he can continue to enjoy his content.

For the PED-DRM design we assume a number of requirements for the various stakeholders of the system. From the point of view of the content provider, the realization of the PED-DRM concept should meet the following requirements: (1) limit proliferation of content that has not been paid for; (2) limit damage in the case of hacked devices; (3) provide support for renewability/revocation of hacked/non-compliant devices; (4) support tracing of devices to facilitate revocation if devices malfunction. The rationale behind these requirements is that the content provider's business model must be sustainable and not break down in the event of an incident with the DRM system that governs the content. From the point of view of the user, the following requirements should be met: (1) in his role as domain and device owner, the user must have control over his domain and devices, i.e. no undesired (de)registrations of his devices to his domain; (2) DRM and domain functionality should work for devices that have limited user interface capabilities; (3) the conceptual complexity for the user must be low, e.g. the user needs to have an overview of his domain and related actions; (4) the solution should be robust, e.g. automatically maintain a consistent state as far as possible; (5) it must be possible to remove broken/offline/stolen devices from the domain. The rationale behind these user requirements is that the user must have maximum control over his devices and content while still not being bothered too much by procedures and technicalities in daily use.

3 PED Threat Model and Attacks

Since PED is a DRM system with a domain concept centered around a user, the typical threat model for DRM systems applies to PED. The DRM threat model assumes that users may be malicious and will attempt to gain unauthorized access to content. To accomplish this goal, the attacker has full control of his local environment, including network and devices, although it is assumed that compliant devices have some form of tamper resistance. Malicious users may use compromised and circumvention devices and software. However, we assume the average attacker has limited computational resources to break cryptography, has only limited capability to disrupt external network communication outside his local environment

and does not have access to professional tools. That said, it is possible that there could be a small number of attackers with the skills, technology and resources to perform such attacks. We continue with a number of attacks with a focus on domain- and person-based aspects, because general DRM aspects are assumed to be known [10].

Active attacks on the realization of the domain concept, i.e. grouping of devices according to a certain policy, include (1) malicious user interference with the domain management protocols in the local environment, (2) malicious user interference with license management and distribution in the local environment, and (3) malicious user interference in the distribution of device compliance status information through the heterogeneous and ad-hoc network of domain devices. Furthermore, there are some attacks that relate more to user behavior and the content owner's business models: (4) 'Content club': a large group of people share an account/identity/domain and obtain lots of content in such a way that it is accessible to all individual members, (5) 'Content cannibalization': realizations of the domain concept that include flexible limits, i.e. limits that can temporarily allow more devices to access content than intended, may be faced with domain extensions just before some premium content is released, (6) 'Content filling station': a rendering/storage device is loaded with domain content and then the device ownership is transferred, leaving the content available to the new and old owner for ever, (7) 'Automated domain or license management': intentional limitation or friction, such as mandatory user confirmation, could be frustrated if such operations are automated, making it seem as if a domain has no limitations.

Attacks that involve binding content to persons through licenses and allowing content access based on user presence include: (1) malicious users exploiting procedural processes for person management, e.g. users maliciously requesting replacement user authentication tokens, (2) malicious users exchanging, sharing, trading or cloning their identification and authentication credentials/tokens (3) simultaneous non-expired user authentication sessions that harm premium content releases. Since user behavior is non-technical, it is hard to detect and counter some of these attacks purely by technical means. The challenge is, therefore, to find the correct balance between attacks, threats, risks, countermeasures and user-friendliness.

4 Functional PED-DRM Architecture and Design

Figure 2 shows a functional and data view of the PED-DRM system. The shaded rectangles are data objects. The ovals above the data objects represent the typical PED-DRM functionality. The typical AD aspect of PED-DRM builds upon the user, device and domain management functions (fig.2, right). We have omitted most of the specific details of the DRM functions, e.g. content protection or license creation, because descriptions of these already exist [11;12] and because we have chosen to solve domain functionality independently so as to limit the effect on the traditional DRM functions (fig.2, left). The relation between rights management and domain management, i.e. the management of the set of permanent devices in the domain, is typically realized by means of a user identifier embedded in the license.

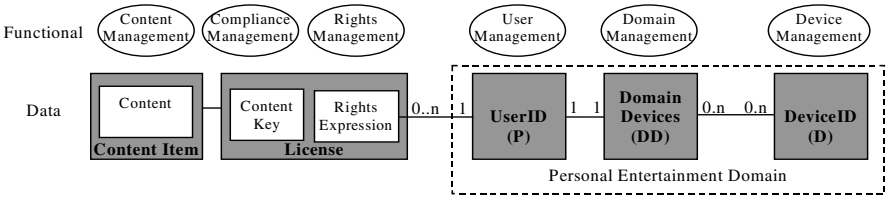

Fig. 2. PED-DRM DRM, identity and domain overview

User/Device Management

The main aspect of user management in PED-DRM is that a user is provided with a UserID certificate and corresponding public/private key pair. The user is not granted access to the private key in order to prevent him from misusing or giving away his private key and enabling someone else to impersonate him. To enforce this, the user's private key must be stored securely on a tamper-resistant user identity device, which also serves as a token that proves the user's presence. The user identity device (hardware and software) must be easy to handle, provide secure computing means and must be hard to clone. Typical solutions for this are smartcards and mobile phones equipped with a SIM card.

Devices in PED-DRM are given a DeviceID certificate. In addition to their identity, devices are also given explicit authorization to fulfill certain functions. This would limit the effects of a security breach by preventing the certificate and keys of a hacked device from being misused for other functions, e.g. keys from a rendering device cannot be used to register other devices to the domain.

The approach outlined for user management contains the most essential elements that provide a working solution. However, this rather straightforward approach also triggers a number of privacy issues. Solutions to this could be found in privacy-enhancing technologies [13], such as pseudonymity services, and use of solutions that for example rely on roles or assertions without revealing the identity involved.

Domain Management

According to the model in figure 2, domain management in PED-DRM concerns the relation between a UserID and a number of DeviceIDs, which is characterized by a DomainDevices (DD) data object. We propose an approach in which DD is a certificate containing a reference to the user of the domain, references to a number of devices, a version number and the signature of the domain manager (ADMCore in fig.3).

$$DD = \{ \text{DomainID, Version, UserID, DeviceID1, …, DeviceIDn, SignDM} \} \quad (1)$$

The first advantage of making DD a certificate is that it shows who issued it. The second advantage of putting all domain members in one certificate is that this allows a simple but secure signaling mechanism to show which devices are in the domain. This can be used effectively to inform deregistered devices and can be used efficiently to obtain and distribute revocation/authorization information. This synchronization mechanism is part of the secure authenticated channel setup, as explained later, which makes it possible for the system to function even when not all devices are online and

reachable. The third advantage of the DD certificate is the ability to report domain information to the user on any domain device at any time.

Domain-based DRM systems often base their security on domain key(s) [4;6;7]. In these systems the content key is typically encrypted with the domain key. This has security advantages if devices are hacked because the accessible content to these devices is limited to the domain content. PED-DRM addresses this threat by limiting license distribution to permanent and temporary domain devices, realizing a similar level of security. PED-DRM could also be extended with a domain key.

System Components and Their Interaction

Figure 3 presents the main components – ADMCore, ADClient, UserIdentity and ADMTerminal – that group PED-DRM functionality and the interaction between them. We defer the descriptions of these interactions to section 5 of this paper. The typical connectivity means that enable interaction between the components are also indicated: combined on the same device (local), connected through a network (IP) or via wired/wireless connection with a strict limitation on the distance (e.g. Near Field Communication (NFC) [14]).

The ADMCore, ADClient and UserIdentity must run on a compliant device which has a DeviceID certificate because they manage domain or content-related data. These components must have a compliant implementation, which means that their implementation is subject to robustness requirements. It is beyond the scope of this paper to go into detail about the robustness and implementation rules for each type. The ADMTerminal component is only responsible for UI and control aspects and ADMTerminal is therefore not subject to DRM compliance requirements.

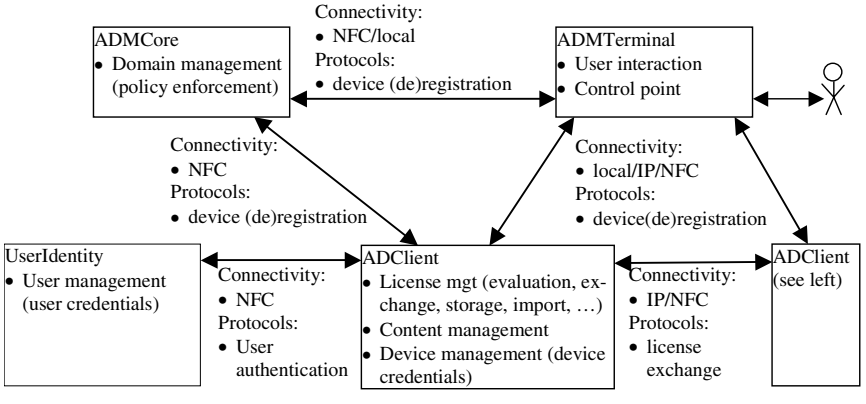

Fig. 3. PED-DRM components and their interaction

With respect to deployment of components over devices and taking into account the characteristics of the PED-DRM system, we foresee that UserIdentity and ADMCore components are combined on one device, e.g. on a smartcard or mobile phone, conveniently referred to as a user identity device in the scenario section. Alternatively, ADMCore runs as a service on the Internet, an approach similar to OMA DRMv2 and Apple's Fairplay. However, in the remainder of this work we

assume that ADMCore runs on a device and not in the network. Some characteristics of local domain management are exploited, such as proximity verification, in the knowledge that some other aspects of central domain management are given up, notably central control and easy and direct audit facilities. Ideally, the ADClient and ADMTerminal would also be combined on one device, allowing straightforward AD management operations using the user interface of the device for interaction with the user. Typical devices are hardware- and software-based media centers, connected renders (TVs), etc. Provision has been made for alternative forms of component distribution, e.g. portable devices that combine the ADMCore, ADMTerminal and UserIdentity.

PED-DRM Domain Policy

The domain policy specifies under which conditions entities are entitled to be part of the domain and thereby largely defines the scale of content proliferation in a domain-based DRM system. It is evident that end users prefer a policy with a relaxed regime, while copyright holders, content providers, etc., prefer more tight regimes. As in most other domain-enabled DRM systems, PED-DRM has a domain policy that is fixed for the system. An exception to this is OMA DRMv2, where the domain policy is left to the individual Rights Issuers. Components that can enforce (part of) the domain policy include the ADMCore and ADClient.

We propose a simple and straightforward basic domain policy enforced by ADMCore. The policy is based on a maximum number of devices per domain. So far, the policy is very similar to Apple's Fairplay limit of 5 authorized PCs. Furthermore, ADMCore only registers ADClients that are in direct proximity. This limits the domain size and content proliferation to places where the user goes. Devices may be a member of multiple domains to support sharing of content between people who share devices.

In addition to the basic domain policy, further measures may be required by the content owners, e.g. to further reduce some of the risks associated with the attacks mentioned earlier or to allow a higher number of permanent domain devices. To accomplish this, a balanced set of the following measures may be added to the domain policy. They should be selected with care so that in normal circumstances a user does not encounter them, keeping conceptual complexity low for end users. Note that it may be hard to estimate the resulting content proliferation when these measures are combined. Some examples are: 'membership liveness' meaning that domain membership stays valid for some time, but must be confirmed regularly by the ADMCore; 'rate-limited domain management' meaning that (de)registrations may not exceed a maximum level per time unit.

Options with respect to the domain policy enforced by the ADClient are: the number of domains of which a device may be a member, which is unrestricted in pure PED-DRM; the validity time of the membership; the rate at which the device may register to different domains. Domain policy enforcement by the ADClient may also interact with other non-domain management actions, such as license management, e.g. limiting the exchange of licenses to distance-limited channels, or user authentication, e.g. a maximum number of authentications over time.

5 PED-DRM Operations

We will now describe the protocols and processes of the PED-DRM that realize the main PED-DRM functionality in a secure and user-convenient way.

Device Registration
The protocol for registering a device starts when, under the user's control, the ADMTerminal instructs the ADClient to register itself at the ADMCore. The ADClient requests registration to the domain at the ADMCore with its DeviceID reference. The ADMCore responds to the ADClient with a new DD certificate, including the ADClient's DeviceID, if the device is registered successfully. The ADMCore processes the registration request using the following steps: it verifies the authenticity of ADClient's request (i.e. verify digital signature or use of secure channel), it verifies that the device with DeviceID is not revoked and is not already a member, it executes the PED policy algorithm to determine whether the device may be added to the domain and creates a new DD certificate with the DeviceID reference included and an increased version number. The ADClient verifies the validity of the DD certificate received, stores it and is subsequently able to render domain content based on possession of the DD certificate.

The device registration protocol supports two deployment configurations explicitly. The first is the trivial case where the ADClient and ADMTerminal are located on the same device with an NFC interface to the ADMCore. The second case concerns a device which has limited UI capabilities and an NFC interface to the ADMCore and which is controlled by an ADMTerminal running on another device in the network. Provision has been made for ADClients without NFC interface to be supported in the future using extensions to the protocol or policy.

The requirement that the device registration protocol should protect the interests of the user in his role as domain and device owner is triggered when domain management can be controlled over a network. Device owner consent can be addressed in two complementary ways. The first solution involves user confirmation at the ADClient for registration, e.g. the ADClient device is put in registration mode by pressing a button. This ensures that a remote ADMTerminal cannot add devices to a domain without confirmation at the physical location of the ADClient. The implementation should ensure that a user confirmation only relates to the intended registration session. The second solution comprises authentication of the device or domain owner when devices are registered remotely to prevent abuse such as registration by third parties on the same network. Domain owner consent is implicitly assumed, because the ADMCore must be in physical proximity of ADClient.

The requirement that the protocol must be robust and must inform the user of the status is supported by including acknowledgements, rollback procedures, and DD distribution in the protocol that prevents the ADClient from considering itself a domain member when the ADMCore does not. The ADMCore has to implement the device registration protocol as a transaction due to the fact that communication could fail, e.g. a smartcard stops functioning when it is removed prematurely from the reader. It must perform a rollback procedure when it is re-activated after removal. To speed up the distribution of the new domain composition and thereby contribute to the robustness and consistency for the user as well, the registered ADClient should

broadcast the latest DD certificate to other ADClients. The latter is a best-effort approach in addition to the forced synchronization as part of the secure authenticated channel required for the license exchange described later.

Device Deregistration
The device deregistration protocol starts when the ADMTerminal under the control of the user instructs ADClient to deregister itself at the ADMCore. The ADClient sends a deregister request with its DeviceID to the ADMCore, which responds with a new DD certificate that indicates the domain composition. If the ADClient is not available, e.g. if it has been stolen, broken or is offline, then the ADMTerminal may send the deregister request on its behalf. The ADMCore verifies that the DeviceID indicated in the deregistration request is listed in DD, it removes the device from the DD certificate and increases its version number. The ADClient performs the following steps for the deregistration response: it checks the validity of the DD certificate and if it is no longer listed it deletes the DD certificate. Before ADMCore replaces its stored DD certificate, it expects a deregistration confirmation from ADClient to ensure that ADClient received the request and deregistered itself. Unconfirmed deregistrations, including deregistrations of offline devices, may be administered differently and used in the domain policy for future device registrations, such that the protocol cannot be misused to allow for lots of new registrations while the old ones are effectively still present. The new DD certificate should be broadcasted so that other domain devices learn the new domain composition as quickly as possible.

The interests of the domain owner in the deregistration protocol are protected by the requirement that the ADMCore and ADMTerminal must be in close proximity or in direct contact with each other. The presence of the ADMCore implies authorization for the deregistration action. Alternatively, the requirement can also be met if the ADMCore is present near the ADClient that is removed, in combination with a user confirmation or explicit deregistration mode on ADClient to thwart unwanted deregistrations from the local network. The device should ensure that the confirmation or deregistration mode corresponds with the correct deregistration session. The implementation must ensure that the user may not be subject to a denial of service attack consisting of many confirmation requests.

The device deregistration protocol fulfills the requirement that devices must be removable from the domain. Stolen and offline devices are removed effectively over time when the new domain composition is distributed in the form of the DD certificate as part of the secure communication protocol discussed later.

ADClient Reset / Local Deregistration
A local deregistration is required when an ADClient needs to be de-registered from a domain but has no opportunity to communicate with its ADMCore, e.g. someone gives his device away without having his ADMCore in the neighborhood but still wants to prevent the new owner from accessing his content. In this case he needs to perform an autonomous ADClient reset action whereby the ADClient deletes the DD certificate. This approach should not be advocated because there is no automatic means to ensure that ADMCore will remove it from the DD certificate as well. A device should therefore indicate to the user that he needs to perform (offline) deregistration using ADMCore as well.

ADMCore Disaster

If the ADMCore device is broken or lost, a user is no longer able to change the composition of his domain. If the ADMCore is stolen, somebody else will be able to add his own devices and access the content belonging to the original owner, assuming that no additional access control mechanisms are in place. To mitigate this problem, a user requests a new ADMCore device. As part of this process the old ADMCore is revoked, with the result that the devices will no longer engage in AD management protocols with the old ADMCore and DD certificates issued by the old ADMCore will not validate correctly, since the old ADMCore is blacklisted. In effect, the old ADMCore and the old domain are revoked. The user needs to register all his devices to a new domain managed by his new ADMCore.

Secure Authenticated Channel and Revocation

The AD management and license exchange protocols require confidentiality, integrity, authenticity and protection against replay attacks; these are provided by a Secure Authenticated Channel (SAC). Although many general purpose SACs exist, e.g. TLS [15], PED-DRM has some specific features that are highlighted here.

An important aspect for PED-DRM is the exchange of DD certificates as part of the SAC setup phase. When a device receives a valid DD certificate with a higher version number than its stored DD certificate, it replaces the stored DD with the new DD, provided that it is still contained in the new DD certificate, otherwise it removes its DD completely. Inclusion of the DD certificate forces the DRM system to function correctly by ensuring that devices have an updated view of the domain composition. Based on this view, they must decide how they can exchange licenses with other devices and what kind of access they can allow to the content. The exchange of the DD certificate as part of SAC setup facilitates the update of a deregistered device that was deregistered from the domain while it was offline. The viral nature of the DD certificate distribution ensures that eventually the deregistered devices are no longer able to render any further domain content, except when a deregistered device no longer has any contact with its former domain members. The viral nature of DD certificate distribution is made more effective by requiring SAC usage for common operations, e.g. license exchanges, domain (de)registrations and user authentication.

A second important aspect of SAC setup is the support for device revocation. Devices only participate in domain management or license management interaction when the other party is still compliant. Lack of space limits us to only sketch the solution: we propose to use a scheme based on authorization lists, i.e. assertions proving that devices are still compliant, which also uses the viral nature of DD distribution to ensure that all active domain devices obtain fresh authorization/revocation status information related to a domain and user, even when some devices do not have global (Internet) connectivity.

User Authentication

The user authentication protocol consists of unilateral authentication based on a straightforward PKI protocol extended with proximity/presence assertions if necessary. Revocation must be supported on two levels: user identity and user identity device/token. The user identity should be revoked when the private key is compromised. User identity devices should be revoked when the device is broken,

lost or stolen. The user obtains a new UserId device and can use both his old and new content. An additional measure could be that for new content the old UserId device is specifically blacklisted in order to tackle cases where propagation of revocation status information may take some time. For both levels of revocation nothing needs to be done with either content or licenses. The organizational and infrastructural aspects of user authentication and identity management have been omitted here.

License Management
Distribution of licenses is straightforward in PED-DRM. The ADClients must exchange licenses with each other using the SAC. The properties of the SAC ensure that only compliant and non-revoked devices can obtain the licenses.

To reduce the effect in the case of hacked devices, licenses in a PED-DRM are only transferred to and stored on devices that are either domain devices or devices to which the domain user has been authenticated recently. It is preferable if the source device receives some proof of the user identity device via the target device such that it can be sure that the user identity device is or was in close proximity to the target device. A pure form of this approach implies that domain licenses are removed from devices upon deregistration or upon expiry of an authentication session, which might be impractical in some cases since it could unintentionally destroy the last domain license for a content item while not achieving any significant increase in security because the device already possesses the licenses. There is a minor drawback to this approach because licenses cannot be distributed upfront as they can when domain keys are used.

6 Related Work

To put PED-DRM into perspective we compare it with other network-oriented DRM systems. Due to limited space we have restricted ourselves to highlighting some advantages of PED-DRM over other systems. Of course, PED-DRM is not free of pre-requisites, e.g. its dependence on user identification and the hardware tokens and interfaces required for this, none of which are currently commonly used in consumer markets and products.

Person-based access to content at any time and in any place is one of the main advantages of PED-DRM over a number of systems that are device based and/or limit content exchanges to the local network, e.g. SmartRight [4], DTCP-IP [5] and Microsoft's WindowsMedia DRM (MS DRM) [16].

An advantage of PED-DRM (and also of SmartRight, for example) is that it separates domain management and domain policy from the license-issuing functionality, which enables a uniform user experience. This is the opposite from OMA DRMv2, where each rights issuer manages the domains for its content according to its own domain policy, which may be confusing for users buying content at different shops.

PED-DRM's approach is based on the equality of rendering/storage devices (ADClients), which is easy to understand for end users. Current systems, such as Apple's Fairplay or MS DRM, put devices in different classes such as PCs, portables, extenders [17], etc. The complicating factor is that policies vary for each device class,

e.g. number of devices allowed per class, permitted functions such as rendering/storage per class, and whether or not a device may further distribute content and licenses to domain devices.

When the relation between persons change, it is quite easy for them to divide up their music in PED-DRM because each person binds his content to his user identity. Solutions such as SmartRight make it more difficult to do this because they are device based and, furthermore, only allow one domain per environment.

7 Conclusions and Discussion

In this paper we have discussed a design and operation of the PED-DRM concept in which (1) content is linked to persons, (2) a person has a number of permanent domain devices, and (3) where content can be rendered on the permanent domain devices or (4) on arbitrary (compliant) devices after authentication. The main characteristics of the design are the seamless access to content on devices that are a member of the domain. Authentication is taken care of by means of a personal smartcard or a device that can act as a user identity device like a mobile phone with a SIM. These two characteristics allow the person of the domain to enjoy his content at any time, anywhere and on any device. It is also possible to share content with relatives or friends by sharing content access devices that can be a member of multiple domains. However, due to the domain policy, which states that the number of devices in a domain is limited and that content can only be accessed on non-domain devices if the owner of the content is in close proximity, the proliferation of the content is still controlled strictly.

The user requirements have been taken into account in the design of the PED management protocols. For example, users have explicit control over what devices are added and/or removed from their PED. The use of close proximity technologies and devices (smartcards) makes the system as user friendly as is possible with the current technology.

The attacks relating to the domain concept realization and person-based content access are addressed by robust protocol and device design and by an appropriate domain policy. With respect to the attacks that deal with user behavior, it is harder to make an assertion. The proposed domain policy for a maximum number of permanent devices per person, and registration of domain devices with a proximity requirement will reduce most threats significantly. However, for some attacks, such as ‘Content filling station’, it is hard to put in place effective protection without adopting less user-friendly policies like time-outs for domain membership.

Technical challenges for PED-DRM lie amongst others in privacy issues for user identities and in the infrastructure, e.g. availability of authentication mechanisms.

Acknowledgements

We would like to thank our colleagues for their comments and help in the design of the PED-DRM system. Furthermore, we would like to thank the reviewers for their valuable comments on draft versions of this paper.

References

[1] R.Vevers and C.Hibbert, *Copy Protection and Content Management in the DVB*, IBC Conference Publication, p458-466, Amsterdam, IBC2002, 15-9-2002.

[2] S.A.F.A.van den Heuvel, W.Jonker, F.L.A.J.Kamperman, and P.J.Lenoir, *Secure Content Management in Authorised Domains*, IBC Conference Publication, p467-474, Amsterdam, IBC2002, 15-9-2002.

[3] DVB-CPT, *DVB-CPT Authorized Domain: Definition / Requirements*, cpt-018r5, 2002.

[4] Thomson Multimedia, *SmartRight*, www.smartright.org, 2003.

[5] 5C, *DTCP Volume 1 Supplement E Mapping DTCP to IP (Informational Version), DRAFT Revision 0.9*, 12-9-2003.

[6] IBM, *IBM Response to DVB-CPT Call for Proposals for Content Protection & Copy Management: xCP Cluster Protocol*, DVB-CPT-716, 19-10-2001.

[7] John Gildred, Ashot Andreasyan, Roy Osawa, and Tom Stahl, *Protected Entertainment Rights Management (PERM): Specification Draft v0.54*, Pioneer Research Center USA Inc, Thomson, 9-2-2003.

[8] Eskicioglu, A. M. and Delp, E. J., *An overview of multimedia content protection in consumer electronic devices*, Signal Processing: Image Communication, Elsevier, 2001.

[9] Willem Jonker and Jean-Paul Linnartz, *Digital Rights Management in Consumer Electronics Products*, IEEE Signal Processing Magazine, Special Issue on Digital Rights Management, 2004.

[10] R.Gooch, *Requirements for DRM systems*, Digital Rights Management: Technological, Economic, Legal and Political Aspects, LNCS2770, Springer-Verlag, 2003.

[11] S.Guth, *A Sample DRM System*, Springer, 2003. Digital Rights Management: Technological, Economic, Legal and Political Aspects.

[12] Open Mobile Alliance, *DRM Architecture: Draft Version 2.0*, OMA-DRM-ARCH-V2_0-20040820-C, 20-8-2004.

[13] Claudine Conrado, Milan Petkovic, and Willem Jonker, *Privacy-Preserving DRM*, SDM 2004, LNCS 3178, Springer, 2004.

[14] Philips Semiconductors, *Near Field Communication*, http://www.semiconductors.philips.com/markets/identification/products/nfc/, 2004.

[15] T.Dierks and C.Allen, *RFC2246: The TLS Protocol (version 1)*, 1999.

[16] Microsoft, *Windows Media Connect - Connectivity Solution for Networked Media Players*, WinHEC2004, 2004.

[17] Microsoft, *Next Generation Windows Media DRM for Consumer Electronics Devices*, WinHEC2004, 2004.