Datavetenskap

Johan Olsson

Implementation och förklaring av DRM-tekniker för media till mobiltelefon

---

Examensarbete, C-nivå

2005:11

Implementation och förklaring av DRM-tekniker för media till mobiltelefon

Johan Olsson

Denna rapport är skriven som en del av det arbete som krävs för att erhålla en kandidatexamen i datavetenskap. Allt material i denna rapport, vilket inte är mitt eget, har blivit tydligt identifierat och inget material är inkluderat som tidigare använts för erhållande av annan examen.

Johan Olsson

Godkänd, 2 juni 2005

Handledare: Stefan Alfredsson

Examinator: Donald F. Ross

Sammanfattning

Begreppet Digital Rights Management, DRM, har på senare år blivit ett känt begrepp. Detta arbete undersöker olika metoder att skydda digital media till mobiltelefoner. Det förklaras vilka mekanismer som ligger bakom och hur det är tänkt att det skall fungera i de olika typer av skydd som finns. Open Mobile Alliance framtagna ramverk till hur DRM för mobiltelefoner skall konstrueras ligger till grund för arbetet. Arbetet visar också hur en implementering av OMAs specifikationer kan se ut. De tre olika skydden Forward Lock, Combined Delivery samt Separate Delivery presenteras och förklaras. Forward Lock hindrar kopiering till annan enhet. Combined Delivery gör det möjligt att restriktioner i renderingen av nedladdat material kan bestämmas. Separate Delivery är det starkare skyddet av de tre, i denna typ av skydd sker kryptering samt att rätten till att rendera nedladdat material skickas separat till mobiltelefonen. Det förklaras hur implementationen testas och hur det kopplas till resultatet. Problem som uppstod samt utökningar till implementationen tas upp och förklaras. Slutsatsen visar på ett blandat resultat och tar även upp tankar om varför så är fallet.

Implementation and explanation of DRM-techniques applied to content used in cellphones

Abstract

The idea of Digital Rights Management, DRM, has become a well-known concept the last few years. The assignment investigates different methods to protect digital media used in cellphones. It will explain the mechanisms behind the concept and how they will function in the different ways of protection that is offered. The framework developed by Open Mobile Association creates the foundation to which this project relies upon. An implementation linked to the framework will be shown. The three ways of protection Forward Lock, Combined Delivery and Separate Delivery is presented and explained. Forward Lock prevents forwarding to other device. Combined Delivery makes it possible to introduce restrictions when it comes to render the content. Separate Delivery is the stronger protection of the three, this way of protection enables ciphering and the right to render the content will be sent separately. How the implementation was tested will be explained and how that connects to the result. The problems that arose and expansions of the implementation is also brought up and explained. The conclusion shows a mixed result and will also address thoughts about why that was the case.

Tack till

Det har varit ett mycket intressant och lärorikt arbete. Att få möjlighet att göra det på ett företag som Milou har skänkt stor glädje, ett stort tack till samtliga på företaget. I synnerhet Christian Levin som agerat handledare och stöttpelare. Det har givit möjligheten att både lära känna människor och den miljö som råder i ett företag. Det har varit skönt att känna stöd från Datavetenskap och då speciellt Stefan Alfredsson som har lagt ned mycket tid på att hjälpa till med denna uppsats.

Innehållsförteckning

1 Inledning 1
1.1 Bakgrund 1
1.2 Syfte och mål 1
1.3 Metod 2
1.4 Disposition 2
2 Begrepp och DRM-tekniker 2
2.1 Digital Rights Management 2
2.2 Open Mobile Alliance 4
2.3 OMA Download 6
2.4 Forward Lock 10
2.5 Combined Delivery 12
2.6 Separate Delivery 14
3 Design 18
3.1 Konstruktion 18
4 Implementation 20
4.1 Miljö 20
4.2 Forward Lock 20
4.3 Combined Delivery 23
4.4 Separate Delivery 26
4.5 Testning 30
4.6 Dokumentering 33
5 Resultat 34
5.1 Funktionalitet 34
5.2 Problem 36
5.3 Utökningar 37

6 Slutsats 38

Referenser 39

Figurförteckning

Figur 1: Modell av OMA Download och DRM 7
Figur 2: De logiska elementens struktur 8
Figur 3: Exempel på Forward Lock 11
Figur 4: Exempel på Combined Delivery 13
Figur 5: Exempel på Separate Delivery 15
Figur 6: Exempel på superdistribution 16
Figur 7: Sammansättningen av DCF 16
Figur 8: Beskrivning av uintvar 17
Figur 9: Tanke bakom konstruktion av Forward Lock 18
Figur 10: Tanke bakom konstruktionen av Combined Delivery 19
Figur 11: Tanke bakom konstruktionen av Separate Delivery 19
Figur 12: GET-request från mobiltelefonen samt svaret från servern 21
Figur 13: Innehållet till ovanstående förfrågan 22
Figur 14: GET-request samt svar till en Download Descriptor 22
Figur 15: Download Descriptor 23
Figur 16: GET-request samt svar till Combined Delivery metod 24
Figur 17: Ett DRM-meddelande av typen Combined Delivery 25
Figur 18: Exempel på restriktioner 26
Figur 19: GET-request anrop av sddd.aspx samt svar 27
Figur 20: Download Descriptorn till Separate Delivery exemplet 27
Figur 21: GET-request av fil med typen DRM Content Format samt svar 27
Figur 22: En krypterad fil, DRM Content Format 28
Figur 23: Hexdump av en .dcf-fil 28
Figur 24: Illustration av sambandet uintvar och motsvarande binärarepresentation 29
Figur 25: Svar från mobiltelefon 30
Figur 26: Alternativet visas i alla tre typer av skydd med OMA Download 35
Figur 27: Anmärkning över lågt antal uppspelningar samt felmeddelande 35

Figur 28: Aktiveringsfas i Separate Delivery och lyckad aktivering av rättigheterna 36

Tabellförteckning

Tabell 1: Testutfall av metoden Forward Lock 31
Tabell 2: Testutfall av metoden Combined Delivery 32
Tabell 3: Testutfall av metoden Separate Delivery 32
Tabell 4: Statuskoders innebörd 33

Terminologi

DCF, *DRM Content Format*, beskriver vad som skall ingå i ett meddelande av typ Separate Delivery.

DRM, *Digital Rights Management*, begrepp som bland annat innefattar metoder att skydda data.

GSM, *Global System for Mobile Communication*, ett nätverk som mobiltelefoner använder sig av för att kommunicera (2:a generationen).

HTTP, *HyperText Transfer Protocol*, ett protokoll som används till att skicka information över Internet.

JAD, *Java Application Descriptor*, innehåller beskrivning av hur innehållet skall laddas ned och installeras bland annat.

MIDlet, en java-applikation som är gjord för små enheter såsom mobiltelefoner.

MIME, *Multipurpose Internet Mail Extensions*, beskriver formatet av meddelandekroppar som används över Internet.

OMA, *Open Mobile Alliance*, en organisation som består av företag som strävar efter gemensamma mål.

URI, *Uniform Resource Identifier*, är en beskrivning eller ett namn som kan användas till att referera till en resurs på Internet.

URL, *Uniform Resource Locator*, är vad vi vanligtvis kallar en Internetadress.

WAP, *Wireless Application Protocol*, en internationell standard för applikationer som använder trådlös kommunikation.

XML, *eXtensible Markup Language*, liknar HTTP men är en mer strikt version och används till att beskriva olika sorters data.

1 Inledning

1.1 Bakgrund

Under andra hälften av 90-talet fick Internet sitt genombrott. Detta globala nätverk växte sig allt större, användarantalet mångfaldigades snabbt och är nu en självklar del av vardagen för många människor. Denna digitala revolution har resulterat i nya sätt att distribuera tjänster på. Information kan snabbt och mycket billigt flyttas långa avstånd och nå många människor.

Allteftersom fler konsumenter fått tillgång till Internet har företag börjat utnyttja detta mycket väl lämpade medie till att sälja och leverera tjänster med. Nackdelen från deras perspektiv är att digitala objekt ofta är enkla att replikera, varpå köparen/mottagaren med lätthet kan sprida identiska kopior vidare. Detta resulterar i en ekonomisk förlust för företagen. Som ett steg mot att kunna skydda sina intressen och upphovsrättigheter i digitala verk har ramverket Digital Rights Management, DRM, tagits fram.

DRM har inte etablerat sig fullt ut ännu men det expanderar hela tiden och begreppet börjar bli känt av den stora massan användare. Allt fler företag har fått upp ögonen för denna teknik och Karlskrona-baserade Milou är ett av dem. De arbetar bland annat inom området som berör mobiltelefoner. Vissa tjänster som byggs för mobiltelefoner är önskvärda att skydda på något sätt. Milou står bakom uppdraget att skapa ett DRM-skydd i enlighet med specifikationer framtagna av Open Mobile Alliance, OMA. Denna mobila aspekt av DRM är vedertagen och har stöd av många stora tillverkare såsom Sony Ericsson och Nokia med flera.

1.2 Syfte och mål

Målen med arbetet är att implementera metoderna Forward Lock [1] och Combined Delivery [1]. Specifikationerna för dessa är utvecklade av OMA och version 1.0 av dem skall användas. Lösningen skall implementeras i C#, dokumenteras, testas och utvecklas för Milous testmiljö.

I mån av tid fanns det två extra mål, implementation av metoden Separate Delivery [1] samt analys av framtida lösningar och OMA 2.0. Analysen samt delar av implementationen av Separate Delivery har ej hunnit utföras.

1.3 Metod

Uppgiften kräver inläsning av OMAs dokumentation, inlärning av programspråket C#, plattformen ASP.NET samt protokollet HTTP. Då Milou har tillgång till flertalet mobiltelefoner från olika tillverkare kommer lösningarna att genomgå tester i aktuell miljö. Dessa element ligger till grund för att kunna slutföra uppgiften och konstruera en lösning som följer uppsatta krav.

1.4 Disposition

Resterande del av uppsatsen är uppbyggd enligt följande. Kapitel 2 tar upp relevanta begrepp som skapar förståelse för uppdraget. Kapitel 3 beskriver designen samt konstruktionen av uppgiften och kapitel 4 behandlar implementationen och miljön. Kapitel 5 innehåller det slutgiltiga resultatet av de olika DRM-skydden samt problem som uppstått. Kapitel 6 avslutar uppsatsen och ger en sammanfattning av uppgiften.

2 Begrepp och DRM-tekniker

2.1 Digital Rights Management

Digital Rights Management är ett begrepp som har växt fram och aktualiserats i och med Internets utbredning. Detta begrepp innefattar ett flertal metoder som tillåter en säljare av digitala verk att kontrollera samt styra hur de tillåts användas. Ofta är innehållet upphovsrättskyddat där säljaren äger spridningsrättigheterna till verket.

DRM är svaret från upphovsrättinnehavarna som historiskt sett har kämpat emot nya sätt att kopiera data. Metoder att kopieringskydda analog data existerar, till exempel Macrovision [2], men det finns ett större behov idag att skydda digital data. Att skapa en analog kopia är relativt tidskrävande, dyrt och ger en kvalitetsförsämring. De analoga kopieringsmetoderna innebär att för varje led av kopiering sjunker kvaliteten av datan där den till slut blir så låg att den blir ointressant. Film- och nöjesbranschen har genom årens lopp alltid varit emot och känt sig hotade av tekniska framsteg som gjort det möjligt att olovligen kopiera skyddade verk. Exempel på innovationer som mött kraftigt motstånd innefattar bland annat radiosändningar, kassett- och videoband [3].

När det kommer till digitala medier kan kopiering av data ske snabbt, till ett lågt pris och utan kvalitetsförsämring. Att det digitalt går att skapa identiska kopior i led efter led anses generera stora förluster för industrin [4]. Internets expansion och tillgänglighet för ett enormt antal användare skapar en liknande rädsla hos dagens digitala distributörer som en gång de analoga distributörerna kände.

DRM ses som ett kopieringsskydd av digital information men ger även stöd för inskränkning av hur användare tillåts använda informationen. Detta har skapat stor debatt, huruvida det anses acceptabelt eller ej. DRM stöds allt mer hårdvarumässigt. Anta att en mp3-spelare har fullt hårdvarustöd för DRM. En användare som köpt rättigheterna till en skyddad låt får den skickad till sin mp3-spelare. Om denne vill flytta låten till sin laptop kan DRM innebära att detta blir omöjligt. Att DRM ska bestämma över användarens egendom ses av många som ett hot, ett intrång i den personliga friheten.

Det skall bli intressant att se hur stor genomslagskraft den nya lagen om upphovsrätt kommer att få i Sverige, då den erhåller laga kraft den första juli 2005 [5]. Lagen om upphovsrätt bygger på EG-direktivet 2001/29/EG och två internationella avtal. Denna lag sätter upp nya bestämmelser vad gäller digitala verk.

DRM kan tillämpas inom åtskilliga områden. Fokus för detta uppdrag är den mobila aspekten. De tjänster och applikationer som används av dagens mobila enheter håller förhållandevis hög kvalitet. Grunden till att vilja skydda något baseras helt enkelt av viljan och syftet med att driva en affärsverksamhet. En affärsverksamhet är beroende av inkomster och tar därför ekonomisk skada av otillåten kopiering. Samarbete krävs på ett globalt plan för att kunna ta fram en tillräckligt generell standard som skall kunna gå att implementera på ett passande sätt. Att enskilda företag eller organisationer försöker skapa egna standarder kommer att skapa inkompabilitet mellan olika tillverkare av mobila enheter och tillverkare av tjänster till dessa. Ett led i att förverkliga en implementering av DRM är att bygga på gemensamma principer och tillvägagångssätt. Initiativet att skapa ett gemensamt forum, att utbyta idéer och skapa gemensamma metoder togs av ett flertal företag och resulterade i Open Mobile Alliance.

2.2 Open Mobile Alliance

I en bransch där det finns enorma ekonomiska möjligheter är det svårt att få fram en global standard. Många viljor betyder många idéer och tankar, ett initiativ till att samla de ledande i branschen togs 2002. Mer än 200 företag gick tillsammans och skapade OMA, Open Mobile Alliance, som har till uppgift att ta fram specifikationer som är tänkta att tillgodose sina medlemmars behov. De har specificerat fyra huvudsakliga mål med sin verksamhet.

- Publicera högkvalitativa, öppna **tekniska specifikationer** baserade på marknadens krav. Dessa skall ställa modularitet, utbyggbarhet och följdriktighet bland utvecklare i första led. Detta för att minska arbetet att skapa nya implementationer.
- Försäkra att OMAs tjänster tillhandahåller **interoperabilitet** mellan olika enheter, operatörer, geografiska områden, nätverk och tjänstedrivande företag. Det vill säga att deras produkter skall kunna fungera i alla tänkbara kombinationer.
- Att driva på förenandet av standarder inom den mobila tjänsteindustrin och **samarbeta med andra standardiseringsorganisationer** för att på så sätt bland annat minska operativa kostnader.
- **Tillhandahålla understöd** till medlemmar i OMA, vilken genre de än må tillhöra, så att de väljer att aktivt ha del i organisationen.

Dessa är högt ställda mål men OMA har lyckats binda många stora globala företag till sig, till exempel Sony Ericsson, IBM, Intel, Microsoft och Vodafone; så visst har de erhållit viss grad av tillit. Sedan starten 2002 har OMA expanderat, idag är antalet medlemmar inte mindre än 420 och den siffran växer fortfarande.

Vad har då OMA kommit fram med för lösningar? De har ett antal olika förslag på lösningar inom många områden, det som är intressant i detta arbete går under namnet OMA Digital Rights Management. Alla lösningar delas upp i tre faser, eller olika grader av interoperabilitet. Fas ett är ett slags kandidatarkitektur, fas två är en godkänd arkitektur och en lösning i fas tre skall garantera interoperabilitet. Det är nämnvärt att ingen lösning ännu nått fas tre. Den av Milou utsedda uppgiften behandlar i första hand OMA DRM 1.0. Denna lösning ligger i fas två och skall således ha ett befintligt stöd och vara en accepterad metod. OMA DRM 1.0 kungjordes 2004-06-25.

OMA DRM 1.0 har som mål att tillgodose önskemål från medlemsföretag som vill ha metoder att skydda sina digitala verk mot otillåten kopiering och otillåten användning. Med hjälp av framtagna metoder skall det gå att styra hur köparen får använda sin införskaffade produkt. Att på så sätt till exempel kunna begränsa antalet gånger mobiltelefonen får visa en bild eller ge mobiltelefonen rätt att spela upp en ringsignal en månad framåt i tiden. Detta skall dock inte ses som en komplett DRM-teknologi, standarden togs i hast fram för att tillfredställa marknadens behov av att kunna erbjuda provlyssningar samt hindra olovlig kopiering. OMA DRM 1.0 använder sig av tre olika komponenter att bygga sin lösning på.

- **Rights Expression Language** används för att uttrycka rättigheter på, det är en XML-struktur som definierar vad som skall vara tillåtet att göra med en tjänst/media på en enhet, till exempel en mobiltelefon.

- **Content Format** beskriver hur ett skyddat objekt/media skall skapas, vad som skall ingå i det skal man omsluter objektet med. Det innehåller själva strukturen för ett DRM-meddelande, bestämda metoder för kryptering och hantering av nycklar samt algoritmer. Om det inte anses befogat med kryptering finns även sätt att använda vanlig text att innesluta och skydda något objekt. Ett objekt/medie som packats på något av dessa sätt sägs vara i DRM-format

- **Metadata** används till att beskriva objekt, OMAs DRM 1.0 egna metaspråk med andra ord. Metadatan används först och främst till två saker.

    - Att ge information till den mobila enhet som tagit emot data.
    - Att förklara för användaren vad det egentligen är för något innehåll i det som skall laddas ned.

På dessa tre metoder grundas de två krav som ställs på funktionsdugligheten av specifikationerna för DRM.

Om ett objekt har blivit skickat inuti ett DRM-meddelande skall det *inte gå att skicka vidare* detta till exempelvis en annan mobil enhet.

Det skall även finnas *stöd för headrarna* ”Content-ID” och ”Content-Transfer-Encoding” i ett DRM-meddelande.

Content-ID kan ses som en stämpel, ett sätt att markera att rättighetsobjekt tillhör det insvepta innehållet. Headern Content-Transfer-Encoding sätter vilken typ av kodning innehållet skickas med, till exempel binär eller base64 [6]. Utöver dessa definieras även ett antal valfria egenskaper.

- Metod som gör det möjligt att skicka både rättigheterna och innehållet tillsammans till en mottagare.
- Metod för att skicka rättigheterna för sig och innehållet för sig samt superdistribution[1] av innehållet.
- Metod att på olika sätt ange hur nedladdat innehåll får användas av enheten, grundade på restriktioner angivna i rättighetsobjektet. Detta är den mest kontroversiella av de tre.

Till OMA DRM används ofta en nedladdningsmetod specificerad av OMA, för att ge understöd till hur Forward Lock, Combined Delivery samt Separate Delivery fungerar beskrivs denna i följande kapitel.

2.3 OMA Download

OMA Download används med fördel när det kommer till förfarandet att ladda ned objekt till mobiltelefon. Figur 1 ger en överblick till hur man vanligtvis modellerar begreppet OMA Download och dess koppling till OMA DRM.

OMA Download är i grund och botten ett ramverk som har skapats för att stödja nedladdning av objekt oberoende av typ och karaktär. Begreppet associeras dock ofta med nedladdningsbart innehåll som kan sätta en personlig prägel på en enhet, det kan röra sig om teman, skärmsläckare och ringsignaler med mera. Ramverket bygger på två redan befintliga mekanismer, protokollet HTTP [7] och MIDlet Download [8].

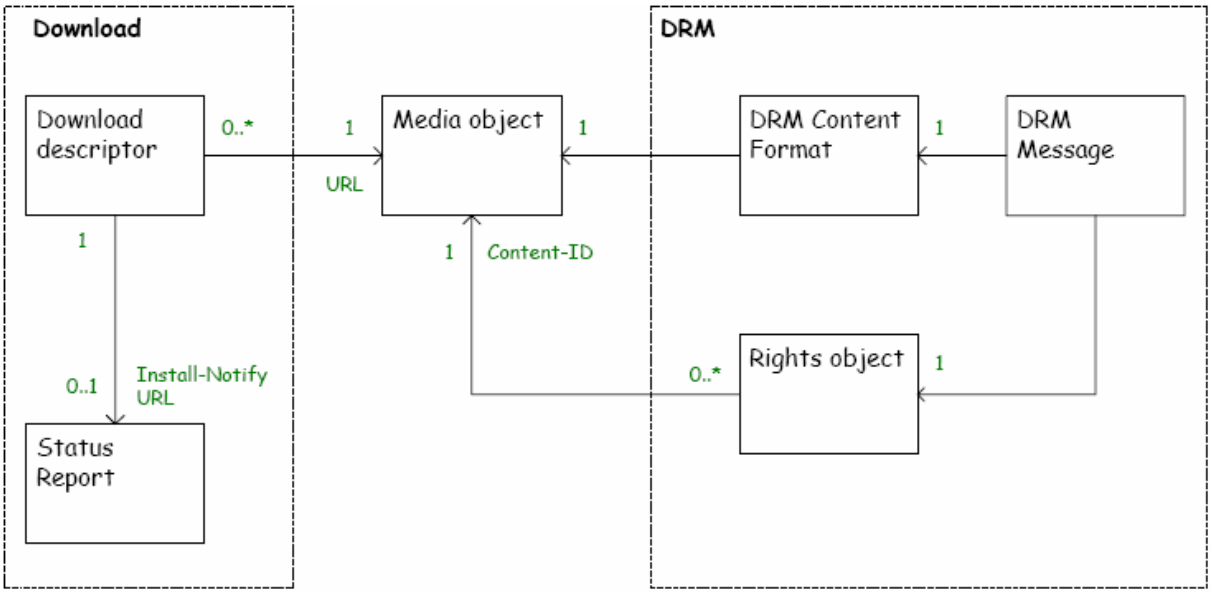
HTTP Download används ofta av företag som säljer nedladdningsbart innehåll vars värde inte är alltför högt eller inte tros orsaka nämnvärda förluster i händelse av otillåten kopiering. Objektet som laddas ned med HTTP, sker med endast en transaktion. Klienten skickar en HTTP_GET-request [7] och servern svarar med att skicka tillbaka innehållet. Denna metod har stöd av samtliga webbläsare, till och med av mobiltelefoner som stödjer WAP [10], och är samma metod som används i vardaglig mening när det gäller att surfa på Internet.

[1] Spridning av DRM-skyddat material som kräver nyckel från tillverkaren för att kunna rendera (visa, spela upp) det.

Javas MIDlet Download har givit inspiration vad det gäller att utöka ovan nämnda transaktion. Vissa mediatyper behöver en nedladdningsmetod som underlättar både för mottagaren och för avsändaren. Innan nedladdningen av en viss MIDlet tas en beskrivning emot, en JAD [8] (Java Application Descriptor). Denna innehåller instruktioner till JAM [8] (Java Application Manager) såsom varifrån just denna MIDlet kan laddas ned. Efter nedladdning (via HTTP) skickas ett svar tillbaka till sändaren av innehållet. MIDlet Download begränsas av att stöd för data av generell typ saknas.

För att på ett översiktligt sätt visa vad OMA Download står för, betrakta Figur 1. Figur 1, Figur 2, Figur 5, Figur 6, Figur 10 och Figur 11 är hämtade ifrån OMAs specifikation av DRM. Figur 3, Figur 4 samt Figur 9 är också hämtade ur specifikationen men är något modifierade av författaren till uppsatsen.

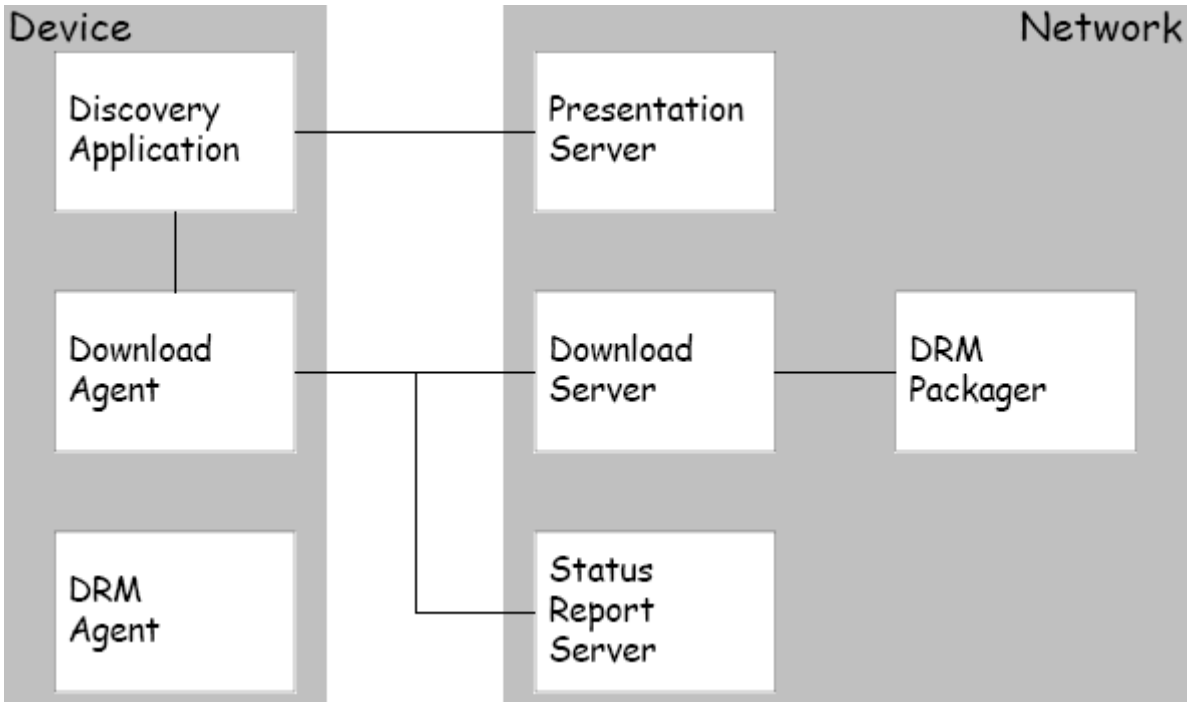
Mediaobjektet står i centrum, ganska självklart då det är något som skall skyddas och tillåtas laddas ned. Pilarna visar i vilken riktning sambandet går, Download Descriptorn[2] kan således innehålla antingen en eller ingen adress till mottagaren av statusmeddelandet. DRM-delen av figuren tolkas så att ett DRM-meddelande skall innehålla ett rättighetsobjekt och ett objekt byggt av DCF[3]. Det kan endast finnas ett mediaobjekt för varje rättighetsobjekt men noll eller fler rättighetsobjekt för varje mediaobjekt.



Figur 1: Modell av OMA Download och DRM

[2] Download Descriptorn innehåller en beskrivning av objektet som skall laddas ned.
[3] DCF förklaras närmare på sidan 16.

OMA Download kan även användas till att överföra oskyddad data. En mer logisk uppbyggd struktur samt förklarande use cases[4] ger förhoppningsvis en klarare bild. Figur 2 visar de olika logiska delarna som ingår i en nedladdning med OMA Download.



Figur 2: De logiska elementens struktur

Presentationsservern kan till exempel vara en hemsida innehållande länkar till olika ringsignaler. Genom mobiltelefonens webbläsare kan exempelvis en länk aktiveras varpå en transaktion påbörjas. Download Descriptorn som antingen finns på presentationsservern eller på nedladdningsservern skickas. Ansvaret för transaktionen på klientens sida överlämnas till Download Agenten. Antingen har innehållet skyddats, varpå DRM-packaren varit aktiv eller har det skickats oskyddat. Nedladdningsservern skickar i vilket fall som helst det önskade materialet. Download Agenten ansvarar även för att statusmeddelandet skall skickas. I mobiltelefonen hanterar DRM-agenten det nedladdade innehållet och ansvarar för att stadgarna för DRM upprätthålls.

För att klargöra vad som har utökats från användarfallet med en transaktion av protokollet HTTP till en OMA Download transaktion ges tre use cases.

[4] Olika fall av användning.

- På Internet idag används **HTTP** ofta till nedladdning. Låt oss säga att en användare klickar på en länk i sin webbläsare och på så sätt initierar en nedladdning. Då händer följande.
    - Genom detta klick får klienten veta att användaren önskar initiera en nedladdning.
    - Ett GET-request skickas till servern dit denna URL pekar och inväntar sedan ett svar.
    - Servern gör objektet redo för leverans och skickar ett GET-response till klienten.
    - Klienten accepterar GET-response och tar emot HTTP-headrarna och medieobjektet från servern.

Detta use case innefattade att medieobjektets URL var giltigt och pekade på något innehåll. Det började med att mobilens användare initierade en nedladdning och slutade med att det laddades ned.

- **OMA Download** utökar ovanstående use case med användandet av en Download Descriptor. Både **medieobjektet och Download Descriptorn** laddas ned med en transaktion, ett så kallat multipart meddelande.
    - Användaren startar en nedladdning, exempelvis genom att klicka på en länk.
    - Ett GET-request skickas till servern dit denna URL pekar och inväntar sedan ett svar.
    - Servern gör objektet redo för leverans och skickar ett GET-response till klienten.
    - Klienten accepterar GET-response och tar emot HTTP-headrarna och medieobjektet från servern.
    - Informationen i Download Descriptorn åskådliggörs och om status efterfrågades skickas ett sådant meddelande.

.

Att ytterligare fördjupa detta fall av användning blir att använda två GET-request, ett för Download Descriptorn och ett för medieobjektet.

- **OMA Download** används i detta fall för att i **förväg** ge mobiltelefonen en chans att **undersöka medieobjektet** och undersöka om kapacitet finns för att rendera det och spara det, med mera.

- Användaren startar en nedladdning, exempelvis genom att klicka på en länk.
- Ett GET-request skickas till servern dit denna URL pekar och ett svar inväntas.
- Servern gör Download Descriptorn klar för leverans och skickar ett GET-response till klienten.
- Mobiltelefonen accepterar GET-response och tar emot HTTP-headrarna och Download Descriptorn från servern.
- Mobiltelefonen analyserar innehållet i Download Descriptorn och användaren ges möjlighet att godkänna nedladdning av medieobjektet.
- HTTP används enligt första fallet till att transportera medieobjektet.
- Om statusmeddelande efterfrågades skickar nu mobiltelefonen ett sådant.

OMA Download är således ett namn på en nedladdningsmetod som till stor del använder HTTP för transaktioner mellan mobiltelefon och webbserver. Det innefattar även medel hur objekt skall tas om hand av mobiltelefonen.

OMA har definierat vad man kan förvänta sig att OMA Download skall klara av. Det mest grundläggande är stöd för HTTP till nedladdning av Download Descriptorn, DRM-meddelandet och uppladdning av statusrapporten genererad vid installationen (om avsändaren har begärt en sådan). Det skall finnas stöd för att kunna ta emot innehåll och Download Descriptorn i olika HTTP-strömmar. De definierar också att tillgång till mediet/innehållet måste stoppas om antingen nedladdningen avbryts eller om sändning av installationsmeddelandet misslyckas. OMA menar med andra ord att ett installationsmeddelande av typen lyckat måste erhållas för att användning av innehållet skall godkännas. I ett tillägg till de egenskaper som måste vara uppfyllda nämner de även en metod som skall vara funktionsduglig, nämligen stöd för samleverans av Download Descriptor och det associerade innehållet.

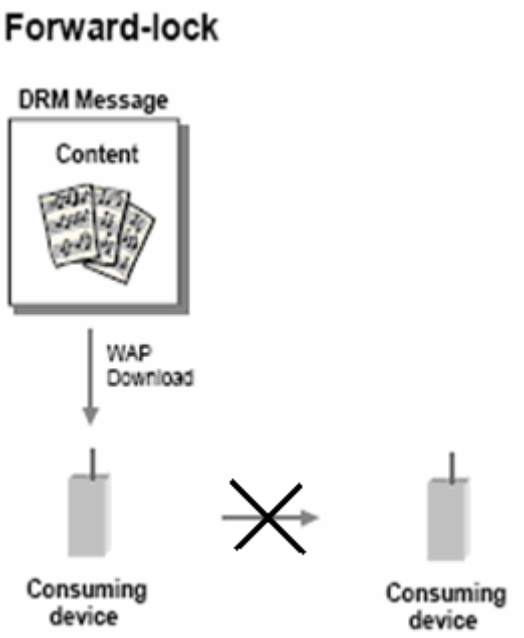
Som en följd av ovanstående metoder och definitioner har tre olika typer av DRM-skydd tagits fram, Forward Lock, Combined Delivery och Separate Delivery.

2.4 Forward Lock

Forward Lock är det enklaste skyddet som har definierats av OMA. Detta är den enda metoden som inte innehåller någon form av rättighetsobjekt. Rättigheter definierar vilka

operationer som är giltiga på skyddat material, men i detta fall appliceras det grundläggande skyddet, där median binds till den enskilda enheten. Med Forward Lock kan därmed inga andra begränsningar vad gäller nyttjandet av median i enheten styras.

Forward Lock är till för att hindra spridning av media till annan enhet. Avsikten är att förhindra peer-to-peer distribution eller superdistribution av material som inte anses behöva ytterliggare skydd. Denna typ av skydd används ofta till tjänster som är prenumerationsbaserade som till exempel nyheter, sport och målservice med mera. Figur 3 exemplifierar Forward Lock.



Figur 3: Exempel på Forward Lock

Mobiltelefonen laddar ned det skyddade objektet från någon källa. Det finns två olika metoder för hur detta kan ske, antingen med protokollet HTTP eller med OMA Download. OMA Download är inget protokoll i sig utan definierar metoder att interagera med enheten och använder HTTP för kommunikation. Att endast använda HTTP ger begränsade möjligheter vad det gäller kommunikation mellan enheten och servern.

I båda fallen skickas innehållet inslaget i ett meddelande som talar om att det är skyddat. I headerfältet sätts Content-Type flaggan till `application/vnd.oma.drm.message` och Content-

Length till storleken på innehållet. Detta paket laddas ned och tolkas av enheten som upptäcker att det är ett DRM-meddelande och hanteras därefter.

OMA har specificerat vad en enhet som säger sig stödja Forward Lock skall anpassa sig till. Ett Forward Lock-skyddat DRM-meddelande tillåts presenteras i enlighet med innehållets MIMEtype, men det får inte skickas till andra enheter. Innehållet i ett DRM-meddelande får ej heller ändras av enheten. Enheten får spara DRM-meddelandet på exempelvis ett minneskort. Ifall minneskortet plockas ut får dock inte enheten ha fortsatt tillgång till det skyddade materialet. Enheten får alltså inte kopiera eller spara det i internt minne.

Det finns i dagsläget inget standardutförande på hur mobiltelefoner hanterar DRM-skyddat material av typen Forward Lock. Det skiljer mellan olika tillverkares implementationer. Denna metod har dock ett ganska brett stöd bland tillverkarna. Vissa mobiltelefoner kräver till och med att ljudfiler skall vara skyddade med Forward Lock för att få användas som ringsignal (exempelvis mobiltelefonen Sony Ericsson S700i).

Forward Lock är inte ett speciellt starkt skydd. Det är tämligen enkelt att implementera och ännu lättare att knäcka. När då tillverkare kräver att ett skydd måste finnas på till exempel en ringsignal för att få använda den skapar detta en del problem för användaren. Man kan tänka sig en användare som har gjort en ringsignal på sin dator. För att kunna använda den i sin telefon måste användaren således sätta sig in i hur Forward Lock fungerar.

2.5 Combined Delivery

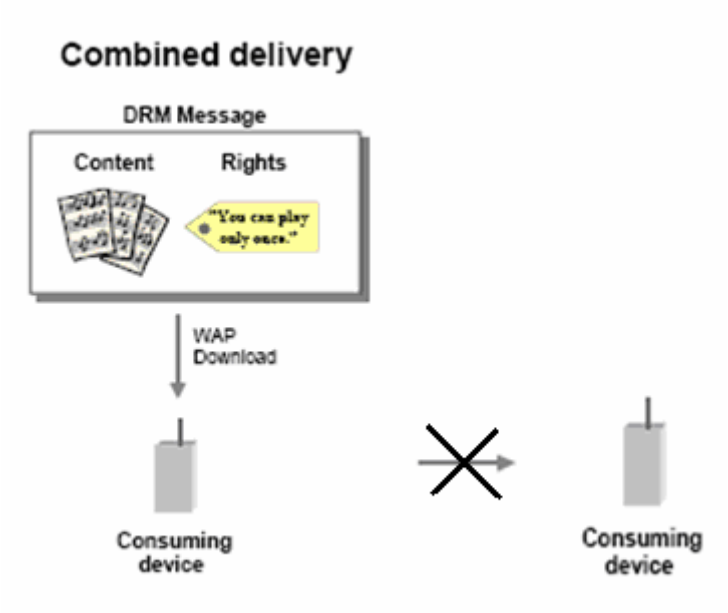
Combined Delivery uppfyller Forward Locks egenskaper men utöver det används ett rättighetsobjekt för att kunna definiera hur mediet kan användas. I och med Combined Delivery har företag som tillhandahåller tjänster möjlighet att begränsa eller styra hur användningen av nedladdat innehåll kan användas. Samma innehåll kan skyddas med olika restriktioner och därmed öppnas marknaden för olika erbjudanden till sina kunder. Detta är en förbättring jämfört med Forward Lock. Combined Delivery, se Figur 4, är en mer avancerad metod där man kan skräddarsy rättigheterna efter önskemål. Användare kan till exempel få en förhandstitt av innehållet, genom restriktionen att innehållet får renderas endast en gång.

Det finns fyra olika klasser av rättigheter; **play**, **display**, **execute** och **print**. Vart och ett av dessa klasser kan endast appliceras på MIMEtyper som renderas på matchande sätt. En bild

kan exempelvis ha rättigheter av klasserna display och print, den kan alltså inte exekveras eller spelas upp.

Till de fyra klasserna kan tre olika sorters restriktioner användas.

- **Count** är antalet gånger något får hända, kanske antalet gånger en ringsignal får spelas upp eller antalet gånger en bild får renderas.
- Det finns en tidsrestriktion, där man kan ange hur lång **giltighetstid** nedladdat material har (räknat från då det laddades ned), det anges på sekunder så när.
- Man kan använda en **starttid** och en **sluttid**, om endast starttid anges tolkas det som att efter detta klockslag går det bra att använda innehållet. Om bara en sluttid anges tolkas det som att man har rätt till innehållet fram till detta klockslag. I det fall då det finns både en starttid och en sluttid måste sluttiden vara senare och det tolkas som att innehållet får renderas mellan dessa klockslag.



Figur 4: Exempel på Combined Delivery

Det kan även förekomma kombination av de olika typerna av restriktioner. Ett exempel kan vara att det finns definierat att du får rendera innehållet två gånger men med tidskravet inom de närmsta fem minuterna.

Vad som händer då är att när du väntat fem minuter och tidsfristen går ut kommer innehållet inte att renderas trots att det finns två visningar kvar. Båda kraven måste således vara uppfyllda för att enheten skall kunna visa innehållet.

Combined Delivery ger samma möjligheter till nedladdning som Forward Lock, det vill säga antingen skickas den med HTTP eller OMA Download. En annan Content-Type omsluter det skyddade objektet vid Combined Delivery, `application/vnd.oma.drm.rights+xml`. De rättigheter som nämns ovan är alla uttryckta med en XML-struktur och den kännetecknas av dess logiska uppbyggnad och ordnade struktur. XML-strukturen innehåller utöver rättighetsobjektet även element som anger vilken version som används och även en stämpel i form av en URI.

Allt detta faller under OMA REL eller Rights Expression Language [9], och dess specifikationer för hur ett sådant meddelande skall vara uppbyggt för att erhålla stöd av enheter. Man kan se Combined Delivery som en utökad variant av Forward Lock, OMA hänvisar till detta genom att förklara att en enhet som stödjer Combined Delivery måste också stödja Forward Lock. De skiljer sig åt knappt alls, förutom REL och den nya Content-Typen då så klart.

Det finns krav som en enhet med Combined Delivery-stöd skall uppfylla. Enheten måste indikera att den klarar av både DRM-meddelandetypen och rättighetsobjekt-typen. Dessutom skall den kunna hantera DRM-meddelanden som innehåller både ett innehållsobjekt och ett rättighetsobjekt. Enheten måste tillämpa rättigheterna givna i REL när innehållet används. Användaren måste tillåtas att spara innehållet, installera och avinstallera DRM-innehåll.

2.6 Separate Delivery

Separate Delivery tillhandahåller ett mycket starkare skydd än Forward Lock och Combined Delivery. De två egenskaperna som bidrar till det förstärkta skyddet är att innehållet krypteras och att rättigheterna skickas separat med så kallad WAP-push [10], se Figur 5.

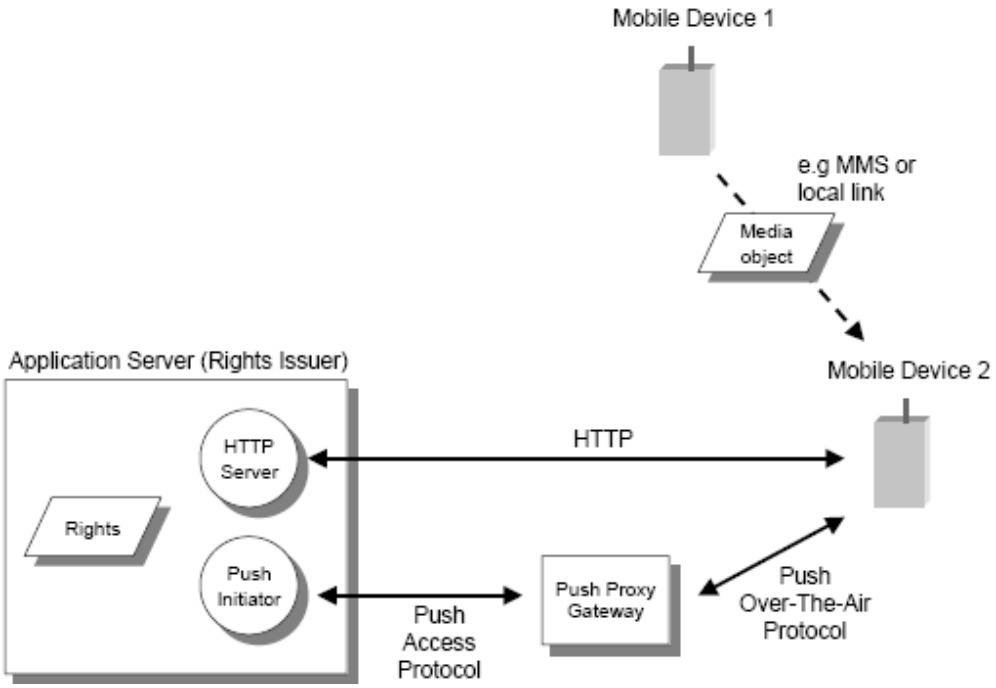
Figur 5: Exempel på Separate Delivery

Det senare innebär dock svårigheter, att på ett bra sätt synkronisera överföringen så pass att inget hindrar sammanstrålningen och att väntetider uppstår.

När det gäller bakåtkompabilitet för en enhet som stödjer Separate Delivery följer den det tidigare mönstret; en sådan enhet skall även stödja Combined Delivery och Forward Lock.

Det man vill skydda sveps in i ett DRM Content Format, DCF [11]. I denna metod krävs symmetrisk kryptering, närmare bestämt AES [12] med 128 bitars nyckel. Kryptering gör innehållet oanvändbart för alla som inte har dess nyckel. Detta gör det möjligt att skicka innehållet i en öppen kanal, eftersom säkrare transportmetod används för att skicka nyckeln och innehållets rättigheter. I ett GSM-nätverk förväntas det att rättigheterna skickas med en obekräftad WAP-push via en uppkopplingslös session, på liknande sätt som en operatör skickar ut inställningar till mobiltelefoner. Ett WBXML-kodat [13] rättighetsobjekt och WAP headrar tar i de flesta fall inte upp mer plats än att de får plats i ett SMS.

I och med att innehållet är omöjligt att rendera utan nyckel öppnas möjligheten upp för superdistribution, se Figur 6. Fördelen är att en användare kan skicka eller kopiera DCF-innehållet till annan enhet, exempelvis till en kompis. En ringsignal som är populär kan då spridas snabbt. När eventuell mottagare vill använda ringsignalen kontaktar enheten angiven URL i metadatan och en session upprättas för att hämta nyckeln och rättigheterna.



Figur 6: Exempel på superdistribution

Specifikationerna för DRM Content Format definierar hur man skall gå tillväga för att konstruera ett krypterat meddelande. Ett DCF-meddelande byggs upp av två huvudsakliga element; datan som skall skickas och tillhörande headrar.

Headrarna beskriver vilken ”Content Type” datan skall mappas till, de innehåller också en unik stämpel som matchas mot rättighetsobjektet. Headrarna visar även krypteringsinformation och vart enheten kan få rättighetsobjektet ifrån. Med hjälp av dessa två samt definitioner i DCF kan man i detalj bygga ett meddelande.

DCF-meddelandet har struktur enligt Figur 7.

Ver.	Cont.Type Storlek	Cont.URI Storlek	Cont.Type	Cont.URI	Headerstorlek	Datastorlek	Headrar	Data

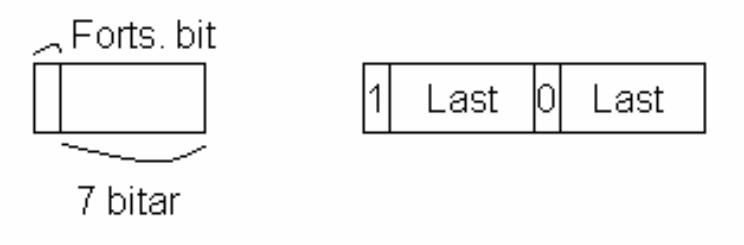
Figur 7: Sammansättningen av DCF

- Version [uint 8]: Anger aktuell version, alltid 1 i OMA DRM 1.0.
- ContentType Storlek [uint 8]: Anger i antal bytes, längden av ContentType-fältet.
- ContentURI Storlek [uint 8]: Anger i antal bytes, längden av ContentURI-fältet.
- ContentType [octets]: Anger MIMEtypen för innehållet som skall skyddas.

- ContentURI [octets]: Anger den unika stämpeln knutet till innehållet.
- Headerstorlek [uintvar]: Anger i antal bytes, längden av Header-fältet.
- Datastorlek [uintvar]: Anger i antal bytes, längden av den krypterade datan plus ytterligare 16 bytes för initialiseringsvektorn som används i AES.
- Headrar [octets]: Beskrivande metadata, det finns sju olika. En av dem är Rights-Issuer headern som definierar vart rättighetsobjektet finns.
- Data [octets]: Den krypterade datan.

Det finns en tanke bakom datatyper i DCF, uintvar används för att med dynamisk längd kunna representera längden av data.

- uint 8: Är en unsigned integer med åtta bitars storlek med värdemängden [0, 2^8].
- octets: Är åtta bitar opak data. Varje octet behandlas som om innehållet är osynlig, man behandlar datan som ett block.
- uintvar: Är kort och gott en uint med varierbar längd. Om den skall representera värdet 2^6 räcker det med ett block av åtta bitars längd, i fallet med 2^9 krävs det två block, totalt 16 bitar, se Figur 8.



Figur 8: Beskrivning av uintvar

För att en enhet skall ha stöd för Separate Delivery krävs det att den kan tyda följande mediatyper; `application/vnd.oma.drm.rights+xml`, `application/vnd.oma.drm.rights+wbxml` och `application/vnd.oma.drm.content`. Rättighetsobjektet som skickas till enheten efter att DCF tagits emot kan variera. Antingen skickas det som vanlig text, jämför med Combined Delivery, eller kodas det med WBXML. Fördelen med WBXML är att mängden data som skickas kan kraftigt reduceras, upp till 16% av motsvarande XML-objekt.

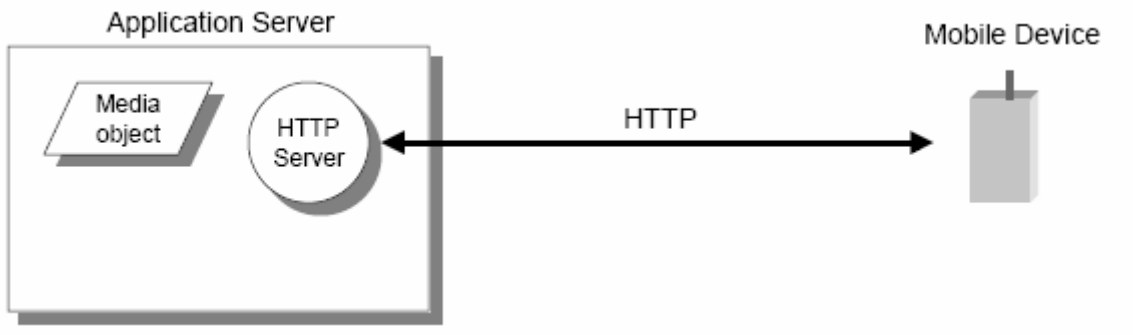
3 Design

3.1 Konstruktion

Hur arbetet skulle kunna konstrueras på ett bra sätt var inte självklart. De medel som fanns tillhands indikerade dock tydligt att någon sorts webbapplikation behövde konstrueras, eftersom mobiltelefoner använder HTTP. Det krävs en webbserver för att tillgodose kommunikation med mobiltelefonen och ett programspråk till att sköta logiken. Ett front- och ett backend. Det måste ingå metoder att omsluta ett objekt av DRM-skydd och därmed skapa ett DRM-meddelande men också ett sätt att låta mobiltelefoner ladda ned dem.

Det finns inget som motsäger att serversidan av Figur 2, de logiska elementen, kan konstrueras som en enhet. Presentationsservern, nedladdningsservern och mottagaren av statusmeddelandet borde utan problem kunna vara en och samma webbapplikation. DRM-förpackningen av ett meddelande kräver metoder att skydda filer, till exempel ringsignaler. Det måste även finnas någon länk mellan front- och backend som möjliggör utbyte av information. De mobiltelefoner som anses utgöra en marknad för ringsignaler och dylikt har stöd för HTTP det finns därmed inget som talar emot denna metod. Att tänka på de olika DRM-skydden var för sig och bryta ned konstruktionen underlättar mycket.

I fallet med Forward Lock kan man tänka sig en lösning enligt följande, Figur 9.

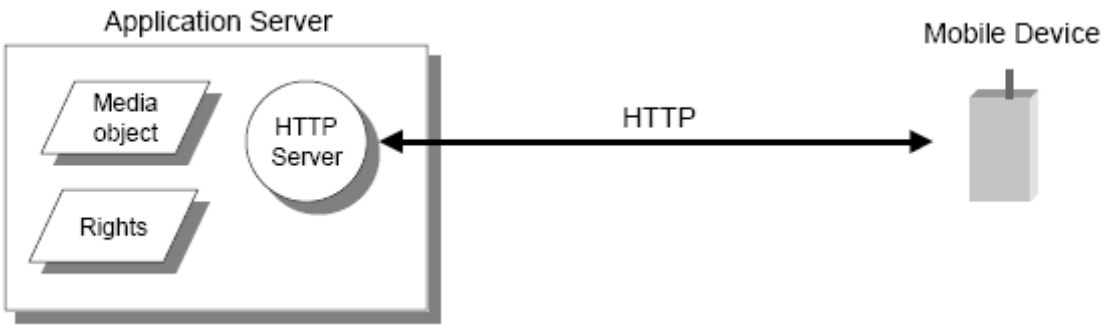


Figur 9: Tanke bakom konstruktion av Forward Lock

Figur 9 beskriver vad som behövs för att på ett okomplicerat sätt kunna bygga en lösning av problemet med Forward Lock. Ett gränssnitt mot mobiltelefonen behövs som på något sätt skall initiera en överföring av önskat innehåll (HTTP). Ett mediaobjekt och metoder att

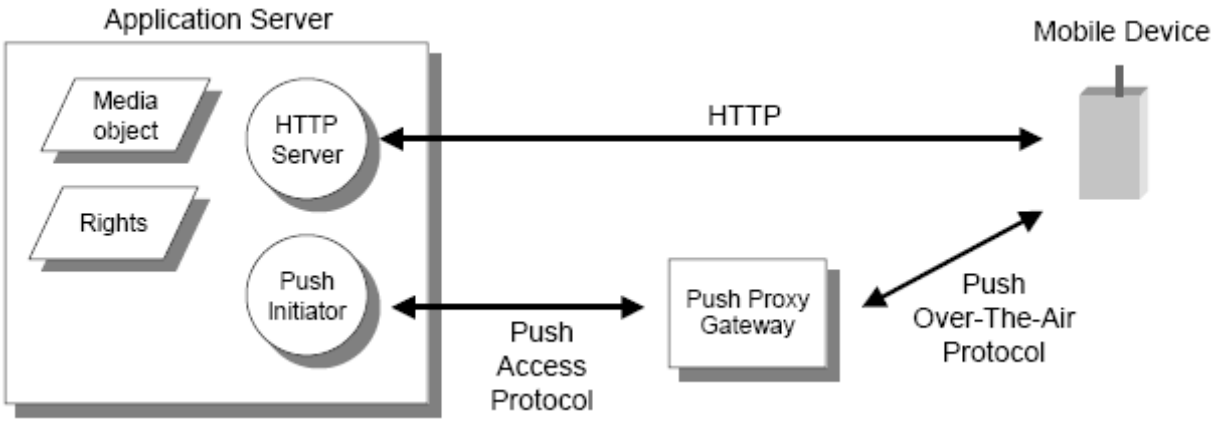
konstruera DRM-skyddet enligt OMAs specifikationer behövs också. De tre grundpelarna är filhantering, DRM-funktionalitet och transport.

Utökningen av konstruktionen till att stödja även DRM-skyddet Combined Delivery, se Figur 10, är inte stor. Rättighetsobjektet skall kunna bestämmas och på något sätt bindas till lösningen.



Figur 10: Tanke bakom konstruktionen av Combined Delivery

Att Separate Delivery, Figur 11, kräver mycket mer av konstruktionen inses klart ur specifikationerna för denna metod. Det behövs någon form av kommunikation mellan webbapplikationen och portalen som skall skicka rättighetsobjektet. Ett tillskott är även den skyddande kryptering som skall finnas till denna typ av DRM-skydd.



Figur 11: Tanke bakom konstruktionen av Separate Delivery

På något sätt skall en WAP-push initieras av applikationen och genom en push-gateway kunna komma fram till mobiltelefonen. Denna gateway innefattas inte i uppgiften att konstrueras, Milou hade tillgång till en sådan. Deras gateway fanns hos en operatör och den ger möjlighet att skicka WAP-push till en specifik mobiltelefon.

För att ha någon möjlighet att ta emot statusrapporten som används vid OMA Download behöver inparametrarna till ett HTTP-anrop läsas av. Detta skulle inte bli något större problem då det bara gäller att sortera inparametrarna på viktig information.

4 Implementation

4.1 Miljö

Milou ansåg att uppdraget skulle byggas till en Microsoft Windows-plattform. I flertalet av Microsofts operativsystem ingår det en webbserver, IIS (Internet Information Services). Denna har använts för att ta hand om webbdelen (frontend) av utvecklingen. Det av Microsoft utvecklade ramverket, ASP.NET [14], agerar länk mellan kodningen i C# (backend) och dess interagerande med IIS. Kodningsmiljön har uteslutande varit Microsofts Visual Studio 2003.

Inga förkunskaper i varken C#, Visual Studio 2003 eller IIS betydde att mycket tid fick tillägnas inlärning. Det fanns en tanke på vad som skulle krävas för att få en funktionsduglig implementation. Avsaknaden av kunskaperna till verktygen bromsade ned arbetets implementationsutveckling och mer kraft tillägnades språket C# samt miljön. Själva konceptet med DRM och då specifikt OMA DRM 1.0 var inte svårt att förstå. När väl tillräckligt med kunskap om miljön erhållits och kännedom om hur dess olika delar kan kombineras kunde implementeringen starta.

4.2 Forward Lock

Till det första delmålet i uppsatsen, Forward Lock, behövdes en webbserver. Milou tillhandahöll webbservern IIS med tillräckliga rättigheter för att kunna börja lösa problemet. De stod även till förfogande med Visual Studio 2003. Med OMAs dokumentation angående Forward Lock som byggsten kunde arbetet påbörjas. Detta DRM-skydd förefaller ganska enkelt att implementera. Begreppet är lättförståeligt och krävde inga komplicerade metoder. Det mest problematiska med implementeringen av denna konstruktion visade sig bli filhantering i C# samt hur godtyckliga headrar i webbserverns HTTP-svar kan läggas till.

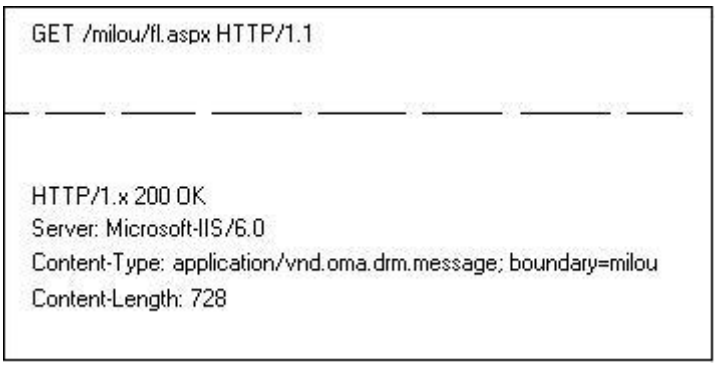
Under arbetets gång har två parallella projekt använts i Visual Studio 2003, ett där webbdelen programmeras (frontend) och ett där programmeringen av C#-delen sker

(backend). I webbdelen av Forward Lock anges vad som skall hända när man matar in en speciell URL, i detta fall **fl.aspx** samt **fldd.aspx**.

Skillnaden mellan dessa sidor ligger i metoden att leverera innehållet till mobiltelefonen. I fl.aspx används HTTP, i fldd.aspx används OMA Download. I det senare laddas det ned en Download Descriptor till enheten och Download Agenten i telefonen utför det metadatan beskriver. Gemensamt för de båda är att säljaren först måste välja vilken fil som skall skyddas och sedermera tas emot av enheten.

I **fl.aspx**, där HTTP används för att skicka det skyddade DRM-meddelandet tas med hjälp av ovanstående valda fil fram filändelsen. Denna mappas mot kända MIMEtyper och skickas tillsammans med filnamnet till backend. Projektets backend ansvarar för att utföra alla aritmetiska och logiska operationer. Till webbservern returneras en bytearray innehållande det nu skyddade filinnehållet. Frontend lägger till två olika headrar, Content-Type och Content-length. Som innehållstyp anges `application/vnd.oma.drm.message`, med tillägget ”`; boundary=milou`”. Typen av innehåll förtäljer att ett DRM-meddelande av typen Forward Lock skall skickas. Den avgränsare som utöver detta specificeras anger vart i DRM-meddelandet det skyddade innehållet börjar och slutar. Bytearrayens längd tas fram och skickas som tillägg till Content-Length headern.

Ett komplett DRM-meddelande av typen Forward Lock samt dess initiering visas nedan i Figur 12 samt Figur 13.

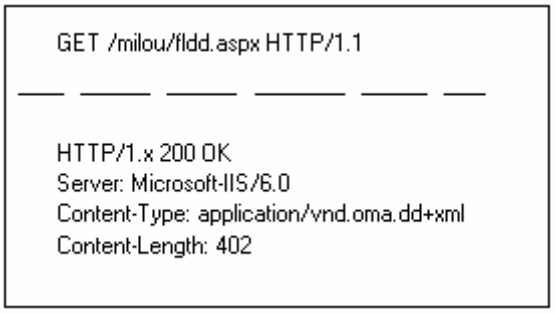


Figur 12: GET-request från mobiltelefonen samt svaret från servern

Figur 13: Innehållet till ovanstående förfrågan

Filen som skickas binärt, därav de oläsliga tecknen, omsluts av gränsen milou, startindikation ges av prefixet --, slutet anges med både ett pre- och postfix. Antalet binära tecken har reducerats för att skapa överskådlighet, DRM-meddelandet är i detta fall egentligen 728 bytes stort. I Figur 13 kan även ses att det som skyddades var en bild av formatet jpeg. Mobiltelefonen som tog emot DRM-meddelandet avlägsnar det skyddade höljet och kan rendera bilden, ett DRM-meddelande av typen Forward Lock har tagits emot.

I **fldd.aspx** skickas istället för den skyddade filen en Download Descriptor till mottagaren som innehåller en mängd metadata. Backend ansvarar i detta fall inte bara för att skapa DRM-meddelandet utan även för att generera Download Descriptorn till en bytearray. DRM-meddelandet sparas tillfälligt på disk och Download Descriptorn returneras till webbservern. De två headrar som frontend skapar blir i detta fall `Content-Type: application/vnd.oma.dd+xml` samt Content-Length med tillhörande storleksattribut, se Figur 14.



Figur 14: GET-request samt svar till en Download Descriptor

```
<media xmlns="http://www.openmobilealliance.org/xmlns/dd">
<objectURI>http://someserver/media/untitled2.jpg.dm</objectURI>
<size>728</size>
<type>image/jpeg</type>
<type>application/vnd.oma.drm.message</type>
<name>Milou exjobb</name>
<vendor>Milou</vendor>
<description>Optional description info</description>
<installNotifyURI>http://someserver/status.aspx?fldd</installNotifyURI>
</media>
```

Figur 15: Download Descriptor

Enheten som tar emot detta meddelande förstår att en Download Descriptor är på väg och läser in längden av den samt tolkar dess innehåll, Figur 15. Det ges möjlighet att med hjälp av metadatan jämföra om den har stöd för angiven mediatyp, har plats i minnet, ge användaren information om vad DRM-meddelandet innehåller och så vidare. Om användaren accepterar innehållet skickas en HTTP-GET request till URLen inom objectURI-taggen i metadatan. Enheten får svar enligt fallet med fl.aspx och nedladdning av DRM-meddelandet sker. Efter nedladdning skickas ett statusmeddelande till angiven URL i metadatan som anger utfallet av hämtningen.

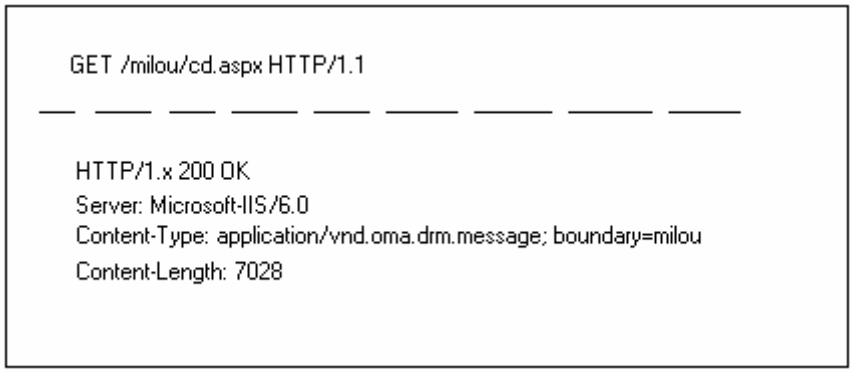
Ur konstruktionen av Forward Lock följde två viktiga ting; en implementation för att skapa ett DRM-meddelande samt hur man skapar en Download Descriptor. Då Combined Delivery har stora likheter med Forward Lock, se Figur 10, kom dessa kunskaper väl till pass.

4.3 Combined Delivery

Översiktligt är Combined Delivery precis som Forward Lock med tillägget att rättigheter kan specificeras för innehållet. Med erfarenhet från något äldre mobiltelefoner finns dock inte stöd för denna metod, många äldre mobiltelefoner har endast har stöd för Forward Lock.

Även i implementeringen av Combined Delivery har både HTTP och OMA Download använts och resultaten blev **cd.aspx** och **cddd.aspx**. Gemensamt i dessa fall blir att de skapar ett innehåll som består av två delar; rättighetsbeskrivningen och innehållet. Mobiltelefonen bryter ned meddelandet och tolkar innebörden av de givna rättigheterna. Bland rättigheterna finns elementet `<o-dd:uid>` som innehåller värdet på en stämpel. Denna måste stämma överens med värdet av Content-ID headern i mediadelen av det multipartmeddelande som Combined Delivery utgör. Om matchningen godkänns kan rendering av innehållet ske baserat på de givna rättigheterna.

Först skapar webbservern en instans av rättighetsobjektet, i .aspx filerna anges de rättigheter som önskas användas. En sträng genereras med XML-strukturerad innehållskaraktär. Den fil som avses att bli skyddad skall också anges. Backend anropas med filnamnet, filens MIMEtype och rättighetsobjektet som inparameter. I stort sett samma förfarande som i fallet Forward Lock sker nu, analogt gäller att skillnaden mellan HTTP och OMA Download metoderna kvarstår. Combined Delivery använder samma MIMEtype som Forward Lock, `application/vnd.oma.drm.message`. Ett exempel på hur ett DRM-meddelande av typen Combined Delivery ser ut, i detta fall **cd.aspx** visas nedan. Förfrågan och svar i Figur 16 samt materialet som hämtades i Figur 17.



Figur 16: GET-request samt svar till Combined Delivery metod

```
--milou
Content-Type: application/vnd.oma.drm.rights+xml
Content-Transfer-Encoding: binary

<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE o-ex:rights PUBLIC "-//OMA//DTD DRMREL 1.0//EN"
"http://www.openmobilealliance.org/tech/DTD/drmrel10.dtd">
<o-ex:rights xmlns:o-ex="http://odrl.net/1.1/ODRL-EX" xmlns:o-dd="http:
//odrl.net/1.1/ODRL-DD" xmlns:ds="http://www.w3.org/2000/09/xmldsig#/">
<o-ex:context>
<o-dd:version>1.0</o-dd:version>
</o-ex:context>
<o-ex:agreement>
<o-ex:asset><o-ex:context>
<o-dd:uid>cid:MilouExjobb@milou.se</o-dd:uid>
</o-ex:context></o-ex:asset>
<o-ex:permission>
<o-dd:play><o-ex:constraint>
<o-dd:count>4</o-dd:count>
<o-dd:datetime>
<o-dd:start>0001-01-01T00:00:00</o-dd:start>
<o-dd:end>2005-06-02T15:50:00</o-dd:end>
</o-dd:datetime>
</o-ex:constraint></o-dd:play>
</o-ex:permission>
</o-ex:agreement>
</o-ex:rights>

--milou
Content-Type: audio/midi
Content-ID: <MilouExjobb@milou.se>
Content-Transfer-Encoding: binary

MThd   ▯   ▯▯àMTrk  ▯r Ö%▯▯▯▯ ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯ ▯▯▯▯▯▯▯▯▯▯▯▯÷ ÿ×▯▯▯▯▯▯
'>P 'BP "J` "Bc ™$Y ™"l<™"  ™$ ▯z'B  '9 ▯'>  ▯J P"B  "J P'9F 'BP '>P "J
•@p▯'-\ '9F '>P 'BP ™(n ™"s ™6s(">  "> ▯™6  ™"  ™( P•@  •< ▯•;Z •?pF'-
▯▯Csx'>  '& ▯▯C  'B ▯'9  ▯Cs▯ '&X '9F '>P 'BP ™(n ™"p ™6f<™6  ™"  ™( P▯
™"  ™(  ™$ ▯▯'+ x'; ▯'> ▯'C ▯—> x"C  ▯G  ▯Ed '"\ ';? '>P 'BP "E^ —>^ ™$
-@N -4D ™$R ™(d ™"j ™6p<™6  ™"  ™(  ™$ ▯H'@ ('<  '& ▯'9 ▯'$HP'$ PÿX▯▯▯▯
--milou--
```

Figur 17: Ett DRM-meddelande av typen Combined Delivery

GET-requesten får ett svar identiskt med det i fallet Forward Lock. Det nedladdade innehållet skiljer sig dock åt på flera punkter. Resultatet och namnet på metoden stämmer bra överens, det blir ett meddelande innehållande två objekt. Ur rättigheterna kan tydas hur filen begränsas. Elementet `<o-dd:play>` beskriver rättigheter att rendera innehållet på sättet spela upp. I en mobiltelefon innebär det att dess uppspelningsenhet för ljud får tillgång att spela upp innehållet. Restriktionerna som ges måste dock vara uppfyllda för att uppspelningen får ske. `<o-dd:count>` har värdet fyra vilket blir antalet gånger ljudsnutten får spelas upp. `<o-dd:datetime>` antyder att det finns även restriktioner på när uppspelning får göras. I detta exempel får det ske från nedladdning till 2/6-2005. I de fall då det finns mer än en restriktion måste samtliga vara uppfyllda för att ge mobiltelefonen rätt att rendera innehållet. Denna midi-fil får alltså spelas upp fyra gånger innan 2/6-2005. Ett exempel på restriktioner där alla specificerade begränsningar har åberopats kan ses i Figur 18.

```
<o-dd:play>
<o-ex:constraint>
<o-dd:count>100</o-dd:count>
<o-dd:datetime>
<o-dd:start>2005-01-27T13:30:00</o-dd:start>
<o-dd:end>2006-06-02T14:50:00</o-dd:end>
</o-dd:datetime>
<o-dd:interval>P1Y1M13DT22H13M7S</o-dd:interval>
</o-ex:constraint>
</o-dd:play>
```

Figur 18: Exempel på restriktioner

Exemplet innehåller fyra olika restriktioner, antal gånger, start- och sluttid samt tidsintervall. Tidsintervallet ger rätten till åtkomst i ett år, en månad, tretton dagar, 22 timmar, tretton minuter samt sju sekunder från det att nedladdning skedde. Detta exempel gick ut på att visa alla restriktioner, det finns inget behov av att ha både starttid och intervall samtidigt då det ena kommer att löpa ut före det andra. Som tidigare angivits finns även elementen `<o-dd:display>`, `<o-dd:execute>` och `<o-dd:print>`, analogt följer att samma typer av rättigheter kan definieras även för dessa element.

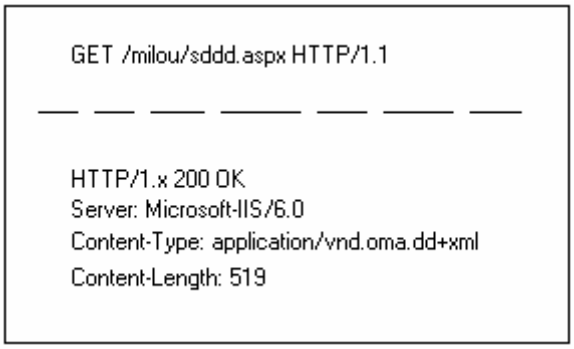
I det fall **cddd.aspx** har använts, OMA Download, sker nedladdningen enligt förfarandet som förklarades i avsnitt 4.2, Forward Lock.

Implementationen av Combined Delivery tillförde kunskapen och möjligheten att skapa ett rättighetsobjekt, detta används även till metoden Separate Delivery för att uttrycka begränsningar i möjligheten till rendering.

4.4 Separate Delivery

Separate Delivery kräver mer från backend än de båda tidigare DRM-skydden. Frontend är i stort sett likadant som i fallet med Combined Delivery. Separate Delivery erbjuder ett ökat skydd till materialet då det krypteras. DRM Content Format, DCF, beskriver hur ett meddelande som skall skyddas med Separate Delivery skall vara uppbyggt. Figur 7 visade vilka byggstenar som utgör en fil av typen DCF. Till valet att använda HTTP eller OMA Download för att skicka det krypterade innehållet rekommenderas OMA Download. Ett bra sätt att initiera pushen av rättigheterna kan tänkas vara ett positivt statusmeddelande som endast kan erhållas om OMA Download används. Implementeringen av Separate Delivery med OMA Download genererar i stort sett en likadan Download Descriptor som tidigare,

skillnaden ligger i MIMEtypen. Anrop av implementeringen Separate Delivery visas nedan i Figur 19, Figur 20, Figur 21 och Figur 22.



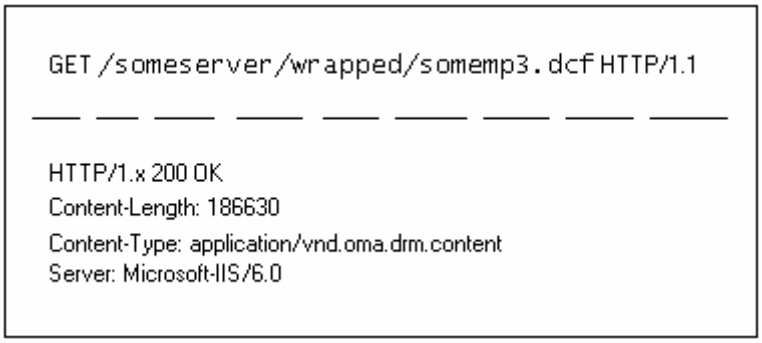
Figur 19: GET-request anrop av sddd.aspx samt svar

Precis som anrop i tidigare metoder till en *dd.aspx sida ger en GET-request ett liknande svar. I och med ovanstående anrop laddar mobiltelefonen ned en Download Descriptor, denna visas i Figur 20.

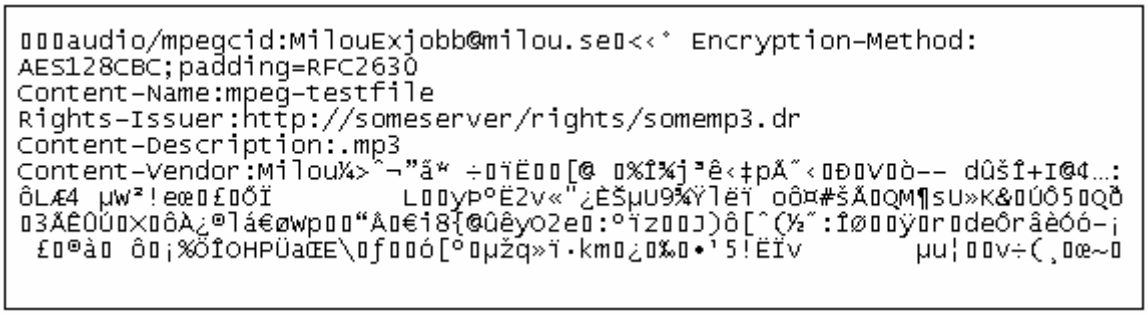
```
<media xmlns="http://www.openmobilealliance.org/xmlns/dd">
<objectURI>http://someserver/wrapped/somemp3.dcf</objectURI>
<size>186630</size>
<type>audio/mpeg</type>
<type>application/vnd.oma.drm.content</type>
<name>Milou exjobb</name>
<vendor>Milou</vendor>
<description>Optional description info</description>
<installNotifyURI>http://someserver/status.aspx?mp3.dcf</installNotifyURI>
</media>
```

Figur 20: Download Descriptorn till Separate Delivery exemplet

Download Descriptorn ger en indikation på vad det är för typ av innehåll som värdet av elementet `<objectURI>` refererar till. Typelementet visar på att det är ett innehåll skyddat av Separate Delivery. Figur 21 visar hur anropet till det värdet av `<objectURI>` sker.



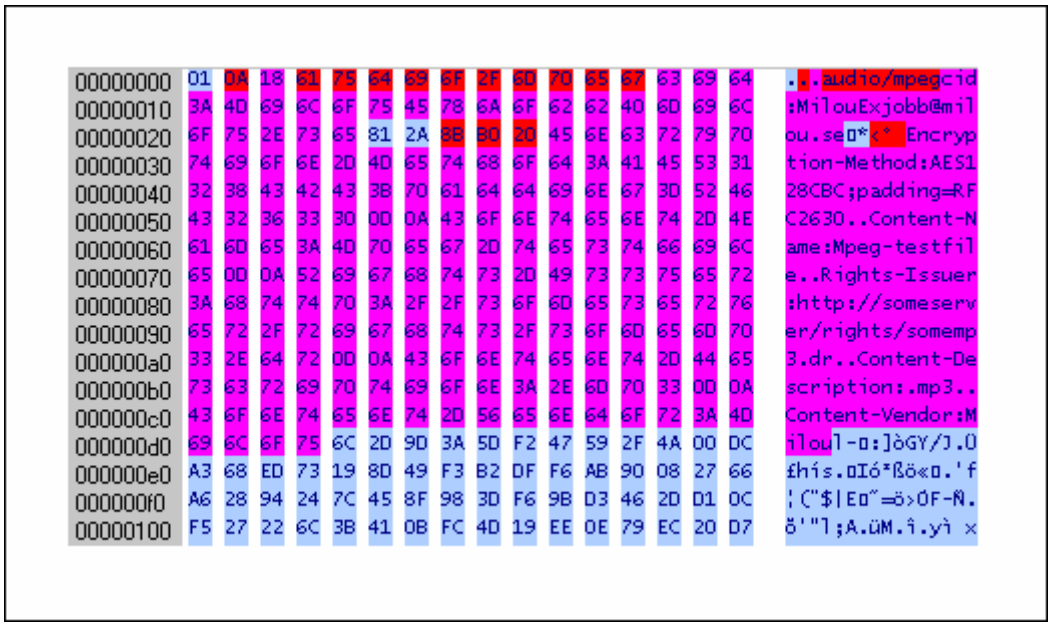
Figur 21: GET-request av fil med typen DRM Content Format samt svar



Figur 22: En krypterad fil, DRM Content Format

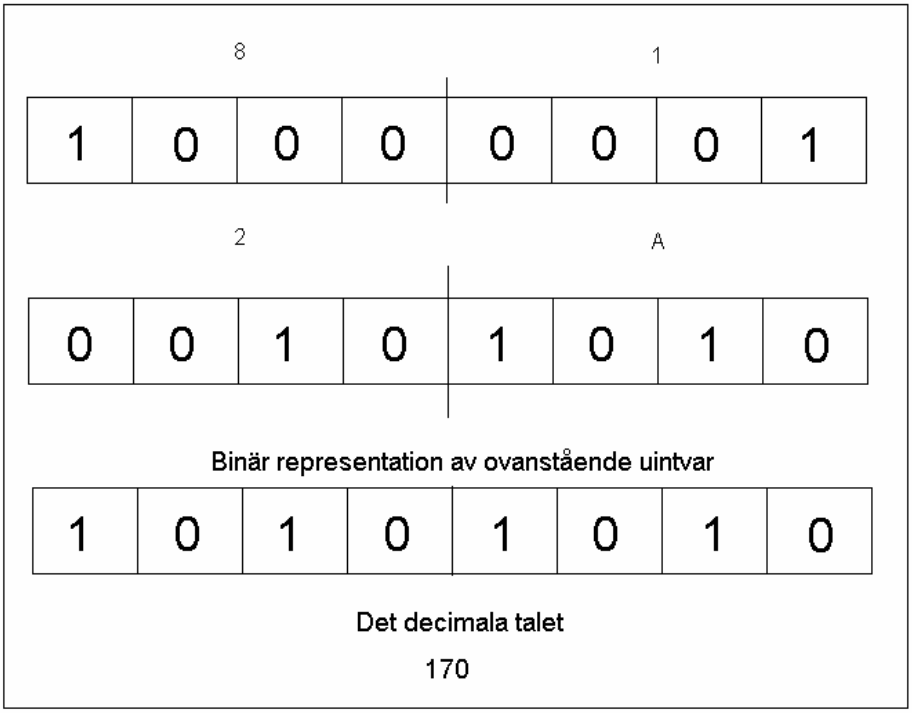
Figur 22 visar innehållet i .dcf-filen, något editerad på grund av detta dokuments begränsade bredd. Andra raden text i figuren skall bygga vidare på rad ett, längden av icke läsbar text är kraftigt bortklippt. Storleken av filen är som Download Descriptorn anger, 186630 bytes.

Den del av implementationen som skapar .dcf-filen är intressant. Backend tar emot precis som tidigare DRM-skyddsimplementationer filnamnet, dess tillhörande Content-Type samt stämpeln (värdet av elementet `<o-dd:uid>`). En filström skapas och skrivning sker till den i följdordningen specificerad av OMA. För att skapa en fil enligt DRM Content Format behöver flera detaljer tas fram, se Figur 23.



Figur 23: Hexdump av en .dcf-fil

- **Versionen** står givet i specifikationen, det skall vara **värdet** 1_{10}, därav värdet 01_{16} (nedsänkt siffra anger basen värdet uttrycks i, 10 för decimalt och 16 för hexadecimalt).
- Från inargumentet **contenttype** beräknas **antalet bokstäver** det innehåller, i detta fall 10_{10}, vilket ger värdet $0A_{16}$.
- **Längden av stämpeln** som skall matchas i rättigheterna skrivs ned till filströmmen, i detta fall 18_{16} vilket motsvarar 24_{10}.
- Den skyddade filens Content-Type skrivs till filströmmen, **audio/mpeg**.
- Stämpeln skrivs till filströmmen, **cid:MilouExjobb@milou.se**.
- **Storleken** av de **headrar** som använts tas fram och värdet beräknas med hjälp av funktioner som genererar passande uintvar representation. 170_{10} tecken användes, detta motsvaras av **812A**, se Figur 24.
- Storleken av **datan**, det vill säga filen som skall skyddas samt AES-specifika **tillägg**, representeras av värdet **8BB020**. I bytes motsvarar det 186400_{10}.
- **Aktuella headrar** som beskriver vilken typ av kryptering, vilken utfyllnadsmetod, vart nya rättigheter kan köpas och så vidare.
- Den **krypterade datan** blir den sista byggstenen till .dcf-formatet.



Figur 24: Illustration av sambandet uintvar och motsvarande binärarepresentation

Krypteringsmetoden AES fanns att tillgå ur .NETs ramverk, det tog dock tid att förstå klassen Rijndael som användes. Det tog även tid att skapa beteendet av datatypen uintvar som användes till att uttrycka storleken av både headrar och den krypterade datan.

Det går inte att jämföra två på varandra skapade .dfc-filer då nyckeln som används till dekryptering slumpmässigt skapas vid krypteringen. Det enda sättet att kontrollera om implementeringen lyckades är att testa den i mobiltelefon.

I specifikationerna till Separate Delivery skall tillhörande rättigheter skickas via WAP-push. På grund av tidsbrist i arbetet pushades aldrig detta ut, tanken var istället att mobiltelefonen skulle ansluta till URLen angiven i .dcf-filen. Detta blev en nödlösning för att se om krypteringen fungerade som den skulle.

Att implementera en sida som skall kunna läsa av inparametrarna som skickas vid ett HTTP-anrop är okomplicerat. Det som behövdes var att lyfta fram intressant information ur det som skickas med vid ett sådant. Query, Useragent samt svarskoden lyfts ut och det gav ett sätt att se vad mobiltelefonen har laddat ned, vilken typ av mobiltelefon som gjorde det och status (se Tabell 4) över installationen. Exempel på ett statusmeddelande visas i Figur 25.

```
Cellphone_status_for_file.mid.dm_with_protection_fl
SonyEricssonT610/R601 Profile/MIDP-1.0 Configuration/CLDC-1.0
900 Success
```

Figur 25: Svar från mobiltelefon

4.5 Testning

I detta arbete är det ganska lätt att testa lösningen. För att se om implementeringen fungerade laddades DRM-skyddat material ned till mobiltelefonen. Det kontrollerades om specifikationerna av OMA upprätthölls. Milou hade tillgång till ett antal olika mobiltelefoner, till dessa laddades det ned bilder, ringsignaler och så vidare. De olika typer av skydd som finns gör att det blev en hel del kombinationer som skulle testas. Om något laddats ned med Forward Lock undersöktes det om mobiltelefonen kunde skicka nedladdat material vidare eller inte. När Combined Delivery användes testades olika kombinationer av rättigheter samt att nedladdat material inte kunde skickas vidare. Med Separate Delivery testas även då hur

olika rättigheter efterföljs samt att superdistribution av materialet tillåts. All testning sammanfattades och sorterades i tabeller. Tabell 1 sammanfattar testning av skyddet Forward Lock, Tabell 2 Combined Delivery och Tabell 3 Separate Delivery.

Tillverkare	Modell	Metod	Testfall	Svar	Resultat
Motorola	V980	HTTP	mp3, jpg, amr, mid	-	OK
Motorola	V980	OMA	mp3, jpg, mid	900	OK
Samsung	**Z107**	**HTTP**	**mid, jpg**	**-**	**OK**
Samsung	**Z107**	**OMA**	**mid, jpg**	**905**	**EJ OK**
Sony Ericsson	F500i	HTTP	mid, jpg	-	OK
Sony Ericsson	F500i	OMA	mid, jpg	900	OK
Sony Ericsson	**S700i**	**HTTP**	**mp3, mid, jpg**	**-**	**OK**
Sony Ericsson	**S700i**	**OMA**	**mp3, mid, jpg**	**900**	**OK**
Sony Ericsson	T610	HTTP	mid, jpg	-	OK
Sony Ericsson	T610	OMA	mid, jpg	900	OK
Sony Ericsson	Z600	HTTP	mid, jpg	-	OK
Sony Ericsson	Z600	OMA	mid, jpg	900	OK

Tabell 1: Testutfall av metoden Forward Lock

I samtliga fall angavs URLen till sidan innehållande skyddet, i detta fall fl.aspx och fldd.aspx.

Tillverkare	Modell	Metod	Testfall	Svar	Resultat
Motorola	V980	HTTP	jpg, mid	-	OK
Motorola	V980	OMA	mp3: count, intervall, tid jpg: count, intervall, tid	900	OK
Samsung	**Z107**	**HTTP**	**mid, jpg**	**-**	**EJ OK**
Samsung	**Z107**	**OMA**	**mid, jpg**	**905**	**EJ OK**
Sony Ericsson	F500i	HTTP	mid, jpg	-	OK
Sony Ericsson	F500i	OMA	mid, jpg, mp3	900	OK
Sony Ericsson	**S700i**	**HTTP**	**mp3, mid, jpg**	**-**	**OK**
Sony Ericsson	**S700i**	**OMA**	**mp3, mid, jpg**	**900**	**OK**
Sony Ericsson	T610	HTTP	mid, jpg	-	EJ OK
Sony Ericsson	T610	OMA	mid, jpg	953	EJ OK
Sony Ericsson	**Z600**	**HTTP**	**mid, jpg**	**-**	**EJ OK**
Sony Ericsson	**Z600**	**OMA**	**mid, jpg**	**953**	**EJ OK**

Tabell 2: Testutfall av metoden Combined Delivery

Noterbart är att i fallet då Combined Delivery testades användes i de flesta fall rättigheterna antalet renderingar bestämt till fyra. Endast med Motorolas mobiltelefon skedde mer grundliga tester, detta då övriga mobiltelefoner fanns till hands under begränsad tid.

Tillverkare	Modell	Metod	Testfall	Svar	Resultat
Motorola	V980	HTTP	mp3, jpg, mid	-	OK
Motorola	V980	OMA	mp3, jpg, mid	900	OK
Samsung	**Z107**	**HTTP**	**mid, jpg**	**-**	**EJ OK**
Samsung	**Z107**	**OMA**	**mid, jpg**	**953**	**EJ OK**
Sony Ericsson	F500i	HTTP	mid, jpg	-	EJ OK
Sony Ericsson	F500i	OMA	mid, jpg	900	EJ OK
Sony Ericsson	**S700i**	**HTTP**	**mp3, mid, jpg**	**-**	**EJ OK**
Sony Ericsson	**S700i**	**OMA**	**mp3, mid, jpg**	**900**	**EJ OK**

Tabell 3: Testutfall av metoden Separate Delivery

Vad det gäller statuskoden till Separate Delivery anger den om .dcf-filen laddades ned korrekt eller inte. Resultat-kolumnen beskriver om det gick bra att rendera innehållet eller inte.

Några av de definierade svarskoder en mobiltelefon kan skicka ges i Tabell 4.

Kod	Innebörd	Förklaring
900	Success	Indikation att innehållet laddades ned och installerades lyckades.
905	**Attribute mismatch**	**Indikation på att innehållet inte stämmer överens med vad som definierats i Download Descriptorn. Mobiltelefonen accepterar inte innehållet.**
953	Non-Acceptable Content	Indikerar att efter nedladdning av innehållet men innan statusmeddelandet skickades beslöt Download agenten att mobiltelefonen inte kan använda innehållet.

Tabell 4: Statuskoders innebörd

4.6 Dokumentering

Kraven Milou ställt på dokumentationen gäller endast programkoden samt summering av testfallen. Koden har dokumenterats på så sätt att det lätt går att förstå tankegången bakom kodningen. Detta innebär en summering av varje klass, det vill säga vad som är dess syfte och vilka metoder som ingår. Till varje metod skall det finnas en summering av vad som är dess syfte, eventuella inparametrar förklaras i korthet samt beskrivning av eventuellt returvärde. Indentering är ett steg i målet med lättläslig kod och har använts med sunt förnuft. Om koden tros vara svår att förstå på vissa ställen har små kommentarer gjorts som förklarar innebörden av operationen. Att välja namn på metoder och variabler har också skett med omsorg, något som ökar förståelsen för innebörden av koden.

Testfallen är samlad information av vilka mobiltelefoner som stöder vilka DRM-metoder i denna implementering. De säger inget om huruvida tillverkaren har specificerat att det skall finnas stöd eller inte. Till testfallen anges vilken mobiltelefon som testas, vilken metod av nedladdning som använts, vad som laddats ned, vilka rättigheter som testades samt övriga kommentarer. I de fall OMA Download använts anges även statuskoden. Vid ytterligare testning är det enkelt att addera utfall till dem som redan finns dokumenterade.

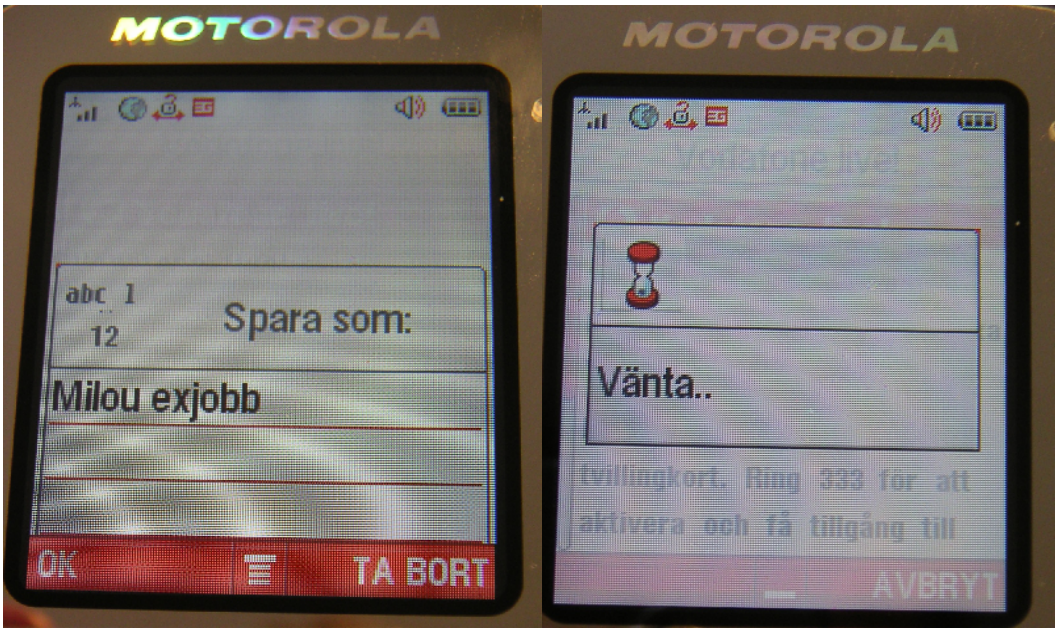
5 Resultat

5.1 Funktionalitet

Grundmålen Forward Lock och Combined Delivery uppnådde önskat resultat, de båda metoderna fungerar som det är tänkt vilket också återspeglas i resultaten av testfallen. Separate Delivery hann aldrig färdigställas enligt OMAs specifikationer vilket måste ses som ett bakslag.

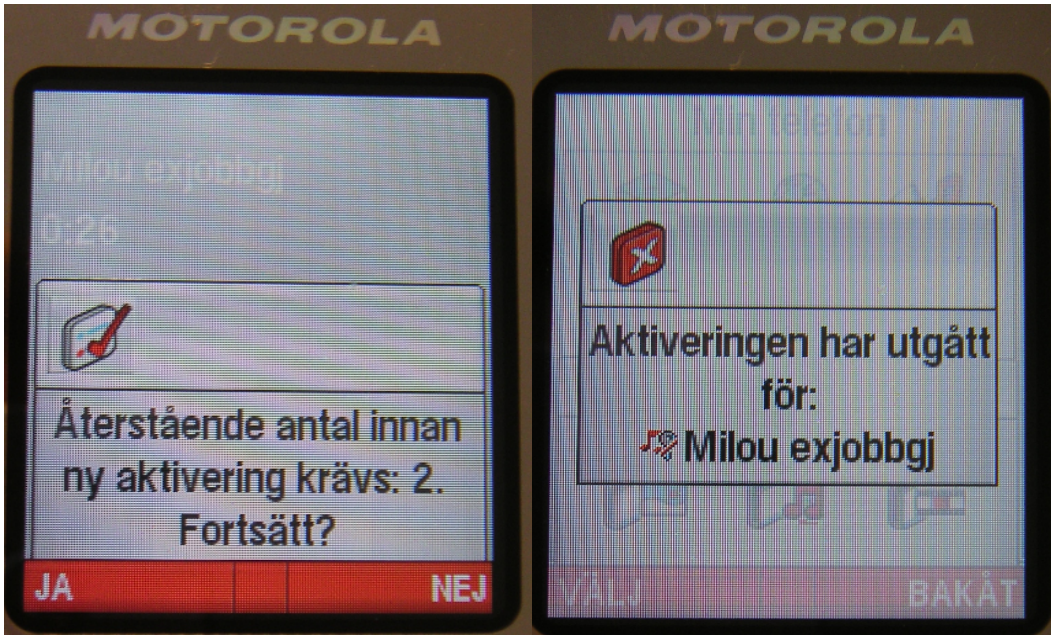
För att skapa ett DRM-meddelande, oberoende av typ, måste den som använder lösningen välja vilken fil som skall skyddas, detta görs genom att i programkoden ange filens sökväg relativt webbprojektets rotkatalog. Det finns alltså ingen möjlighet för användaren av mobiltelefonen att välja vad som skall laddas ned. Till Combined Delivery samt Separate Delivery krävs också att rättigheterna bestäms. Detta sker genom att i .aspx-sidorna, det vill säga programkoden, ange vilka typer av rättigheter som önskas till vald fil. Till vart och ett av de fyra olika renderingsmöjligheterna går det att bestämma definierade restriktioner. För att ändra fil som skall skyddas eller byta typ av rättighet måste webbprojektet uppdateras för att denna skall ses även utåt. Det finns en klass som enbart används till att definiera konstanter. I programkoden används dessa på alla ställen där det anses befogat, till exempel sökvägen till webbprojektet. På så sätt kan nya kataloger och strukturer läggas till projektet och ändring behöver bara ske på ett ställe.

Här följer bilder på vad implementationen resulterade i. I de tre fallen med OMA Download kommer följande bilder upp, se vänstra delen av Figur 26. När det gäller OMA Download skall det i denna implementation efter nedladdning skickas ett statusmeddelandet över nedladdningen. Detta visas i den högra delen av Figur 26.



Figur 26: Alternativet visas i alla tre typer av skydd med OMA Download

I de fall då rättigheter används skall mobiltelefonen se till att de efterföljs, i nedanstående fall hade antalet uppspelningar bestämts till fyra. När det återstår två uppspelningar av denna ringsignal meddelar Motorolan detta, se Figur 27 som även visar överskridningsmeddelandet.



Figur 27: Anmärkning över lågt antal uppspelningar samt felmeddelande

Fortsatta uppspelningar resulterar i att antalet uppspelningar sjunker till noll. I detta fall kan inte ringsignalen spelas upp längre.

Implementationen av Separate Delivery hanteras av mobiltelefonen trots att OMAs specifikationer inte har uppfyllts. I Figur 28 har nedladdning av en .dcf fil skett och när

uppspelning sker av ringsignalen uppger mobiltelefonen att rättigheter behöver laddas ned. Om alternativet ja anges ansluter den till Internet och laddar ned rättigheterna. Aktiveringen lyckades och följande indikation ges på det.

Figur 28: Aktiveringsfas i Separate Delivery och lyckad aktivering av rättigheterna

Något som försvårade uppgiften var att det knappast fanns några dokumenterade fall av de olika DRM-skydden att gå efter. OMAs specifikationer var inte direkt generösa på exempel som dokumenterat fungerade till mobiltelefoner. Det är värt att nämna att OMAs medlemmar har tillgång till skyddade dokument och det är möjligt att de har tillgång till sådan information. Detta ger känslan av att OMA kanske inte är så öppna ändå. De försök som gjordes att kontakta OMA för svar på ett antal frågor resulterade tyvärr i att inga svar erhölls.

5.2 Problem

Under arbetets gång har ett flertal problem eller hinder uppstått. Vissa har visat sig vara väldigt lätta att förstå och lösa men det har även uppstått svårare problem. Det övergripande problemet för implementationen av DRM-skydden har varit att få olika mobiltelefoner att fungera till samma lösning. Det har framkommit att mobiltelefoner skiljer sig hur de tolkar DRM-skydd. Vissa mobiltelefoner försöker strikt följa OMAs specifikationer medan andra frångår den och är mer förlåtande i tolkningen av DRM-meddelanden. Detta resulterar i att ett

obefintligt radavslut kan resultera i att en mobiltelefon inte kan tolka innehållet medan andra kan det.

Det största problemet som uppstod tog väldigt lång tid att hitta lösningen till. Problemet yttrade sig i att vissa mobiltelefoner inte kunde använda implementationen av DRM-skyddet Combined Delivery. Det som försvårade felsökningen var att implementationen fungerade till Motorolas mobiltelefon V980 men inte till Sony Ericssons S700i eller K700i. Efter nära en och en halv månads frustrerande felsökning kunde det lösas. Det låg två orsaker bakom problemet som innefattade dels radavslut och dels Download Descriptorn. Då Sony Ericsson är en av initiativtagarna till OMA har deras implementation av DRM fått agera mall till hur ett DRM-meddelande skall se ut. Det som förvirrade var att deras egen implementation inte hade stöd av deras mobiltelefoner. Ovanstående modeller visade ingen information förutom ett meddelande ”downloading failed”, som status angav de 952. 952 betyder att mobiltelefonen avbröt nedladdningen av mediaobjektet trots att det borde kunna renderas. Det gick inte att tyda av felmeddelandet vad som var fel, bara att något var fel. Lösningen blev att lägga till ett radavslut i DRM-meddelandet samt ta bort elementet `<type>application/vnd.oma.drm.rights+xml</type>` ur Download Descriptorn. Typelementet förvarnar mobiltelefonen vad den behöver stödja för att hantera det nedladdade materialet på ett korrekt sätt. Eftersom OMAs specifikationer anger att multipla `<type>` element skall hanteras korrekt samt att Sony Ericssons egen DRM-lösning använder ovanstående element fanns det ingenting som gjorde att misstanke uppstod. Av en ren tillfällighet råkade elementet tas bort och Combined Delivery fungerade då även på dessa mobiltelefoner. Ett mindre problem utspelade sig i att ta reda på vilka headrar som skickades till svar på mobiltelefonens förfrågan. Det uppdagades att IIS skickar ett antal onödiga headrar och en tanke uppstod att dessa skapade besvär för DRM-implementationen. Utan kunskap i hur mobiltelefoner tolkar svaren de får från webbservern fick testning och åter testning avgöra om de påverkade hanteringen av meddelanden eller inte. Den tid det tog gjorde det tyvärr inte möjligt att hinna med alla av Milou ställda mål.

5.3 Utökningar

För att få en väl fungerande DRM-lösning som skall vara gångbar bör en del utökningar implementeras. Några ändringar i hur hanteringen med den unika stämpeln skall fungera krävs för att generalisera lösningen. I det fall Separate Delivery är tänkt att användas kan en databas fungera som lagring av rättigheter med matchande stämpel samt krypteringsnyckel. Då användaren vill förnya rättigheterna till den nedladdade ringsignalen, behöver dessa tas

fram på nytt. DRM-skyddet behöver troligen anpassas efter vilken typ av mobiltelefon det är som begär nedladdning. Arbetet har visat att mobiltelefoner inte följer specifikationerna till samma grad. Detta innebär att om lösningen skall vara tillgänglig för den stora massan av mobiltelefoner måste det finnas testfall av mobiltelefoner inom tänkt målgrupp. Att generera DRM-skyddet baserat på information om vilken mobiltelefon som gör förfrågan kommer således att bli ett måste. Färdigställa metoden för WAP-push är också något som behövs göras, detta inte minst med anledning av resultaten över testfallen till metoden Separate Delivery samt specifikationerna i framtagna av OMA. Arbetet gick inte ut på att skapa en tjänst till kunder utan att skapa metoder att skydda media. För att få en lösning att uppmuntra användning behövs också något bra sätt att presentera nedladdningsalternativen på. Det vill säga någon typ av hemsida som användaren kan surfa in till och välja mellan vad som erbjuds.

6 Slutsats

Det finns en anledning till att OMAs DRM 1.0 inte har nått fas tre, det finns märkbart skillnader i hur olika mobiltelefontillverkare har valt att implementera DRM-stödet. Det borde inte vara svårt att komma överens hur DRM specifikationerna skall följas men det kanske är det egna vinstintresset som tar överhand. En annan aspekt är att eftersom utvecklingstiden för en mobiltelefon är lång och att OMAs DRM 1.0 nådde fas två relativt nyligen så stödjer inte många av dagens mobiltelefoner det. Forward Lock är den metod som helt klart har bredast stöd, till och med Sony Ericssons T610 (en två år gammal mobiltelefon) har stöd för den. Den huvudsakliga testtelefon som använts under arbetet, en Motorola V980, har hanterat samtliga DRM-skydd på ett godkänt sätt. Skillnaden i kompabiliteten skapade tvekan i hur OMAs specifikationer egentligen skulle tolkas. Detta och den sparsamma dokumenteringen som finns att tillgå försvårade uppgiften.

Referenser

[1] Open Mobile Alliance. *Digital Rights Management*. http://www.openmobilealliance.org/release_program/docs/DRM/V1_0-20040625-A/OMA-Download-DRM-V1_0-20040615-A.pdf, 2004 (hämtad 2005-01-14)

[2] Macrovision. Analog content protection for DVD. http://www.macrovision.com/products/macrovision_acp/index.shtml, 2005 (hämtad 2005-04-27)

[3] S.J Liebowitz. *The impact of reprography on the copyright*. Research and International Affairs Branch, Bureau of Corporate Affairs, Consumer and Corporate Affairs Canada, 1982.

[4] Helen Riise. Piratkopiering hotar exportörer. http://www.swedishtrade.se/svenskexport/nr4_2005/Piratkopiering.aspx, 2005 (hämtad 2005-04-27)

[5] Justitiedepartementet. *Upphovsrätten i informationssamhället - genomförande av direktiv 2001/29/EG, m.m*. Proposition Prop.2004/05:110, 10 mars 2005.

[6] S. Josefsson. *RFC 3548 - The Base16, Base32, and Base64 Data Encodings*. http://www.faqs.org/rfcs/rfc3548.html, 2003 (hämtad 2005-03-07)

[7] Balachander Krishnamurthy, Jennifer Rexford. *Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching, and Traffic Measurement*. Addison-Wesley Professional, 1st edition, 2001

[8] Sun Microsystems. *Sun Java Studio Mobility 6 2004Q2 Tutorial*. http://docs.sun.com/source/817-2343/index.html, 2004 (hämtad 2005-05-02)

[9] Open Mobile Alliance. *Rights Expression Language*. http://www.openmobilealliance.org/release_program/docs/DRM/V1_0-20040625-A/OMA-Download-DRMREL-V1_0-20040615-A.pdf, 2004 (hämtad 2005-01-15)

[10] Charles Arehart, Nirmal Chidambaram, Shashikiran Guruprasad med flera. *Professional WAP*. Wrox Press, 1st edition, 2000.

[11] Open Mobile Alliance. *DRM Content Format*. http://www.openmobilealliance.org/release_program/docs/DRM/V1_0-20040625-A/OMA-Download-DRMCF-V1_0-20040615-A.pdf, 2004 (hämtad 2005-01-15)

[12] Joan Daemen, Vincent Rijmen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer, 1st edition, 2002.

[13] Bruce Martin, Bashar Jano. WAP Binary XML Content Format. http://www.w3.org/TR/wbxml/. 1999 (hämtad 2005-03-07)

[14] Richard Anderson, Brian Francis, Alex Homer med flera. Professional ASP.NET. Wrox Press, 2001