Protecció de continguts

Toni Comerma Paré

PID_00198471

Universitat Oberta de Catalunya

Índex

Introducció

L'objectiu d'aquest mòdul és explicar quins són els diferents mecanismes disponibles per a protegir el contingut audiovisual que es distribueix per Internet cap als diferents dispositius. I per *protegir*, hi entenem controlar l'ús que els usuaris en poden fer (reproduir-lo, copiar-lo, quantes vegades, des de quins dispositius, etc.). La seguretat és un concepte molt ampli, i fins i tot si ho centrem a contingut audiovisual, ens trobarem amb un munt de casuístiques diferents.

Objectius

L'objectiu general d'aquest mòdul consisteix a tractar les casuístiques que afecten la seguretat en la distribució de continguts per a la Internet oberta als diferents dispositius (mòbils, ordinadors, televisors, etc.), deixant de banda altres camps com la protecció de contingut fora de línia (*off-line*) (DVD, Blu-ray) o els canals de televisió de pagament.

En concret, amb l'estudi d'aquest mòdul, assolireu els objectius següents:

1. Adquirir una visió tan pràctica com sigui possible, orientada a entendre la tecnologia subjacent a la protecció de continguts, però especialment saber quins productes hi ha avui en el mercat i quines possibilitats ofereix cadascun d'aquests productes.

2. Aprendre a valorar quines són les mesures de protecció necessàries en cada cas i evitar caure en una sobreprotecció dels materials que pot representar un cost i una complexitat extres.

1. Alguns aspectes de la protecció de continguts

1.1. La necessitat de seguretat

Per què hi ha aquesta necessitat de seguretat? Probablement la pregunta sembla naïf avui en dia, però també seria molt estrany començar un mòdul que parla de seguretat sense analitzar el perquè. El contingut que gestionem presenta les característiques següents:

a) Té un cost de producció, que pot ser baix (com en un vídeo domèstic) o increïblement alt (com en una superproducció de Hollywood). Això significa que algú ha invertit esforços, en temps o diners, per crear aquest contingut, i pot tenir la intenció de preservar-ne la propietat.

b) És un producte que no és material, que és simplement un conjunt de bits, i que per tant el cost de còpia és negligible i la facilitat de distribució és molt elevada. Això significa que, si algú és capaç d'obtenir una sola còpia i la publica a Internet, és pràcticament impossible aturar-ne la difusió.

Impacte i dificultat de controlar una filtració

Un exemple de l'impacte i la dificultat de controlar una filtració es pot trobar el 2009, quan es va publicar en un web una versió no acabada de la pel·lícula *X-men Wolverine* un mes abans de l'estrena, i malgrat els esforços de la productora i de l'MPAA, no se'n va aconseguir aturar la distribució.

Queda per valorar l'impacte que pot haver tingut en la recaptació de la pel·lícula aquesta filtració. És a dir, el perjudici real que va causar als ingressos aquesta filtració, a part de fer-ho al prestigi. Aquest tema és d'eterna discussió entre parts enfrontades.

c) És un producte de consum puntual. Es reprodueix una o potser dues vegades i a partir d'aquí, en la majoria de casos, perd l'interès per a l'usuari. Aquesta característica es pot observar clarament en els models de negoci que s'han desenvolupat a l'entorn d'aquest producte: els cinemes, els videoclubs, el consum per Internet han triomfat molt més que la compra de contingut (vídeos abans, DVD o Blu-ray actualment). Això requereix centrar la protecció en la reproducció del contingut més que no en la possessió.

d) És un producte que pot canviar de format. Podem trobar des de formats d'alta qualitat per a reproducció a cinemes fins a còpies de baixa qualitat per a mòbils (o còpies pirates enregistrades en un cinema d'alguna exrepública soviètica amb una càmera, de pèssima qualitat). El valor de cadascuna d'aquestes còpies és diferent, però en el fons són el mateix contingut. Els formats de més

MPAA

La Motion Picture Association of America (MPAA) és l'associació que uneix les sis principals productores de pel·lícules dels Estats Units d'Amèrica i que té com una de les activitats principals la lluita contra la pirateria.

qualitat requereixen una protecció més alta, ja que se'n poden generar fàcilment altres còpies de la mateixa qualitat o més baixa, però l'usuari s'adapta de seguida a una qualitat inferior si el cost és més baix (o gratis).

e) És un producte que no volem conservar dins una caixa forta, on seria fàcil de protegir; el volem posar a l'abast de l'usuari, però només per als usos i amb les condicions que nosaltres volem imposar. Aquí resideix el problema.

Podem concloure que hi ha una necessitat objectiva de protegir continguts i controlar-ne la difusió.

1.2. Nivells de protecció

Els continguts no tenen tots el mateix valor, i es pot assegurar que protegir té un cost, que s'incrementa a mesura que volem incrementar la seguretat.

Hem de partir de la premissa que la seguretat absoluta no existeix. En informàtica, igual que en el món real, es poden invertir esforços a aconseguir nivells de seguretat superiors, però fóra ingenu (i no haver mirat mai enrere en el temps per trobar exemples en què això ha passat) assegurar que "el sistema de seguretat X és inviolable".

Vegem què passa en el món real. És un fet que robar en un banc és més complicat que robar a casa nostra. Per què? Doncs perquè el banc ha invertit molts més diners en mesures de seguretat que nosaltres: càmeres, agents de seguretat, cambres cuirassades, alarmes, etc.

- Això vol dir que hauríem d'implantar les mateixes mesures a casa? No. Per què? Doncs perquè el valor del contingut que ens poden robar (i per tant la motivació dels lladres) és més baix i cal ponderar riscos amb costos. Les mesures de seguretat que hem d'implantar a casa han d'estar en consonància amb les mesures implantades per la resta de cases similars (per a no ser un objectiu més fàcil que la resta) i assumir que sempre hi ha uns riscos que s'han de valorar. Si intentem aplicar les mateixes mesures que un banc, no ens robaran, però no ens quedaran diners per a menjar.

- Vol dir que el banc és segur i no hi poden robar? No. Vol dir que els esforços que ha de fer algú per a robar-hi han de ser molt més grans. El banc també fa un càlcul de riscos entre el que li costa la seguretat i el valor del contingut i pren les mesures que considera adequades. Però al final, alguns bancs acaben essent robats.

- Sempre és tan difícil robar? A vegades no és que algú aconsegueixi superar les mesures de seguretat implantades per a fer un robatori, sinó que s'aprofita d'una mala implantació o d'un mal ús d'aquestes mesures. Exemples? Doncs a vegades ens hem trobat que sortint amb presses de casa hem deixat la porta oberta; aleshores tota la inversió en portes blindades i

panys de tres punts no serveix de res. O algú pot xerrar el codi per a desactivar l'alarma a un amic o en veu massa alta en un bar i arruïnar el sistema.

Aquests fets del món real tenen translació directa a la seguretat informàtica:

- El nivell de protecció del contingut ha de ser proporcional al valor del contingut, i ha d'estar alineat amb els nivells de protecció que implanten serveis similars. Veurem que hi ha diferents mesures de seguretat i que totes (o pràcticament totes) es poden vulnerar. Només que l'esforç i la complexitat per a aconseguir-ho són diferents. I les mesures a les quals no s'han trobat vulnerabilitats, probablement se n'hi acabaran trobant.

- Les mesures s'han d'implantar d'una manera tècnicament apropiada i s'ha de protegir tot el cicle de vida amb el mateix nivell de seguretat. S'ha d'evitar invertir enormes esforços a distribuir un contingut de manera segura si hem d'acabar descobrint que algú s'ha introduït per una via alternativa en els sistemes informàtics i està robant a l'origen, que no està tan protegit.

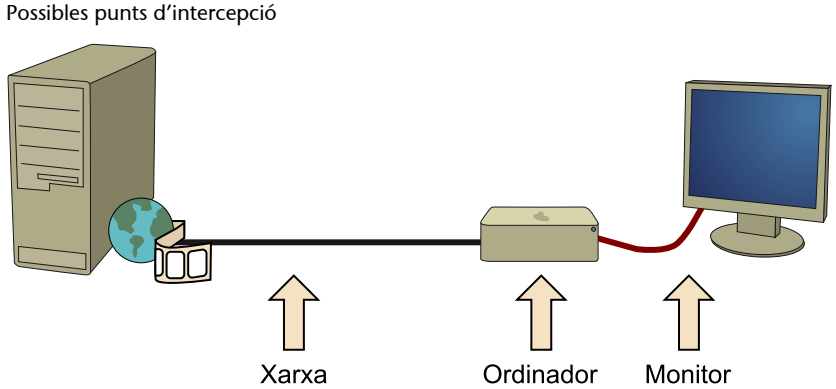
Estudiarem els diferents mecanismes de seguretat en ordre de menys a més seguretat.

1.3. Què es pot protegir

Bàsicament hi ha dos aspectes que cal protegir: les **còpies** i les **reproduccions**.

Les **còpies no controlades** són la principal font de pirateria de continguts i la principal amenaça que cal evitar.

Per a poder copiar el contingut s'ha de trobar un punt de la cadena entre l'origen i la pantalla de l'ordinador on es pugui interceptar el contingut i copiar-lo. Els punts on això és possible són aquests:



a) En la transmissió. Si la informació circula desprotegida durant la transmissió, és possible capturar-la i extreure'n el contingut.

b) En el receptor. És el punt on es pot capturar el contingut alterant el dispositiu. De possibilitats, n'hi ha moltes. Si el receptor és un ordinador, la intercepció en aquest punt és més fàcil, ja que és un equip on es pot instal·lar programari específic per a la funció. Si per contra és un telèfon intel·ligent (*smartphone*) o una televisió connectada, les possibilitats són més limitades. La majoria de serveis es basen en la reproducció directa del contingut, però alguns permeten la descàrrega del contingut a l'equip per a reproduir-lo després. Aquests serveis representen un repte extra ja que s'ha de protegir el contingut mentre és a l'equip.

Possibilitats de captura de contingut

- Sol·licitar la descàrrega des d'una altra aplicació que permeti gravar el contingut (per exemple, hi ha multitud de programes que permeten descarregar vídeos de youtube, tot i que youtube no proporciona la possibilitat de fer-ho).
- Capturar el contingut quan entra a l'ordinador interceptant el trànsit de xarxa.

c) De camí cap al monitor. És possible, finalment, interceptar el contingut quan circula des del reproductor cap al monitor. És complex, perquè calen dispositius especials, però viable. Per això els estàndards de cablejat de connexió digital incorporen mesures de seguretat per intentar evitar-ho (el mecanisme més difós és el protocol HDCP que implementen les connexions per HDMI).

Hi ha mesures per a protegir cadascun d'aquests punts.

Les **reproduccions** són un altre aspecte sobre el qual es vol establir un control.

Fins i tot si s'evita la còpia, en alguns models de negoci pot interessar controlar:

- La quantitat de vegades que es reprodueix el contingut.
- En quina franja temporal es pot reproduir.
- Des de quins dispositius es pot fer.

Els models de negoci del tipus videoclub en línia són els que estan més preocupats per aplicar aquestes limitacions.

Hi ha un altre control sobre la reproducció, més comú i estès, que és la **reproducció fora del context original**. Això s'aplica a llocs web o serveis en general que ofereixen el contingut gratuïtament, però que han desenvolupat un negoci a l'entorn d'aquest contingut, habitualment basat en publicitat, o bé volen preservar la imatge de marca o el control sobre la manera com es consumeix el contingut i on es consumeix. Aquests llocs web o serveis es poden trobar que un tercer lloc fa accessible aquest contingut des d'un servei no relacionat amb el titular del contingut. Això pot provocar pèrdues d'ingressos per publicitat, mentre que s'assumeixen els costos de distribució. Amb uns exemples ho veurem més clar.

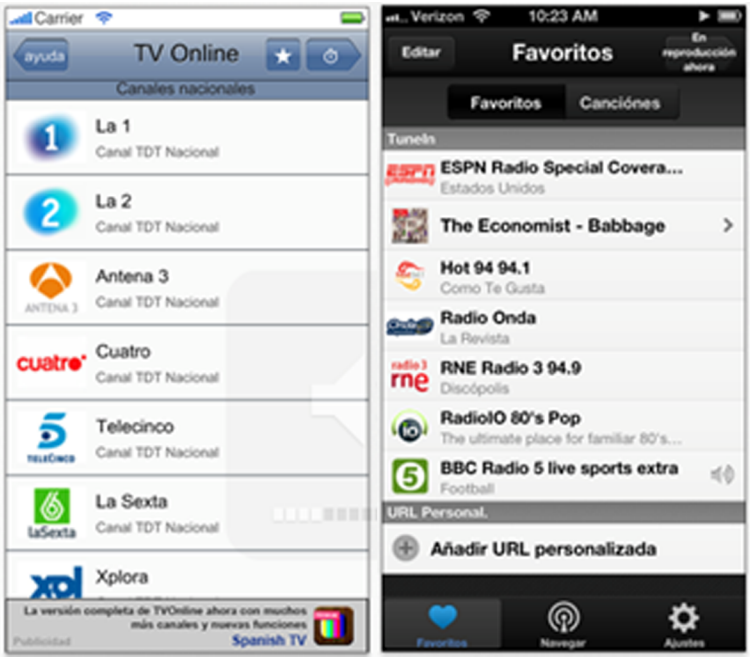
Una retransmissió de Fórmula 1

Una televisió compra els drets per emetre la Fórmula 1 per televisió i també per Internet. A part dels costos, la Formula One Group posa unes condicions estrictes sobre tots els aspectes de la distribució, incloent-hi la presència dels patrocinadors, l'aspecte visual, etc. La televisió fa l'emissió en obert i es finança amb la publicitat que ha posat en el lloc web.

Algú, però, agafa l'URL del vídeo en directe (o dels diferits publicats) i el posa en una pàgina web feta per ell amb publicitat que cobrarà ell, i ja tenim el problema plantejat. La televisió no vol impedir que els usuaris consumeixin el contingut, però es vol assegurar que el consumeixen en el lloc esperat.

Un altre exemple d'aquesta classe de reproducció són les aplicacions agregadores de canals de televisió o ràdio. Aquestes aplicacions utilitzen els fluxos de dades (*streams*) publicats per les emissores i els inclouen en la seva aplicació (de la qual obtenen ingressos per publicitat o venda), mentre que dels costos de distribució se'n fan càrrec les emissores.

Aplicacions per a iOS, TVOnline i tunein radio



En alguns casos hi ha acords entre emissores i publicadors d'aplicacions, però en altres no.

Es poden implantar mesures de seguretat orientades a controlar des d'on es fa la reproducció.

Reflexió

Paradoxes de la vida, la majoria de llocs web per a compartir vídeos (*video sharing*) –s'anomenen així els llocs web on els usuaris pugen les sèries de televisió i pel·lícules perquè les puguem veure sense pagar–, com nowvideo.eu, allmyvideos.net, implanten mecanismes de seguretat per limitar la descàrrega de contingut que no han creat ni comprat. Ho fan perquè el seu negoci es basa en ingressos de publicitat i usuaris que paguen per descarregar-se contingut. I, per tant, volen limitar el que es pot descarregar gratuïtament un usuari. Les mesures de seguretat que tenen són vulnerables (i cercant per Internet es pot trobar com es pot fer), però suficients per a una majoria d'usuaris.

No volem acabar aquesta introducció als aspectes de seguretat sense esmentar que els temes de la protecció del contingut audiovisual, la pirateria, l'adequació dels models de negoci de les empreses titulars dels drets a la realitat tecnològica i social estan de plena actualitat i és molt difícil estar-ne aliè

o no tenir-ne una posició pròpia. Però aquestes opinions no influeixen en la tècnica, i evitarem entrar-hi en aquest material. Això sí, poden ser motiu d'un interessant debat al llarg del curs.

Per a fer el tema més pràctic, en lloc de fer una explicació teòrica i després veure com s'aplica, ho farem a l'inrevés. Repassarem quines són les tecnologies que hi ha en el mercat per a protegir contingut i sobre la marxa, a mesura que en sorgeixi la necessitat, introduirem els conceptes teòrics necessaris.

2. Tecnologies de protecció de continguts

2.1. Orígens i convergència de les tecnologies de protecció de continguts

Les tecnologies de protecció de contingut audiovisual tenen dos orígens diferents, que en marquen la idiosincràsia i els punts forts i febles: les que provenen d'Internet i les que provenen del món de la televisió.

a) Tecnologies de protecció que provenen del món d'Internet

El vídeo en línia per Internet va començar pràcticament sense sistemes de protecció de continguts. Durant molts anys, la majoria de contingut s'ha emès en obert i només se n'ha protegit una petita part. Per això, els fabricants de tecnologia s'han centrat més a proporcionar eines per a desenvolupadors i poder crear una experiència agradable per a l'usuari que no pas a proveir d'eines per a protegir-ne el contingut. No volem que això s'entengui com un "no absolut"; des del principi hi ha hagut fabricants que han desenvolupat mecanismes de seguretat, però sempre han estat solucions de veta de mercat que s'han aplicat a projectes de poca repercussió per al públic en general. Actualment, però, hi ha solucions que ofereixen les màximes garanties de seguretat, equivalents a les que es poden trobar en qualsevol dispositiu.

b) Tecnologies de protecció que provenen del món de la televisió

En aquest entorn, des de fa molts anys, abans que Internet arribés al gran públic, ja hi havia canals de pagament, amb els quals es van desenvolupar mecanismes per a controlar-ne l'accés. Tecnològicament eren molt diferents dels sistemes actuals, però amb l'arribada de la digitalització i dels canals d'IPTV (Imagenio de Movistar, OrangeTV d'Orange, etc.), que comparteixen algunes característiques amb la Internet oberta, les tecnologies són cada cop més similars. Aquest entorn ha tingut sempre unes característiques concretes:

- Són negocis que mouen unes grans inversions econòmiques. Engegar un servei com un canal de televisió requereix una inversió de capital enorme i uns projectes d'enginyeria per a posar-los en servei mastodòntics.

- Es paga pel servei, i això significa que les empreses tenen uns ingressos importants que han de compensar la inversió que fan, naturalment.

- Les empreses controlen tota la cadena de valor. Des de l'emissió i la distribució fins a l'equip que ens instal·len al domicili, cosa que els permet triar

quins elements tecnològics volen utilitzar i garantir la interoperabilitat de tots aquests elements.

- El contingut que s'hi distribueix té un alt valor (com ho demostra que els usuaris estan disposats a pagar per aquest contingut) i per tant és temptador per a molta gent poder-hi accedir gratuïtament. I com dèiem, com més valor, més esforços es faran per aconseguir accedir-hi il·legítimament i més protecció cal posar-hi.

Aquest escenari ha desembocat en unes tecnologies fiables, que integren tots els elements de la cadena i d'un cost elevat.

Contraposem aquest model amb el de la Internet oberta, en què els dispositius (ordinador, telèfon intel·ligent[1], etc.) són propietat de l'usuari, amb potències de càlcul, programari, sistema operatiu instal·lat, formats, mida de pantalla, etc., força diferents. És un ecosistema molt variat, fora del control d'una única empresa, com és el cas de la televisió (quan us doneu d'alta en una televisió de pagament, us subministren un descodificador controlat per l'empresa, que no fa cap altra funció i que no es pot manipular, i si esteu donats d'alta en dos canals, teniu dos descodificadors; imaginem això a Internet i que necessitéssim un ordinador per a accedir a YouTube i un altre per a accedir a Vimeo). No és possible el control d'extrem a extrem i com que s'han adaptar a un entorn més heterogeni, compartit i controlat per l'usuari, la compatibilitat és un requeriment imprescindible.

(1)En anglès, *smartphone*.

També canvien els models de negoci (habitualment comencen amb una inversió relativament petita, i uns marges més reduïts), cosa que obliga a solucions més incrementals i d'implantació més ràpida.

I finalment també canvia el tipus d'usuari (o la seva actitud, ja que la mateixa persona no té les mateixes expectatives o els mateixos requeriments en diferents circumstàncies). Amb el seu telèfon intel·ligent o ordinador, no està disposat a renunciar al control. Vol controlar què s'hi instal·la i és reticent que el forcin a instal·lar aplicacions; tolera malament que aquestes noves aplicacions generin problemes o que li imposin restriccions extres (l'ús del navegador X, o la versió de sistema operatiu Y). Aquestes preferències de l'usuari acaben tenint molta importància a l'hora de seleccionar quina tecnologia s'ha d'utilitzar.

Convergència

Aquests dos orígens, les tecnologies de protecció de continguts provinents d'Internet i les provinents del món de la televisió, es van acostant, van convergint.

Aplicació de Youtube XL en un Sony Bravia

Això passa tant perquè els fabricants intenten ampliar el negoci cap a altres camps com perquè tecnològicament els dos mons s'acosten:

a) Les televisions de pagament es van movent cap a sistemes d'IPTV, que utilitzen la mateixa tecnologia de transmissió que Internet (IP), de manera que es produeix una convergència que els fabricants de tecnologia provinents d'Internet aprofiten per a intentar arribar a les televisions.

Aplicacions disponibles en un Samsung SmartTV

b) Les televisions es comencen a connectar a Internet –fenomen que es coneix com a televisions connectades[2]– i es pot començar a traslladar el contingut disponible a Internet cap a aquests dispositius, fet que provoca que:

[(2)]En anglès, *connected TV*.

- D'una banda, els fabricants de televisió –que tradicionalment han treballat amb els proveïdors de tecnologia de seguretat del seu entorn– els prefereixen, ja que hi ha unes relacions sòlides i experiències compartides. Per això intenten portar els proveïdors de continguts cap a aquestes televisions.

- D'altra banda, els proveïdors de continguts que vénen d'Internet, i que han estat treballant amb les tecnologies d'aquest entorn i s'hi troben més còmodes, intenten que aquests fabricants entrin en el terreny de les televisions i que els fabricants de televisió implantin tecnologies provinents d'Internet.

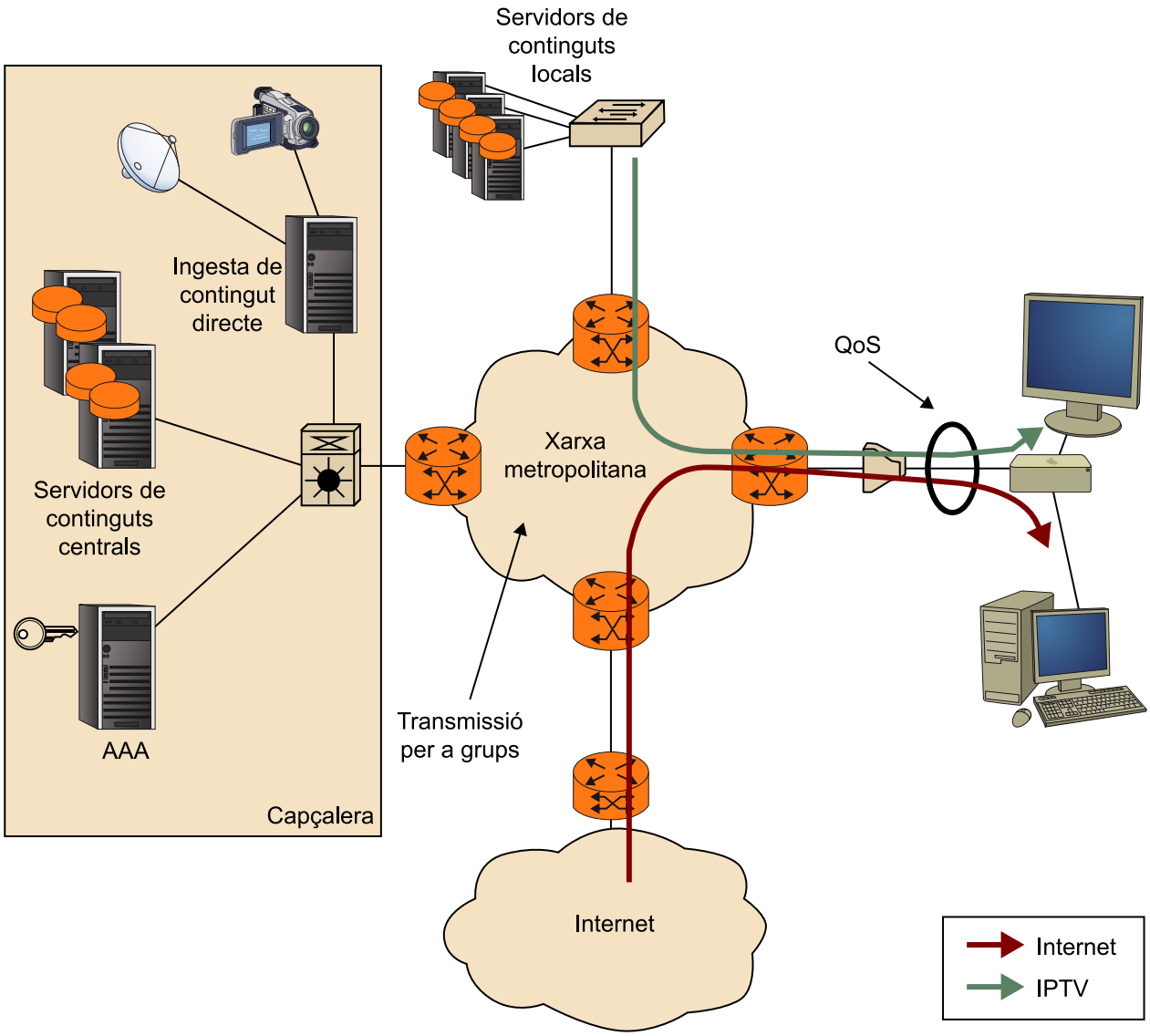
IPTV i *over the top*

Aquests dos noms defineixen dues formes diferents de transmissió de continguts cap a un televisor utilitzant xarxes de comunicació basades en IP. Tenen en comú el fet d'utilitzar IP, però a partir d'aquí tot són diferències, tant tecnològicament com de model de negoci:

1) La **IPTV** (**televisió per IP**) és un model en què l'operador de comunicacions de banda ampla (el que habitualment ens dóna accés a Internet) utilitza la seva infraestructura pròpia per a fer arribar als domicilis canals de televisió via IP per una xarxa pròpia. Arriba al receptor pel mateix cable que Internet i possiblement que el telèfon, però les infraestructures estan completament separades.

Un sistema d'IPTV com el que es veu en l'esquema següent consta d'un espai central on es gestionen tots els continguts –tant en directe com a la carta– i hi ha tota la infraestructura de gestió necessària (monitoratge, seguretat, facturació, etc.) des d'on s'envien els continguts a centres metropolitans que s'encarreguen de la distribució als usuaris finals. Aquests centres poden tenir còpies locals del contingut per a descarregar la infraestructura central i la xarxa. Finalment, l'usuari final es connecta a aquesta xarxa mitjançant l'encaminador (*router*) que té instal·lat a casa cap al DSLAM de la central més propera al domicili.

Esquema simplificat d'una xarxa d'IPTV



L'operador connecta la seva xarxa a Internet i dóna accés als usuaris autoritzats, però, com hem dit, controla tota la xarxa. Això permet dues coses fonamentals:

- Pot utilitzar dins la seva xarxa els protocols de transmissió per a grups (*multicast*), cosa que li permet una difusió dels canals en directe molt més eficient que per Internet (circula un flux de dades únic per a cada canal fins a cada node de la xarxa, en lloc d'un flux per a cada usuari). En contraposició a això, la transmissió per a grups és un protocol que no funciona amb Internet.

- Pot assignar QoS a tots el fluxos de dades, fet que li permet assegurar la qualitat de la recepció. Fins i tot en el tram final cap a l'usuari, on el trànsit d'IPTV comparteix el canal amb el d'Internet o del telèfon, es poden separar aquests canals, reservar amplada de banda per a cadascun i configurar circuits virtuals d'ATM per a cada servei.

La part negativa d'això és que, si esteu connectats al proveïdor A, no podeu utilitzar el servei de televisió del proveïdor B perquè les xarxes no es comuniquen.

2) L'***over the top*** (**OTT**), per contra, fa referència als serveis que utilitzen una infraestructura de comunicació que no tenen controlada. És el cas de qualsevol servei que utilitza Internet, des de Google fins a un blog personal, ja que tots són usuaris d'una xarxa de comunicacions gestionada per diferents empreses. La transmissió de dades en aquests entorns es caracteritza pel tant-com-puc (*best effort*), sense una garantia ni de no pèrdua de paquets ni d'amplada de banda. Encara més, les condicions de la xarxa són molt canviants.

Per exemple, es pot estar reproduint correctament un vídeo fins que un altre equip de la mateixa xarxa es posa a descarregar un programa i ocupa tota l'amplada de banda. Aquests serveis no es beneficien dels avantatges que proporciona la IPTV i han de bregar amb unes comunicacions sense QoS i sense transmissió per a grups. En contrapartida, no cal fer cap inversió en infraestructura, atès que ja està disponible, i això simplifica molt la creació de negocis, en minimitza el cost i en facilita la profusió. A més, el fet que la xarxa sigui comuna, fa que amb una única connexió es pugui accedir a múltiples serveis.

Malgrat aquestes dificultats, els serveis d'OTT funcionen i han triomfat decididament, com es pot veure en el fenomen del vídeo per Internet. Pel que fa al trànsit, algunes previsions auguren que el 2016 el 86% del trànsit d'Internet serà de vídeo (vegeu Cisco Visual Networking Index: Forecast and Methodology, 2011-2016), i ja en l'actualitat un únic servei de vídeo de pagament com NetFlix consumeix més del 30% del trànsit d'Internet als Estats Units d'Amèrica (vegeu Sandvine Global Report).

I existeix també el món dels telèfons i les tauletes[3], en el qual hi ha hagut també un xoc de tecnologies, però en aquest cas el resultat ha estat força clar. Els fabricants tradicionals de telèfons (Nokia, Ericsson, Alcatel, Motorola, etc.) s'han vist escombrats del mercat pels productes que tenen un origen en el món Internet/ordinador. Apple (amb l'iPhone) i Google (amb l'Android) han revolucionat el mercat. Ara és més important el que es pot fer amb el dispositiu que no pas les prestacions físiques que afegeix (pantalla, càmera, etc.), i això ho marca el sistema operatiu que porten dins i l'ecosistema de creadors d'aplicacions que hi ha al voltant (Apple Store, Google Play). Hi ha hagut sistemes més oberts, menys controlats pel fabricant, que han tingut èxit i han portat continguts audiovisuals a aquests dispositius bàsicament provinents d'Internet. El fet que tant els continguts com la tecnologia provinguin d'aquest entorn ha fet també que hi hagi solucions de seguretat més implantades que provinguin d'Internet.

(3)En anglès, *tablets*.

Android

L'Android no és un telèfon, sinó el sistema operatiu que fa funcionar el telèfon, però és l'element que tenen en comú tota una gamma de terminals de diferents fabricants, i el que ha fet que destaquin i tinguin èxit en el mercat.

2.2. Principals fabricants de tecnologies de protecció de continguts

Parlarem de dos tipus de fabricants de tecnologies de protecció de continguts: els que provenen del món d'Internet i els que vénen del món de la televisió.

2.2.1. Provinents del món d'Internet

Com a principals fabricants de tecnologies de protecció de continguts que provenen del món d'Internet esmentarem Adobe, Microsoft i Apple.

Adobe

Adobe és una prestigiosa empresa de programari orientada a la producció de continguts audiovisuals i a Internet amb una gamma molt àmplia de productes en què destaquen aplicacions com Photoshop, Premiere o Acrobat (PDF). Però cap d'aquestes aplicacions no ens interessa en aquest context.

A mitjan anys noranta, a l'inici de la Internet comercial, quan la web era text i imatges, l'empresa Macromedia –adquirida després per Adobe– posa al mercat el Flash Player, un complement[4] que s'incrusta al navegador web i permet la

(4)En anglès, *plug-in*.

creació d'animacions amb un entorn de desenvolupament simple i orientat als dissenyadors, els usuaris primers. Les animacions en moviment triomfen i l'ús del producte s'estén fins al punt que a hores d'ara Flash està instal·lat en gairebé tots els ordinadors. En versions posteriors, el producte evoluciona i incorpora la capacitat de reproduir vídeo i àudio i un llenguatge de programació (ActionScript) orientat a objectes. Es pot dir que crea una categoria d'aplicacions nova: les *rich internet applications*. Al mateix temps que evoluciona el Player, Adobe desenvolupa altres productes per cobrir les necessitats de distribució del contingut (Adobe Media Server) i incorpora mesures de seguretat als productes (Adobe Access).

Lectura recomanada

Podeu trobar un extens article sobre Flash Player a la Viquipèdia anglesa:

Wikipedia. "Adobe Flash".

Tot i que recentment el Flash Player, l'aplicació que s'instal·la en el dispositiu de l'usuari, ha rebut crítiques –especialment pel que fa a dispositius mòbils– i Adobe ha anunciat que deixarà de desenvolupar-lo per a aquestes plataformes (és interessant fer una cerca a Google per "flash for mobile" pel debat encès que hi ha) i que Apple va decidir vetar-lo en els seus dispositius basats en iOS, en l'entorn dels ordinadors personals és omnipresent.

El veto d'Apple a Flash

Llegiu unes declaracions de Steve Jobs sobre el suport de Flash a iOS:

Thoughts on Flash

Percentatge d'ordinadors amb Flash Player instal·lat

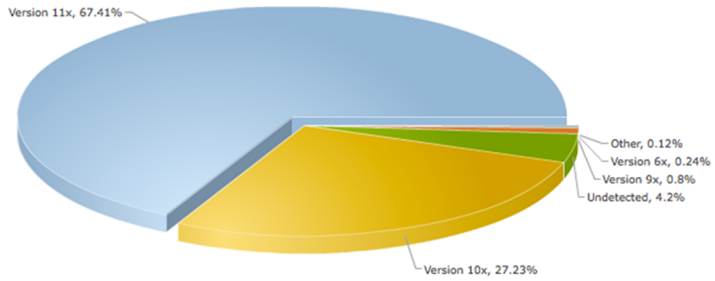
OVERALL FLASH PLAYER SUPPORT	TOTAL AVG	JAN '12	FEB '12	MAR '12	APR '12	MAY '12	JUN '12
Flash Support (all versions)	95.80%	95.98%	95.41%	95.56%	96.02%	96.00%	95.79%

Report Filters
Click here to drill-down to find the information you are looking for!

DATE RANGE	INTERVAL	CHART TYPE	BROWSER	OPERATING SYSTEM	SEARCH ENGINE	CONNECTION
Jan '12 - Jun '12	Monthly	3D Pie	All Browsers	All Operating Systems	All Search Engines	All Connections

TIMEFRAME » INTERVAL » CHART TYPE » EXPORT » SHARE »

Flash Plugin Version Support



Font: statowl.com

Els dos factors que han contribuït més a aquesta omnipresència han estat aquests:

- Va ser dels primers; va aparèixer en un moment en què els usuaris reclamaven una solució per a millorar la interactivitat de la web i fer-la més atractiva i se'n va universalitzar l'ús.

- La manera de programar, molt simple i adaptada als dissenyadors, va contribuir a crear una gran comunitat de desenvolupadors.

Aquests dos factors es van realimentar fins a posar-lo al capdavant, sense pràcticament competència.

Adobe ha sabut evolucionar i mantenir el lideratge, incorporant noves funcionalitats i desenvolupant el llenguatge de programació cap a un model orientat a objectes per atreure els desenvolupadors.

Reflexió

Actualment el Flash Player va perdent la posició de privilegi. Hi ha discussió respecte al perquè, però podríem apuntar tres grans causes:

1) Hi ha alternatives viables per a aconseguir el mateix efecte. Amb l'aparició i evolució del llenguatge JavaScript i CSS, es poden fer webs amb la mateixa interactivitat i espectacularitat que amb Flash. Justament per a reproduir vídeo és on continua essent necessari, però recentment la versió 5 de l'estàndard d'HTML ha incorporat l'etiqueta (*tag*) <video> (de la qual parlarem més endavant), que pretén resoldre aquesta carència.

2) Adobe ha fracassat en els dispositius mòbils. No ha aconseguit traslladar el producte dels ordinadors personals als telèfons mòbils i les tauletes. N'ha fet versions amb funcionalitat limitada, però que no han acabat de satisfer. Això ha portat Adobe a anunciar que deixa d'intentar portar Flash Player als telèfons mòbils.

3) No oblidem la competència del mercat. Arguments tècnics objectius a banda, Adobe ha ocupat una posició de quasi monopoli en aquest segment de mercat i això ha fet que la resta de competidors busquin maneres de descavalcar-lo.

Microsoft

Microsoft és un veterà en el sector de vídeo per Internet. Molt abans que Adobe desenvolupés els seus productes, Microsoft era –juntament amb una empresa quasi desapareguda, que també tenia una bona tecnologia, anomenada *RealNetworks*– líder en el mercat, amb un conjunt de productes que cobrien totes les necessitats, incloent-hi la protecció del contingut; però va ser relegada a una posició de comparsa amb l'aparició del Flash Player.

Microsoft, però, és una gran companyia, amb capacitat financera i d'innovació i ha tornat a la lluita amb una nova versió de la seva solució. Cal dir que, tant a principis de la dècada del 2000 com en l'actualitat, la tecnologia de Microsoft és igual o superior en prestacions a la d'Adobe. Tot el que es pot fer amb la segona es pot fer amb la primera. El que va marcar la diferència i va decantar la balança en favor d'Adobe va ser el Flash Player: la possibilitat de personalitzar la reproducció i arribar a l'audiència amb una experiència d'usuari més atractiva i integrada amb el contingut de la pàgina web.

En la imatge següent es pot veure una captura d'una antiga pàgina web amb el reproductor incrustat (emmarcat en un requadre verd). L'aspecte del reproductor era poc configurable, i la barra de controls de la part inferior era sempre la mateixa. Per no parlar de les dificultats de reproducció amb Apple o Linux, o simplement amb altres reproductors que no fossin l'Internet Explorer.

Reflexió

Aquesta situació per què va passar Microsoft, de tenir un producte tecnològicament avançat però que no coincideix amb el que l'usuari demana, és una constant en el món d'Internet. Segurament Microsoft tenia uns objectius estratègics (potenciar el sistema operatiu, diferenciar-se de la competència, qui sap) però sens dubte no hi va reeixir. En menor grau, això ens pot passar en engegar qualsevol projecte. Sigui quina sigui la nostra idea, la tecnologia que utilitzem o la visió que tinguem de les necessitats dels usuaris, si no encaixen amb la realitat, les possibilitats de reeixir-hi són minses.

Microsoft va fer un canvi d'estratègia incorporant una tecnologia equivalent al Flash Player, anomenada *Silverlight*, i donant suport al còdec H.264 per guanyar en compatibilitat, i actualment va guanyant presència. També ha fet un canvi en la tecnologia de distribució per a evolucionar cap a la reproducció en temps real adaptativa (*streaming adaptive*) i al mateix temps estendre's fora de l'àmbit dels ordinadors –cap a telèfons, tauletes i televisors. A part del reproductor, disposa de peces de programari per a tota la cadena de creació i distribució.

Apple

Apple és el tercer fabricant amb un pes important en el vídeo per Internet. Tot i tenir unes eines molt més limitades que els anteriors, té una veta de mercat important en els dispositius iOS, en què el consum de vídeo és molt important, i això el fa rellevant.

Apple va ser dels primers, juntament amb Microsoft, a tenir productes per a la distribució de vídeo per Internet, i va ser un dels que es va ajustar més als estàndards existents (utilitzava còdec MPEG-2 i després va ser dels primers a adoptar h.264 i RTSP com a protocol de transmissió), però aquests productes no van aconseguir una posició predominant en el mercat, bàsicament pel mateix problema que Microsoft, això és, requerien un reproductor Quicktime– poc personalitzable, que no permetia la personalització del Flash Player, fins que, juntament amb el llançament de l'iPhone i l'iPod, va desenvolupar un nou protocol de transmissió, l'HTTP Live Streaming (HLS).

Les opcions de protecció són més limitades que en els anteriors. Les veurem més endavant.

2.2.2. Provinents de l'entorn de difusió àmplia

Els principals fabricants de tecnologies de protecció de continguts que provenen de l'entorn de difusió àmplia[5] que esmentarem són Marlin i Widevine.

(5)En anglès, *broadcast*.

Marlin

Marlin no és pròpiament una empresa, sinó una agrupació de fabricants de televisors (Sony, Philips, Samsung i Panasonic) més una empresa d'R+D (Intertrust) que s'uneixen per impulsar conjuntament una solució tecnològica estandarditzada i segura per a la distribució de contingut audiovisual en línia a dispositius d'electrònica de consum. Amb aquest objectiu, es plantegen els objectius següents:

- Especificar un conjunt de protocols i especificacions que defineixin com han de ser les interaccions entre les diferents parts del sistema (per exemple, com s'ha de comunicar el reproductor amb el servidor de continguts, com s'ha de protegir el contingut o com s'ha de codificar el contingut).

- Crear els elements de programari que implementin l'estàndard anterior i comercialitzar-los (d'aquesta part, se n'encarrega Intertrust). Pel que fa a productes, disposa per exemple de reproductor de continguts (Wasabi Marlin, la peça de programari que s'incorpora als dispositius i reprodueix el contingut) i programari per a la distribució del contingut (Bluewhale Marlin) i per a la protecció (Octopus DRM).

El fet que separin l'especificació de la implementació permet que hi hagi desenvolupaments de tercers que puguin interoperar amb els seus, i està aconseguint guanyar presència en el món de les televisions connectades.

Recentment l'Associació Espanyola d'Empreses de Televisió Interactiva (AEDETI) ha escollit Marlin i Microsoft com a productes per a implantar la seguretat en les televisions connectades (DRM[6]).

(6)*DRM* és la sigla de *digital rights management* (gestió de drets digitals).

Widevine

Widevine és un fabricant amb una llarga experiència en la creació de solucions per a la indústria de l'electrònica de consum. Com Intertrust, té productes per a totes les necessitats: des del reproductor, que adapten i llicencien a cada dispositiu, fins a servidors per a la distribució i solucions de seguretat (DRM).

Recentment, Widevine ha estat adquirida per Google i el sector està a l'expectativa de què pot significar aquest moviment, tant tècnicament com estratègicament.

Lectura complementària

Podeu llegir l'anunci de l'AEDETI en l'article següent:

"La industria propone la adopción de los DRMS de Marlin y Microsoft"

Vegeu també

De la DRM en tractarem en l'apartat 4 d'aquest mòdul didàctic.

Reflexió

Una diferència entre l'entorn d'Internet i el de l'electrònica de consum és que en el d'Internet l'usuari té el control del dispositiu i la llibertat de decidir quin programari hi instal·la: els ordinadors, les tauletes i els telèfons donen la flexibilitat d'instal·lar-hi programari.

Per contra, els dispositius d'electrònica de consum són més tancats, i habitualment l'usuari no hi pot instal·lar res. Això implica també una forma diferent de relació entre els fabricants de tecnologia i els seus clients en funció de l'entorn:

En l'entorn d'Internet, els fabricants han d'intentar convèncer l'usuari o el creador d'un servei de la utilitat del seu producte. Això dóna més flexibilitat tant a usuaris com a desenvolupadors. Aquí el client és el creador d'un servei o l'usuari final.

En canvi, en l'entorn d'electrònica de consum, els fabricants com Marlin o Widevine han de convèncer el fabricant del producte (Sony, LG, Philips, etc.) perquè incorporin la seva tecnologia. Són contractes amb el fabricant, més estratègics, grans i de més import.

3. Tipus de protecció de continguts

En aquest apartat, que podem considerar el principal, analitzarem les diferents opcions que tenim per a protegir el contingut, i en valorarem els pros i els contres en cada cas.

3.1. Prevenció del *hotlinking*

El *hotlinking* és una pràctica que consisteix a enllaçar des d'un web contingut d'un altre web sense el permís d'aquell. Un exemple d'això és el cas de la Fórmula 1 que hem presentat abans, però a part de vídeo, també es pot donar amb imatges o qualsevol altre recurs. El lloc web d'origen carrega els costos de distribució (ja que cada usuari accedeix als seus servidors) però no n'obté el crèdit. És diferent de fer una incrustació[7] d'un contingut, que és una pràctica legítima ja que hi ha llocs que ho permeten i encoratgen (es pot incrustar el reproductor de YouTube a qualsevol pàgina, però el que no es pot fer és obtenir l'accés al vídeo i publicar-lo al nostre web amb un reproductor diferent).

[7] En anglès, *embed*.

Vegeu també

Vegeu el cas de la Fórmula 1 en el subapartat 1.3 d'aquest mòdul didàctic.

Tots els continguts que es publiquen en línia, tant directes com diferits, s'acaben presentant com un URL que tindrà una forma o una altra en funció del protocol que utilitzi. Vegem-ne un exemple.

Un lloc web com http://www.longtailvideo.com/jw-player/, que són els autors d'un reconegut reproductor de vídeo, tenen en aquesta pàgina un reproductor per a mostrar un vídeo d'exemple; en aquest cas no és tan important el contingut com el reproductor en si, ja que és el que ells volen mostrar.

Captura de la pàgina amb el reproductor incrustat

Si s'observa el codi font de la pàgina (amb el botó dret, quasi tots els navegadors tenen una opció que permet veure el codi font de la pàgina), enmig de tot l'HTML de la pàgina es pot trobar el codi JavaScript següent:

Codi Javascript per a incrustar el reproductor

```
<script type='text/javascript'>
jwplayer('player_1').setup({
   file: "http://content.bitsontherun.com/videos/lWMJeVvV-364767.mp4",
   width: "876",
   height: "470",
   image: "/content/images/jw-player/lWMJeVvV-876.jpg",
   logo: {
          file: "http://p.jwpcdn.com/6/0/logo.png",
          link: "http://www.longtailvideo.com/jwpabout/?a=l&v=" +
                jwplayer.version + "&m=f&e=a"
   },
   abouttext: "JW Player " + jwplayer.version,
   aboutlink: "http://www.longtailvideo.com/jwpabout/?a=r&v=" +
              jwplayer.version + "&m=f&e=a",
   sharing: {
         code: encodeURI("<iframe
     src='http://content.bitsontherun.com/videos/lWMJeVvV-364767.mp4' />"),
         link: "http://www.longtailvideo.com/jw-player/"
       }
});
</script>
```

Aquest codi, que s'executa en carregar la pàgina, fa que es mostri el reproductor. No és important entendre'l; simplement observem-ne l'URL destacat en vermell:

http://content.bitsontherun.com/videos/lWMJeVvV-364767.mp4

Aquesta URL indica el fitxer del vídeo que el reproductor invocarà per a descarregar-lo i reproduir-lo. En les pàgines HTML, com que el codi és visible, és ben fàcil inspeccionar-ne el contingut. Si es posa aquest URL en un navegador, el vídeo es reproduirà en el reproductor per defecte (i no el de longtailvideo.com), i es pot utilitzar qualsevol utilitat per a descarregar-lo al nostre ordinador.

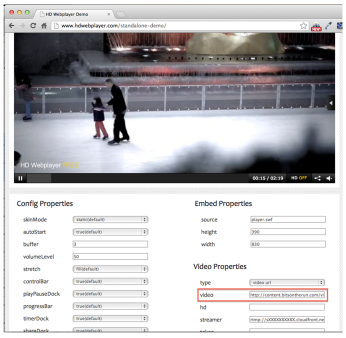
Descàrrega del fitxer utilitzant wget

```
$ wget http://content.bitsontherun.com/videos/lWMJeVvV-364767.mp4
--2013-01-25 23:45:35--  http://content.bitsontherun.com/videos/lWMJeVvV-
364767.mp4
Resolving content.bitsontherun.com (content.bitsontherun.com)..q. 66.132.221.169
Connecting to content.bitsontherun.com
(content.bitsontherun.com)|66.132.221.169|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 24377712 (23M) [video/mp4]
Saving to: 'lWMJeVvV-364767.mp4.3'

100%[=============================================================================
=====================>] 24,377,712   627KB/s   in 38s

2013-01-25 23:46:15 (627 KB/s) - 'lWMJeVvV-364767.mp4.3' saved
[24377712/24377712]
```

Vegem com es pot fer un *hotlinking* fàcilment. Anem al web d'un altre desenvolupador de reproductors (*players*), HDwebplayer; tenen una pàgina que permet provar les característiques del reproductor:



Reproductor reproduint un vídeo d'un altre lloc web

http://www.hdwebplayer.com/standalone-demo/

Si en aquesta pàgina introduïm aquest URL en la casella *Vídeo*, podrem reproduir el vídeo en aquest reproductor.

I podeu provar a incrustar el reproductor en una pàgina web amb el codi que proporciona la mateixa pàgina:

Codi per a incrustar HDwebplayer en una pàgina web

```
<object id="player" classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
name="player" width="830" height="390">
<param name="movie" value="http://www.hdwebplayer.com/standalone-
demo/components/com_webplayer/player.swf"/>
<param name="wmode" value="opaque" />
<param name="allowfullscreen" value="true" />
<param name="flashvars" value="&skinMode=static&autoStart=true&volumeDock=true&
playPauseDock=true&timerDock=true&controlBar=true&shareDock=true&
stretch=fill&logoPosition=center&
video=http://content.bitsontherun.com/videos/lWMJeVvV-
364767.mp4&title=undefined&buffer=3&volumeLevel=50" />
<param name="allowscriptaccess" value="always" />
<object type="application/x-shockwave-flash"
data="http://www.hdwebplayer.com/standalone-
demo/components/com_webplayer/player.swf"  width="830"  height="390">
<param name="movie" value="http://www.hdwebplayer.com/standalone-
demo/components/com_webplayer/player.swf"  />
<param name="wmode" value="opaque" />
<param name="flashvars" value="&skinMode=static&autoStart=true&volumeDock=true&
playPauseDock=true&timerDock=true&controlBar=true&shareDock=true&
stretch=fill&logoPosition=center&
video=http://content.bitsontherun.com/videos/lWMJeVvV-
364767.mp4&title=undefined&buffer=3&volumeLevel=50" />
<param name="allowscriptaccess" value="always" />
</object></object>
```

En un dispositiu mòbil o en un televisor connectat no és tan fàcil, ja que probablement no es pugui veure el codi font, però hi ha un munt d'eines per a inspeccionar el trànsit HTTP entre l'equip i els servidors (per exemple, configurant un servidor intermediari[8] per a la inspecció de trànsit com Paros o ZAProxy).

(8)En anglès, *proxy*.

Com a conclusió, és molt fàcil que algú pugui utilitzar el contingut que s'envia des dels equips d'un servei per a reproduir-lo en un altre. Amb aquesta tècnica no es vol impedir la reproducció, sinó assegurar que només es fa des del lloc web o l'aplicació de l'emissor, i evitar que altres webs pirategin el contingut.

3.1.1. Verificació del reproductor

La verificació del reproductor és una tècnica creada per Adobe que intenta assegurar que el contingut només es reprodueix des d'un reproductor concret. El funcionament de la verificació del reproductor[9] és simple:

(9)En anglès, *swf verification*.

Més detalls

Trobareu més detalls de la manera com es configura una verificació del reproductor en un apunt (*post*) de l'Adobe Developer Connection:
"SWF verification for Protected HTTP Dynamic Streaming"

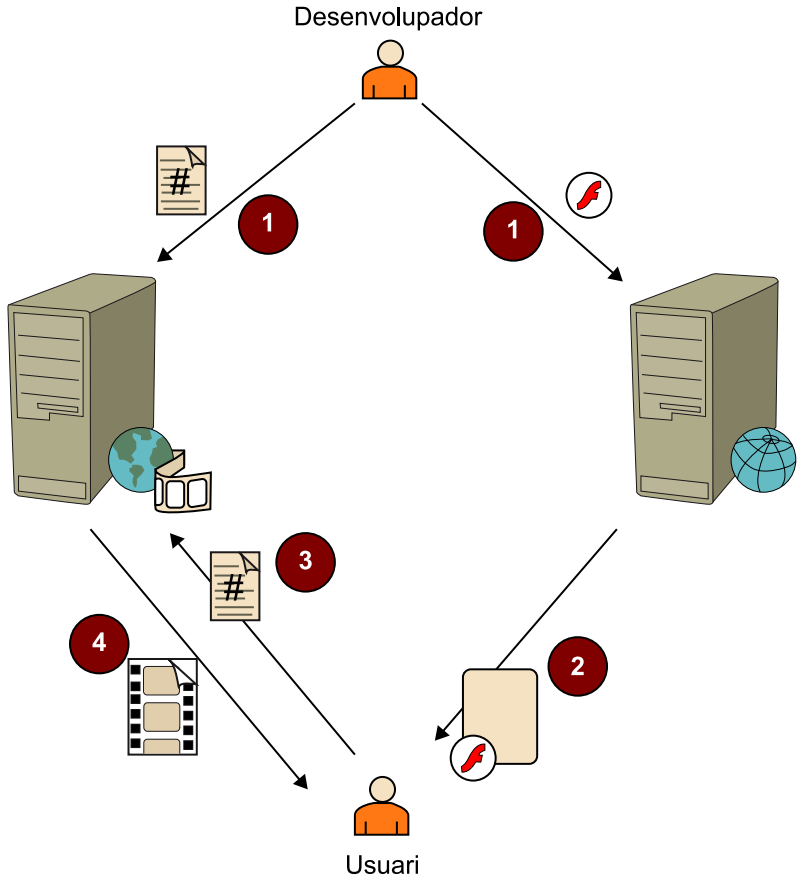
- El desenvolupador que crea el reproductor genera una "signatura" utilitzant una eina subministrada pel proveïdor. La signatura consisteix en una funció resum[10] (SHA256) del reproductor, i la diposita al servidor que distribueix els vídeos (1). El reproductor s'instal·la al servidor web (1).

(10)En anglès, funció *hash*.

- Quan l'usuari que ha navegat al lloc web (2) utilitza el Player per a accedir al vídeo, aquest envia al servidor del contingut una "signatura" generada a partir del Player (3).

- Si aquesta signatura coincideix amb una de les que hi ha emmagatzemades en el servidor, s'autoritza la connexió (4). Altrament es denega l'accés.

Cicle d'ús de la verificació del reproductor



Aquesta tècnica requereix l'ús dels productes següents:

- Un reproductor basat en Flash.
- Un servidor d'*streaming* Adobe Flash Media Server.
- Un protocol de transmissió propietari d'Adobe RTMP (o alguna de les variants que té).

Pros:

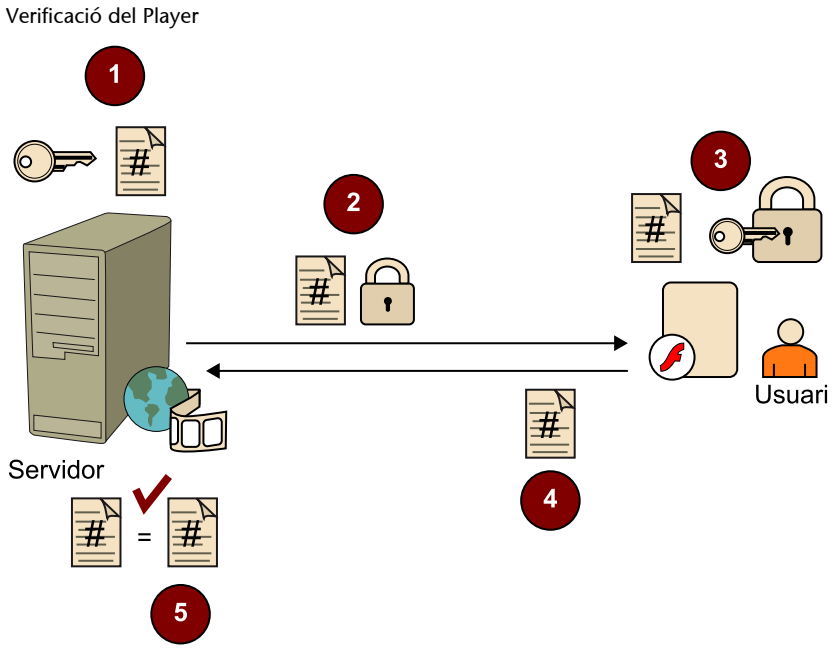
- És fàcil d'implementar. No requereix canvis en el desenvolupament ni implica limitacions addicionals.

Contres:

- Està limitat a l'ús de servidors d'Adobe.

3.1.2. Testimoni de seguretat

El testimoni de seguretat[11] és una solució similar a l'anterior, però desenvolupada per Wowza[12]. Es basa en l'intercanvi d'una informació testimoni[13] xifrada entre servidor i reproductor mitjançant una clau compartida que tenen tant el servidor com el reproductor.



El servidor genera un codi que és encriptat amb una clau (1) i s'envia al client a l'inici de la connexió (2). El reproductor, utilitzant la mateixa clau, extreu el codi (3) i el retorna al servidor (4), que comprova si és igual que la que ell ha enviat (5) i autoritza o no la connexió.

Aquesta tècnica requereix l'ús dels productes següents:

- Un reproductor basat en Flash.
- Un servidor d'*strearming* Wowza Media Server.
- Un protocol de transmissió propietari d'Adobe RTMP (o alguna de les variants que té).

Pros:

- Impedeix el *hotlinking*.

Contres:

- Cal modificar el reproductor per a implantar aquesta tècnica.

(11) En anglès, *secure token*.

(12) Wowza Media Server és un servidor d'*streaming* compatible amb Microsoft, Adobe i Apple.

(13) En anglès, *token*.

Més detalls

Trobareu més detalls de la manera com es pot implementar la tècnica del testimoni de seguretat en l'article següent:

"How to protect RTMP streaming using SecureToken (ModuleSecureToken)"

- El codi Flash és fàcil de descompilar, i per tant d'accedir al codi font i trobar la clau utilitzada per a desxifrar. Amb això es pot violar la seguretat. Es pot millorar la seguretat utilitzant ofuscadors de codi, que fan més difícil el procés de descompilar, però no impossible.

Implementació de la descompilació

Una cerca de Google per "swf decompiler" retorna tot el necessari per a saltar la protecció.

- Està limitat a l'ús de servidors de Wowza.

3.1.3. Desenvolupaments a mida

Per a tecnologies que no implementen aquestes opcions, es poden desenvolupar solucions com el testimoni de seguretat sense gaire dificultat, afegint codi desenvolupat a mida en servidor i client, però amb uns nivells de protecció baixos.

Com a conclusió, el *hotlinking* es pot prevenir fins a un nivell, però les tècniques exposades, sense combinar-les amb d'altres, no ofereixen un nivell de seguretat important. Aquests desenvolupaments a mida estan pensats per a evitar el *hotlinking*, no pas la còpia del contingut.

3.2. Restricció de domini

La restricció de domini és una problemàtica similar a l'anterior. Pretén evitar el cas que algú pugui agafar el codi que permet incrustar el reproductor en una pàgina web i copiar-lo en un altre lloc. En aquest cas les tècniques de prevenció del *hotlinking* no funcionarien (ja que no s'intenta evitar el reproductor; s'utilitza el reproductor legítim, però des d'un altre lloc web). Aquí apareixen les restriccions de domini, que es basen a especiar una llista de dominis "de confiança" des dels quals es permet accedir al contingut. Aquesta tècnica s'implementa de diferents maneres en funció de la tecnologia utilitzada.

3.2.1. Flash

Una aplicació basada en Flash, abans d'establir una connexió a un servidor diferent del servidor des del qual s'ha descarregat el codi, fa una petició HTTP en què sol·licita un fitxer anomenat crossdomain.xml, que ha de ser a l'arrel del servidor. En aquest fitxer s'indiquen la llista de dominis autoritzats. El complement de Flash fa la comprovació, i si no apareix el domini de la pàgina en què està incrustat, rebutja establir la connexió. Aquesta protecció és aplicable, no solament a connexions per a accedir a contingut audiovisual, sinó també a qualsevol tipus de connexió.

Més detalls

Els detalls de la sintaxi del fitxer crossdomain.xml els podeu trobar a l'adreça següent:

"Cross-domain policy for Flash movies"

3.2.2. Microsoft Silverlight

El complement de Microsoft incorpora una solució similar, només que en aquest cas el fitxer s'anomena clientaccesspolicy.xml.

Més detalls

Trobareu més detalls sobre el fitxer clientaccesspolicy.xml a l'adreça següent:

"Making a Service Available Across Domain Boundaries"

Aquestes dues solucions es basen en el fet que el complement (la peça de programari instal·lada a l'equip de l'usuari i subministrada pel fabricant) implementa aquesta seguretat i rebutja fer accions no autoritzades. Això és un fet molt habitual (per exemple, els navegadors tenen restriccions similars respecte a quines connexions poden fer els programes en Javascript que executen) però té les limitacions que ja hem vist, i és que, si s'accedeix al contingut sense utilitzar els complements corresponents, aquests mecanismes no protegeixen. Unes eines de descàrrega com rtmpdump o wget poden ser suficients per a saltar la protecció. Són un primer nivell de protecció, però per si es vol seguretat, cal combinar-les amb altres proteccions.

3.2.3. Comprovació del Referer

Hi ha una altra solució per als casos en què:

- el reproductor no està basat en un dels complements anteriors (com Javascript, o una aplicació de mòbil, una consola o un televisor connectat);
- el protocol de transmissió és HTTP, i per tant, el servidor és un servidor web genèric (Apache, nignx, etc.).

Per a entendre bé la base d'aquesta tècnica cal entendre un aspecte determinat del protocol HTTP, que és el que utilitzen els navegadors. El navegador emmagatzema l'URL de la pàgina que presenta a l'usuari (i posa aquesta informació a disposició del codi que s'executa dins la pàgina, com hem vist en el cas anterior), però en totes les crides HTTP que fa mentre s'està en la pàgina, aquesta informació s'envia al servidor en la capçalera del missatge, dins el camp Referer.

Exemple de capçalera HTTP amb Referer

```
GET /dd361754.MVP_DKurata.jpg HTTP/1.1
Host: i.msdn.microsoft.com
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_8_2) AppleWebKit/537.17
(KHTML, like Gecko) Chrome/24.0.1312.56 Safari/537.17
Accept: */*
Referer: http://msdn.microsoft.com/en-US/aa497440
Accept-Encoding: gzip,deflate,sdch
Accept-Language: en-US,en;q=0.8
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.3
```

En l'exemple anterior es pot veure una crida per a obtenir una imatge jpg, i entre altra informació, apareix el camp Referer amb un URL que correspon a la pàgina activa.

Amb aquesta informació, el servidor utilitza regles de configuració (que la pràctica totalitat de servidors permeten) o desenvolupa una petita aplicació per a identificar d'on arriba la petició, i l'accepta o no, de manera que obté un resultat com l'anterior.

L'exemple següent és una mostra de la manera com es limitaria l'accés als fitxers amb extensions .mp4 i .mp3 d'un servidor només a peticions que arribin de www.elmeudomini.com.

Regles per a limitar l'accés per Referer a Apache

```
RewriteCond %{HTTP_REFERER} !www.elmeudomini.com [NC]
RewriteRule \.(mp4|mp3)$    -    [F,NC]
```

Aquesta protecció funciona correctament mentre el client tingui el comportament esperat. Això és així habitualment, però és molt fàcil, utilitzant eines disponibles per a tothom, enviar peticions a un servidor i alterar el camp Referer a voluntat i saltar-se així la protecció. Com les anteriors, és una protecció per a l'usuari habitual, però no es pot considerar que ofereixi protecció contra un usuari amb uns coneixements tècnics mínims. Per exemple, utilitzant curl, una utilitat molt popular i disponible per a quasi tots els sistemes, es pot fer una petició per a canviar el Referer com segueix:

Petició amb un camp Referer inventat

```
curl --referer http://www.elmeudomini.com/inventada.html -o video.mp4
http://www.elmeudomini.com/video.mp4
```

3.3. Encriptació

Un altre punt de vulnerabilitat és el camí entre el servidor i el reproductor. Els bytes que circulen passen per múltiples xarxes, cap de les quals està sota el control de l'emissor, de manera que són punts en què el contingut pot ser interceptat i copiat. Per a evitar aquest problema, la solució és encriptar la comunicació entre aquests dos punts.

El procés d'encriptació varia en funció del protocol utilitzat, però generalment cal encriptar:

- RTMP per a reproducció en temps real[14] amb tecnologia Flash.
- HTTP per a HLS, Smooth Streaming, descàrrega progressiva o Adobe HDS.

(14)En anglès, *streaming*.

La tendència actual del mercat

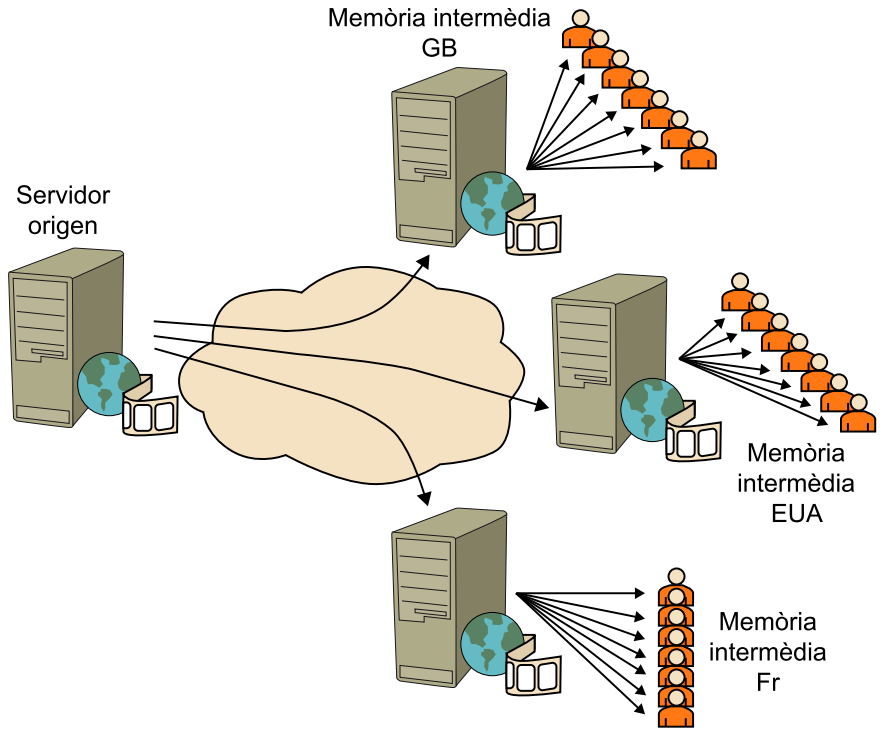
Actualment hi ha una tendència en el mercat d'evolucionar vers la descàrrega en temps real per Internet basada en HTTP i abandonar protocols dissenyats nativament per a descàrrega en temps real –com RTSP, RTMP (Adobe) o MMS (Microsoft). Aquest fet es dóna perquè HTTP és el protocol més utilitzat a Internet i hi ha una infraestructura de servidors gegantesca que pot distribuir contingut utilitzant aquest protocol. Si els protocols de reproducció en temps real utilitzen HTTP, es pot utilitzar aquesta infraestructura comuna

i no és necessari instal·lar i mantenir equipament diferent per a cada tecnologia. Això és un gran estalvi per a empreses com les CDN (*content distribution network*), que han d'invertir molts diners per a donar serveis de reproducció en temps real.

El fet que diferents tecnologies utilitzin HTTP no pressuposa que siguin iguals o compatibles. Cada fabricant desenvolupa una solució diferent, però totes les solucions tenen en comú que poden compartir una base de servidors HTTP estàndard.

L'altre gran avantatge d'HTTP és que funciona amb el *caching* de continguts, és a dir, es poden emmagatzemar còpies dels vídeos (complets o parts) en servidors prop de l'usuari final i així minimitzar els costos de transmissió de dades i la càrrega de la xarxa, cosa que permet una reducció de costos i una millora de la qualitat.

Ús de memòria cau (*cache*) per a optimitzar la distribució



S'entén més bé amb un exemple. Suposem que volem distribuir un contingut des d'un servidor situat a Barcelona, però una bona part dels nostres usuaris són a altres països. Això significaria que per cada usuari concurrent tenim un flux de dades que travessa diversos països. Si tenim 1.000 usuaris concurrents des de França, la Gran Bretanya i els Estats Units, i el vídeo té una velocitat de transmissió (*bitrate*) de 1.000 kBps, significa 1 GBps de trànsit internacional. Si podem tenir servidors de memòria cau (*cache*) prop de l'usuari final, en el millor dels casos tindrem tres fluxos de trànsit internacional entre el servidor origen i els tres servidors de memòria cau i la resta de trànsit serà local dins de cada país.

Òbviament:

- el cost és més baix (local enfront d'internacional);
- el servidor origen no ha de resistir tota la càrrega, sinó només una part petita, cosa que en facilita l'escalabilitat;
- la qualitat del servei serà més bona, ja que una transmissió internacional té més risc de presentar problemes (pèrdua de paquets, latència, retard –*jitter*–) que una de local.

En tots dos casos, el xifratge es fa utilitzant el protocol de seguretat TLS. En el cas d'HTTP, això vol dir el ja conegut HTTPS; i en el cas d'RTMP, RTMPS (Secure RTMP).

Aquest protocol permet garantir que:

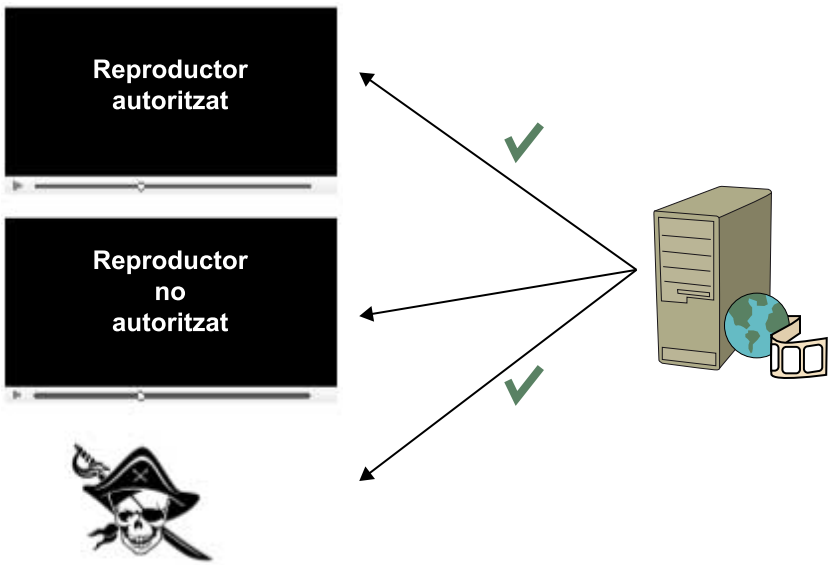
- La comunicació no és espiada.
- La comunicació no és alterada pel camí.
- L'origen és realment qui afirma que és (no hi ha usurpació d'identitat).

Malgrat això, però, l'encriptació per si sola no protegeix d'una descàrrega indeguda. El motiu és que els mateixos reproductors o aplicacions, legítims o no, poden fer crides tant a URL segures com no segures. Fent un símil per a entendre-ho, des d'un navegador podem fer una crida a http://www.google.com. Si Google decideix canviar i utilitzar un protocol xifrat (per a evitar que es puguin espiar les consultes que els usuaris fan al cercador i garantir-ne la confidencialitat[15]) i passa a **https**://www.google.com, no hi ha res que impedeixi accedir al servei. El que l'encriptació garanteix és que un no espiï l'altre, però no limita l'accés.

(15)De fet, això ja és així. En trobareu una explicació dels motius a "Making search more secure".

3.3.1. Autentificació

Les tècniques i aplicacions concretes de tècniques que hem vist fins ara tractaven tots els usuaris igual. O més ben dit, no tractaven un aspecte bàsic com el control de "qui" intenta utilitzar el servei. El resultat és que podíem controlar alguns accessos mentre s'utilitzés un reproductor de confiança o el complement (Flash o Silverlight) subministrat pel fabricant. Però si s'utilitzava alguna altra aplicació, era fàcil saltar la protecció.



La introducció d'una identificació de l'usuari, a part d'una necessitat de molts serveis, ens permet millorar el control de l'accés al contingut, ja que, a part de controlar el dispositiu que es connecta, podem validar qui ho fa.

La identificació de l'usuari s'utilitza en les situacions següents:

a) Per a la pràctica totalitat de sistemes que requereixen pagament. La majoria ho fan per una modalitat de subscripció, en què l'usuari es dóna d'alta i s'ha d'identificar cada vegada que accedeix al servei.

b) Una variant del cas anterior són els sistemes que funcionen per micropagaments, és a dir, en lloc d'haver-hi una subscripció, es fa un pagament puntual cada vegada que es vol accedir al servei (per exemple, pagar per llogar una pel·lícula), i per tant és més anònim. Aquesta modalitat es va popularitzar fa anys amb micropagaments per SMS i després ha evolucionat cap a pagaments per targeta de crèdit o PayPal. El fet de no conèixer el nom de l'usuari no té un impacte tècnic, ja que en realitat es pot considerar que tenim un usuari vàlid només per a una acció.

c) Finalment hi ha serveis que no exigeixen pagament, però sí que demanen registrar-se com a usuari, amb l'objectiu de a) poder protegir el contingut i b) fer un perfil de l'usuari i poder-li oferir contingut (i publicitat, que en el fons és el que sustenta econòmicament el servei) personalitzat.

Autentificació en HTTP

El fonament de l'autentificació en HTTP, que és un protocol sense estat, es basa en els elements següents:

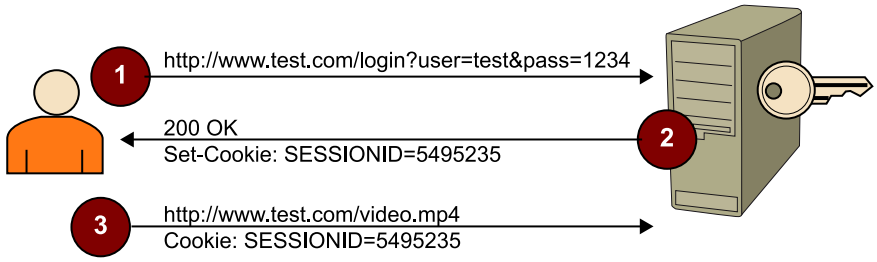
- El client envia una petició amb les dades per identificar l'usuari (normalment un nom d'usuari[16] i una contrasenya[17]) al servidor.
- El servidor valida que siguin correctes (segons la lògica de negoci pròpia de cada cas) i contesta al navegador client, adjuntant una galeta[18] que aquest navegador emmagatzemarà i enviarà en totes les peticions HTTP següents que faci a aquest servidor.
- Un cop validat l'usuari, el servidor emmagatzema una informació d'estat per recordar que l'usuari està connectat i aquesta informació està enllaçada amb el valor de la galeta, de manera que per cada petició podrà saber de quin usuari es tracta. La quantitat d'informació d'estat que tingui depèn de les necessitats de cada aplicació.

(16)En anglès, *login*.

(17)En anglès, *password*.

(18)En anglès, *cookie*.

Procés d'autentificació HTTP



Aquest és el funcionament definit en l'estàndard d'HTTP (RFC-6265), que hem simplificat per tal que sigui fàcil d'entendre. S'utilitza tant en entorns web com en mòbils o televisions connectades, ja que tots utilitzen HTTP com a base.

Més informació

Una explicació més entenedora del funcionament de les galetes la podeu trobar a la Wikipedia:
"HTTP cookie"

Un cop feta aquesta autentificació, ja tenim una seguretat de qui és el client, molt més del que podíem controlar amb els mecanismes anteriors.

L'escenari, però, no acostuma a ser tan simple, perquè habitualment el servidor que fa la validació de l'usuari (que conté l'aplicació) i els que proporcionen la reproducció en temps real no són els mateixos. Això es deu als fets següents:

- Els diversos servidors poden utilitzar diferents tecnologies. Els llenguatges que s'utilitzen per al desenvolupament web (PHP, Java, Ruby, .NET, etc.) requereixen servidors web que no han pas de coincidir amb els dels servidors d'*streaming*. A més, per a protocols diferents d'HTTP, calen servidors específics.

- La tipologia de trànsit de pàgines web i la de reproducció de vídeo en temps real són molt diferents. La primera fa moltes peticions petites, amb un volum de trànsit baix, i la segona en fa poques de gran volum. Per això és adequat separar els serveis en equips diferents.

- Per escalabilitat, acostuma a ser necessari instal·lar múltiples servidors per a la distribució de *media*.

En aquesta situació tenim els servidors d'*streaming* separats dels de l'aplicació, cosa que obliga a buscar un sistema per a validar l'accés al contingut.

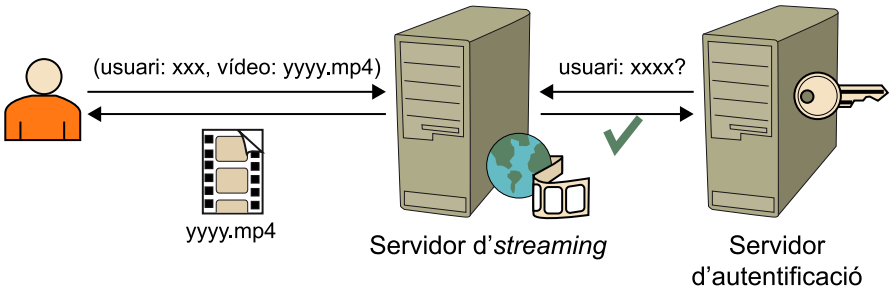
També cal afegir a això, abans de veure com es resol aquesta situació, un factor addicional. La distribució de continguts és una tasca que requereix molta amplada de banda i una gran quantitat de servidors especialitzats en funció de la tecnologia utilitzada. Una gran inversió que moltes empreses prefereixen evitar i per això lloguen el servei de distribució de continguts a empreses especialitzades anomenades *xarxes de distribució de contingut* (*CDN*). Per a poder utilitzar serveis de tercers, és molt important poder separar la lògica de l'aplicació (pròpia de cada client) de l'accés autoritzat al contingut (que pot ser comú a múltiples clients).

El que s'intenta fer és implementar un procés de validació de la manera més simple i genèrica possible, afegint una informació extra a l'URL del vídeo (en una cadena de consulta –*query string*– o en una galeta) que permeti validar l'accés. Les dues opcions més habituals són les següents:

Opció A. El servidor d'*streaming* verifica els permisos

En aquesta opció, el servidor d'*streaming* està interposat entre l'usuari i el servidor d'autentificació.

Autentificació amb servidor interposat



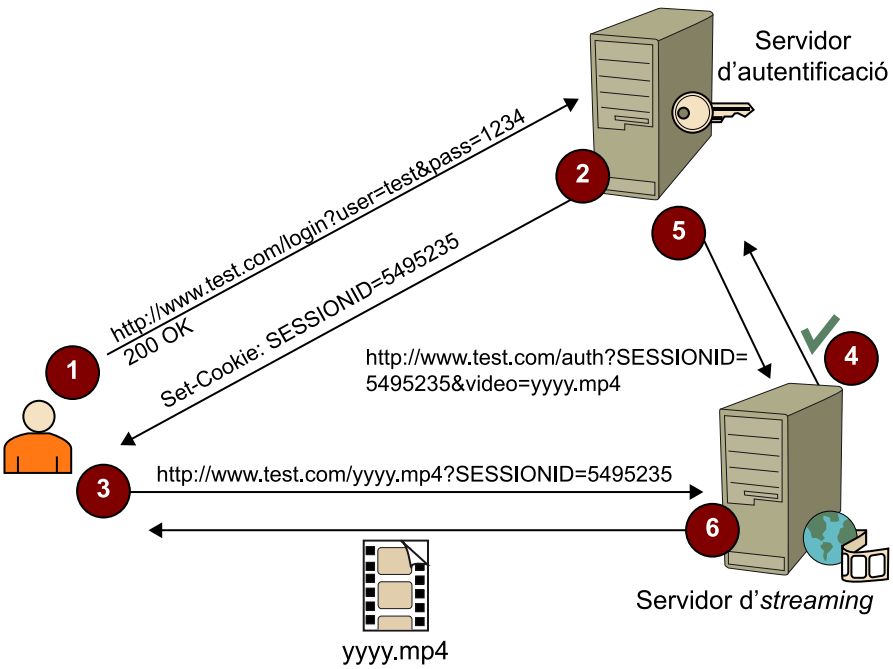
La informació que envia el client al servidor d'*streaming* pot ser un nom d'usuari més una contrasenya, però és una opció poc segura. Més aviat s'acostuma a enviar alguna identificació de sessió que el navegador ha obtingut d'una fase prèvia d'autentificació. El que és important és que el servidor d'*streaming* no té un lògica de validació d'usuaris; simplement recull els paràmetres rebuts del client i construeix una crida HTTP cap al servidor d'autentificació que retorna una resposta d'OK o KO.

Més informació

Un exemple de servidor que implementa aquesta tècnica és erlyvideo. Podeu trobar una descripció acurada del procés que descrivim aquí a:

"Authentication and authorization".

Autentificació amb servidor interposat, detallat



Quan el servidor d'*streaming* és proveït per una CDN, aquesta xarxa de distribució de contingut defineix una API amb els paràmetres que enviaran al servidor d'autentificació i la resposta que espera; en configurar el servei, s'ha d'indicar quina és l'adreça del servidor d'autentificació del client i ja es pot operar. El problema d'aquesta opció és que si es produeixen problemes amb el servidor d'autentificació, aquests problemes es traslladen al servidor d'*streaming*. És un sistema altament acoblat.

Opció B. Autentificació per testimoni

Un mètode diferent per a aconseguir el mateix objectiu. Es basa en el fet que el servidor d'*streaming* sigui capaç de fer una validació simple de manera autònoma. Per això, el servidor d'autentificació i de reproducció en temps real comparteixen una clau que permet xifrar un missatge de manera segura:

1) El client sol·licita l'URL del vídeo.

2) Es construeix l'URL del vídeo amb una cadena de consulta[19] que conté les dades que indiquen l'accés que es concedeix al client. Per exemple (cada implementació d'aquest mètode pot variar els paràmetres):

(19)En anglès, *querystring*.

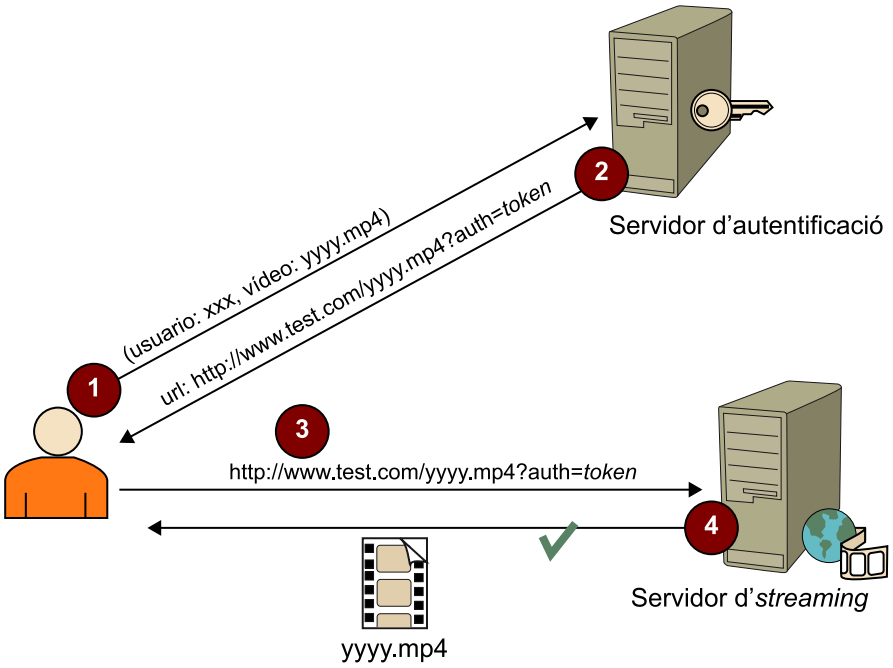
- Inici del període de visualització
- Final del període de visualització
- IP autoritzada

Aquests paràmetres (als quals a vegades s'afegeix algun text a l'atzar per a donar més solidesa al procés d'encriptació) s'encripten amb la clau compartida i s'adjunten com a cadena de consulta.

3) El client fa la crida al servidor d'*streaming* passant l'URL anterior, que desencripta la cadena de consulta amb la clau compartida i verifica si la petició compleix els criteris indicats.

4) Si és així, inicia la reproducció.

Autentificació per testimoni



En aquesta opció s'aconsegueix dotar d'autonomia el servidor d'*streaming*, que per si sol pot verificar les peticions, i s'acostuma a preferir més que la primera que hem vist.

En aquests dos esquemes, el punt de vulnerabilitat rau en la validació de l'usuari. Si podem garantir que només els usuaris vàlids poden accedir al sistema des dels reproductors vàlids, el fet d'afegir autentificació a la reproducció en temps real dóna un bon nivell de seguretat. S'ha d'evitar que es pugui obtenir una URL autentificada, ja que aleshores tindrem les mateixes possibilitats d'accés il·legítim al contingut.

Per això, cal combinar diferents mecanismes per a protegir el contingut: verificació del reproductor per a garantir que s'utilitza el client adequat i encriptació per a evitar que es pugui interceptar el contingut. Si es poden combinar les tres tècniques, aleshores es pot oferir un bon nivell de seguretat, fins i tot en serveis sense identificació nominal d'usuari:

- El servidor d'*streaming* només entregarà el contingut si hi ha testimoni, i al reproductor que tenim verificat (si el reproductor no deixa enregistrar el contingut, no hi haurà fugues).

- L'encriptació garanteix que no s'intercepta.

Tots dos sistemes presenten dificultats d'implementació en dos casos:

1) HTTP Streaming. En aquests protocols, no hi ha una única connexió entre reproductor i servidor, sinó un munt. Això implica dues coses:

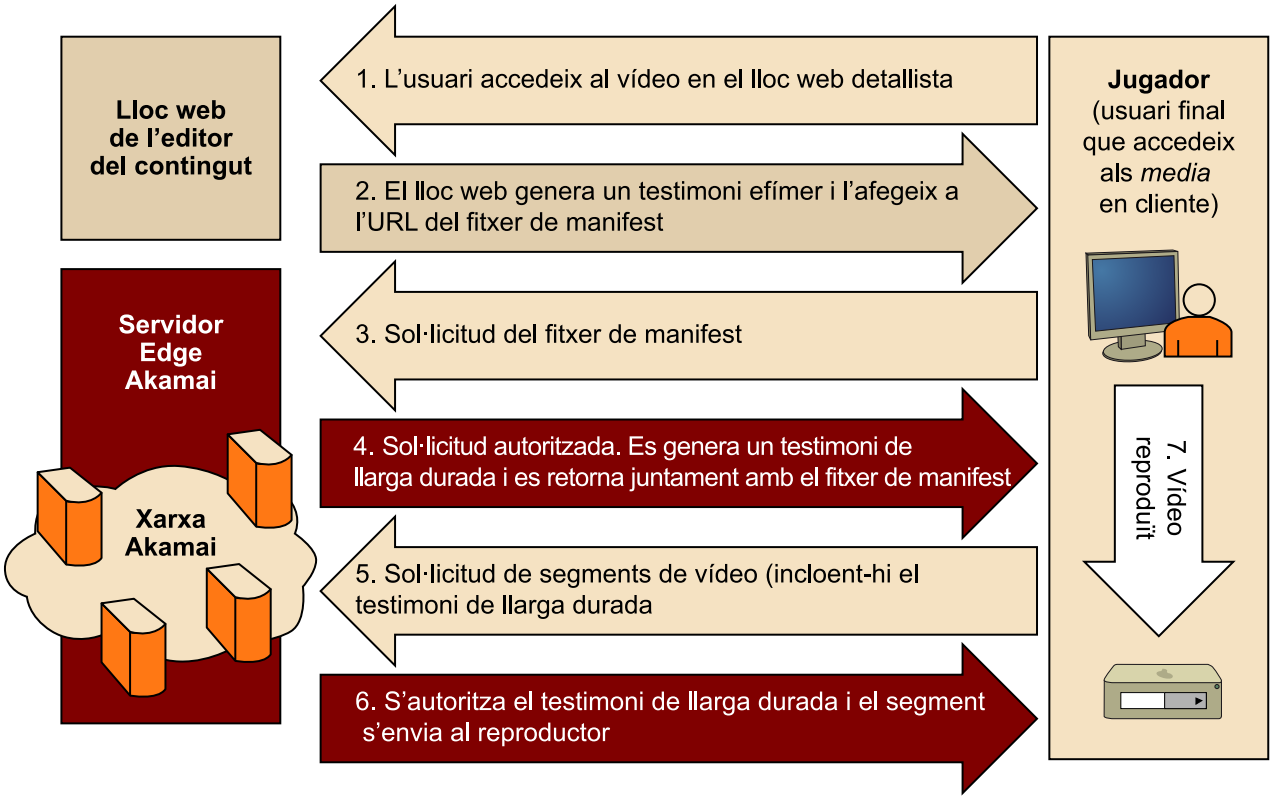
a) Per a poder garantir la seguretat, cal afegir el testimoni de seguretat a cada petició, i el servidor ha de comprovar cadascuna d'aquestes peticions. Una sobrecàrrega important. És viable, però més complicat.

b) Apareix un problema de temps de vigència del testimoni d'autorització. Habitualment s'acostuma a donar un temps curt, per a evitar que algú pugui copiar l'URL i reutilitzar-la, o enviar-la a una tercera persona. Això funciona si només hi ha una connexió al principi, però en HTTP Streaming es fan múltiples connexions, durant tot el temps que dura la reproducció. Això es pot resoldre si el servidor d'*streaming* genera testimonis de durada més llarga per als segments de fitxer.

Més informació

En HTTP Streaming, un fitxer de vídeo és segmentat en fragments de pocs segons, cadascun dels quals se sol·licita en una crida HTTP separada i és el reproductor el que torna a unir els fragments. Això permet que el client pugui optar per sol·licitar fragments de diferents qualitats en funció de les condicions (amplada de banda, CPU, etc.). Per a més detalls, podeu consultar: "Adaptive bitrate streaming".

HTTP Streaming autentificat - Implementació d'Akamai



2) Memòries cau. Abans hem comentat que el gran avantatge de la descàrrega progressiva i de l'HTTP Streaming era la facilitat d'emmagatzemar el contingut en memòries cau a prop del client. Però la necessitat d'autentificar l'accés complica la situació, ja que ara s'ha d'impedir que les memòries cau emmagatzemin el contingut, si no són capaces de validar les peticions. És viable, però ja obliga a afegir la lògica de validació i distribuir les claus secretes a totes les memòries cau. A això, hi hem d'afegir que, si s'utilitza HTTPS, l'ús de memòries cau es fa molt més complicat.

Autentificació en HLS

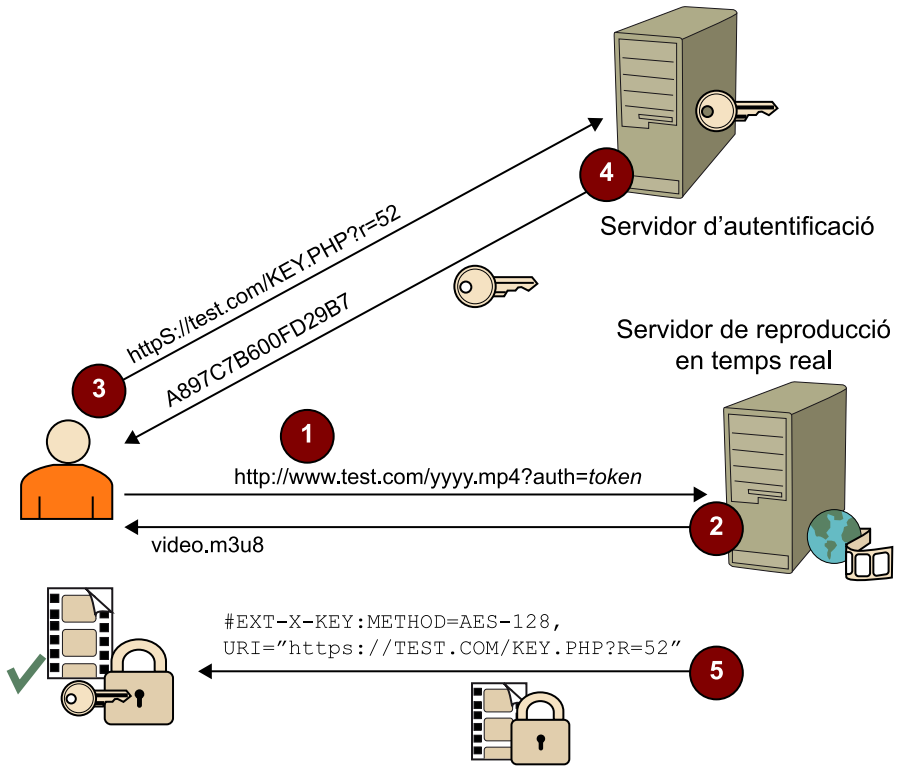
HLS, el protocol d'HTTP Live Streaming desenvolupat per Apple i després alliberat com un estàndard, incorpora un mecanisme de seguretat addicional que es combina amb l'encriptació per a garantir un nivell de seguretat elevat. Es basa en principis similars als exposats més amunt, però incorpora la protecció en l'estàndard.

Més informació

Trobareu més detalls sobre el protocol HTTP Live Streaming a "HTTP Live Streaming". HLS és una de les variants d'HTTP Streaming, com Smooth Streaming o HDS.

HLS permet encriptar cadascun dels fragments de vídeo utilitzant un algorisme de clau simètrica i ofereix una manera d'indicar en el fitxer de manifest un URL que el reproductor podrà invocar per obtenir la clau per a desencriptar el contingut.

Flux en HLS autentificat



La sintaxi de la capçalera que especifica l'encriptació és aquesta:

```
#EXT-X-KEY:METHOD=<method>[,URI="<URI>"][,IV=<IV>]
```

I un exemple d'utilització:

```
#EXT-X-KEY:METHOD=AES-128,URI=https://priv.example.com/
key.php?r=52
```

El procés és aquest:

1) Es demana el fitxer de manifest al servidor.

2) El servidor retorna el fitxer que indica que el contingut està encriptat i l'URL per a obtenir la clau de desencriptació.

3) Es fa una crida a aquest URL.

4) El servidor retorna la clau per a desencriptar.

5) Se sol·liciten els fragments, que el reproductor desencripta amb la clau.

Es pot observar que aquest sistema presenta el mateix problema que els anteriors. Si podem obtenir l'URL i no disposem de validació del reproductor (i HLS no ho incorpora), podem invocar l'URL, descarregar-ne el contingut i desencriptar-lo. Per a fer-lo segur, cal combinar-ho amb una tècnica d'autentificació i filtratge per usuaris. Però, si més no, el problema amb les memòries cau i el temps de vida del testimoni estan resolts.

La manera d'implementar-ho ja és decisió de cadascú, però, per exemple, el reproductor pot afegir un paràmetre més a l'URL que envia al servidor d'autentificació perquè incorpori una galeta de sessió, de manera que el servidor no donarà la clau de desencriptació, si l'usuari no s'ha identificat.

Un altre problema resideix en el fet que, si la clau d'encriptació és fixa i es produeix una filtració d'aquesta clau, tot el contingut queda compromès. Per a afrontar aquest problema cal optar per claus d'encriptació diferents per cada fitxer, o variables en el temps, però a costa de complicar l'escenari, ja que requereix una base de dades que relacioni cada fitxer amb la clau de desencriptació corresponent.

Un avantatge d'aquest mètode és que no té impacte negatiu en les memòries cau, ja que aquestes memòries emmagatzemen un contingut xifrat, i no han de comprovar a qui l'entreguen (encara més, no requereix l'ús d'HTTPS perquè el contingut ja està protegit per l'encriptació). No cal implantar cap mecanisme de protecció a les memòries cau.

Reflexió

Volem tornar a incidir en la diferència entre encriptar la comunicació i encriptar el contingut. El primer cas, del qual HTTPS és un exemple, s'utilitza per a evitar la manipulació i intercepció de la transmissió, però no es requereix a l'usuari la possessió d'una clau per a accedir al contingut de la transmissió. L'encriptació és transparent per a l'usuari. Per contra, l'encriptació del contingut exigeix a l'usuari que disposi d'una clau per a accedir al contingut.

Conclusions sobre l'autentificació

En aquest punt creiem que és un bon moment per a fer una reflexió sobre una cosa que probablement ja heu observat:

a) Hi ha una gran fragmentació en el mercat. A part de diferents tecnologies (HLS, RTMP, Smooth Streaming, etc.), cadascuna té diferents opcions de protecció que, a més, varien en funció del programari que s'utilitzi (per exemple, utilitzant RTMP, es pot escollir entre servidors d'Adobe, Wowza i altres).

[20] En anglès, *open source*.

b) Algunes de les opcions de protecció que hem vist no són estàndard, ni tan sols tecnologies desenvolupades pels fabricants, sinó que són pràctiques d'ús comunes que els desenvolupadors acostumen a implementar, cadascun ajustant-les a les necessitats concretes del seu cas.

c) És un sector molt canviant. Els fabricants treuen noves versions dels productes amb molta freqüència, a la qual cosa cal afegir l'aparició de moltes implementacions de codi obert[20] que aporten més opcions. Probablement, quan llegiu aquest text, ja hauran aparegut versions noves dels principals productes.

En resum, ens trobem amb un trencaclosques de difícil solució. Per a aconseguir seguretat hem de combinar tècniques que depenen tant de la tecnologia utilitzada com del fabricant dels servidors. Si en lloc d'implantar la nostra pròpia solució, lloguem serveis a una CDN, també es genera una dependència pel que fa a quines opcions ofereix aquesta xarxa.

3.3.2. Geoblocatge

En el món audiovisual hi ha una dita que diu que "el contingut és el rei", i que ve a dir que l'element més important és el contingut, per sobre de la resta d'elements. Això és el que atreu l'audiència. I els drets sobre aquests continguts es negocien quasi sempre per a un territori concret. Això obliga a implementar un sistema que garanteixi que el contingut només pugui ser accessible dins el territori per al qual es disposa de drets. En això consisteix el geoblocatge.

Els fonaments del geoblocatge són bases de dades que contenen les ubicacions geogràfiques de totes les adreces IP existents, amb més o menys precisió en funció del país i de l'empresa que recull la informació. Aquestes bases de dades es poden comprar –o bé se'n pot llogar l'accés per un període– a múltiples empreses que ofereixen el servei. Les més rellevants són Neustar, MaxMind, IP2location o IPLigence. D'aquestes, MaxMind i IPLigence tenen versions gratuïtes, amb informació reduïda, com també HostIP, que és una base de dades de codi obert.

Hi ha dues maneres d'utilitzar aquestes bases de dades:

- Descarregar-la, instal·lar-la en un equip i fer un desenvolupament per a accedir a aquesta informació des de les aplicacions pròpies.

- Usar el servei directament, utilitzant serveis API REST o serveis web[21] que la majoria ofereixen (per exemple: http://api.hostip.info/get_html.php?ip=12.215.42.19).

(21)En anglès, *web services*.

El resultat és el mateix, però.

La implementació de la seguretat es pot fer en dos llocs:

- **En el reproductor**. En aquest cas, és el reproductor el que, dins la seva lògica de negoci, comprova que està executant-se dins la zona geogràfica amb drets abans de sol·licitar la reproducció del contingut. Aquesta és l'opció més flexible però presenta un forat de seguretat: potser el reproductor no reprodueix el contingut, però si algú envia l'URL a una altra persona fora de l'àmbit geogràfic autoritzat, el podrà reproduir. Cal combinar aquesta tècnica amb la següent o amb la verificació de reproductor.

- **En el servidor del contingut**. En aquest escenari, és el servidor mateix que envia el contingut el que fa la verificació de la IP del client i per tant accepta o rebutja la petició. És més segur que l'anterior però presenta problemes de flexibilitat (cal configurar les polítiques de geoblocatge en el servidor de vídeo).

Si és possible, si la configuració de les polítiques de seguretat és prou simple, la segona opció és més bona, ja que proporciona un nivell de seguretat superior.

4. Els sistemes de gestió de drets digitals

Solucions segures?

Entenem per *segur* que no s'hagin trobat vulnerabilitats que facin que, amb una implementació correcta de la tecnologia, es pugui obtenir un accés no autoritzat al contingut.

Fins ara, totes les solucions de protecció de continguts que hem vist han presentat vulnerabilitats. Algunes són molt evidents i fàcils d'explotar, altres requereixen més esforç, però cap no es podia considerar segura.

A més, totes pateixen d'una manca de capacitat per a expressar els tipus d'accés de què pot gaudir l'usuari; es concedeix o es denega l'accés, però per a poder modelitzar tipus d'accessos més complexos (per exemple, que es pugui veure dues vegades), cal fer desenvolupaments a mida. I el que és més important, cap de les tècniques vistes no permet protegir contingut fora de línia[22]; és a dir, si el contingut és descarregat al dispositiu client per a visualitzar-lo més tard –escenari que s'utilitza en dispositius mòbils (que no sempre tenen connexió) o en contingut HD (que no es pot reproduir en temps real per limitacions d'amplada de banda)–, se'n perd el control. La gestió de drets digitals apareix per solucionar aquests problemes.

[22]En anglès, *off-line*.

El terme **gestió de drets digitals** (*digital rights management*, *DRM*) fa referència a un conjunt de tècniques, eines i regles per a controlar el consum de contingut digital, entenent per *contingut digital* principalment vídeo i àudio, però també aplicable a imatges, documents o fitxers en general. L'objectiu és entregar contingut al consumidor i alhora mantenir el control dels drets, i possibilitar diferents models de negoci.

La DRM intenta expressar les regles que regeixen diferents models de negoci audiovisual (lloguer, pagament per visió[23], compra, etc.) que abans s'implementaven físicament (s'anava al videoclub, s'agafava una pel·lícula, es visualitzava i es tornava) però en l'entorn digital, tot garantint-ne la seguretat. La DRM abraça molt més que unes tècniques de protecció; és un suport tecnològic per a un **tipus de negoci**, i per a entendre'n el funcionament, cal començar per explicar aquest tipus de negoci.

[23]En anglès, *pay-per-view*.

4.1. Els agents de la DRM

Els agents implicats en el tipus de negoci esmentat són els següents:

- **Consumidor**. L'usuari final que "consumeix" el producte, sia per descàrrega o reproducció en temps real.

- **Propietari del contingut**[24]. La persona o organització que posseix els drets[25] sobre el contingut. Poden ser grans productores, petits estudis, canals de televisió o usuaris particulars.

- **Distribuïdor**. L'entitat que s'encarrega de la comercialització i distribució del contingut (un cop protegit) cap a l'usuari final (consumidor). Adquireix els drets pertinents als propietaris del contingut i això li permet comercialitzar el contingut cap al consumidor.

- **Preparador de contingut**[26]. L'entitat que s'encarrega de preparar els continguts per a la distribuir-los per Internet (codificació als formats necessaris per a la distribució pels diferents mitjans, xifratge, aplicació de DRM i addició de metadades). Aquesta funció la pot fer una empresa que s'hi dedica o a vegades la fa el propietari o el distribuïdor.

(24) En anglès, *content owner*.

(25) En anglès, *copyright*.

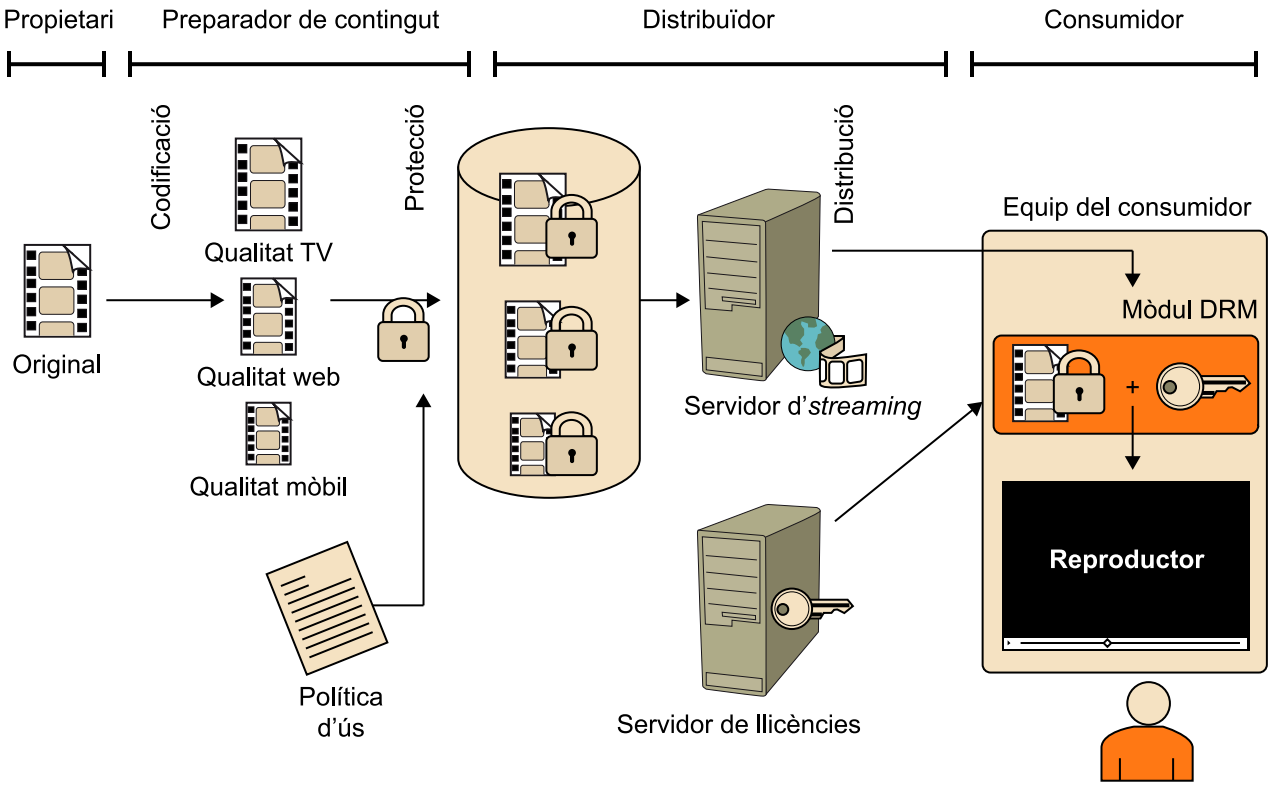
(26) En anglès, *content packager*.

4.2. El cicle de treball de la DRM

Els actors interaccionen els uns amb els altres en un cicle de treball[27] com el següent. Aquesta versió és una versió simplificada que cada empresa acaba desenvolupant adaptada als criteris propis.

(27) En anglès, *workflow*.

Cicle de treball general de DRM



- El propietari disposa d'un contingut en format original que vol distribuir.

- Aquest contingut és recodificat als formats i qualitats necessaris per a distribuir-lo, segons les plataformes objectiu.

- El contingut és encriptat i enllaçat a una o més d'una polítiques d'ús; s'obté així un **contingut protegit**. La **política d'ús** és una definició formal de la manera com els consumidors poden utilitzar el contingut, i reflecteix els drets que el distribuïdor ha obtingut del propietari d'una manera que el reproductor és capaç d'interpretar i complir.

- Aquests fitxers es posen a disposició dels consumidors per mitjà de servidors d'*streaming* real o d'HTTP.

Fins aquí tenim el **cicle de treball de preparació de contingut**, que acostuma a produir-se una vegada per cada contingut nou que s'incorpora al sistema.

A partir d'aquest punt, es desencadena el **cicle de treball del consum**, que es fa per cada accés al contingut:

- En l'equip del client, el reproductor té un programari acoblat que gestiona la DRM. Quan el consumidor vol reproduir un contingut, el reproductor passa la sol·licitud a aquest contingut, que a la vegada inicia la descàrrega del servidor.

- Com que el contingut està protegit, requereix una llicència per a poder-hi accedir; per això fa una sol·licitud al servidor de llicències amb la informació dels drets que té del contingut (si té una subscripció al servei, l'identificador, etc.). El servidor de llicències valida les dades presentades i decideix si el client pot accedir al contingut o no. En cas afirmatiu, envia la llicència al client. Aquesta llicència és una clau, específica per a aquell reproductor, que desencripta el contingut rebut.

- El mòdul de DRM desencripta el contingut i el posa a disposició del reproductor, que el visualitza.

Alguns ja deveu pensar: Això és segur? I si algú intercepta la llicència? Les claus són les mateixes? He obviat de moment aquests aspectes per poder fer una explicació més clara del cicle. Paciència! Més endavant veurem amb més detall com es fonamenta la seguretat.

4.3. Models de negoci

En el món físic, el mercat audiovisual ha desenvolupat al llarg del temps una sèrie de models –maneres– de fer negoci. La DRM intenta fer possible aquests models en el món digital (majoritàriament en línia, però també fora de línia). La manera que té per a aconseguir-ho passa fonamentalment per la llicència que hem esmentat abans.

La **llicència** conté els drets i les restriccions que defineixen com es pot utilitzar el contingut i en quines condicions.

Per exemple, una llicència pot indicar que es pot tenir dret de "reproducció" per a un vídeo X, però no dret d'"enregistrament en CD". I que la reproducció és possible des del reproductor des d'on s'ha descarregat però no des del reproductor del mòbil. I que el dret de reproducció expira el 31 de maig de 2013. Si totes les condicions es compleixen, el reproductor accedirà a descodificar el vídeo i reproduir-lo.

Amb aquesta capacitat, es poden traslladar els models tradicionals al món digital. Per exemple, un model de lloguer com el d'un videoclub passa a modelitzar-se com una llicència per a la reproducció d'un vídeo per un període de X dies, o per a Y reproduccions. Un model de venda de contingut es modelitza amb la generació d'una llicència indefinida de reproducció associada a un dispositiu concret (per a evitar la còpia a altres reproductors).

Reflexió

Parlem de DRM i models de negoci amb casuístiques del món audiovisual, però aquests conceptes també són aplicables a altres sectors. Un bon exemple per a comparar amb el que hem vist és el món del llibre digital, que també funciona amb sistemes de DRM adaptats als models de negoci –més simples– del sector.

En realitat, la versatilitat dels sistemes de DRM permet models de negoci més variats que els que eren possibles en el món físic. Cada fabricant ofereix el seu propi ventall de possibilitats, però les més habituals són aquestes:

- **Subscripció**. El proveïdor del servei (el distribuïdor de contingut habitualment, tot i que a vegades la comercialització la fa una altra empresa) cobra una tarifa plana a un consumidor per poder accedir a un ventall de continguts.

- **Compra**. En aquest model, el consumidor compra el dret per un contingut determinat per un temps indefinit, associat a un dispositiu o un conjunt de dispositius propietat del mateix consumidor.

- **Pagament per visió**. El consumidor accedeix al dret per a reproduir un contingut concret una vegada o unes quantes.

- **Regal**. Variant de les possibilitats anteriors en què una segona persona pot comprar un dels drets anteriors en nom d'algú altre, que rebrà aquest dret com a regal. Aquest model també es pot utilitzar per a casos de revenedors de continguts.

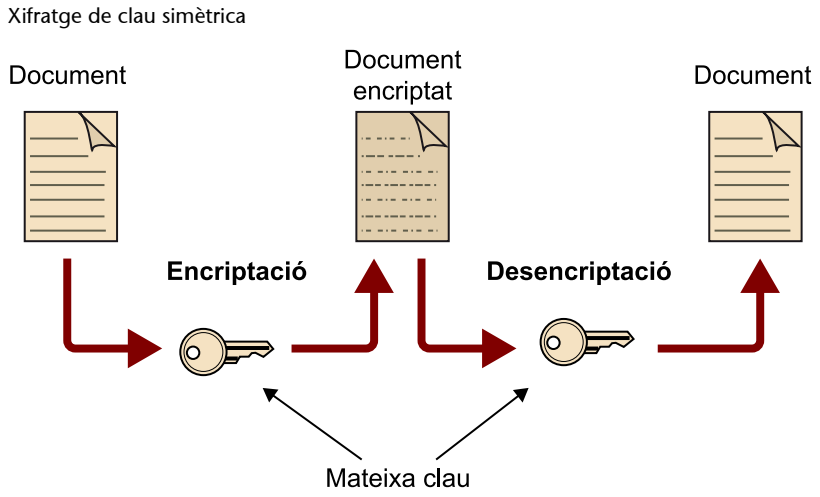
4.4. Fonaments criptogràfics

Vegem quines tècniques criptogràfiques s'utilitzen en la DRM per a aconseguir el grau de seguretat requerit. Aquí no explicarem la base matemàtica en què se sustenten, i que hi confereix la seguretat que s'afirma que tenen. Per als descreguts i els inquiets, tindreu referències complementàries per a aprofundir en la matèria.

Així mateix, identificarem tipus d'algorismes, però per cada tipus se'n poden trobar diferents implementacions. Cada fabricant de solucions de DRM utilitza el (o els) que considera més apropiats. No us preocupeu gaire per això: els fabricants estan prou interessats a fer que la seva solució sigui segura per a haver de dedicar els esforços d'experts en la matèria a triar, entre els que hi ha possibles, un conjunt d'algorismes que ofereixin prou seguretat.

4.4.1. Xifratge de clau simètrica

El xifratge de clau simètrica són un conjunt d'algorismes que permeten encriptar un contingut utilitzant una clau, i desencriptar-lo utilitzant la mateixa clau. Conceptualment és fàcil d'entendre i s'explica visualment en la figura següent:



Aquests algorismes tenen la particularitat que són computacionalment econòmics; és a dir, la capacitat de càlcul necessària per a encriptar o desencriptar és reduïda, cosa que es especialment interessant quan el volum d'informació és elevat i es requereix velocitat.

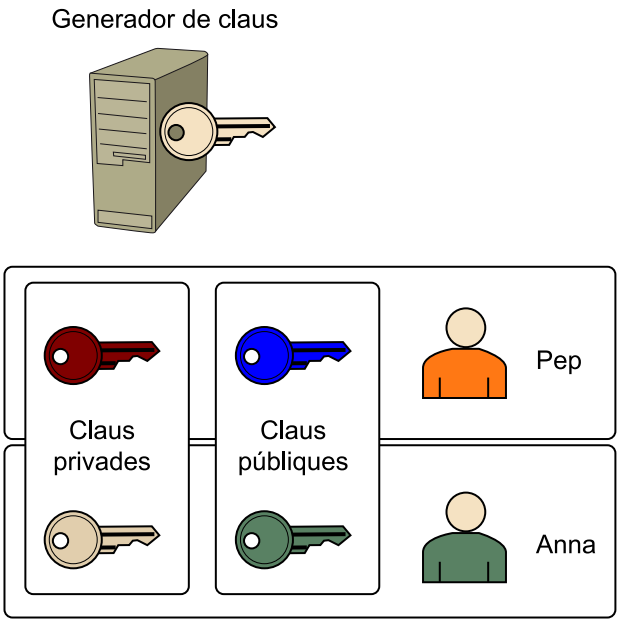
Per contra, presenten una dificultat clara. L'emissor i el receptor han de tenir coneixement de la mateixa clau, i això representa un problema. Com se subministra, de manera segura, aquesta clau al receptor de manera que no sigui interceptada? A més, el receptor amb coneixement de la clau pot xifrar contingut, i podria suplantar la identitat de l'emissor.

Alguns algorismes d'aquest tipus són AES, 3DES i IDEA.

4.4.2. Xifratge de clau asimètrica

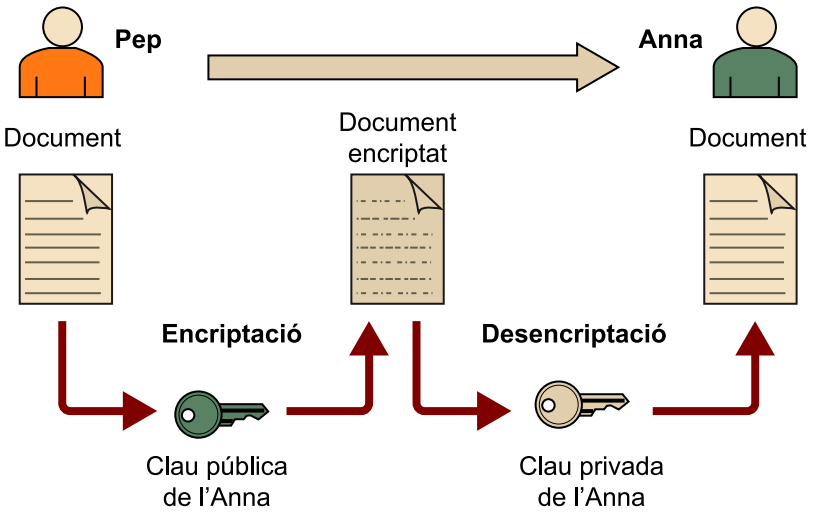
El xifratge de clau asimètrica (o clau pública, com també és conegut) és una tècnica molt més creativa. Per a evitar el problema del secret compartit que tenia el sistema anterior, es desenvolupen aquests algorismes en què tant l'emissor com el receptor utilitzen dues claus diferents, lligades matemàticament entre si. Una d'aquestes claus s'anomena **clau secreta** i l'altra **clau pública**. La primera s'utilitza per a desxifrar (i cal mantenir-la en secret, només la té qui ha de rebre el missatge), mentre que l'altra s'utilitza per a xifrar el missatge, i s'entrega a tothom que ha de poder enviar missatges.

Claus en un algorisme de clau asimètrica



Això fa desaparèixer el problema de la distribució de claus; les claus públiques es poden lliurar a qualsevol i no cal cap mecanisme segur de transmissió. Vegem-ne un exemple d'ús, suposant que en Pep vol enviar un missatge a l'Anna:

Xifratge de clau asimètrica



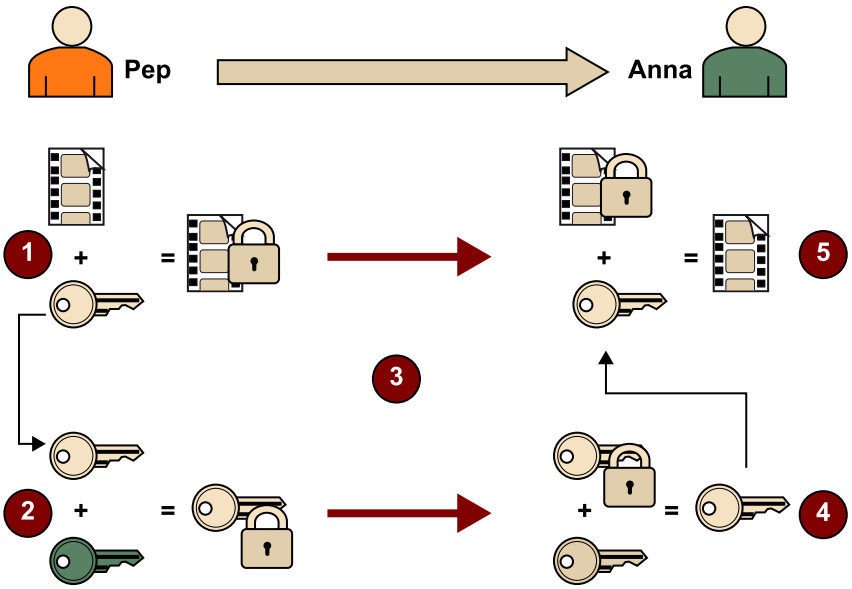
En Pep agafa la clau pública de l'Anna, que l'Anna ha donat a tots els seus coneguts, i la utilitza per a encriptar el missatge. Aquest missatge encriptat, l'envia a l'Anna per un mitjà no protegit, ja que només l'Anna, amb la seva clau privada, pot desencriptar el missatge.

La contrapartida és que les operacions que utilitzen aquests algorismes són computacionalment costoses. Per això s'intenta evitar xifrar grans volums de continguts. Però us deveu preguntar: quina utilitat tenen, si no es pot xifrar un contingut pesant com un vídeo? Doncs el que s'acostuma a fer és utilitzar xifratge simètric per a encriptar el vídeo, i xifratge asimètric per a encriptar la clau i fer-la arribar al receptor.

La base d'aquests algorismes

Fent una simplificació que un matemàtic podria qualificar fàcilment d'insultant, la base d'aquests algorismes és que hi ha operacions matemàtiques fàcils i altres de difícils. Per exemple, multiplicar dos nombres és fàcil (especialment per a un ordinador) mentre que trobar els dos nombres originals a partir del resultat de la multiplicació (factorització) és molt més complex: s'han de fer moltes més operacions. Els algorismes de clau asimètrica es basen en problemes de factorització de nombres primers molt grans o funcions logarítmiques discretes que tenen aquestes mateixes propietats, però la dificultat de les operacions és tan gran que es requeririen molts anys per a trobar el resultat.

Xifratge de clau asimètrica + simètrica



Tornem a l'exemple d'en Pep, que aquesta vegada vol enviar un fitxer de vídeo a l'Anna. El que es fa és això:

1) Genera una clau simètrica nova per a aquesta transmissió, i s'utilitza per a encriptar el fitxer.

2) La clau de xifratge s'agafa com si fos un text qualsevol i s'encripta amb la clau pública de l'Anna.

3) S'envien les dues informacions a l'Anna. En aquesta transmissió no hi ha risc, ja que tots dos continguts estan xifrats.

4) L'Anna utilitza la seva clau privada per a desencriptar la clau simètrica.

5) Finalment, l'Anna pot utilitzar aquesta clau per a desxifrar el vídeo.

Uns exemples d'aquests algorismes són RSA i ElGamal.

4.4.3. Signatures digitals

El xifratge de clau asimètrica permet fer una altra operació a part de xifrar, que és la signatura digital. Aquesta operació s'utilitza de manera anàloga a les signatures manuscrites del món físic: per a provar la identitat d'una persona i per a validar la integritat d'un document.

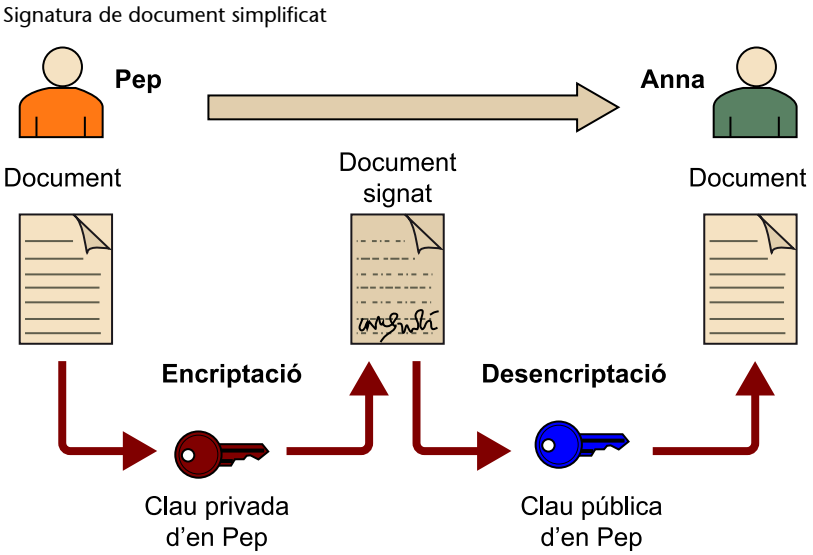
Tots estem acostumats a signar documents: des de l'acceptació del pressupost per a arreglar la rentadora fins a una hipoteca. El fet de signar implica, com dèiem, que:

a) el document signat no s'ha canviat per un altre, ja que altrament no tindria la signatura, i

b) només el signant és capaç de fer aquella signatura concreta, i per tant és ell i només ell el qui ho ha fet.

Bé, en el món físic, afirmar que un document no pot ser modificat un cop signat és agosarat, i que no es pot reproduir una signatura també, però en el món digital, el procés és molt més exacte i segur.

El procés de signar un material es fa justament a la inversa del procés de xifratge. Hem dit que de les dues claus que teníem, la pública es distribuïa lliurement a tothom, mentre que la privada es mantenia en secret i sota la custòdia del propietari. Per tant, si en Pep xifra un text amb la seva clau privada (fet que només pot fer ell) i l'envia, qualsevol persona el podrà desencriptar (ja que tothom té la clau) i verificar que l'ha enviat en Pep. Ja tenim un procés anàleg a la signatura en el món digital.



Falten uns detalls per resoldre, però:

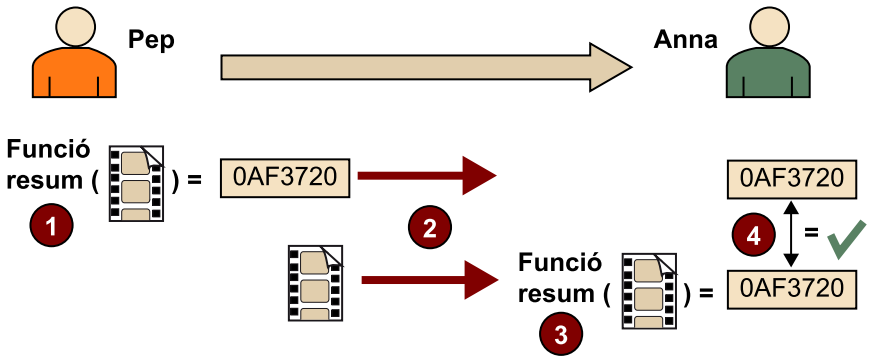
a) Havíem dit que xifrar grans quantitats de continguts amb criptografia de clau asimètrica era costós, i ara estem xifrant documents només per poder-los signar.

b) Estem enviant els documents xifrats, que a vegades no és l'objectiu; el que es vol és acompanyar el document (no encriptat) amb la signatura, però si la signatura és tan gran com el document, no sembla pràctic.

c) No hem resolt totalment el problema de la integritat (no modificació) del document. Si es modifica el contingut encriptat, és possible que el procés de desencriptació funcioni, i l'original es vegi alterat. Potser no podrem controlar la modificació, però és possible fer-la i que el receptor cregui que en Pep l'ha enviat així.

Per resoldre aquest problema apareix una altra tècnica criptogràfica anomenada **funció resum**, que tothom coneix pel nom anglès de **funció *hash***. En aplicar aquest tipus de funcions sobre un contingut, en retornen un resum (*hash*) de longitud més curta (l'algorisme MD5 genera resums de 128 bits, SHA-1 de 160 bits i SHA-2 de 224, 256, 384 o 512 bits), amb una particularitat important: dos continguts diferents a l'entrada, encara que només sigui per 1 bit, donaran resultats de la funció resum diferents. Això permet establir una relació unívoca entre el document original i el resum d'aquest document. En l'exemple següent es pot veure el procés de verificació d'una signatura:

Verificació de la funció resum



1) En Pep aplica la funció resum al fitxer i obté el resum.

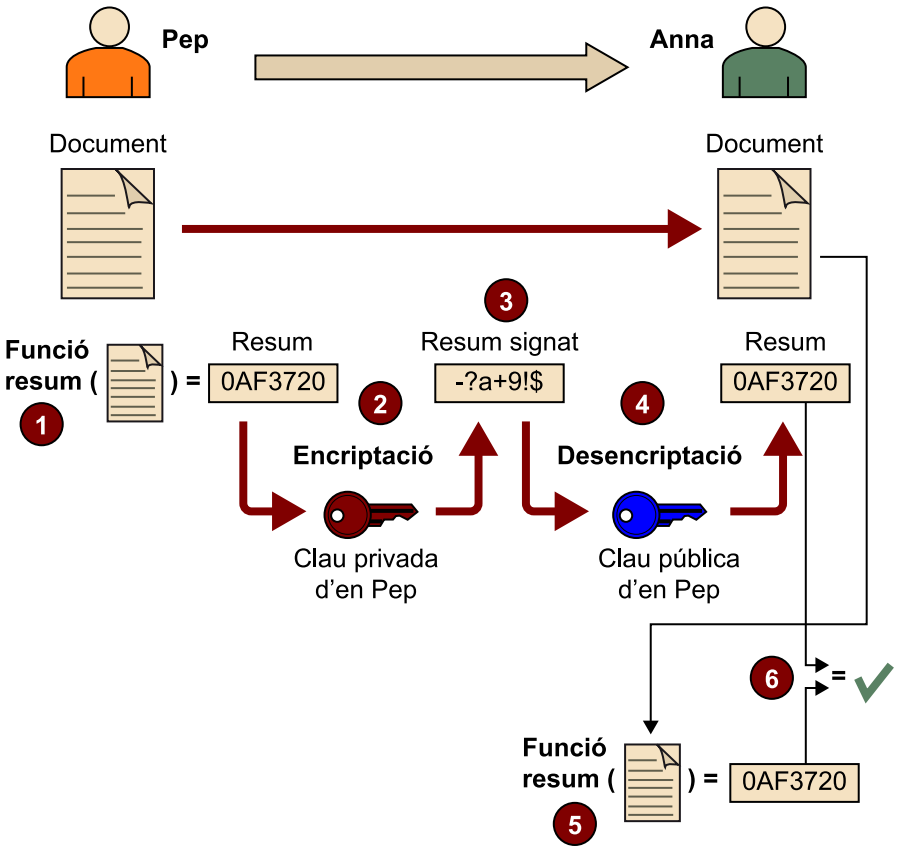
2) Envia el fitxer i el resum a l'Anna.

3) L'Anna aplica la mateixa funció resum al fitxer i obté un resum.

4) Compara aquest resum amb el que ha rebut d'en Pep. Si són iguals, significa que el fitxer no s'ha modificat.

El procés, com l'hem descrit en aquest exemple, no és gaire segur, ja que si algú vol alterar el document, només ha de tenir la precaució de modificar també la signatura. Però si combinem aquesta funció resum amb la signatura anterior, podrem obtenir una solució fiable i computacionalment assequible. Vegem-ne els passos en l'esquema següent:

Signatura amb resum



1) En Pep, que vol enviar el document, hi aplica una funció resum per obtenir una signatura.

2) A continuació, encripta el resum amb la seva clau privada. Com que el resum són uns pocs bytes, el procés és ràpid. Això s'anomena **signatura del document**.

3) En Pep envia el document i la signatura. Aquest cop no hem duplicat la quantitat d'informació, sinó que només hi hem afegit una petita signatura.

4) L'Anna, amb la clau pública d'en Pep, desencripta la signatura.

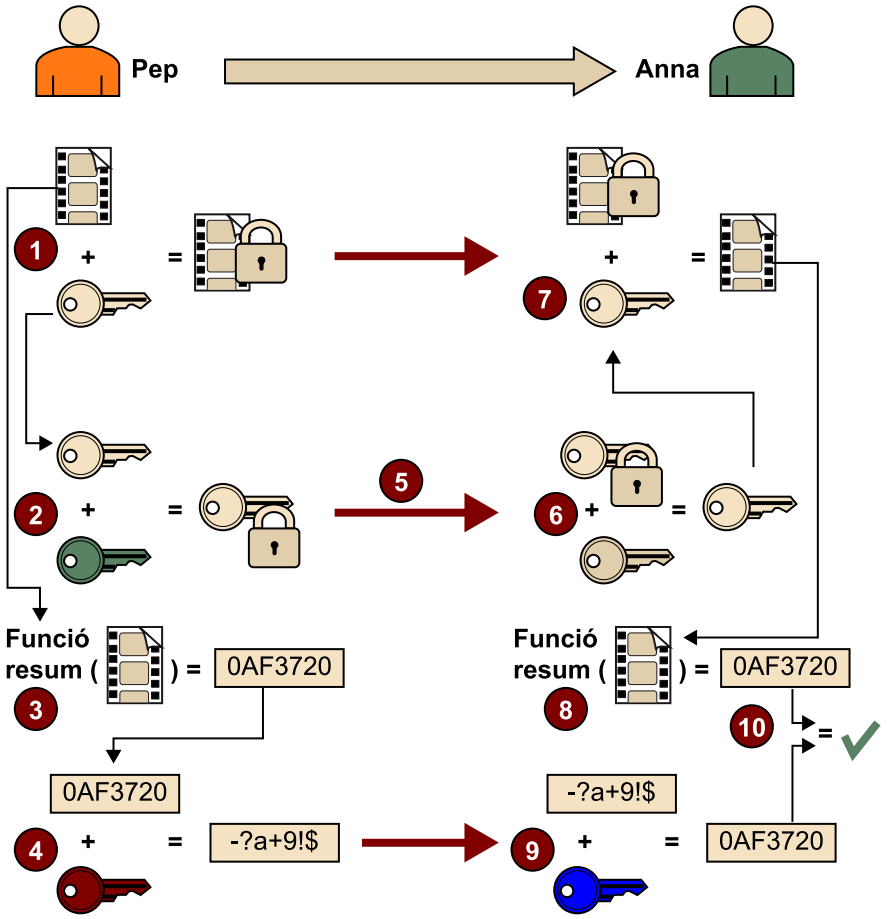
5) L'Anna aplica la funció resum al document rebut i calcula un segon resum.

6) Compara les dues signatures. Si són iguals, es pot afirmar que a) el document no ha estat modificat i b) ha estat enviat per en Pep, ja que la signatura del resum només la pot haver fet ell.

4.4.4. Xifratge + signatura

Combinant els tres darrers elements que hem vist podem obtenir l'objectiu d'una comunicació segura: contingut secret, no alterat i de l'origen esperat. Vegem com resulten tots els passos alhora:

Xifratge més signatura



1) En Pep crea una clau compartida i la utilitza per a encriptar un contingut (un vídeo, en aquest cas).

2) A continuació, encripta la clau compartida amb la clau pública de l'Anna (per tal que només ella la pugui desencriptar).

3) Aplicant la funció resum, genera un resum del vídeo.

4) Encripta el resum amb la seva clau privada per construir la signatura.

5) S'envia a l'Anna el vídeo xifrat (amb la clau compartida), la clau compartida xifrada i la signatura.

6) Quan l'Anna rep les tres peces, comença per desencriptar la clau compartida amb la seva clau privada (només ho pot fer ella).

7) Amb aquesta clau compartida, pot desencriptar el vídeo.

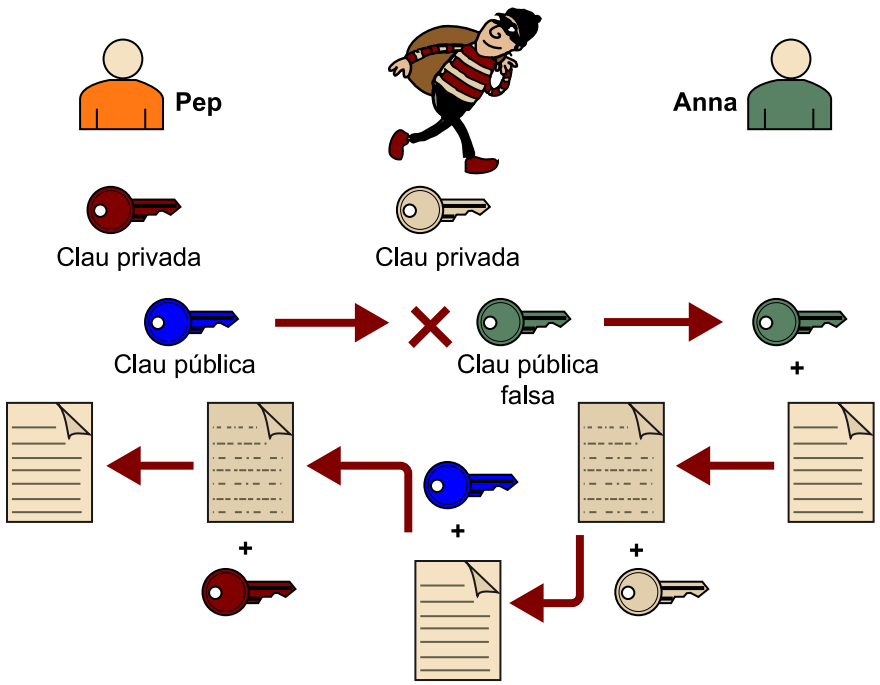
8) Ara li falta assegurar-se que el vídeo no ha estat alterat i ha estat enviat realment per en Pep; per això desencripta la signatura amb la clau pública d'en Pep i obté un resum.

9) Aplica la funció resum al vídeo i obté un segon resum, que compara amb el que ha rebut d'en Pep. Si són iguals, el procés ha acabat amb èxit.

4.4.5. Certificats

La solució criptogràfica presentada fins ara és quasi perfecta, però hi falta resoldre un problema: quan els usuaris no es coneixen entre si, apareix el problema de "com puc estar segur que l'altra part és qui afirma que és". Les claus públiques que utilitzem per a xifrar contingut poden ser interceptades i canviades per altres i fan possible així la intercepció i la manipulació de les comunicacions.

Intercepció de la clau pública



Si algú intercepta la publicació de la clau pública d'en Pep, la canvia per una de seva i la fa arribar a l'Anna com si fos d'en Pep, l'Anna la utilitzarà per a xifrar el missatge, l'espia pot desencriptar el document (ja que té la clau privada aparellada), veure'l, manipular-lo i tornar-lo a encriptar amb la clau pública d'en Pep (la vertadera) i reenviar-lo. En Pep no notarà la diferència.

Si pot fer el mateix amb la clau pública de l'Anna, podrà encriptar i signar, i farà l'engany perfecte. Tinguem en compte que no parlem d'obtenir les claus privades, sinó les públiques, que se suposa que es difonen obertament, de manera que no és una possibilitat inconcebible.

La solució d'aquest problema prové d'una altra analogia amb el món físic. Com s'assegura que una persona és qui diu que és en el món físic? Doncs perquè hi ha una autoritat en la qual tothom confia que valida les identitats i emet quelcom que ho certifica i al qual està lligat. En abstracte, no queda gaire clar, però en realitat parlem del DNI o algun document similar. L'autoritat en la qual es confia (de bon grat o per obligació) és l'Estat, que emet un document (el DNI) que lliga la persona amb la seva signatura (apareix al document); la policia, abans d'emetre el document, verifica les dades de la persona (partida de naixement, fotos, etc.) de manera que, un cop emès, la possessió del document i la capacitat de fer la signatura permeten assegurar la identitat de la persona. Aquesta figura, traspassada al món digital, es coneix pel nom d'**autoritat de certificació** (o simplement **CA**).

El DNI electrònic

El DNI actual, que incorpora un microxip, conté a dins un certificat digital i la clau privada per a poder-lo utilitzar. El mateix Ministeri actua com a autoritat de certificació. Podeu trobar més informació a "DNI electrónico".

Les autoritats de certificació emeten **certificats**, que són documents electrònics que lliguen la clau pública d'una persona o entitat amb la seva identitat i que permeten a una tercera part verificar aquesta identitat. La idea fonamental és que, en un món de milers de milions de persones que no ens coneixem, no podem confiar que una clau pública que ens enviïn correspongui realment a una persona, però sí que podem confiar en un petit grup d'entitats (autoritats de certificació) que tenen com a missió verificar aquestes identitats i després emetre'n el certificat. Vegem-ne els diferents passos:

A) Obtenció del certificat

Per a obtenir el certificat cal primerament que l'usuari generi un parell de claus, una de pública i una de privada. A la pública s'hi afegeix la informació d'identitat que es vol presentar al món (nom, adreça, adreça de correu si és una persona, o URL i nom de l'empresa si és un lloc web) i s'envia a l'autoritat de certificació, que valida les dades (la policia us obliga a anar personalment a identificar-vos; altres CA permeten fer el tràmit per Internet, enviant documentació) i, si són correctes, hi afegeix més informació (identificant-se com la que ha validat les dades, afegint dades de validesa) i signa la petició amb la seva clau privada i la retorna.

Per a validar que el certificat d'en Pep és correcte, només cal agafar el certificat de la CA i aplicar-lo al certificat d'en Pep; si ha estat manipulat es detectarà. La quantitat i tipologia de documentació que s'ha de presentar depèn de l'ús que es vulgui fer del certificat; no és el mateix un certificat per a identificar-se en un servei de vídeo en línia que un certificat per a signar contractes de milions d'euros entre empreses.

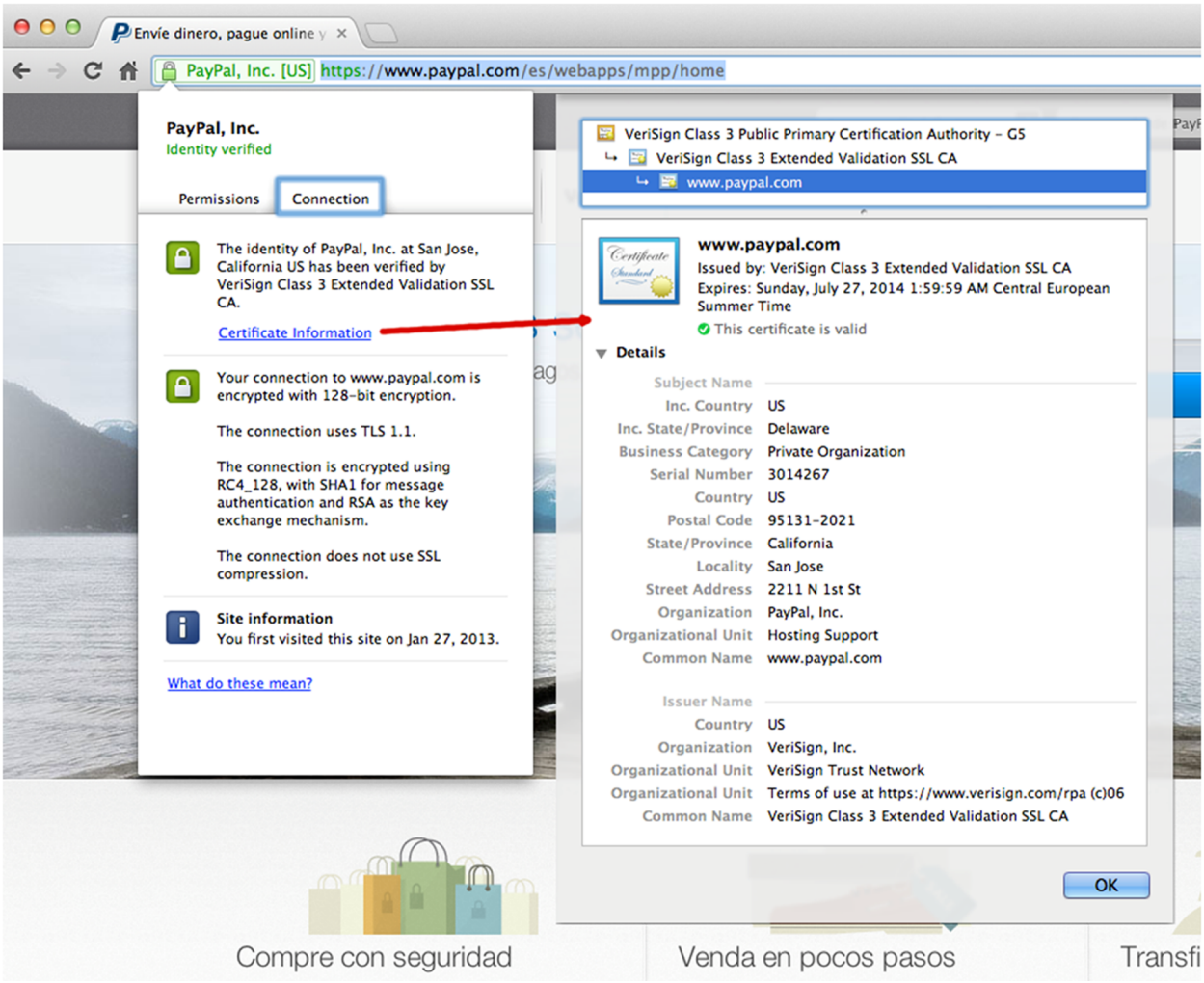
El problema de l'ou i la gallina

I les autoritats de certificació (CA), d'on obtenen el seu certificat? És el problema de l'ou i la gallina. L'usuari obté el certificat de la CA (que el signa), però d'on l'obté, aquesta? Bé, algunes CA l'obtenen d'altres CA que els generen un certificat que permet generar-ne més, de manera que es crea una cadena de CA que confien en la de nivell superior, fins a arribar a les primeres que han signat els seus propis certificats. Aquí acaben les relacions

de confiança verificables i s'ha de fer l'únic acte de fe en tot el procés i confiar en el seu certificat.

Habitualment, aquests (pocs) certificats ja ens arriben inclosos per tots els productes que utilitzen criptografia de clau pública (els navegadors, les biblioteques de seguretat, els llenguatges de programació), de manera que fan possible verificar qualsevol certificat fins a l'origen. En la imatge següent es pot veure el certificat d'un lloc web (que s'utilitza per a garantir que realment ens hem connectat al lloc web esperat, i no a un altre que simula ser-ho) i es veu la cadena de certificats fins al certificat arrel. El fet que el navegador mostri el cadenat, el color verd i el text "Identity verified" indiquen que s'ha verificat el certificat i que la connexió és correcta.

Certificat d'un lloc web



B) Signatura

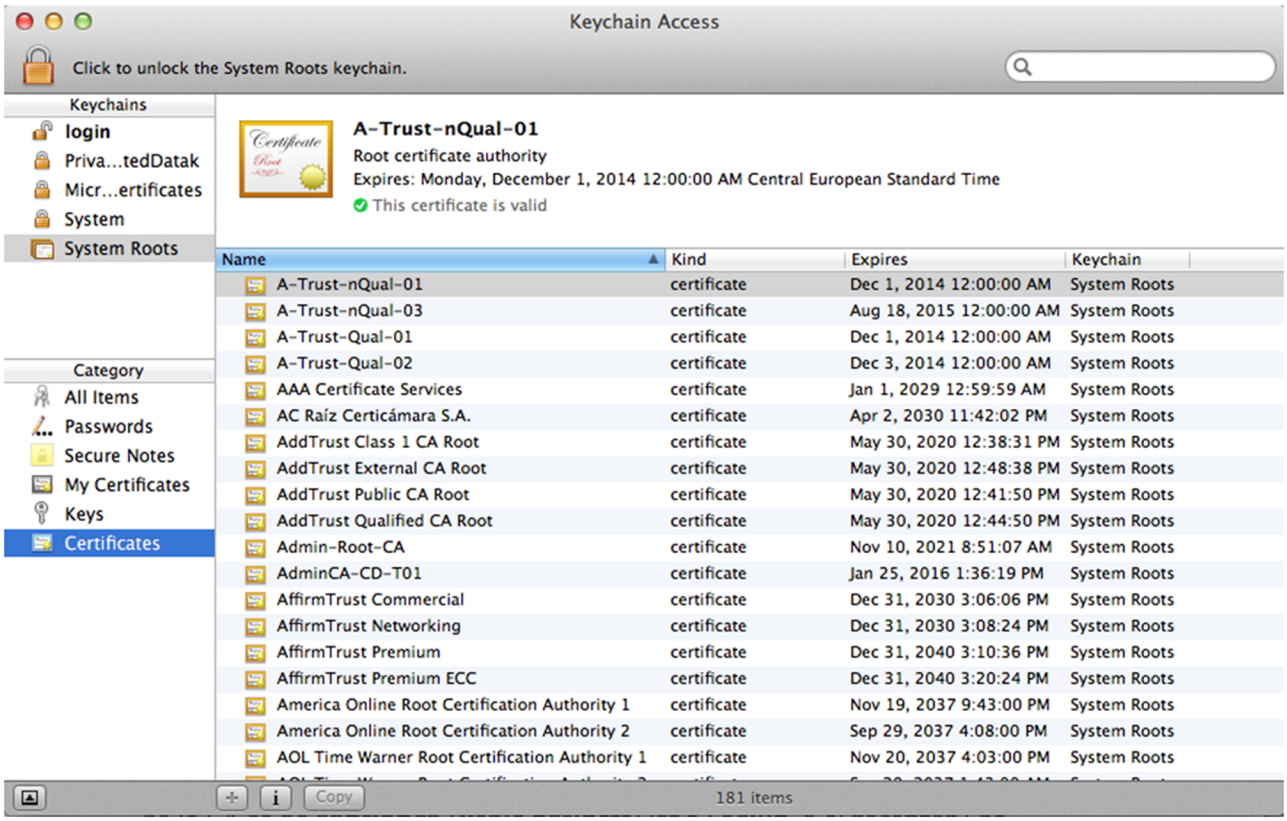
La signatura és el procés que ha canviat menys. El qui vol signar un contingut utilitza la seva clau privada per a xifrar el resum i envia el document, la signatura i el seu certificat al receptor.

C) Verificació de signatura

El receptor, abans de verificar la signatura, verifica el certificat. Per a fer-ho, obté el certificat de la CA i l'utilitza per a verificar. Si el certificat de la CA és **de confiança** (venia preinstal·lat a l'equip, o el receptor l'ha instal·lat perquè hi confia), ja hem acabat. Si el certificat de la CA està signat per una altra CA, hem de repetir el procés de validació fins a arribar a una CA de confiança.

Un cop validat el certificat, ja sabem que la persona que envia és realment qui afirma que és, i podem validar la signatura amb la clau pública inclosa en el certificat.

Certificats arrel inclosos en un ordinador



L'estàndard que regeix els formats i processos de signatura digital és l'X.509.

Amb els certificats, completem el calaix d'eines que ens permeten protegir el contingut en sistemes de DRM.

Reflexió

El món de la criptografia és més ampli, i hi ha altres figures a l'entorn de la certificació digital (servidors de temps, autoritats de registre), que en conjunt formen el que s'anomena *infraestructura de clau pública* (*PKI*, de l'anglès *public-key infrastructure*), però per al propòsit d'aquest mòdul, que és poder explicar la base en què es fonamenta la DRM, amb el que hem vist n'hi ha ben prou i no ens hi estendrem.

4.5. Procés de la DRM

Vegem com funciona amb més detall un sistema de DRM utilitzant les eines criptogràfiques acabades d'introduir. Cada implementació fa variacions sobre aquest model general i afegeix algunes característiques extres per cobrir necessitats específiques dels diferents models de negoci, però els fonaments són comuns.

Abans de començar, però, cal fer notar que hi ha un punt especialment important en tot el procés. Si l'aplicació de la criptografia es fa correctament, un cop encriptat el fitxer hi ha garanties de seguretat fins al final del camí, on hi ha el **reproductor**. Aquest element és clau, perquè:

- està instal·lat dins l'equip que té l'usuari final, i per tant, més exposat a manipulacions i atacs, i

- és el punt on s'ha de descodificar el contingut, i per tant, el lloc ideal per a accedir-hi.

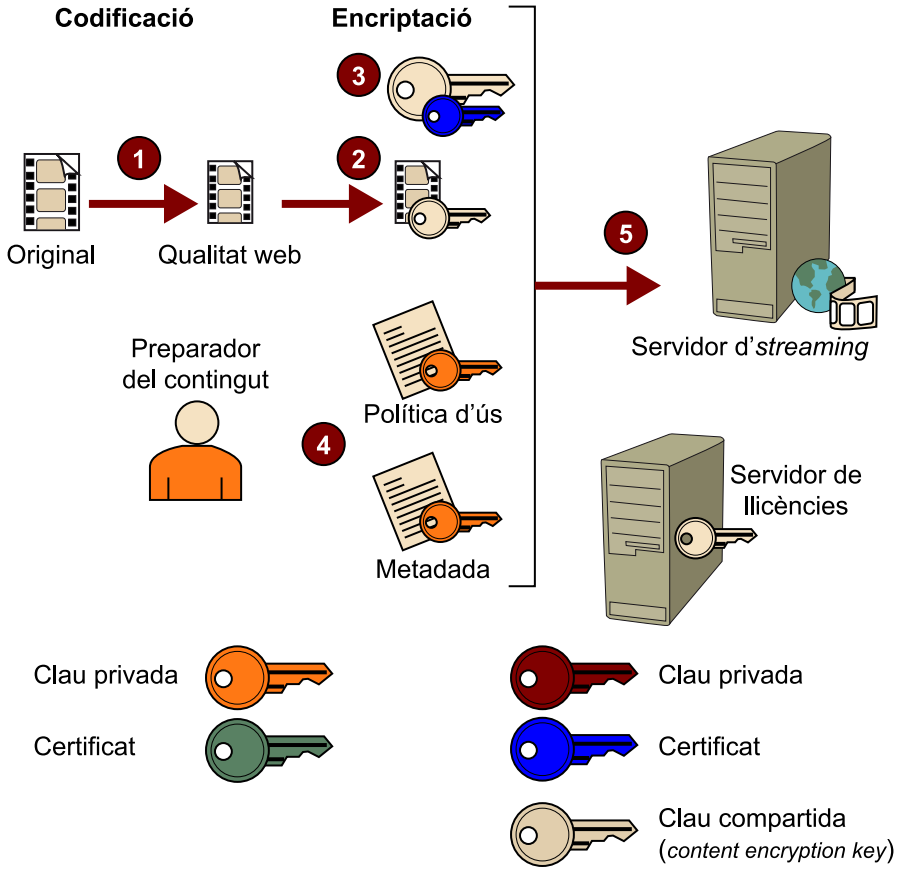
Per això és una peça clau de qualsevol sistema de DRM; tots els fabricants posen molta cura en el desenvolupament d'aquest element i a saber si és viable, implementen en el maquinari dels equips la lògica de seguretat (especialment en televisors, descodificadors d'Internet –*set-top boxes*– i telèfons). El reproductor ha de seguir escrupolosament les normes que trobarà descrites en la llicència per a garantir que tot funciona com està previst.

Vegem-ne el procés des de la generació de contingut fins a l'usuari final; per a fer-ho més digerible, dividirem el cicle de treball en dues parts: preparació del contingut i accés al contingut.

4.5.1. Cicle de treball de preparació del contingut

El procés de preparació del contingut només es fa una vegada per a cada contingut que s'incorpora al sistema protegit per DRM.

Cicle de treball de preparació de contingut



1) El primer que es fa és la codificació del contingut original al format o formats necessaris per a la distribució.

2) A continuació es xifra el fitxer amb una clau compartida anomenada *content encryption key* (CEK), que pot ser la mateixa per a tots els continguts o diferent per a cada fitxer, amb l'objectiu de millorar la seguretat.

3) La CEK es xifra amb la clau pública del servidor de llicències (de manera que només hi pugui accedir aquest servidor).

4) La política d'us i la metadada associada al fitxer es xifren amb la clau privada del preparador del contingut:

- La política indica quin ús es pot fer d'aquest contingut (el mínim dret possible és poder reproduir, però pot incorporar dades límit, tipus de dispositius acceptats, si es pot copiar, etc.).

- La figura del preparador de continguts té entitat (i per tant té claus i certificat per a poder fer accions criptogràfiques) perquè pot ser una entitat o persona diferent de la que farà la distribució, i limita els drets que es podran donar als consumidors sobre el contingut, de manera que fa els contractes de cessió de drets entre creador i distribuïdor.

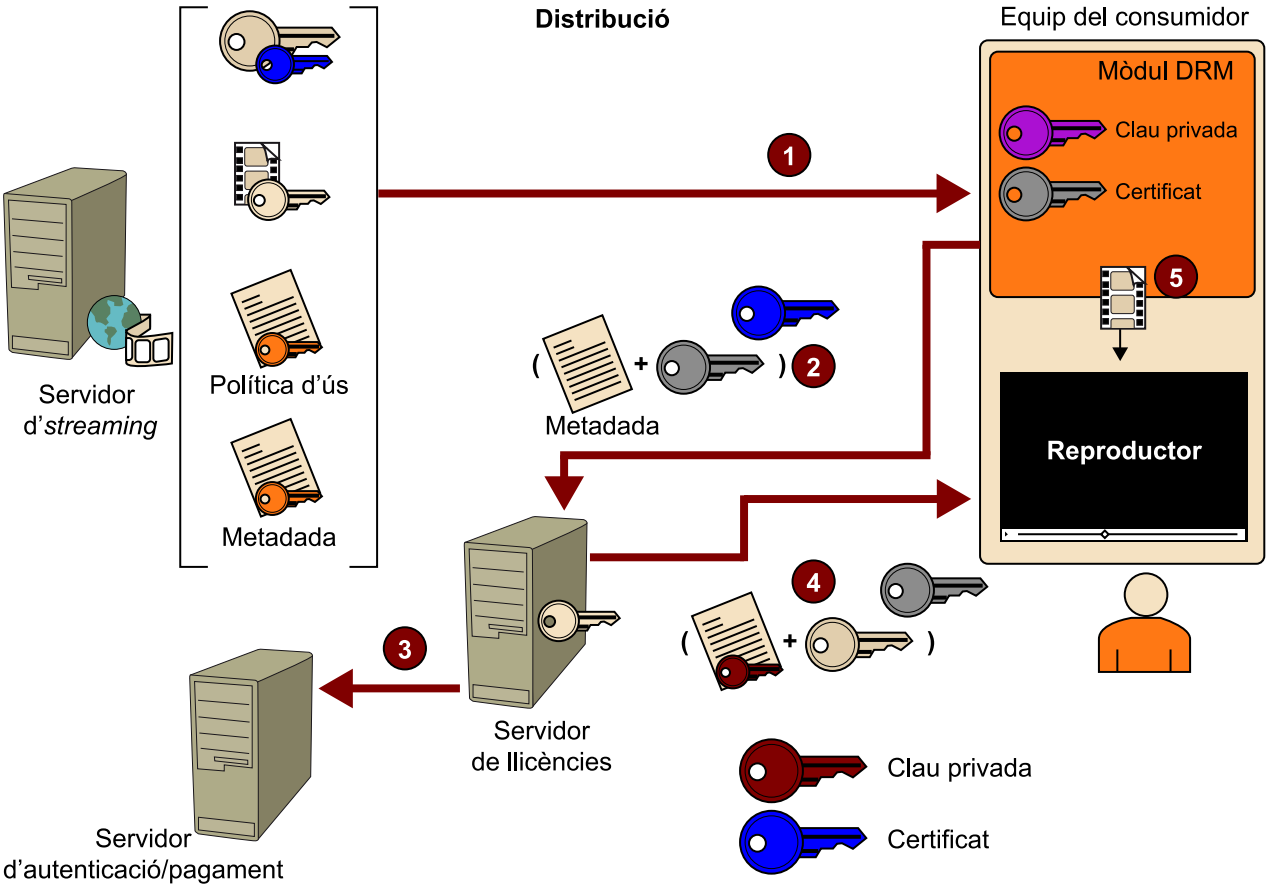
- La metadada és informació sobre el contingut. Com a mínim inclou l'URL on el reproductor podrà trobar el servidor de llicències, un identificador únic del contingut i la CEK encriptada en el pas 2. A part d'això, pot incorporar qualsevol altra dada descriptiva (títol, autor, etc.).

- El fet de signar la política es duu a terme perquè en tot moment es pugui comprovar que es compleix la política i que no ha estat alterada.

5) Aquests elements s'empaqueten conjuntament i es posen a disposició del consumidor en un servidor.

4.5.2. Cicle de treball d'accés al contingut

Cicle de treball d'obtenció de contingut



Abans d'explicar el cicle de treball, cal precisar que en els sistemes de DRM és necessari que cada reproductor estigui identificat individualment, amb l'objectiu de poder lliurar contingut exclusivament a aquest reproductor. I hem vist que la manera que tenim per a gestionar les identitats digitals són els certificats; per això a cada reproductor s'hi assigna una parella de clau privada i certificat. Aquest certificat pot venir generat de fàbrica o ser generat la primera vegada que s'accedeix a un servei. Aquest certificat acostuma a estar

signat per la CA del fabricant de DRM, cosa que permet verificar-lo en qualsevol moment, i compartir una identitat única encara que s'utilitzi el mateix equip per a diferents serveis.

1) El consumidor accedeix al lloc web del proveïdor i escull visualitzar un contingut. En aquest moment, el mòdul de DRM del reproductor sol·licita la descàrrega del paquet format pel vídeo, la política, les metadades i la CEK xifrada.

2) Un cop el reproductor té les metadades i ha validat que no han estat manipulades (desencriptant-la amb la clau pública del preparador de continguts), extreu l'URL del servidor de llicències, fa una petició de llicència al servidor i hi adjunta el certificat (prova d'identitat) i les metadades del vídeo, tot encriptat amb la clau pública del servidor de llicències. Quan arriben al servidor de llicències, aquest servidor desencripta la petició i n'extreu les dades.

3) Primer utilitza les dades per a invocar el servidor d'autentificació i l'autorització del proveïdor per a comprovar si el reproductor té drets per a accedir al contingut (per exemple, si té la subscripció al corrent de pagament).

4) Si aquest reproductor respon positivament amb una llicència, desencripta la CEK (recordem que en la fase de preparació del contingut havia estat encriptada amb la clau pública del servidor de llicències justament per a aquest moment) i la torna a encriptar juntament amb la llicència, ara amb la clau pública del reproductor. Finalment, encripta tot el contingut amb la seva clau pública i el retorna.

5) El reproductor desencripta el missatge amb la clau pública del servidor de llicències (i verifica així que el contingut no s'ha alterat) i extreu la CEK que li permetrà accedir al contingut. Però abans de donar-hi accés verifica que l'acció sol·licitada (reproduir, enregistrar, etc.) és admesa per la llicència entregada.

Fixem-nos que la clau que permet accedir al contingut ha estat emmagatzemada i ha viatjat encriptada tota l'estona. Només hi té accés el servidor de llicències, i al final del procés, el reproductor autoritzat. Per això és tant important protegir aquest darrer element ja que, si se'n pot extreure la CEK, es posa en compromís tot el sistema.

4.5.3. Dominis

Un cas de negoci freqüent que alguns sistemes de DRM preveuen és la possibilitat d'associar diferents dispositius a una mateixa persona (per exemple, es pot llogar una pel·lícula que es pot reproduir al televisor, al mòbil i a l'ordinador).

Aquesta agrupació de dispositius a una mateixa persona s'anomena **domini**. Per a implantar aquesta opció cal afegir un nou element, el **servidor de domini**, que gestionarà les associacions entre persones (dominis) i dispositius. Cada domini, com qualsevol identitat digital, significarà un certificat i una clau. Els dispositius es registraran en aquest servidor i es lligaran amb un o diversos dominis.

Quan el servidor de llicències emeti una llicència, aquesta llicència pot estar adreçada al dispositiu (cas que ja hem vist) o a un domini; en aquest cas, el dispositiu no podrà desencriptar la CEK (ja que caldrà la clau privada del domini X, i no del dispositiu) i li caldrà enviar-la al servidor de domini perquè emeti una CEK encriptada amb la seva clau.

Si el servidor de domini ha d'emetre certificats, significa que ha de disposar d'un certificat de CA que li permeti justament fer això.

Les associacions dispositiu-domini varien en funció de cada proveïdor.

Els sistemes de DRM no acostumen a fixar regles sobre els drets i les limitacions dels dominis, i es deixa això a la lògica de negoci de cada proveïdor de serveis. Només proporcionen les primitives per a poder gestionar addició i eliminació de dispositius del domini i generar autoritzacions.

4.6. Principals DRM en el mercat

Al principi de l'apartat, on descrivíem els principals actors, ja hem esmentat alguns fabricants de solucions de DRM. Acabarem l'apartat recapitulant les diferents opcions que podem trobar en el mercat.

En fer una cerca, es poden observar clarament tres orígens que marquen els terrenys on són més forts, i també unes certes maneres de treballar.

4.6.1. DRM provinent del mercat dels ordinadors personals

Hi ha dos grans fabricants que provenen d'aquest entorn:

1) Microsoft té una llarga experiència en solucions de DRM. Primerament va treure un producte anomenat Windows Media DRM, la primera versió del qual va aparèixer el 1999, i la darrera el 2004: totes dues utilitzaven Windows Media Player com a reproductor, i per tant estaven centrades en els PC amb Windows. El 2005, el Windows Media DRM va començar a tenir problemes de seguretat que Microsoft va anar apedaçant, fins que va substituir el producte per un altre, PlayReady, que va aparèixer el 2007. Les principals novetats que aportava eren les següents:

- PlayReady és independent de la plataforma, i funciona amb equips que no són de Microsoft, que ha creat un equip (*kit*) de desenvolupament per permetre a fabricants de dispositius incorporar la tecnologia.

- Funciona amb models de negoci més avançats, com ara dominis i la distribució de llicències amb el contingut per accés sense connexió al servidor de llicències.

Més informació

Podeu trobar més informació sobre PlayReady en les adreces següents:
- Lloc web de Microsoft
- Documents introductoris

2) Adobe, com a fabricant de solucions de reproducció en temps real, va llançar el 2009 l'Adobe Flash Media Rights Management Server, la primera versió de DRM, que es va renovar el 2009 i va canviar el nom per **Adobe Access 2.0**. El 2012 ha aparegut la versió 4.0, que funciona amb les plataformes Windows, Mac, Linux, iOS i Android.

Més informació

Podeu trobar més informació sobre Adobe Access en les adreces següents:
- Lloc web d'Adobe
- Introducció a Adobe Access
- Documentació del producte
- Pàgina on es pot trobar un reproductor que utilitza DRM

Tots dos fabricants tenen uns punts forts importants:

a) Tenen una gran presència en les plataformes en què actualment es consumeix més vídeo: ordinadors i terminals amb android o iOS. En televisors connectats, Microsoft ha aconseguit convèncer força fabricants per a incorporar la seva tecnologia i posicionar-se més bé que Adobe.

b) Hi ha una gran base de desenvolupadors amb experiència en aquestes plataformes, cosa que facilita la creació de productes.

c) Tots dos han independitzat les seves solucions de DRM de les seves tecnologies de reproducció en temps real amb l'objectiu de poder ampliar el mercat.

Més informació

Per als qui hi estigueu interessats, podeu trobar un article –ja una mica antic– de Tim Siglin que compara aquestes dues tecnologies al web streamingmedia.com:

"DRM: The big two"

4.6.2. DRM provinents del mercat de mòbils

La indústria del mòbil va definir un estàndard de DRM el 2002 (OMA DRM) més centrat en contingut de petites dimensions, com ara politons i fons de pantalla. Des d'aquella primera versió, l'estàndard ha anat evolucionant per donar suport a les noves necessitats. Actualment, la darrera versió publicada és la 2.1.2.

El procés és dirigit per l'OMA[28], que engloba els principals fabricants i que defineix estàndards per a una gran quantitat d'aspectes relacionats amb la tecnologia de mòbils.

[28] *OMA* és la sigla de l'entitat Open Mobile Alliance.

Més informació

Les especificacions de l'estàndard OMA DRM les podeu trobar al web de l'OMA:

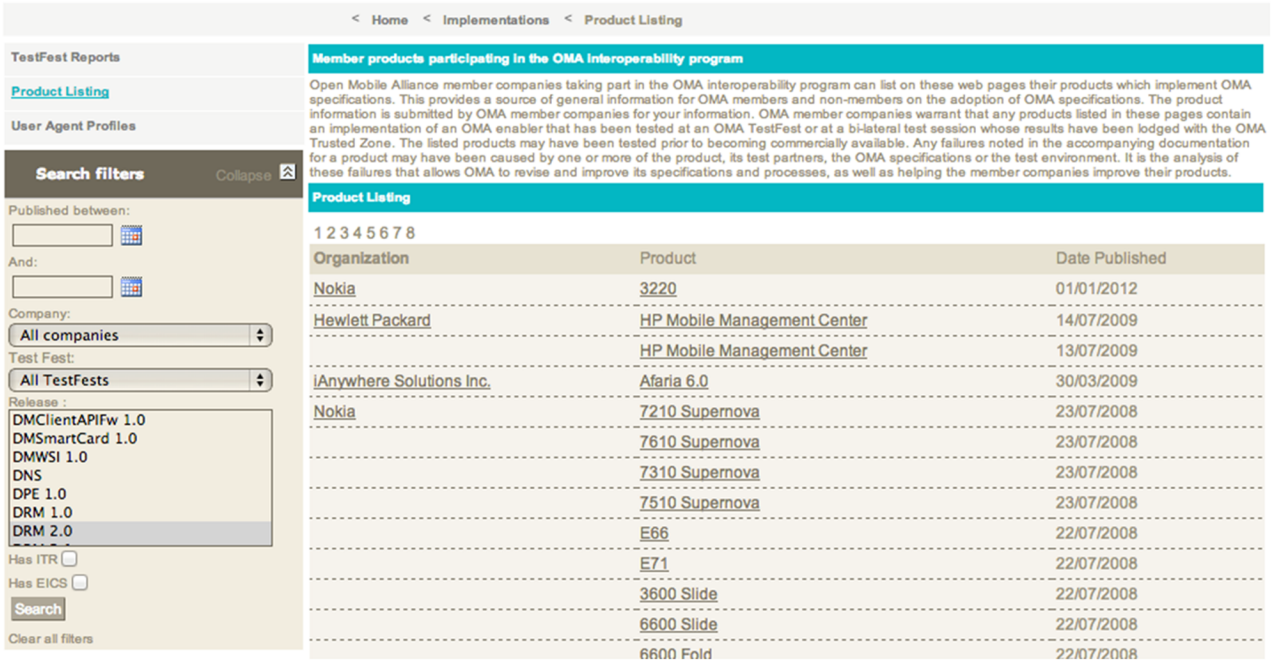
"OMA Digital Rights Management V2.1.2"

El que l'OMA defineix és un estàndard, no un producte desenvolupat i operatiu. Els diferents fabricants poden implementar el programari com vulguin, seguint les especificacions definides, i segons la Wikipedia ("OMA DRM"), n'hi ha diversos que ho han fet.

El problema d'OMA DRM és que el mercat de mòbils ha fet un salt enorme i els fabricants "tradicionals" n'han estat escombrats, no tant pel que fa a la fabricació com als sistemes operatius que incorporen, i al final és aquesta peça la que determina què s'executa en el terminal. Actualment, els líders són iOS, Android i Windows, i no s'adhereixen a aquest estàndard.

Fent una consulta al web de l'Open Mobile Alliance per als dispositius que funcionen amb DRM, només apareixen dispositius del 2009, i només un del 2012.

Darrers dispositius que funcionen amb OMA DRM



4.6.3. DRM provinents del mercat de la televisió

En el mercat de la televisió, la DRM fa temps que hi és present. Les plataformes d'IPTV l'han utilitzat per a protegir l'accés als continguts i hi ha nombrosos fabricants que en tenen solucions. En aquests sistemes, la implementació s'acostuma a fer per maquinari. L'empresa que vol llançar un canal adquireix descodificadors d'Internet, que distribuirà als clients, els quals controlaran tot el procés. Un dispositiu és per a un únic servei.

El punt d'inflexió es produeix en aparèixer el vídeo per Internet –*over the top*, com s'acostuma a anomenar en el sector–, en què hi ha diversos serveis que poden proporcionar contingut al mateix dispositiu. Això obliga a la interoperabilitat, i els fabricants de dispositius intenten donar suport a diverses tecnologies. I és per aquí per on els fabricants del món Internet, especialment Microsoft, han començat a entrar en aquest mercat.

D'altra banda, els fabricants provinents d'aquest sector han començat a desenvolupar solucions per a dispositius mòbils i ordinadors personals, cosa que ha ampliat la competència. I dins d'aquests moviments estratègics, Google compra el 2010 una de les empreses més importants del sector (Widevine) i encara s'està en espera de saber quins són els objectius d'aquesta compra.

Els principals fabricants d'aquest sector són Widevine, Irdeto i NDS.

A part d'aquestes iniciatives, n'hi ha una altra d'interessant: un conjunt d'empreses compost per Intertrust, Panasonic, Philips, Samsung i Sony, que es van unir per formar la Marlin Developer Community amb l'objectiu de desenvolupar un sistema de DRM, alliberat com a codi obert, que s'ha implantat a nombrosos televisors connectats. Un dels aspectes que fan interessant aquest producte és que ha estat escollit, juntament amb PlayReady, com a solució per a protegir la televisió hibrida a Espanya per l'associació que agrupa fabricants i cadenes de televisió (AEDETI).

S'anomena **televisió híbrida** la televisió que combina emissió en difusió àmplia (per la TDT) amb els continguts a la carta distribuïts per Internet i està estandarditzada amb l'especificació HbbTV, vigent tant a Espanya com a molts altres països europeus.

Més informació

Podeu obtenir més detalls sobre la televisió híbrida a l'adreça següent:

"HbbTV® = More entertainment at your command"

Marlin està guanyant rellevància ràpidament i sembla tenir el favor del sector de la difusió àmplia.

Resum

Com podeu veure, la seguretat en el vídeo per Internet presenta diferents problemes:

- Totes les opcions que no inclouen DRM tenen deficiències.

- Hi ha una gran fragmentació de tecnologies, fabricants i dispositius. Fins i tot independentment de la seguretat, trobar una solució per a portar un servei a diferents plataformes pot requerir l'ús de més d'una tecnologia. Si a això hi afegim la necessitat de protecció, encara tenim més problemes.

- Els sistemes de DRM tenen mala fama, no pel funcionament sinó perquè són complexos, cars d'implantar i limiten molt l'accés al contingut. Fins i tot alguns serveis que tenen contingut de valor intenten evitar l'ús de DRM.

No és simple prendre decisions en aquest context. Habitualment s'utilitza una tècnica iterativa que passa pels punts següents:

- Definir plataformes objectiu: on volem que sigui present el contingut.

- Identificar les tecnologies que permeten complir l'objectiu.

- Definir les necessitats de seguretat.

- Analitzar les possibilitats que ofereixen les tecnologies del punt 2 i veure si se'n troba una que compleixi les condicions de seguretat.

- Si no es troba una solució que encaixi, sia perquè no compleix els requeriments de seguretat o perquè per a cobrir-los cal més d'una tecnologia, s'ha de tornar al punt 1 per a reduir el nombre de plataformes o al 3 per a reduir les necessitats de seguretat.

- La iteració es fa fins a arribar a un compromís satisfactori.