Cryptography and DRM

Martin Persson & Alexander Nordfelth

Cryptography, spring 2008

Uppsala Universitet

Table of contents:

Introduction

The concept of Digital Rights Management is a broad one. It encompasses business models, national and international laws and technology. For us to be able to look at DRM in a fairly short amount of pages we need to limit ourselves.

Following a brief historic overview, we will strictly look at the technical side of DRM. We will delve into a few common areas where DRM is implemented today and we will look at what algorithms and structures that are used. We will also look at a few cases when DRM has been broken and what methods the attackers have employed.

Brief History

For as long as there has been physical media, there has been attempts to acquire said media illegally. Copyright holders have become increasingly concerned with this theft over the past decades as the means to make copies of the media has become increasingly easy.

In the past, the quality of the physical media, deteriorated with each copy being made and the "pirate" was only able to produce a finite amount of copies. Today copyright holders face the threat of a single "pirate" being able to make an infinite number of perfect copies with little or no effort. Adding to this, the ease of sharing content through broadband-internet connections, the need for a sophisticated content protection system has become obvious.

DRM was first implemented on compact discs in the year 2002 when the constellation of record companies consisting of BMG, Arista and RCA started to ship promotional cd:s with DRM protection.

In 2005, Sony BMG introduced a more obtrusive technology which would install DRM software on the customer's computer without their consent. Their attempt failed as the software would only run on certain computers and because of the controversial way of which it was handled.[1]

As the music industry is shifting their channel of distribution from cd:s to online music stores, they are now focusing on how to implement DRM on downloaded music. The largest online distributor of digital music is the iTunes Store run by Apple Inc. and they have since the introduction in 2003 used a system called Fairplay for protecting the downloaded content.[2]

When it comes to film, copy-protection has been in place since the eighties. Hollywood was spearheaded by the company Macrovision that had developed a method to make duplication of analog videocassettes harder. Copy protection was further enforced when DVD-titles entered the market around 1996.[3]

The Content Scrambling System was introduced as the common way to protect DVD:s and it is still in use today. CSS will be further investigated in a later portion of this PM. In 2005 Macrovision introduced RipStop, which is a technology to prevent "ripping" (transferring the content of a DVD-disc to a computer).

As the movie industry is shifting towards High-Definition media such as Blu-Ray and downloadable movie rentals, new methods for content protection have been introduced. The technology used for Blu-Ray is called AACS and we will take a closer look at it in the next few pages.[4]

As the computer has evolved to a full-blown media center, content protection is today implemented to various extent in the hardware and software. Microsoft has gone to great lengths to ensure a protected environment in their latest release of Windows.

---

[1] http://en.wikipedia.org/wiki/Digital_Rights_Management

[2] http://en.wikipedia.org/wiki/ITunes_Store

[3] http://en.wikipedia.org/wiki/Macrovision

[4] http://en.wikipedia.org/wiki/Digital_Rights_Management

How it is used

There are a number of different ways to implement DRM on different kinds of systems. Portable music players, dvd-players, media-center pc:s and cellphones utilise different mechanisms to ensure that the DRM works.

Examples (portable music players):

- **Apple's iPod:** You can only buy music to your iPod using Apple's iTunes Store, and this is managed through iTunes, hence iTunes handles the DRM and checks that it can be uploaded to the iPod of choice, iTunes uploads the necessary keys to the iPod so that it can play the encrypted files.[1]
- **Microsoft Zune**: Microsoft response to Apple's iPod is their Zune, this media player is a bit different from the iPod since it has the Zune Marketplace built right into the unit. This marketplace allows you to buy music on the spot (you've got to have WiFi-access of course), and then download it straight to the Zune. This means that the Zune has to take care of the DRM-related information in the Zune software. So each time you play a song or watch a movie, it tries to read the DRM information and decide whether or not you can play it.[2]
- **Sony Walkman** uses a proprietary protocol to obfuscate the audio files before downloading them to the music player. [3]

When it comes to the PC, manufacturers are still figuring out the optimal implementation. Microsoft is aiming to implement a hardware-based DRM-system. Their attempt to do is called The Next-Generation Secure Computing Base (NGSCB) or formerly known as Palladium. This platform follows the Trusted Computing hardware specification, which consists of:

- Trusted Platform Module: This holds the cryptographic keys in a secure area and a cryptographic co-processor that does all the calculations.
- A curtained memory which is implemented in the CPU so that no application, except the one who allocated that space in the memory, can access it.

The NGSCB is implemented in software by:

- Nexus: This is a security kernel that is implemented in the operating system. This kernel provides a secure area for trusted applications to run in.
- Nexus Computing Agents: These are trusted applications that runs within the Nexus. [4]

---

[1] http://www.roughlydrafted.com/RD/Home/728B5C4B-0A35-40BE-A2E7-E5464C68B80A.html

[2] http://en.wikipedia.org/wiki/Zune

[3] http://drmnews.com/archives/2005/04/11/sony-walkman-digital-rights-management/

[4] NGSCB: http://en.wikipedia.org/wiki/Next-Generation_Secure_Computing_Base

CSS (Content Scrambling System)

Context

CSS is a DRM-technology that is used in most DVD-video discs, so almost every DVD-player has support for this technology. It was introduced in the mid-nineties and is still used today.

Implementation

CSS uses a proprietary 40-bit stream cipher algorithm. The CSS key set used for encrypting/ decrypting the DVD-video is licensed to the manufacturers who then incorporate them into their DVD-players.
The CSS key set consists of authentication key, disc key, player key, title key, second disk key set, and/or encrypted key.

Description of the keys:
- Authentication key is used by the DVD drive to authenticate itself with the CSS decryption module before reading data.
- Title keys are used for scrambling or de-scrambling the actual data on the DVD.
- Disc keys are used to decrypt the title keys on the DVD.
- Player keys are used to decrypt disc keys on the DVD.

Attacks on CSS

CSS was introduced around 1996, but it took until 1999 before a working solution, that could completely decrypt and show the content of a DVD, appeared. It was done by, the now quite famous hacker, Jon Lech Johansen, and two of his anonymous friends who reverse engineered the algorithm and then released it to the public. After the algorithm was released, it was analysed to be very suspicable to brute-force attacks. One of reasons for this was a law in the US which forbade the export of cryptographical keys in excess of 40 bits. There was also a structural flaw in the algorithm which reduced the effective key-length to about 25 bits, which at that time, could be broken by brute force in less than a minute on a 450 MHz machine.[1]

AACS (Advanced Access Content System)

Context

AACS is the next-generation DRM standard for optical discs. The AACS standard was introduced in 2005 and was then adopted by Blu-ray and HD DVD. AACS is controlled by the AACS Licensing Administrator, LLC (AACS LA), which consists of some of the biggest content-providers and hardware-manufacturers in the world.

Implementation

The main difference between AACS and CSS is how the keys are stored. CSS give all players of the same model the same shared decryption key. With AACS, each individual player has their own unique set of decryption keys which are used in a broadcast encryption scheme, similar to the X. 509 certificate scheme used for e-mail and website certification. This scheme allows the AACS LA to "revoke" certain decryption keys and thus certain players. So if a player is compromised, hence the decryption keys extracted, the AACS LA can revoke these keys from future releases.

AACS also has a system for traitor tracing. This system allows the AACS LA to track down decryption keys even though only the titles keys have been released to the public. They can do

[1] http://en.wikipedia.org/wiki/Content_Scrambling_System

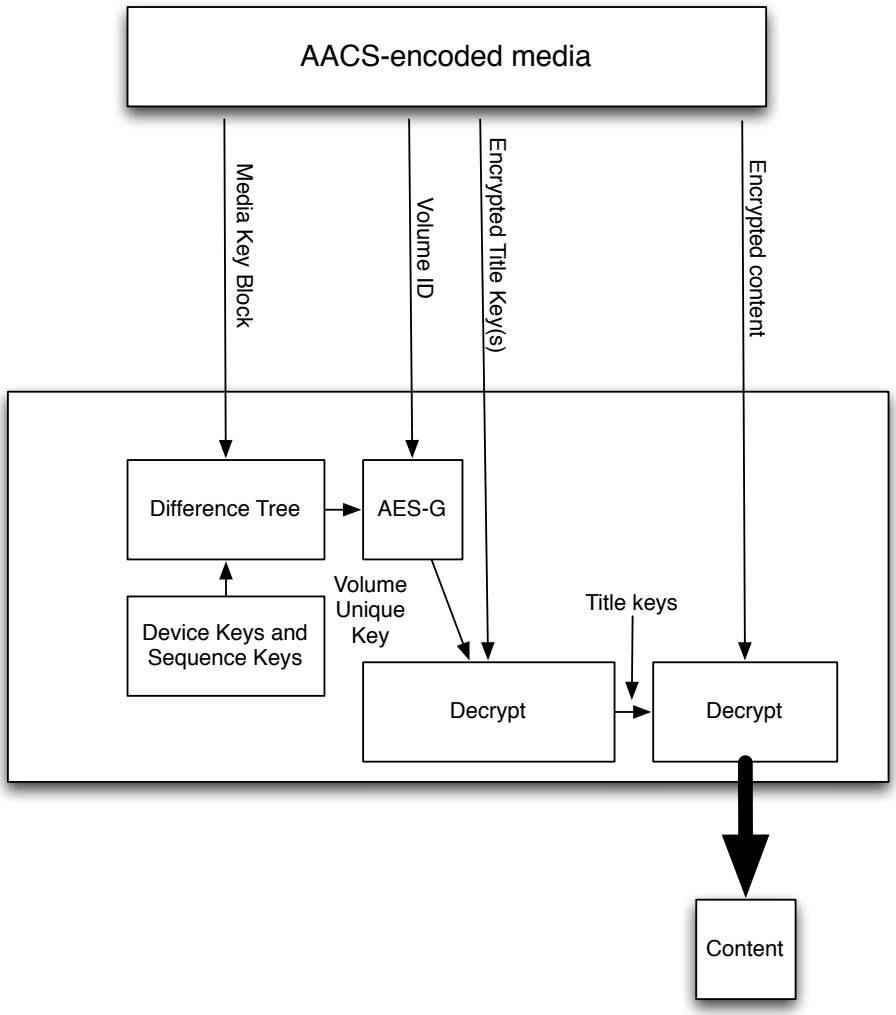
this since the AACS standard allows short sections of a movie to be decrypted with different keys. By analysing the digital watermark in this section, the AACS LA can track down the decryption key and revoke it.

AACS uses one or more Titles Keys to encrypt the content using Advanced Encryption Standard (AES). These keys are derived from a combination of the Media Key and Volume ID (basically the ID of the media).

The Volume ID is a type of unique identifier of the disc, which cannot be duplicated on consumers disc. To read the Volume ID, a certificate signed by the AACS LA is required.

The encrypted media must first be decrypted before it can be watched, this is done using the Media Key Block, Volume ID and the Encrypted Titled Keys. Essentially, a set of keys is arranged in a tree structure such that every key can find every other key except its parent key. So to revoke a player, the Media Key Block only has to be encrypted using the parent key.

When the Media Key Block has been decrypted you will get the Media Key, this key is combined with the Volume ID using the one-way encryption scheme, AES-G, to produce the Volume Unique Key, this key is used to decrypt the Title Keys, which then decrypt the final content.



Attacks on AACS

AACS has been partly broken in the sense that people has been able to extract decryption keys from weakly protected software players, they have also found device keys and other keys used in different processes during AACS decrypting. Other than these approaches, breaking AACS is considered very hard. [1]

[1] http://en.wikipedia.org/wiki/Advanced_Access_Content_System

Fairplay

Context

Apple Inc. introduced the iPod portable music player in 2001. In 2003 they announced the iTunes Online Music Store (iTMS). Songs purchased and downloaded through the iTMS have built in DRM through a technology called Fairplay. FairPlay is built into the QuickTime software which is the multimedia foundation for running the iTunes client which is used to connect to the music store, organising the downloaded content and for connecting to iPod portable music players.

The store has caused much controversy in its limitations that downloaded songs may only be played on hardware running iTunes and only on Apple's own line of media players. The controversy has gone as far as certain countries declaring the system, and the monopoly that follows as a consequence, to be illegal.[1]

The company's view on DRM has since the introduction shifted and they are currently broadening their range of downloadable media that is free of DRM.[2]

Implementation

Songs downloaded through iTMS consists of an encrypted AAC audio stream residing in an mp4-container. The encryption algorithm used is the AES algorithm in combination with MD5-hashes.

In order to download and decrypt songs, the user needs to create an account at the iTMS and authorise a computer for running the iTunes client software. Upon authorisation, Apple assigns a globally unique ID for the computer that has been authorised. The ID is kept with the account records of that particular user on Apple's servers. The user has the ability to authorise up to five computers at once to use with the same iTunes account.

The system uses two keys:

- A master key that is used to decrypt and playback the audio content. It is embedded in the mp4-container.
- A user key that is used to decrypt the master key in the audio file.

Both master key and user key are 12 bytes in length.

Decryption happens in several stages: The first step is to produce an initialisation vector for decrypting the area holding the master key. This vector is acquired by applying a MD5 hash function to the user id and a special initialisation field in the file. When the master key is decrypted, we use it along with a second initialisation vector that has become available through decryption, to decrypt each sample of audio data.[3]

When a user makes a purchase, the following takes place:

- A random user key is generated
- The file is downloaded
- The master key residing in the file is encrypted using the user key

The user needs to authorise all computers and devices that will play the file. Upon authorisation, all user keys associated with the particular user is transferred from the iTunes Server to the user's computer. Worth noticing is that all encryption happens locally in the iTunes client.[4]

---

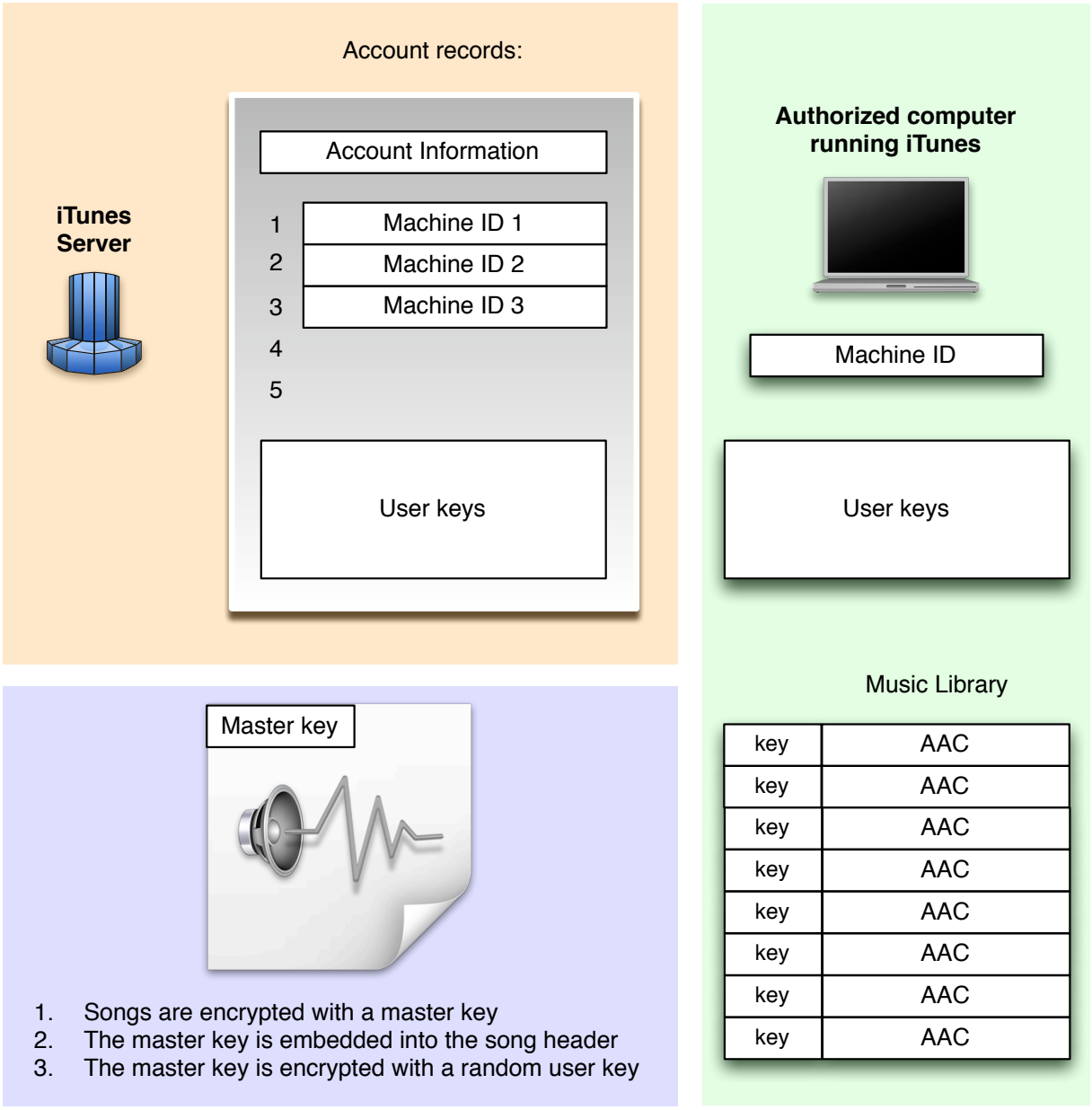
[1] http://www.macnn.com/articles/07/01/24/norway.rules.against.drm/

[2] http://en.wikipedia.org/wiki/ITunes_Store

[3] http://www.hymn-project.org/docs/hymn-manual.html

[4] http://www.roughlydrafted.com/RD/RDM.Tech.Q1.07/2A351C60-A4E5-4764-A083-FF8610E66A46.html

An overview of the iTunes ID- and key system



Attacks on Fairplay

There have been several attempts at extracting the audio content from an encrypted file. The limitation of only being able to play a song on Apple's iPods or in iTunes is what drives the attackers. The methods used so far has been aimed at extracting the AAC audio stream while the song is being decompressed.

A user typically holds the user key for that song, so there is no need to try and de-scramble the encrypted audio. By using these methods, the user can dump the decompressed audio to a new file that is free of the DRM protection. The user may then play the file on any media player.

Other methods have involved using a iTunes client that doesn't lock the songs after downloading them. Other software can be used to impersonate the iTunes client and can request the user keys from the iTunes Server.

The important thing to point out is that the Fairplay encryption scheme has not yet been broken in the sense of an attacker being able to extract the content of an encrypted audio file. There has been no point in attempting this since iTunes provides other weaknesses that the attacker can exploit.[1]

[1] http://www.roughlydrafted.com/RD/RDM.Tech.Q1.07/2A351C60-A4E5-4764-A083-FF8610E66A46.html

Windows Vista DRM

Context

Microsoft have spent plenty of time and effort on upgrading the support for protecting content in their latest release of Windows. They have focused on creating a "protected environment" for video and audio in its operating system.

The protection is embedded deep in the software architecture of the operating system and consists of several modules. The operating system monitors all processes trying to access the protected content and if an unauthorised process is present, the system will stop playing back high-quality media. The key here is quality. The system will prevent anyone trying to capture data from this protected environment. Data leaving the environment will be of lesser quality and thus less valuable to the user. For example, older VGA-connections cannot be trusted and users holding high-resolution VGA-connected monitors will not be able to play back their content in full resolution.

For this to work, all components in the system needs to be certified by Microsoft. Graphics cards and their drivers need to provide protected outputs and abilities to switch them off. Connections that cannot guarantee protected delivery causes the system to downscale the quality of the data stream.

In software, before playback of high quality content can begin, the protected environment needs to be set up and the hardware and drivers needs to be authenticated. When this is done, the operating system has established a protected video path (PVP) for the video stream. We'll look at how the play-back of video is implemented.

Implementation

A combination of algorithms are used depending on the nature of the data stream. The minimum requirement is that graphic cards implement a 128-bit AES algorithm in counter mode. This works for data streams of up to 50 MB/sec and is adequate for compressed video. It takes about 20 clock cycles to encrypt one byte and this is possible on modern computers. However, if we need to play back full resolution content with little or no compression (HD-video), the data stream will be upwards to 250 MBytes/sec and this is too much even for the processors of today.

Researchers at Intel devised of an algorithm, called the Intel Cascaded Cipher, which reuses the 128-bit blocks coming out of the AES encryption engine in a serpent encryption setting. This allows the processor to encrypt one byte using only 4-5 clock cycles. Implementation of the Intel Cascaded cipher is optional for graphics card vendors but it is required in order to supply playback of HD-video content. Implementation of the 128-bit AES counter algorithm is required in order to receive a certification.

For establishing the session key, both ends of the PVP needs to agree on a key. When the first key has been determined, the following keys can be sent encrypted over the path. The session key is established using a 2048-bit Diffie Hellman key exchange. This initialisation key is relatively time-consuming to compute (around 1 sec) but is acceptable since it only happens on boot and wake from sleep. The computation can also be done effectively using the graphics card processor.

The 128-bit key used in the AES-scheme is derived from the 2048-bit D.H-key by pushing the 2048-bit key through a AES Davies-Meyer hash process. The 2048-bit key from the Diffie Hellman process is "forgotten" once the 128-bit AES-key is extracted.

The above mentioned algorithms are the central cryptographic measures that are used to protect the data stream. AES ciphers are further used in other parts of the system such as encrypting/decrypting page-outs to system-memory. Similar protection is also put in place for ensuring a safe delivery of audio streams.[1]

---

[1] http://www.microsoft.com/whdc/device/stream/output_protect.mspx

Conclusions

What have we learned?

The problem of determining the need for DRM falls outside the scope of this PM - thankfully! We have strong feelings about the topic and there is a very lively debate taking place where traditionally the media companies are the ones pushing for the use of DRM.

We have looked at some of the currently common software and hardware environments in which DRM is put in place. Algorithms such as the AES-cipher keeps reappearing when studying encryption/decryption of content. Diffie Hellman key exchanges makes an appearance in Windows Vista and we have bumped into MD5-hashes and other more exotic methods while looking into these case studies.

When we look at the attacks that have been made on these systems we can make a conclusion that today, attackers will probably try extracting keys or circumventing the encryption before attempting brute-force attacks on the encrypted content. This points in the direction that the encryption methods used today are secure but the supporting architecture for distributing keys and storing these is still lacking security.

Future of DRM

Looking into the future and supposing a continued existence of DRM, we can envision software and hardware systems that are more closed and confined. If media companies have their way, content on media players and computers will be kept in air-tight stuctures that will only allow extraction of the content through certified hardware components.

Manufacturers of Blu-ray players for personal computers, demand to put their own DRM kernel modules into the operating systems that will utilize the players. Companies like Apple have opposed this and are thus not yet offering Blu-ray equipped hardware. The same skepticism is present in the *nix world where openness is one of the foundations. Hardware manufacturers will have a hard time requiring the open source community to implement mechanisms similar to Microsoft's Protected Video Path.