Group: Indigo

FairPlay: Effectiveness and Weaknesses of Apple's Digital Right Management Technology

Abstract

In this paper, we reported our study of Apple's Digital Right Management technology, also known as FairPlay. We reviewed current literature about the effectiveness and weaknesses of FairPlay. We also reviewed a set of known anti-DRM tools that exploits FairPlay. We performed a root cause analysis of the known attacks. We believed that Apple faced a dilemma between changing their business model and improving FairPlay's security model. We recommended future that improve security of FairPlay.

Introduction

Apple's iPod is the most ubiquitous portable digital music player in the market today [1]. The iPod is available in several models that use either a hard drive or flash memory to store multimedia content. Almost all iPod owners use a proprietary application called iTunes [2] that is developed by Apple. iTunes can connect to the Apple own online music store -- iTunes Music Store -- where users can purchase music and recently video and download the multimedia content to their iPod's.

Copy protection for the iTunes Music Store's media is provided by Apple Digital Rights Management system, also known as FairPlay. Fairplay limits the usage of the media to a number of computers and iPods. Another limitation is that the only operating systems that are accessible to iTunes are Mac OS and Microsoft Windows, leaving some major systems such as Linux out in the dark. These complaints have led to the "hacking" of Apple's FairPlay DRM system. Two of the most well known hacks are PyMusique and Hymn. Both were developed under the guiding hand of Jon Lech Johanson [3]. PyMusique allows users to employ a software of their choice, while Hymn decrypts the music so it can be played an unlimited amount of times.

Like most computer systems and software, the security model of Apple's FairPlay is not completely bullet proof. As a matter of fact, during the course of this project, we have learnt that FairPlay has a number of weakness. We present a summary of current literature in the next section. We also have learnt a number of tools developed to exploit these weaknesses. We refer to these tools as "anti-DRM tools". We present a detailed analysis of these anti-DRM tools in a separate section. We perform a root cause analysis of these attacks and summary our findings in a separate section.

Literature Review

The following presents several literary pieces that were in the media regarding Apple FairPlay DRM technology and its known vulnerabilities. A short description of each provides an overall picture of how FairPlay works and its vulnerabilities.

Geer (2004) discussed that most DRM technologies, including Apple FairPlay, are closed

systems. In general, there is little or no interoperability between one system and the another. Incompatible DRM technology prevents a song protected in one DRM technology from being played on portable music players using other a different DRM technology. In other words, a consumer can only play a song that is purchased and downloaded from Apple iTunes Music Store on Apple's own iPod portable music player but not Dell's music player. Interoperability between DRM systems is a major drawback in the digital market, and cause Some consumers refuse to purchase DRM-protected media because of its lack of transparency. Geer also noted that DRM technology vendors are unwilling to share their design or implementation because they are afraid of the discovery of new security vulnerabilities.

Unlike many other DRM technologies, Arnab and Hutchison (2004) found that Apple FairPlay's DRM controller is implemented using the software virtual machine technique. They noted that Apple DRM controller runs at the normal application level. Compared to other DRM controllers that run at operating system or hardware levels, application level controller has two inherited weaknesses. The first weakness is that application-level DRM controller is hard to generalize and prone to incompatible implementations. The second weakness is that the controller is vulnerable to exploits at the operating system level.

Based on previous reverse-engineered efforts, Swartz (2005) documented the XML interface to iTunes Music Store as well as its cryptography usage. Specifically, iTunes program uses HTTP XML messages to communicate with iTunes Music Store. The communication is encrypted using AES128 CBC algorithm. The encryption key was determined to be "8a 9d ad 39 9f b0 14 c1 31 be 61 18 20 d7 88 95". Hengeveld (2005) maintained a list of common XML commands of iTunes Music Store. Because these technical information was previously private only to Apple engineers, the public disclosure will likely result in future exploit.

Swartz (2005) also described a weakness, in the overall iTunes architecture, that can be used to circumvent the five-computer limitation of iTunes program. As a result, a user can authorize as many computers as he or she likes to play the protected music he or she downloaded from iTunes Music Stores. This effectively circumvents a major restriction of FairPlay.

Greek (2005) reported an exploit with Apple's DRM that dealt with the communication protocol between Apple iTunes and Apple iTunes Music Store. The attacking program interfaced directly with Apple iTunes Music Store online service, bypassing the security measurement within the Apple iTunes program. When a song was purchased, the attacking program was able to download an unprotected audio file from Apple's servers because the addition of DRM was performed in the Apple iTunes program.

Rosenblatt (2003) accounted an attack of FairPlay by exploiting an implementation weakness in Apple QuickTime player. The attack did not break the encryption scheme used in FairPlay. Instead, the attack cleverly retrieved the decrypted and uncompressed audio data from a temporary memory location used by QuickTime. The uncompressed audio data can be easily converted to the a compressed file without any DRM information.

Singer (2004) reported that RealNetwork has introduced a DRM translation technology,

called Harmony Technology after successfully reverse-engineered Apple FairPlay technology. Harmony allows creation of FairPlay-compatible music files – a task that was only capable by Apple at the time. In doing so, music purchased from online music stores besides Apple's own iTunes Music Store, could be played on Apple own iTunes or iPod devices. This "hack" solved one of the interoperability problems.

Knight (2004) described another attack on Apple iTunes program, where the attacker was able to determine the secret encryption key used to secure communication between Apple iTunes and Apple Express base station. According to the author, the secret key was probably obtained from an undisclosed vulnerability in a bug in Apple's program. Therefore, the attack would still be possible even if Apple updates the secret key.

Apple has addressed most of the attacks in newest version of iTunes program and iTunes Music Store. In some cases, Apple forced their customers to upgrade the iTunes program in order to continue using iTunes Music Store. Apple also performed firmware upgrade of some versions of iPod to counter RealNetwork's Harmony Technology.

Anti-DRM Tools Review

An introduction to tools employed to defeat Apple's DRM scheme, needs to mention famed hacker Jon Lech Johanson. As a teenager from Norway, Johanson first rose to the public's attention by developing a hacking program called DeCSS. The program broke CSS (Content Scrambling System) encryption, a weak encryption program for movie DVD's. The episode opened up a Pandora's box of sorts for the digital copyright community as it opened up discussion of user's privacy versus a company's copy right protections. This was Jon's first of many trysts into the controversial world of DRM.

After nearly being jailed for developing DeCSS, he set his sights on developing applications that exposed holes in Apple's DRM. Johanson played a hand in the development of PlayFair, SharpMusique, and [10] also influenced others to develop the Hymn project with his FairKeys program. This section will touch upon some of Johanson's tools to defeat Apple DRM. A more thorough analysis of PyMusique and Hymn will also be presented.

In December of 2003, Johanson released on his website, PlayFair (written in C), which was the first decoder for iTunes FairPlay. His website stated that he reverse engineered Apple FairPlay. PlayFair works by first decoding the file using user key from iPod or the Windows system [11]. It then creates new mp3 with the same Meta data intact without the DRM code. This technique was imitated by RealPlayer's Harmony project in order to play m4p files. PlayFair was shortly followed by DeDRMS, a C# implementation of PlayFair to be played in the Windows platform.

Apple countered PlayFair by successfully banning it from existing in both the SourceForge and Sarovar websites. Johanson immediately came back with FairKeys, a tool to extract user key from the iTunes music store server, given the correct username and password. FairKeys influence has broadened as it is currently used by the Hymn project (http://www.hymn-project.org) to decode m4p file using user key retrieved from iTunes music

store server. The Hymn project's goal is to "exercise your fair-use rights under copyright law."

Working with Travis Watkins and Cody Brocious, Johanson went a step further and in March 2005, introduced PyMusique, an open source iTunes music store client [12]. We were able to download the program from www.drmnews.com, but were not able to test it because of our unfamiliarity with Phyton. PyMusique exposed another hole in Apple's DRM, song file encoding. DRM occurs in the client machine after the download of the file has been completed.

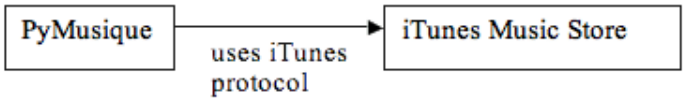

Figure 1: PyMusique accessing iTunes Music Store

Music files transferred from iTunes music store do not get encrypted by the server, but by the user iTunes' software. PyMusique mimics the iTunes application. It connects to iTunes music store server and purchase (even register) using real iTunes own user identification. As a result, during the purchase of the song, PyMusique does not encrypt the file.

Another user advantage of PyMusique is that unlike iTunes, user can re-download the song just in case the user unintentionally deleted the original file. PyMusique also performs iTunes functionalities like browsing through and previewing the available songs (see figure 1). It just lacks the aesthetic feature of iTunes, which has a more detailed graphic user interface.

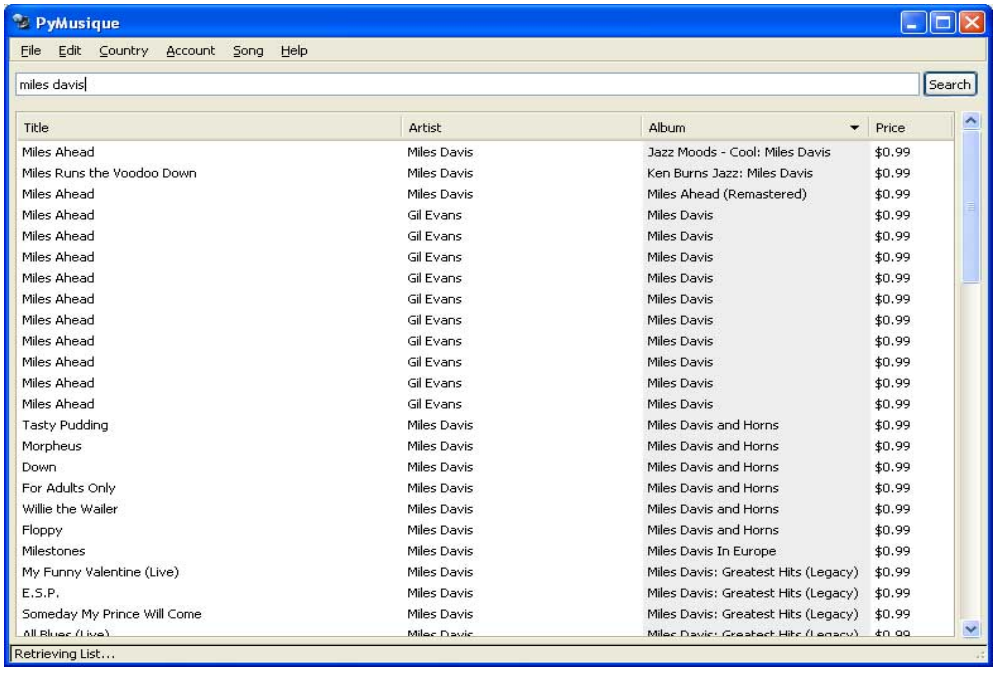

Figure 2: PyMusique User Interface

Johanson compares PyMusique to an e-mail client that connects to mail servers and retrieves or fetches email that was destined to that user. Thus he rationalizes that it is similar to iTune's and is therefore legal. Moreover, Johanson said that PyMusique uses the same protocol as iTunes [13].

iTunes version 4.6 was especially vulnerable to PyMusique, but Apple has fought back by upgrading to iTunes 4.7. Not one to be denied, Johanson and his partners released PyMusique version 0.4, the last version for Windows according to Cody Brocious. And in trying to keep a step ahead, Johanson released PyMusique's descendent SharpMusique[14]. SharpMusique is currently available through Johanson's personal website.

Influenced by Johanson's work, Anand Babu maintains and owns Hymn, which stands for Hear Your Music aNywhere [15]. Hymn is also legally supported by FSF (Free Software Foundation) of India [16]. Hymn converts m4p file files to mp3 or wav format (See Figure 2). It works successfully like its predecessor, PlayFair. The current version of Hymn can convert m4p files downloaded by iTunes 4.6. Babu has not been able to crack though DRM for files downloaded using the newest iTunes version 6.0.

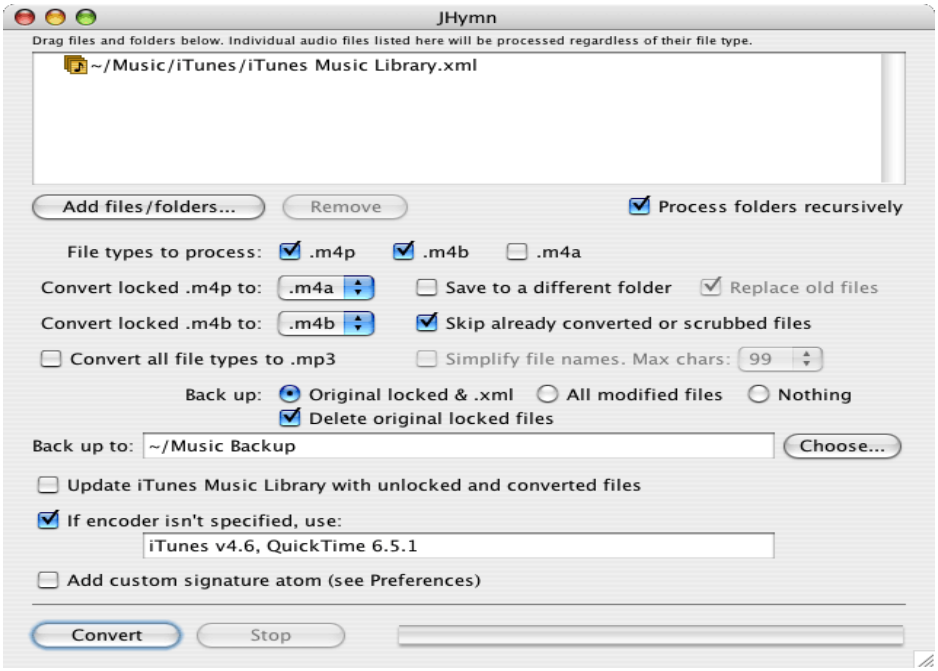

Figure 3: JHymn User Interface

Hymn works in both Windows and OS X. The JHymn Windows version appears to be about the same as JHymn in OS X. Hymn has 3 interfaces that work in OS X:

1) JHymn, the Java based application (Figure 3). Of the three, this is the most user friendly because of the neat OS X windowing interface.
2) Hymn simpler drag and drop interface using Cocoa (Apple's objective-C object oriented programming for OS X applications) [17].

3) Lastly, it can simply be used employing the command line.

From our analysis, it is apparent that this application is based on FairKeys application. FairKeys, a program for extracting your iTunes DRM FairPlay keys from apple server [18]. In implementation with Hymn, FairKeys is employed to access iTunes Music Store and obtain a user key to decrypt the file.

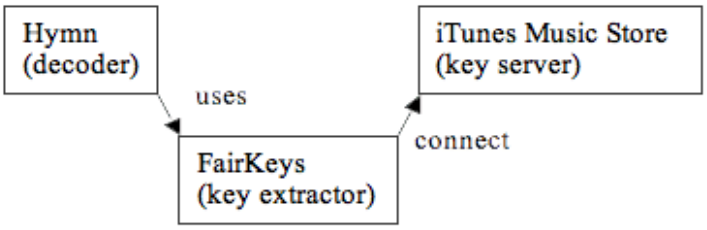

Figure 4: Relationship among Hymn, FairKeys and iTunes Music Store

What is interesting about the application is that unlike PlayFair, it will not try to find the existing user key from the attached iPod or user machine. In the Apple machine, we suspect iTunes to store those information in KeyChain. The application will simply ask for the user key from the music store server, given user name and password. It already knows the user name. We suspect that it does this by probably reading the m4p file itself.

When the user is prompted to enter his password, the prompt window will state that it will access the iTunes music store to get the keys. Given that you are using iTunes 4.6 when you purchase the song and you enter the right identification and password as stored in the iTunes server(music store), then Hymn will convert your m4p file(s) to mp3 or wav.

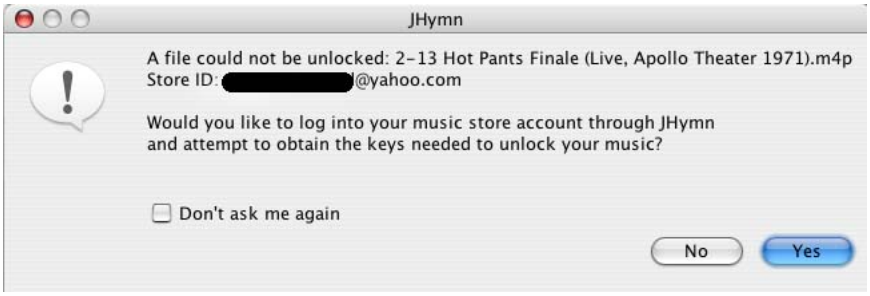

Figure 5: JHymn Warning Interface

Hymn does not rip the m4p file into new mp3 file. Instead, it basically opens up the m4p file and create new mp3 file. Thus, information about who purchase and own the original file still exists in the mp3 file. This fact is what leads Anand Babu to think that this program is not piracy and is therefore legal. Babu said that it is not piracy if the Meta data is still intact in the file (mp3) [16]. The Meta data in the new file includes the user name of purchaser and album cover (cover art) of the song.

This fact about Meta data has been tested and shown with the following scenario:

1. Alice buys a song from iTunes (songA.m4p) and plays the song on her computer.

The album cover appears on iTunes.
2. Bob copies the file to his computer. Bob was unable to play this file because it belongs to Alice.
3. Bob then downloads Hymn and runs it on his computer
4. Bob asks Alice to type in her iTunes user name and password. Hymn is able to successfully convert the m4p file to mp3.
5. When Bob plays this song in his iTunes, the album cover appears.

Regular mp3 file does not have this information so it was concluded by Babu that the mp3 still has the Meta data intact.

Hymn also lets the user add information to the resulting file. For example, the user can put string of a text to it. This is basically the same as watermarking your file. Hymn user can use this feature by filling the custom signature field in the preference window. The manual stated that the user has to enter atom value (4 character string) and string of text [19]. But there is a catch. The atom value should not be the same as what Apple used because it may corrupt the file. However, there seems to be no application out there that would analyze song file signature or Meta data unless you want to go binary with it.

Hymn packages are still available to be downloaded from www.hymn-project.org and was tested successfully on an m4p file bought using iTunes 4.6. The interface tested was JHymn on OS X running Panther with iTunes 4.6. JHymn was also tested unsuccessfully on m4p file bought using iTunes version 6.

Root Cause Analysis

As we were reviewing the current literature and anti-drm tools, it became obvious to us that Apple and the attackers are engaged in a mouse and cat chase. Whenever a new venerability or a new tool to circumvent FairPlay was released, Apple would react by patching the security hole in the system (when feasible) or pursuing legal means to protect the system.

In the course of CSCI E-170, we learnt that law exists in some countries, e.g. Digital Millennium Copyright Act (DMCA) of United States, prohibits any attempt to circumvent a DRM technology. So, we are puzzled by the fact anyone would choose to engage in illegal activities. Our curiosity led us to perform a root cause analysis of the problem.

We attempted to understand the attackers and their motivation. In our analysis, we identified three user groups as the most likely suspect. The first group is a small population of users who historically do not respect intellectual property rights. This group often do not follow the general fair use guidelines. In the past, this group might have performed illegal copying of music CD or tapes. This group was also likely among the first who participated in illegal sharing of mp3 music files on popular file sharing services. DRM technologies primarily exist to deter or minimize the activities of this group. However, this group would likely attempt to break or circumvent any DRM technology used to protect the music content. This group is also the primary users of anti-DRM tools.

The second user group is likely created as a direct result of the FairPlay technology and Apple's closed design. This group represents typical music CD consumers who find the FairPlay-protected music has a far more restrictive usage model. For instance, a CD music can be played on an unlimited number of computer systems and by whoever that own the physical music CD disc. On the other hand, a song purchased from Apple iTunes Music Store can only be played on up to five personal computers or by five users. Moreover, this group may desire to purchase music currently not available on iTunes Music Store and play the music, encoded in other DRM technology, on their iPod music players. While Apple has been successful with iPod and the current balanced usage model, a restrictive and closed usage model will always entice some users with the technical know-how to exploit any potential weakness.

Apple iPod's tremendous success and popularity is a magnet attracting interest of the third group. Some call this group the "hackers". Of the three groups, this group has the most technical expertise and is capable of undermining FairPlay through reverse engineering efforts. This group is highly motivated to break FairPlay because a vulnerability discovery of FairPlay will always certainly generate worldwide press coverage. Hackers such as Jon Lech Johanson who continually try to seek glory and acknowledgement are naturally drawn to FairPlay. As FairPlay exploits continues to grow and attract attention, more hackers will be attracted to this group. As more people try to break FairPlay, the more likely someone will come up with a new vulnerability.

In addition to the three motivated groups, our analysis led us to believe that the FairPlay security model is fundamentally weak. First, the FairPlay DRM controller is implemented at the application level and therefore is susceptible to attacks at the operating system level. Theoretically, an attacker may install a sound device driver that intercept decrypted audio stream and convert the stream back to an unprotected music file. While no known exploit of such exists at this time, we believe this type of attack will become popular in near future.

Second, Apple continues to rely on secrecy as part of the overall security model of FairPlay. Such security design is generally considered to be unsound. Once a sensitive piece of information is revealed, the system may be easily rendered insecure. Some of the recent attacks were possible because of the availability of previously undocumented technical information. E.g. anti-DRM tool, e.g. Hymn, utilized the XML interface of iTunes Music Store. These exploits based on reversed-engineered technical information proved the architecture itself has many holes.

Ironically, we believe allowing a wider audience, particularly the academic researchers, to review its current design may be Apple's best chance of improving the FairPlay technology for the long run. The additional independent reviewers will likely find existing design or implementation flaws that are yet known to Apple engineers. Apple could use the information to improve FairPlay and overall iTunes architecture.

However, we do not believe Apple will pursue such course because it does not want to risk others creating FairPlay-compatible devices or services. A market of iPod- or iTunes-clones will likely destroy the exclusiveness and dominance of Apple iPod's franchise. Unless Apple

changes their closed business design, Fairplay will likely remain a closed design and is subjected to future attacks. We see this as the dilemma of Apple and its FairPlay security model.

Recommendation for Future Works

Given our analysis of the attackers' motiviation and Apple's present dilemma, we present the following ideas for future consideration. First, we think that Apple should move away from a DRM solution that uses a application-based DRM controller. A hardware-based DRM controller could eliminate several classes of attacks, e.g. software bugs and operating system intercept attack, and therefore offer a far greater security. As Swartz (2005) noted, each FairPlay-protected song has a private key that is stored in an unknown secret location on each authorized computer. It is possible that someone will find out the secret location and breaks the FairPlay security model. By storing the secret keys at the hardware layer, FairPlay significantly reduces the possibility that the secret keys are compromised.

Furthermore, we propose that migrating FairPlay to a security model that does not involve private key exchange. Such model may use the public and private key system. Specifically, each iPod device and iTunes authorized computers may each have a pair of public and private keys. The keys may be encoded at the hardware layer. The iTunes Music Store also has a pair of public and private keys. The audio file targeted at a specific device will be encrypted at the server side using the device's public key. Because private key is kept locally, no other device can decrypt the audio file. Nonetheless, we suspect it may be difficult to retrofit existing iPod devices to support this model.

In addition to strengthening security design of FairPlay, Apple could consider making FairPlay a lesser target by addressing the motivation of the attackers. Currently, FairPlay prevents iTunes customers from playing their purchased music from iTunes Music Store on any portable digital music players other than Apple own iPod players. As a result, some consumers are unwillingly locked into Apple's own online music store, program and portable music players. We propose that Apple licenses the FairPlay technology to some portable music manufacturers. That will satisfy the end users who wish to use other portable players (like Dell's players) to play their music downloaded from the iTunes Store. More importantly, the move eliminate the need for the paying customers to circumvent FairPlay in order to play their music anyway.

We also proposes Apple licenses other's DRM technologies and incorporate them into iPod. Similarly, this will satisfy the end users who wish to use their iPod players to play music downloaded from the other online music stores. This also eliminates the need for the paying customers to circumvent FairPlay in order to play their music anyway.

Furthermore, we believe that there should be more interoperability between different DRM technologies. Any portable player should play a song encoded any DRM format. Existing DRM vendors, e.g. Apple wants to lock in their user base, a standard DRM is unlikely to happen any time soon. However, we propose that Apple forming a alliance among the major players. The alliance might produces greater interoperability among various DRM technologies and further stimulate the growth of digital music industry.

We also recommend creating a consortium of all the major stakeholders including IT professionals, media content providers, law makers, consumer groups and the device manufacturers. The consortium would be charged with tackling the above three issues of interoperability, DRM integrated hardware and open design. With all the major stakeholders in dialogue, major progress can be made in at least one of these fronts. Submission of any standard proposal can be submitted to a standard governing body.

Our last two proposals on alliance and consortium are made based on our root cause analysis. We believe that by addressing the primary motivation of the attackers, i.e. giving freedom of choice, the majority of music consumers will find little or no need to compromise FairPlay even given an anti-DRM tool.

Conclusion

It is easy to see how Apple is using DRM to protect its own business model. As more and more users get hooked on iPod and iTunes, their reliance on it grows. As a result, Apple's control over the market will increase. A downside is that this often challenges hackers such as Jon Lech Johanson to try to beat Apple at its own game. PlayFair, PyMusique and Hymn are just a few examples that he has had a guiding hand in fostering. RealNetworks and Anand Babu have picked up Johanson's enthusiasm in trying to crack Apple's system. We are certain that if Apple continues its current course, others will join the hacking bandwagon.

Apple's iPod and its go it alone attitude with FairPlay is not just hindering its own progress, but also the progress of digital music industry as a whole. As shown in this paper, its secretive and its popularity team up to create cauldron of tension between themselves and the end users. While the end users expect more transparency with the rise of the iPod, Apple wants to make it harder for interoperability.

Apple can take several steps to improve its security model. Instead of using a DRM closed design such as FairPlay, Apple should consider opening their design. They can also move towards a DRM solution that is integrated into the hardware where each iPod has a unique private key for decrypting music streams.. There should be more liberal interoperability between different DRM's. Downloading one DRM should allow you to access another.
And lastly, a consortium of all the major stakeholders should be created to at least open up dialogue about the direction of DRM. In an area so broad in scope, there needs to be some standardization. Maybe one standard will not be possible, but several encompassing ones can be accomplished in the near future.

BIBLIOGRAPHY

[1] Brad Gibson, "First on TMO: Apple Exec: Shuffle Grabs 58% of Flash Player Market; What Cell Phone Threat?", 4 May 2005,
[Online] Available: http://www.macobserver.com/article/2005/05/04.4.shtml

[2] Wikipedia, "ITunes", last modified 18 Sep. 2005, [Online] Available:
http://en.wikipedia.org/wiki/ITunes

[3] Anonymous, "Apple's Music Services Under Attack?", 19 Aug 2004, The Berkman Center for Internet and Society at Harvard Law School, [Online] Available:
http://cyber.law.harvard.edu/home/filter?wid=379&func=viewSubmission&sid=560

[4] David Geer, "Digital Rights Technology Sparks Interoperability Concerns", IEEE Computer Society, 2004

[5] Alapan Arnab and Andrew Hutchison, "Digital Rights Management – An Overview of Current Challenges and Solutions", Information Security South Africa (ISSA) Conference 2004, 2004

[6] Dinah Greek, "FairPlay as crackers get to core of iTunes", 21 Mar. 2005, [Online], Available:
http://www.computing.co.uk/computeractive/news/2012404/fairplay-crackers-core-itunes

[7] Bill Rosenblatt, "DVD Hacker Turns to iTunes", 2003, Nov 23, 2003 [Online] Available:
http://www.drmwatch.com/drmtech/article.php/3113431

[8] Michael Singer, "Real's Harmony Taps Apple's Core", Jul 24, 2004, [Online], Available:
http://www.internetnews.com/infra/article.php/3386411

[9] Will Knight, "iTunes wireless music streaming cracked", 13 Aug 2004, [Online], Available:
http://www.newscientist.com/article.ns?id=dn6282&lpos=home1

[10] Jon Lech Anderson, "About Me", 2005 [Online] Available: http://nanocrew.net/about

[11] Cd-rw.org, "Jon Johansen's PlayFair decrypter for Apple iTunes", Afterdawn, 6 April 2004 [Online] Available: http://www.afterdawn.com/news/archive/5118.cfm

[12] Thomas Mennecke, "Getting Around iTunes DRM", 19 Mar. 2005, Slick Times, [Online] Available: http://www.slyck.com/news.php?story=706

[13] Peter Cohen, "PyMusique lets you buy iTunes songs without DRM ", Playlist Magazine, 18 Mar. 2005 [Online] Available: http://playlistmag.com/news/2005/03/18/pymusique

[14] Wikipedia, "PyMusique", last modified 26 Oct. 2005 [Online] Available:
http://en.wikipedia.org/wiki/FairPlay

[15] Wikipedia, "FairPlay", last modified 15 Nov. 2005 [Online] Available:

http://en.wikipedia.org/wiki/FairPlay

[16] Ketola, "PlayFair resurrects as hymn", 12 May 2004 [Online] Available: http://www.afterdawn.com/news/archive/5236.cfm

[17] Apple Developer, "Cocoa" [Online] Available: http://developer.apple.com/cocoa/

[18] Anonymous, "DVD Jon realeases FairKeys", 8 Jul. 2004, BoingBoing [Online] Available: http://www.boingboing.net/2004/07/08/dvd_jon_releases_fai.html

[19] Jhymn Info and Help version 0.9.1, [Online] Available: http://www.hymn-project.org/jhymndoc/

[20] WWDC, "Apple to Use Intel Microprocessors Beginning in 2006", 6 Jun. 2005, Apple Public Relation Library [Online] Available: www.apple.com/pr/library/2005/jun/06intel.html.

[21] Aaron Swartz, "Behind iTunes Music Store: A Technical Description of iTMS and FairPlay", Nov 20, 2005, [Online], Available: http://www.aaronsw.com/2002/itms

[22] Willem Jan Hengeveld, "about the apple music store protocol", Nov 20, 2005, [Online], Available: http://www.xs4all.nl/~itsme/projects/misc/itunes.html