SELF-PROTECTING DIGITAL CONTENT

— A TECHNICAL REPORT FROM THE CRI CONTENT SECURITY RESEARCH INITIATIVE —

Paul Kocher, Joshua Jaffe, Benjamin Jun, Carter Laren, Nate Lawson

Keywords: Piracy, risk management, watermarking, renewability, programmable security, forensic marking

EXECUTIVE SUMMARY

Introduction

Despite the high public profile of piracy as a threat to intellectual property owners, surprisingly little useful research has been done to understand the range of technical solutions that are feasible. This paper presents results from a study sponsored by Cryptography Research, Inc. to determine how cryptographic systems can provide the most effective long-term deterrent to the piracy of digital video and other content distributed on optical media.

Although numerous products and technologies have been advertised as solutions to the problem of piracy, most commercial security systems fail catastrophically once an implementation is compromised. These designs can work in limited deployments, but any technology deployed as part of a major standard will inevitably attract extremely determined attacks – and some implementations will get broken. The long lifespan of media formats, diversity of player implementations, complexity of security/usage models, and constantly-changing risk scenarios provide attackers with numerous avenues of attack and the time and resources to explore them. As a result, effective content protection systems must be able to survive compromises and adapt to new threats.

Risk Managing an "Unsolvable Problem"

Risk management approaches often provide the only way to manage security problems in situations where unbreakable solutions are unavailable or impractical. For example, the major credit card networks are based on fundamentally insecure magnetic stripe technology, yet risk management efforts have held fraud rates below 0.1 percent. Similarly, computer security flaws are discovered frequently, but users can manage (though not eliminate) their risk by applying software updates and by using anti-virus programs. Without risk management tools, neither credit/debit networks nor the Internet could survive.

Piracy, like credit card fraud and computer security, is a problem that cannot be solved completely. Our research identified technical systems that give content owners the ability to control their risk. The most practical and effective of these combine programmable code with encrypted digital content. This code would be distributed as part of the content, execute dynamically during playback, and enforce each title's security policies. Publishers could then control security for their own content.

Programmable Security: Smart Content

Programmable security approaches give publishers the freedom to add new countermeasures and improve security after a standard has been widely

Examples of correctable problems with existing content protection systems:

- After players are sold, security is static and cannot evolve as new attacks and new threats appear.
- Compromises beyond the decoder (digital output devices, software device drivers, etc.) are not recoverable.
- Product vendors do not receive clear benefits for investing in security.
- Copies cannot be traced to decoders for revoking equipment, reducing pirates' anonymity, or helping with prosecution.

adopted. Players would include a simple virtual machine with APIs that provide data about the playback environment, such as player information, software versions, output device types, and user commands. The content-specific code would analyze this data and control whether and how decoding would proceed. The code can also use player APIs to authenticate output devices, support player-specific security features, validate user actions (e.g., copy vs. play), check whether media is consumer-recordable, and implement locale-specific requirements. Content being decoded by software-based PC players could even check for malicious software or device drivers. Playback can be prevented if the environment is unacceptable.

The Chess Game: Avoiding Checkmate

The security flaws in the system used to protect DVD video cannot be fixed without abandoning compatibility with the installed base of DVD players. Programmable protection systems have a unique ability to avoid this category of problem by shifting responsibility for security from players to the content itself. While compromises will still occur, new titles can carry security code that corrects for past vulnerabilities. As a result, each attack has an effective response. Content owners will be able to constantly upgrade security over time – even to correct for risks that were not known when the original system was designed.

Although risk management can control problems, no security technology can eliminate piracy. Some attacks, such as copying from analog outputs (speakers, displays, etc.) using general-purpose recording devices, are impossible to prevent completely and will always remain a threat. Similarly, no player or media technology can eliminate piracy using Internet-based file sharing networks. When problems do occur, self-protecting content can be used to correct security weaknesses and to identify/revoke pirates' equipment, although responses to the most determined pirates will continue to require law enforcement.

Economics of Security

Today, product manufacturers generally bear the costs of providing security, but do not receive the benefits. As a result, vendors lack incentives for making significant investments in controlling piracy. Placing security code on the media helps correct this economic imbalance by giving content owners responsibility for the security software used by their own content. This also gives manufacturers incentives to become active participants in security because only well-designed players will be trusted by publishers with their most compelling content. Publishers can use their control over each title's security to manage their risk and maximize profits.

Forensic Marking: Uncovering Pirates

Effective risk management requires the ability to detect and to respond to problems. While media-based security code makes it possible for new content to resist known attacks, publishers must also be able to gather information from past compromises. Watermarks have been proposed for carrying security-related information. Unfortunately, it appears to be infeasible to make a watermark that is secure against removal by adversaries who have reverse engineered the mark detector More generally, we do not believe that conventional ("public") watermarks will prove effective as a robust way to block copying in widely-deployed standards.

Fortunately, a new class of steganographic marks provides an attractive alternative to conventional watermarks for risk management purposes. These "forensic marks" are embedded dynamically and can carry detailed information about the decoding process. Unlike conventional watermarks, forensic marks can be provably secure, efficient to embed, imperceptible, and extremely robust.

Publishers can analyze mark contents to determine the specific equipment and methods used to make each pirated copy. This data is essential for rights holders to be able to revoke devices used for piracy, improve the security of future content, and prosecute pirates. Because forensic marks embed identifying information in decoded (analog) output, they have the

Table of Contents

psychological benefit of reducing the perceived anonymity and safety of piracy without affecting the privacy of legitimate users.

Need for Leadership

Investments in security have been inadequate relative to the major economic threat posed by piracy. After successfully lobbying for the Digital Millennium Copyright Act, publishers have failed to present a coherent long-term technical strategy.

Efforts to improve security will require strong technical leadership. Without clear objectives, standards efforts tend to degenerate into unwieldy and ineffective committees with short-term focus. Leadership is also needed to verify that security needs are met before products ship and to help secure designs succeed in the marketplace. We conclude that only rights holders can provide this leadership; no other participants have the motivation, expertise, or resources to ensure the deployment of effective anti-piracy technologies.

* * *

Cryptography Research, Inc. provides consulting services and technology to solve complex security problems. In addition to security evaluation and applied engineering work, CRI is actively involved in long-term research in areas including tamper resistance, content protection, network security, and financial services. This year, security systems designed by Cryptography Research engineers will protect more than $50 billion of commerce for wireless, telecommunications, financial, digital television, and Internet industries. For additional information or to arrange a consultation with a member of our technical staff, please contact Jennifer Craft at 415-397-0329 or visit www.cryptography.com.

Paul Kocher is President and Chief Scientist of Cryptography Research. His work includes designing numerous cryptographic applications and protocols, including SSL v3.0, the world's most widely used security protocol. In addition to leading the team at CRI that discovered differential power analysis and designed the record-breaking DES key search machine "Deep Crack", he is also credited with discovering timing attack cryptanalysis and co-founding ValiCert, Inc. (NASDAQ:VLCT). His work has been reported in forums ranging from technical journals and Scientific American to CNN and the front page of the New York Times. Paul can be contacted via e-mail at paul@cryptography.com.

Josh Jaffe is a Security Architect at Cryptography Research, Inc. who specializes in signal processing applications, cryptographic implementations. He holds B.S. degrees in computer science and physics/astronomy from Brandeis University. Josh can be reached via e-mail at josh@cryptography.com or at 415-397-0324.

Benjamin Jun is Vice President of Cryptography Research and is responsible for consulting services and content protection efforts. He has developed and evaluated numerous systems for the protection of financial transactions, audio content, and pay television. Prior to Cryptography Research, Ben worked at IDEO Product Development on Secure Content Distribution Systems. He has also held positions at Bain & Company, the National Institute of Standards and Technology, and the Institute for Defense Analysis. Ben holds B.S. and M.S. degrees in Electrical Engineering from Stanford University, where he is an NSF Graduate Fellow and a Mayfield Fellow. Ben can be contacted at (415) 397-0323 or at ben@cryptography.com.

Carter Laren is a System Architect at CRI with a background in electrical engineering and extensive experience designing and implementing hardware and software cryptographic components. Prior to joining Cryptography Research, Carter worked at L-3 Communications where he designed secure communication systems for both government and commercial applications. He also held the position of Weapon Systems Engineer at Lockheed Martin, where he designed and tested portions of the AEGIS Combat System. Carter is a Chancellor's Scholar at the University of Pittsburgh, where he received a B.S. degree in Electrical Engineering. Carter can be contacted at (415) 957-2667 or at carter@cryptography.com.

Nate Lawson is a Senior Security Engineer at Cryptography Research, Inc. with a background in systems engineering and network security. Prior to joining Cryptography Research, Nate designed and implemented network devices, intrusion detection systems, SAN appliances, and media distribution networks for companies including ISS, Decru, and Nifty Devices. He also co-founded Elite Networking (http://elite.net). Nate can be contacted at (415) 397-8662 or nate@cryptography.com.

The Content Security Research Initiative is an ongoing effort funded by Cryptography Research, Inc. to solve security problems for the content distribution industry. This effort has yielded significant advances in securing pay television broadcasts, Internet downloads, and optical media. Results from the study (including approaches in this paper) are protected by U.S. patents #6,298,442, #6,327,661, #6,304,658, #6,188,766, #6,289,455, #6,381,699, and/or #6,278,783; other U.S. and international patents are pending, including U.S. patent application 20020141582 and U.S. provisional application 60/279,323 (which specifically cover programmable self-protecting content technologies). Please contact Cryptography Research for more information about the initiative, other research results, or technology/patent licensing.

1. INTRODUCTION

If hard drive densities continue to double annually, a drive costing $250 in 2012 will be able to store 160 terabytes – enough for over 10,000 full-length high-definition movies plus 100,000 uncompressed CDs.[1] Similar improvements in communication technology will provide users with the bandwidth required to utilize this storage capacity. These advances are presenting increasingly complex risks and challenges for those wishing to limit piracy and profit from their intellectual property.

Some have argued that the pirates will prevail, because all content will eventually be available as "unprotected bits" that can be copied easily and anonymously. For example, one cryptographer has argued that, "All digital copy protection schemes can be broken, and once they are, the breaks will be distributed... Average users will be able to download these tools from Web sites that the laws have no jurisdiction over."[2]

Our research challenges these dire predictions and examines the question of how security technologies can most effectively control piracy in the long-term while satisfying the needs of consumers and device manufacturers. Although our results support the view that the total elimination of piracy is not a realistic objective, we believe that properly-designed technical systems can provide an effective deterrent and prevent piracy from destroying the value of digital content.

Cryptography developed from the need to keep information private. In many ways the field is very advanced – the best modern cryptographic algorithms are flexible, efficient, reliable, and virtually unbreakable. Even an attacker with the entire world's computing power, access to virtually unlimited amounts of encrypted data, and the best known attack methods cannot break a single strongly-encrypted message.

Strong algorithms do not necessarily make systems secure. Weaknesses in the protocols and products that manage keys and decrypted content make it unnecessary for attackers to break the underlying cryptographic algorithms. Unfortunately for content distribution systems, implementation weaknesses are so common that compromises are virtually inevitable.

The primary technical challenge is therefore to design architectures that maintain their effectiveness even after individual devices or implementations have been compromised. Protection measures that fail catastrophically when attacked are clearly not acceptable as long term solutions. In contrast, even relatively easy-to-break approaches may be useful if they provide a lasting deterrent to low-budget or casual piracy and limit the problem to professional operations that can be targeted by investigative and legal efforts.

This paper presents results from a study sponsored by Cryptography Research, Inc. to determine

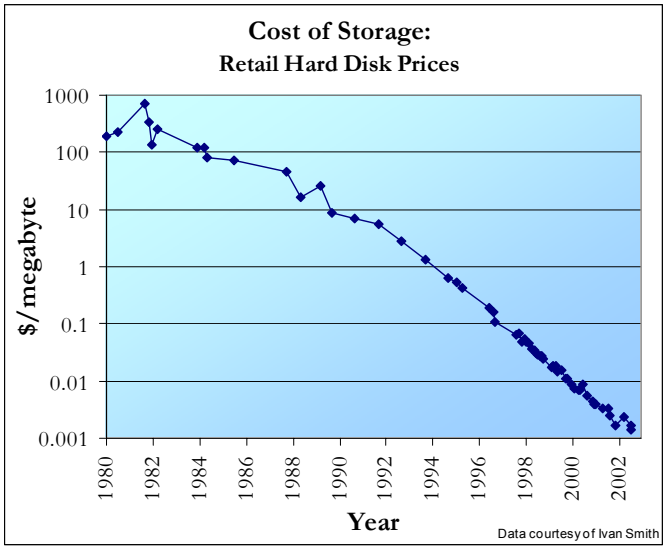

Figure 1: Cost of storage – advertised hard disk prices.

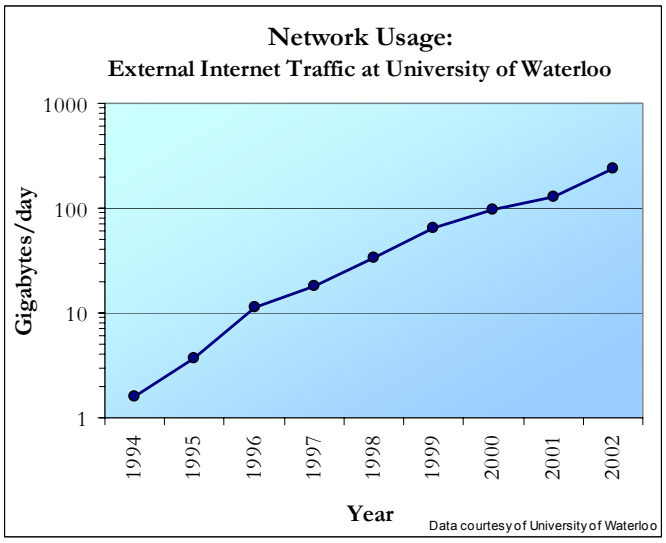

Figure 2: Internet usage at University of Waterloo.

[1] (10,000 movies × 9 gigabytes) + (100,000 CDs × 650 megabytes) = 155 terabytes. A 160 gigabyte drive cost $250 in July 2002. A similarly-priced drive in 2012 is expected to hold 160 terabytes.

[2] Schneier, Bruce, "The Futility of Digital Copy Prevention," Cryptogram, May 15, 2001.

whether technical systems can provide a meaningful long-term deterrent to piracy. The examples in this paper focus primarily on the problem of securing video distributed on conventional (passive) optical media, although our results are also applicable to broadcast/Internet distribution and other content types. We do not address philosophical questions such as whether artists should be able to apply copy protection to their work.

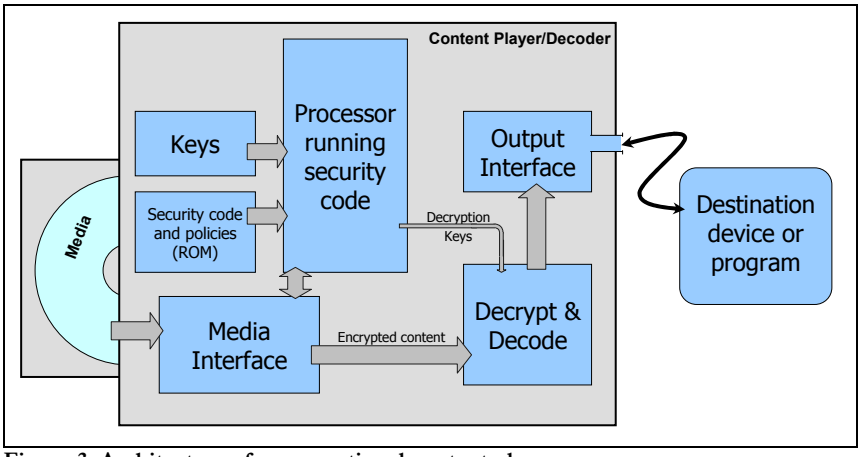

Figure 3: Architecture of a conventional content player.

2. CSS & OTHER CONVENTIONAL ARCHITECTURES

The Content Scramble System[3] (CSS) used for DVD video is noteworthy because of its widespread use and poor design. CSS is implemented in the player and provides a simple, fixed security policy for all content: any device with valid keys can decrypt all media valid in its region.

Figure 3 shows the architecture of a typical player implementing a conventional content encryption scheme such as CSS. The content is compressed, encrypted, then distributed on read-only media. To allow off-line playback, every player is pre-loaded with all keys required to decrypt all media it will ever decode. The security scheme is defined in the player, typically as software, and enforces a set of fixed security rules. After decryption, the content is sent to an output interface, which is typically unprotected or has protection features that are independent of the protection used on the media.

CSS failed to meet even its limited security objectives. Although CSS contains many design flaws, the most catastrophic was the use of proprietary cryptographic algorithms which proved trivial to break. After a player compromise, CSS was supposed to allow new DVDs to be mastered so that they could not be decoded by players with revoked manufacturer keys. Poor use of cryptography allowed attackers to circumvent this capability. Today, circumvention software is widely available, but CSS cannot be repaired without making the entire installed base of DVD players obsolete. In practice, CSS would probably have failed even without the obvious cryptographic weaknesses, as consumers would not have tolerated the revocation of a major manufacturer. Other limitations of CSS include its inability to revoke individual decoders, adapt security policies to new threats, secure/revoke digital output formats, or trace pirated content back to a compromised device.

The security problems in CSS can be traced back to the design process. CSS was developed by product companies without major exposure to piracy or adequate experience designing secure systems. The Copy Protection Technical Working Group (CPTWG), which was supposed to ensure the security of DVD, was politically divided and lacked leadership or active participation by experienced cryptographers or security engineers. As a result, the CSS specification failed to provide adequate assurance of its own security, yet unrealistically assumed bug-free implementations.

Because CSS failed to give implementers clear incentives to ensure security, implementation quality became an increasingly major problem after the success of the DVD format was assured.[4] Some vendors even appear to have intentionally produced insecure products to help users circumvent the CSS region coding. For example, the region coding on many players can be defeated by pressing a "secret" sequence of buttons.[5] The source of the problem is that manufacturers profit from sales to people who circumvent the region coding, but do not incur losses when their products are broken.

[3] The official specifications for CSS (also called Content Scrambling System) are confidential and are licensed by the DVD Copy Control Association (http://www.dvdcca.org).

[4] Cryptography Research ultimately discontinued auditing CSS implementations because vendors wanted documentation that their products were not the "least secure" on the market, and were not interested in identifying and correcting security problems.

[5] Numerous web sites specialize in documenting these sequences. See, for example, http://www.regionfreedvd.net and http://regionhacks.datatestlab.com.

3. DESIGN CHALLENGES

Content protection systems must address many technical challenges. Although a complete requirements analysis is beyond the scope of this paper, Figure 4 lists several of the major security and design requirements reflected in our analysis. The feasibility of meeting these requirements will be reviewed in detail at the conclusion of this paper (Section 10).

Renewability
Playability
End-to-End Security
Cost
Openness
Player Diversity
Migration Path
Assurance
Incentives for Security
Forensic Reporting

Figure 4: Design challenges for content protection systems.

4. RISK MANAGEMENT FUNDAMENTALS

Although cryptographic algorithms and some other elements used in copy control systems can be extremely secure, other components are much more difficult to protect. For example, determined adversaries will find ways to copy media, modify players, and redistribute data. As a result, we have little optimism that any complete copy protection system will survive unbroken throughout the life of a successful media format. The lack of perfect security does not necessarily support claims that rights holders need to adopt new business models because "copy protection efforts are doomed"[6] and rampant piracy is inevitable.

Risk management approaches have the potential to provide a long-term deterrent without perfect security. Instead of trying to anticipate and prevent every possible attack, risk management systems are designed to respond to dynamic threats and recover from compromises.

Other industries depend on risk management to control security problems that cannot be solved completely. For example, software vendors have largely failed to produce defect-free programs, but provide users with patches to address security risks as they are discovered. Similarly, anti-virus programs require frequent updates in order to detect newly-discovered viruses. Although reactive approaches will never eliminate security risks, attacks can be prevented from getting out of control. Without security updates, the Internet as we know it could not exist because each new flaw or virus would be catastrophic.

Financial institutions also rely on risk management techniques. Although credit card networks are based on fundamentally insecure magnetic stripe technology, risk management tools have been able to hold credit card fraud rates below 0.1% of transaction volume.[7] In practice, even lower fraud rates could be achieved by adjusting credit scoring and transaction risk management parameters, but doing so would tend to decrease profits by denying more valid transactions and increasing costs.

Risk management systems are only effective if they provide the ability to detect attacks and to respond. For example, software companies actively seek out information about new viruses and security flaws, then respond by issuing updates. Similarly, credit card companies detect fraud by using neural networks and other risk assessment tools to analyze data collected from point-of-sale terminals. When a high-risk transaction is identified, actions are taken to mitigate the risk, such as declining the transaction, obtaining additional cardholder verification, or suspending the account. Because responses incur costs (such as the loss of customers whose transactions were declined), risk management approaches try to maintain a steady state that balances risks and mitigation costs (see Figure 5).

Content protection systems have several important advantages over credit card security systems. For example, fraud rates considerably higher than 0.1%

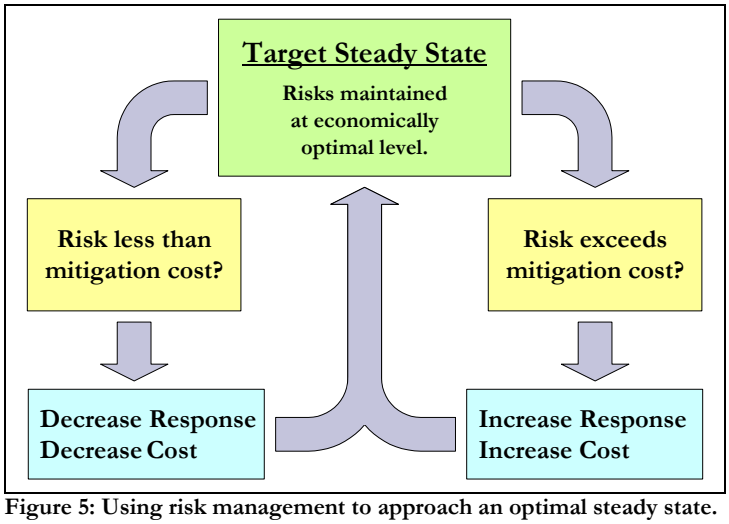

Figure 5: Using risk management to approach an optimal steady state.

[6] Chmielewski, Dawn, "Andreessen: Copy protection efforts are doomed," The Mercury News, Apr. 9, 2002. (Available on-line from http://www.siliconvalley.com.)

[7] "Fraud Rates Decline with Visa's Innovative, End-to-End Solutions", Visa USA media release, September 2001.

Requested Actions	Player Information	Media Information	Output Information	User Information
Play	Model/version	Format	Type	Name
Copy	Form factor	Recordable	Manufacturer	E-mail address
Record	Memory contents	Pre-recorded	Quality/bit rate	Telephone #
Export/Convert	Revision status	Capacity	Version	Payment card #
Eject	Playback history	Manufacturer	Device keys/certs	Registration #
Delete	Serial number	Serial number	Serial number	IP address
⋮	⋮	⋮	⋮	⋮

Figure 6: Examples of player information on which risk management decisions can be made.

are generally tolerable (though undesirable) because piracy represents lost opportunity instead of lost money.[8] Similarly, while stolen credit cards can be used to buy goods that can be fenced, more effort is required to convert stolen intellectual property into cash.

Despite these advantages, content protection technologies must be able to operate without on-line notification and authentication when content is rendered. As a result, risk management systems must be specially designed to enable content owners to detect problems and to respond effectively.

5. PROGRAMMABLE SECURITY

Threats against anti-piracy systems are dynamic and unpredictable. Although some existing systems can detect or respond to specific types of attack, approaches that address a limited aspect of the problem (such as decoder compromises) are of little use if attackers can simply target other parts of the system (such as digital outputs). To be effective, content protection systems must have the ability respond effectively to an extremely broad range of threats – including attacks that were not anticipated when the system was originally designed.

Existing anti-piracy systems generally use static decoding processes that are defined as part of the media format and implemented in every player. Of these schemes, some newer ones (such as CPPM[9] used for DVD-Audio) support the revocation of individual players, although it is unclear how compromised devices would be identified. Static systems also generally lack the flexibility required to address security risks beyond the decoder itself, such as compromises of digital output devices or software device drivers. If a static system is widely broken, as occurred with DVD-CSS, the problem cannot be remedied without replacing the installed base of players.

We believe that future formats must be able to mitigate unexpected risks. Instead of implementing the security system solely in the player, much of the content's protection system and decoding software can be *distributed as part of the content itself*. Having each title carry its own security logic, policies, and countermeasures makes it no longer necessary to anticipate and prevent all possible attacks when the media format is designed. Deferring security decisions until the content is mastered (or, in some cases, decoded) allows security problems to be corrected without changes to the media format or the installed base of players.

> The content's protection system and decoding software can be distributed as part of the content itself.

Under this type of security architecture, the player provides an execution environment for the security code that is distributed with the content. The player component would typically be implemented as an interpreter or virtual machine (as used by languages such as Java™ or BASIC). The player would also provide the content's code with access to cryptographic primitives and detailed data about the playback environment, such as the information in Figure 6.

Although the player provides raw information, the content's code controls how this information is used. For example, if a player has marginal security or if the user is making a copy, the content might decide to play at standard quality. High-definition playback could be reserved for players with superior security. If a player is

[8] For an interesting economic analysis, see: Liebowitz, Stan, "Policing Pirates in the Networked Age," Policy Analysis No. 438, Cato Institute, May 15, 2002.

[9] "Content Protection for Prerecorded Media Specification", available from the 4C Entity, June 28, 2000.

known to be compromised or cannot be trusted to provide correct information, the content could refuse to play, at least until the player's security is upgraded. Of course, for titles where piracy is not a concern, code could allow unrestricted playback on all players.

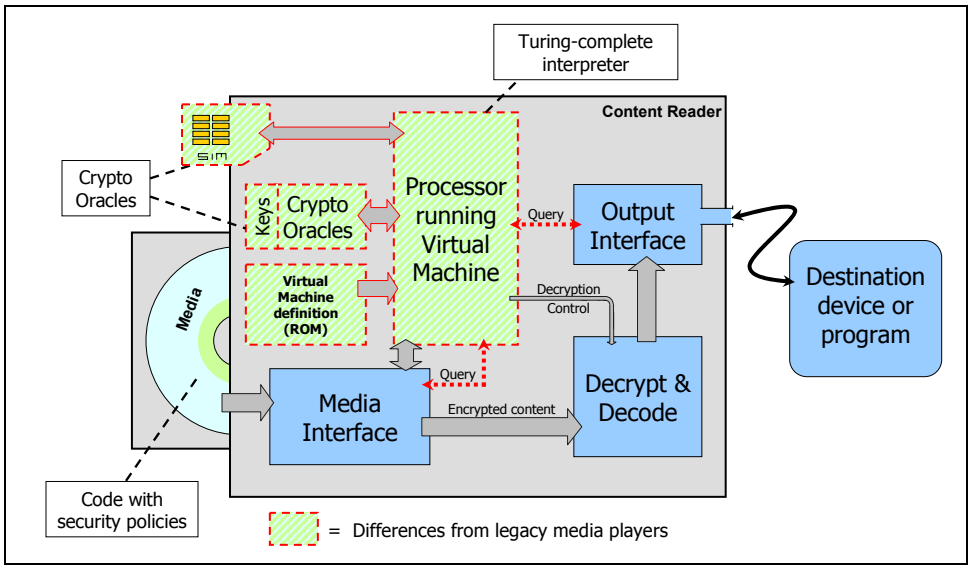

Figure 7: Architecture of a programmable content player.

The flexibility gained by separating the player design from the security code can improve both security and the user experience. For example, existing systems often allow only system-wide, irreversible, all-or-nothing choices about whether to revoke players with marginal security. In contrast, programmable systems allow flexible responses such as allowing playback at reduced quality, adding user verification steps, or displaying customized warning messages.

Programmable systems can also solve unexpected problems. For example, even though this capability was not planned, a publisher could prevent discs in multi-disc sets from being sold or rented separately by checking for the first disc of the set in the player's history. Greater flexibility can also help with antitrust issues by allowing participants to make their own security decisions. Although this paper focuses on security issues, programmability can also be used for non-security purposes.[10] For example, content-based code can be used to overcome format limitations or provide user interactivity.

6. IMPLEMENTATION

Figure 7 outlines the general architecture of a typical programmable content player. The player ROM contains code for an interpreter (virtual machine) instead of the static security policies used by legacy systems. As described previously, the interpreter would also provide the content's code with information about the playback environment as well as cryptographic support. If desired, some keys could be placed on a removable security module, such as a smart card.

The content's code needs to have access to cryptographic functions that use the player's keys, but the code should not have access to the keys themselves. Architectures that do not provide this separation are vulnerable to compromise by poorly or maliciously designed content. Hardware-based players should ideally separate player keys in a separate EEPROM memory that is accessible only by the player's cryptographic module. Software-only decoders would typically store keys in obfuscated form. Drives for use in general-purpose PCs could also include cryptographic keys and support in the drive itself.

Prior to deployment, the playback process needs to be standardized. This effort would include defining the interpreter, the programming interfaces (APIs) that provide the content code with information about the playback environment, and the key management system. Considerable technical expertise is required to produce good specifications, particularly for highly-constrained and complex systems. Although often neglected, careful testing and verification are also necessary to provide high assurance in a design's security.[11]

Compared to legacy designs, hardware-based decoders will tend to use slightly more silicon area. Software-based decoders are likely to incur a modest performance overhead and use slightly more RAM. These differences should be minor, however, when

[10] Note that adding simple programmability to a platform is not sufficient for security purposes. For example, existing video game players lack security-related APIs and key management capabilities necessary to enable secure device revocation and forensic marking.

[11] Careful evaluations reduce the chance of unexpected failures and help relying parties understand their risks. Cryptography Research encourages third-party evaluations of all security designs, including our own. For critical systems, testing can exceed the design effort by a factor of 10 or more.

compared to the advances predicted by Moore's Law.[12] The additional storage space required for security code should be negligible given the storage capacities available on modern optical media.

For basic security capabilities, an interpreter capable of 1 MIPS with 128 kilobytes of memory would be minimal but adequate. As with non-programmable systems, a small nonvolatile memory for storing keys and a higher-speed cryptographic module would also be needed. The nonvolatile memory should also include room for carrying software updates, player information, cryptographic certificates, identifiers of revoked devices/media, and historical information about previous media and attached devices. In theory, a basic design should not add more than a few cents to the incremental manufacturing cost of a high-volume hardware-based player,[13] and nothing for a software-only player. Other costs for product vendors include product design and technology licensing, although these are partially offset by transferring responsibility for security policy implementations to rights holders.

More expensive designs could offer better performance, security, and features. For example, players that store and manage their keys and historical data in separate dedicated hardware can offer better tamper resistance. Players with Internet or telephone connectivity could support on-line security verification, downloadable security updates, and alternative business models such as pay-per-view. Secure internal clocks could also enable subscription-based pricing models. Higher-performance systems with video displays could even support general-purpose computing applications such as web browsers, interactive content, or video games.[14]

These features, and virtually all others, could be optional. Manufacturers could add extensions or features to their products and offer them to publishers. The content's code would determine what capabilities are supported and decide whether and how to use them. Even security itself can be optional, since rights holders could control whether products such as unsecured open-source software decoders or disc copiers could decode their content. In practice, coordination between product vendors and rights holders is also important to ensure a consistent and positive customer experience.

While publishers would be responsible for mastering their own content, we expect a market to develop for third-party tools. These tools could range from simple protection systems to full-featured digital rights management systems (DRMs).[15] Vendors would compete to provide publishers with the best features, security, and cost.

Although the content would control its own security, some key management processes should be centralized to help ensure compatibility. This service would provide product manufacturers with certificates describing their products' capabilities, and would provide publishers with information about players. It would also supply keys to enable new products to decode older content (subject to the content's security policies). It would also provide data to publishers so that their content could be decoded by players issued in the future (again, subject to the content's security policies). If desired to stimulate competition, multiple key management services could exist in parallel.

7. POINT-TO-POINT VS. END-TO-END SECURITY

The models pursued by the SDMI committee and most other anti-piracy standardization efforts are based on providing point-to-point security. Content is encrypted when it is stored on media or communicated between devices. Each device decrypts the input it receives, decompresses the data, and (for digital outputs) re-encrypts it for the next component. Additional devices decrypt, process, and re-encrypt the content until it is ultimately sent to an analog output. Figure 8 shows an example of a point-to-point system with three devices.

Point-to-point systems are only secure if all supported devices and protocols are secure. For example, if the keys from one device's input are cracked and published on the Internet, other devices will continue to output content encrypted using these keys. Even if content owners are aware of the attack, nothing

[12] Every 18 months, the number of transistors per square millimeter is predicted to double, and the cost per transistor will fall by half.

[13] As of July 2002, retail DRAM costs are below 0.03 cents/kilobyte, flash memory prices are below 0.04 cents/kilobyte, and CPU prices are below 10 cents/MHz. Actual costs could be higher if a new chip was required, or lower if the necessary hardware was already available.

[14] It is important to note that programmability is necessary but not sufficient for decoders to support self-protecting content. For example, conventional computers or video game machines would at least require additional software.

[15] The DRM industry is currently struggling due to the difficulty of simultaneously and ubiquitously deploying compatible players and content. Programmable systems can help by eliminating the need for explicit player support for each DRM.

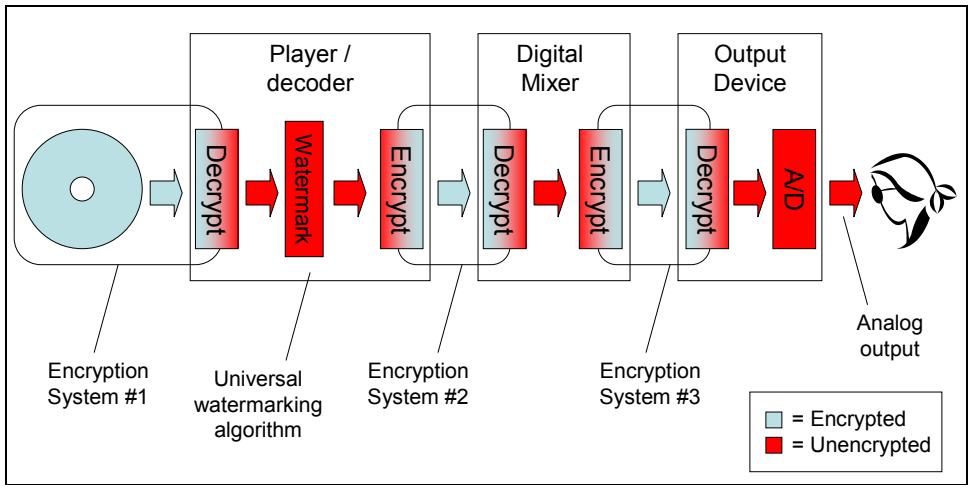

Figure 8: Point-to-point content protection system.

can be done to address the problem without losing compatibility with all fielded devices.

Although some existing schemes allow the revocation of individual player devices, player revocation is generally not effective against downstream attacks. For example, if the output device in Figure 8 is compromised, the content cannot prevent intermediate devices from using the compromised keys. In fact, the player device is unaware of how the content will ultimately be used. Player revocation features are also of limited use unless there is a practical way to detect compromises and respond to situations where a large number of devices share a security flaw.

Systems providing end-to-end validation can provide much better risk management capabilities than those with only point-to-point security. Figure 9 diagrams the operation of a sample system with end-to-end security using the program-based approaches described previously. Although links between devices are still encrypted individually, the initial decoder device validates how the content will be used downstream.

End-to-end validation can be implemented by having the player/decoder provide an interface through which the content's security code can identify and query downstream objects. The code can use this information to control whether and how playback would proceed and to deliver security parameters or even security code to downstream devices.

In Figure 9, the plaintext (decrypted) content does not leave the validated environment until the final analog-to-digital conversion. Compromises prior to the analog conversion can be handled using the content-controlled programmable risk management approaches described in Section 5, while forensic marking techniques (see Sections 8 and 9) can help prevent piracy from analog outputs.

In general, we believe that point-to-point designs are unlikely to provide a long-term deterrent in major deployments due to their lack of risk management capabilities. End-to-end systems are not necessarily any less likely to be broken, but are likely to prove much more effective over the long-term because recovery is possible from a much broader array of compromises.

8. PUBLIC (CONVENTIONAL) WATERMARKING

Watermarks have been proposed as a way to detect and control copying. For example, the SDMI committee planned to use an audio watermark to convey a "do not copy" signal to recording devices. This design implies a "public" watermarking system, consisting of a mark embedding algorithm (which can be public or private) and a public detection algorithm. The detection

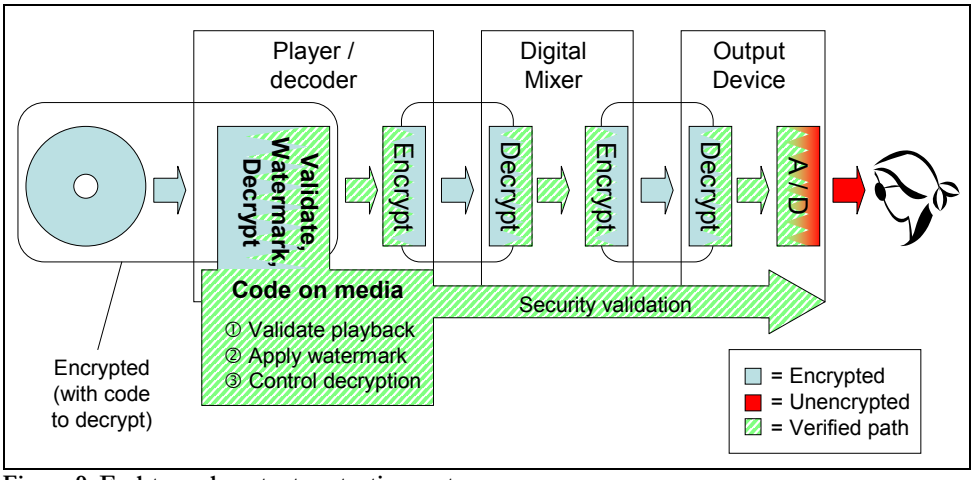

Figure 9: End-to-end content protection system.

algorithm is assumed to be public (known to attackers) because it must be standardized and deployed in large numbers of recording devices, some of which will eventually be reverse engineered.[16]

Although secure public watermarking systems would be enormously useful in combating piracy, there are convincing arguments that they are impossible to construct for audio, video, images, and other normal content. The basic challenge is that knowledge of the detector allows attackers to determine when the mark has been removed. For example, a simple automated attack that will break all schemes we know about is to use successive approximation (also called sensitivity analysis) to construct unwatermarked versions of marked content by repeatedly making tiny changes until the mark is no longer detected (see Figure 10).[17]

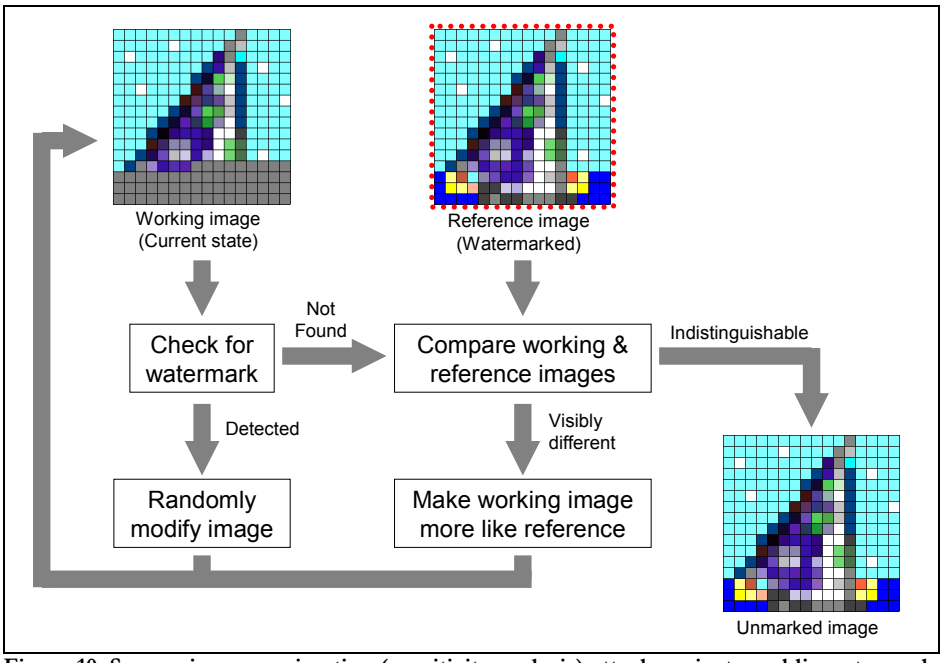

Figure 10: Successive approximation (sensitivity analysis) attack against a public watermark.

In addition to security concerns, current watermarking proposals are computationally complex, making them expensive to embed and to detect. Other common problems include distracting artifacts and the inability to survive common transformations such as cropping and compression. Although some progress is being made at improving robustness and efficiency, we are not optimistic that a practical and secure public watermarking scheme is possible.

9. FORENSIC MARKING

For effective risk management, publishers must be able to respond to attacks. Although programmable security capabilities can provide a flexible response mechanism, appropriate responses require knowledge about the specific equipment and processes used to make pirated copies. Methods used to convey this information need to be secure, efficient, and respect users' privacy.

Because players must be able to operate off-line, the only practical and effective channel for reporting information is the output content itself. Although conventional watermarks could theoretically be adapted for this purpose, forensic marks provide a practical and provably secure alternative. Forensic marks embed identifying and diagnostic information in outputs, but do not use a fixed detector. As a result, they are able to avoid the security problems with conventional watermarks, but cannot be used in systems such as SDMI where the detection algorithm must be standardized and deployed widely.

To embed each bit of a typical forensic mark, the player device decrypts and outputs one of two (or more) versions for a portion of the content (see Figure 11). From even a heavily-degraded analog recording, the embedded data can be recovered by determining which of the versions is present. Because the detection process is not fixed, each mark bit can be represented by virtually any difference in the output. If the decoding process is controlled by content-specific security code, this code can choose what to output and can also generate decryption keys to secure the selection. The actual information that is encoded in the forensic marks could include any data available during playback, such as the parameters listed in Figure 6 (page 7).

[16] In practice, many systems can often be broken without even reverse engineering the detector. For example, see: Craver, Scott et al., "Reading Between the Lines: Lessons from the SDMI Challenge", Proceedings of 10th USENIX Security Symposium, August 2001.

[17] For more information about this attack and several others, see Cox, I., Miller, M., and Bloom, J., Digital Watermarking, Morgan Kaufmann Publishers, 2002, pages 307-317.

In a simple example using video, the media might carry two versions (polymorphs) for a small portion of each of 500 video frames. During playback, the content's security code first obtains data identifying the player device and any output devices. The code uses this data to select which version of each polymorphic frame to decrypt. Given a recording of the decoded content, the publisher can determine which version of each marked frame is present, and use the recovered data to identify the devices used to make the copy.

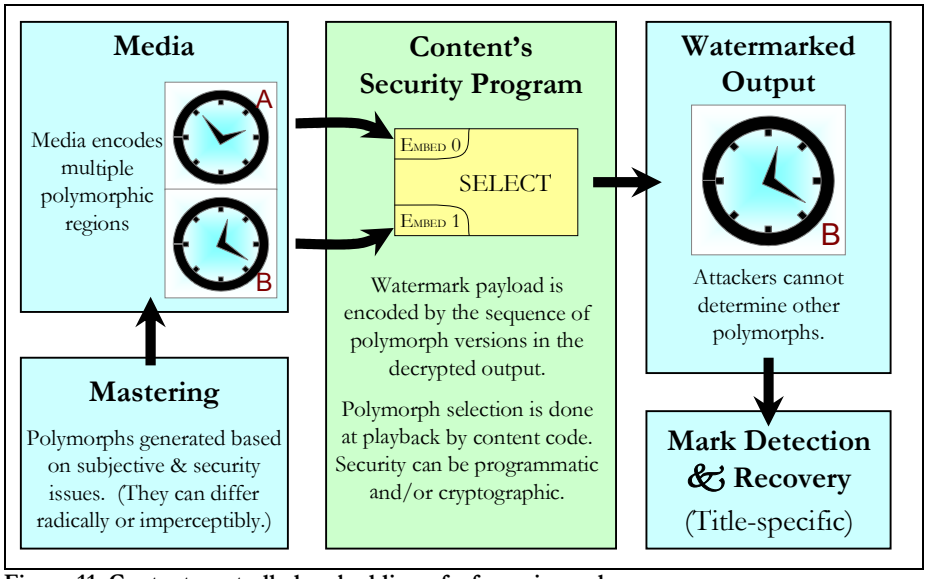

Figure 11: Content-controlled embedding of a forensic mark.

Forensic marks can be both provably secure and provably robust. Because no constraints are placed on the variations (polymorphisms) in the content, knowledge of one does not enable attackers to determine others. The polymorphs are stored on the media or generated on-the-fly by the content's code, and can be protected using conventional cryptographic or programmatic security measures. The locations of variations can also be concealed securely by encrypting portions of the decoding software. Without knowledge of what variations are present or where they are located, attackers cannot reliably remove forensic marks without destroying the content. (See Figure 12 for an example.)

The plaintext content is divided into portions $P_1..P_n$. A randomly-selected portion P_i ($1 \le i \le n$) is modified to create an alternate version P_i' such that the change cannot be identified from the context ($P_1..P_{i-1}$ and $P_{i+1}...P_n$). Portions $P_1..P_n$ and P_i' are encrypted with random keys $K_1..K_n$ and K_i' then stored on the media in random order. A first decoding program D_1 is constructed that includes keys $K_1..K_n$ and indexes for locating the encrypted $P_1..P_n$ on the media. A second decoding program D_2 is constructed with $K_1..K_{i-1}$, K_i', $K_{i+1}..K_n$ and indexes to $P_1..P_{i-1}$, P_i', $P_{i+1}..P_n$.

Programs D_1 and D_2 are encrypted with program keys K_{P1} and K_{P2}, respectively, and stored on the media. Finally, the values of K_{P1} and K_{P2} are placed on the media encrypted so that the set of players that should embed the bit '0' in the mark can determine K_{P1} (and only K_{P1}), while all other valid players (which embed '1') can only recover K_{P2}.

A player decrypts either D_1 or D_2 using K_{P1} or K_{P2}. Because D_1 decrypts the content with P_i while D_2 decrypts with P_i', the value of the marked bit can be recovered by analyzing the output.

An attacker with either D_1 or D_2 (or their outputs) cannot determine which portion has multiple versions or what the differences are. As a result, the adversary cannot reliably destroy the mark without also destroying the content so extensively that all possible changes become undetectable (i.e., completely obliterating the work). An adversary with both D_1 and D_2 can produce an output containing both P_i and P_i' or that omits P_i and P_i', but this reveals even more information to the publisher, notably that the copy was made by combining outputs from at least one device in each group.

Figure 12: Example of a provably-secure, provably-robust forensic mark.

Because content-specific code can control the decryption process, publishers can choose during the mastering process what data will be encoded in each mark, where marks will be placed, and how marks are encoded. For example, variations can be chosen to accommodate artistic or subjective requirements. Marking can also be disabled if piracy is not a concern.

When a pirated copy is recovered, mark data can be extracted and used to master future content so that it cannot be played or decrypted using the same compromised or misused devices. Copies produced by combining multiple outputs can even be traced (see Figure 13).[18] This detection and revocation capability forces pirates to put their equipment at risk and can provide evidence for prosecution. Finally, because people are more likely to misbehave in situations where they feel anonymous, simply making users aware that

[18] For a detailed analysis, see: Boneh, D., and Shaw, J., "Collusion-Secure Fingerprinting for Digital Data", IEEE Transactions on Information Theory, Vol 44, No. 5, 1998, pp. 1897-1905.

copies are traceable is expected to reduce piracy. At the same time, forensic marks avoid the privacy concerns associated with other data collection approaches because no information is revealed about users who do not redistribute copies.

> "Absolute anonymity breeds absolute irresponsibility."
> — Scott McNealy, Chairman & CEO, Sun Microsystems

General information gathered from forensic marks can also help publishers make appropriate risk management decisions. For example, if piracy using a particular software decoder becomes widespread, a content owner might prevent it from decoding future content at high resolution until users install a security upgrade.

10. REVIEW OF DESIGN OBJECTIVES AND REQUIREMENTS

Figure 4 in Section 3 lists major requirements and objectives for content protection systems. This section reviews these issues and the feasibility of addressing them using self-protecting content with forensic marks.

- **Renewability** – Security must be reestablished after individual devices are compromised or flaws are found in product designs.

  No limitations are imposed on number of compromises or attacks that can be survived. Many compromises can be repaired using code updates. Unaffected products are not impacted.

If k out of N decoders collude to try to remove a forensic mark, there are $\binom{N}{k}$ possible sets of colluders. A set of colluders can be excluded if no set members could decode the observed version of a polymorph. If each version of each polymorph can be decrypted by an independent random 50% of decoders, each polymorph in the output excludes $(½)^k = 2^{-k}$ of the collusion sets. If a total of p polymorphs are present, the expected number of non-excluded collusion sets is $\left(1-2^{-k}\right)^p\binom{N}{k}$.

For example, a 90-minute movie at 30 frames/second has 162,000 frames. For 1% of the frames (p=1620), two polymorphs are included. Even if an adversary produces a pirate copy by combining outputs from 4 decoders (k=4) chosen from a population of 1 billion decoders (N=10^9), the content owner can identify all of the compromised devices with probability >99.9999999%, since the expected number of ambiguous collusion sets is:

$$\left(1-2^{-4}\right)^{1620}\binom{10^9}{4} < \left(\tfrac{15}{16}\right)^{1620}\left(10^9\right)^4 = e^{1620(\ln 15-\ln 16)+36\ln(10)} < e^{-21.6} < 4\times10^{-10}.$$

Figure 13: Simple traitor tracing (collusion detection) example.

- **Playability –** All valid players must be able to play all valid content, subject to security policies.

  Operation is fully configurable by publisher, but security would normally be hidden and automatic. Flexibility allows publishers to block unauthorized actions while minimizing any impact on legitimate users.

- **End-to-End Security** – Content should be protected through the entire distribution and playback process.

  Security code can validate all information available during the playback sequence, including decoder types, media types, software device drivers, devices connected to digital outputs, etc. Forensic marks deter copying from analog and other outputs.

- **Cost** – Cost should be minimized.

  Modest impact on player complexity; manufacturing cost today should be less than costs for CSS when DVD was introduced. Effort to develop/procure security code would increase content mastering costs. Fixed costs include administration, technology licensing, player engineering, and standards development.

- **Openness** – Because implementations will eventually be reverse engineered, security must not rely on the secrecy of the system's design.

  All system design documents could be made public; only players' production keys need to be secret.

- **Player Diversity** – Security must be provided across a broad range of decoding devices.

  Support for all player types is practical, including those that are software-based, portable, and off-line. Future player types and security features can be supported in future content. Because publishers/artists can decide where their content will be played, content code can range widely in features and security policies.

- **Migration Path** – Transitions from one format to another should be as smooth as possible.

  To support migration from insecure designs, players can support both legacy and self-protecting content formats. Legacy standards can be implemented in updateable code running on the player's interpreter. Upgrades and transitions from programmable formats can be done by adding appropriate code to content.

- **Assurance** – System-level designs must provide high assurance of security, while assuming that individual implementations may be insecure.[19]

  System design assurance is only limited by the standards process, quality of documentation, and third party

[19] Security products are uniquely difficult to evaluate because security flaws are invisible during normal operation and vendor claims are notoriously unreliable. Careful due diligence of all security claims (including our own) is strongly encouraged.

evaluations. Cryptographic components and forensic marking can be provably secure. Security flaws in content code do not affect other titles. Player flaws can affect older content, but can be avoided or repaired in new content.

- **Incentives for Security** – Vendors must have tangible market-based incentives to ensure security, even after a format has been adopted.

  Programmable designs give manufacturers an ongoing incentive to invest in security, since publishers will trust products with better security with their most compelling, highest-quality, and newest content.

- **Forensic Reporting** – It should be possible to identify the specific devices and methods used by pirates.

  Forensic marks allow content to embed arbitrary information about the decoding process in the output. Publishers can recover this data from even a degraded analog copy and use it to revoke pirates' equipment, improve the security of new content, and prosecute pirates.

In addition to these design issues above, some attacks cannot be prevented completely by any player or media technology. Although these will always remain sources of piracy, risk management approaches can provide useful responses:

- **Media cloning –** No technology can distinguish between original media and a perfect copy.

  Although players can detect user-recordable media and reject media with revoked IDs, law enforcement efforts will be required to stop professional pirates who obtain access to equipment for making exact copies to non-consumer-recordable media. (Proprietary media features may help in the short term, but will eventually be reverse engineered or circumvented by professional pirates.)

- **Analog Recording –** No technology can eliminate recording from analog or unprotected outputs.

  Although general-purpose recording devices will always be able to record from analog outputs, forensic marks can trace copies back to specific devices, which can then be revoked.

- **File Sharing –** No technology can eliminate copying of content that has had its protection removed.

  Once content has been converted to a format that lacks security features, it can be redistributed, e.g. via computer networks. Although player security features and forensic marking may help deter this piracy or trace its source, we do not suggest that improvements in player security alone will solve the problem of piracy over Internet file sharing networks.

11. CONCLUSIONS

It is impossible to predict the specific attacks and threats that anti-piracy systems will face. Conventional static security approaches are ineffective because they lack the flexibility required to respond to unexpected problems. In contrast, programmable systems eliminate the need to anticipate all future threats by separating critical security design choices from the media format and player design. When failures occur, as we expect they inevitably will, publishers can mitigate their risk by revising security systems and policies without losing compatibility with the installed base of players.

> "Failure is only the opportunity to begin again more intelligently."
> — Henry Ford

Programmable systems can adapt and evolve as technical advances yield new threats and opportunities. This provides content owners with the ability to respond and to recover from attacks that would have otherwise been catastrophic. The intended result is a chess game of pirate attacks and publisher countermeasures. Newer content will benefit from newer security measures, while older content is more likely to be pirated. Piracy will not be eliminated, but programmatic responses such as forensic marking, equipment revocation, and code upgrades can provide an ongoing deterrent by increasing the risk, cost, and effort of piracy.

Publishers have been effective at lobbying, but have not presented a long-term technical strategy. While new anti-piracy systems could be far more effective than any in use today, investments in security have been inadequate relative to the major economic threat posed by piracy.

Efforts to improve security will require strong technical leadership. Otherwise, standards efforts will tend to degenerate into unwieldy and ineffective committees with short-term focus. Leadership is also needed to prevent ineffective proposals from wasting time and momentum, to verify that security needs are met before products ship, and to help secure designs succeed in the marketplace. We conclude that only rights holders can provide this leadership; no other participants have the motivation, expertise, or resources to ensure the deployment of effective anti-piracy technologies.

* * *