SECURING DIGITAL CONTENT – STRENGTHS AND WEAKNESSES OF SOFTWARE AND HARDWARE IMPLEMENTATIONS

Robin Wilson
Nagravision

Abstract

Conditional Access (CA) and Digital Rights Management (DRM) are implemented in a number of ways in software (SW) and hardware (HW). Often these schemes are described as either "HW" or "SW" based security or rights management systems. Since SW requires HW on which to execute, and HW has necessarily SW running on it, the terminology is often thoroughly confusing and misleading. Since we are not dealing with locks and keys or hypothetical systems, both hardware and software elements must be present, and work together smoothly, in any sophisticated content security system.

Further confounding the confusion is the frequent use (and abuse) of terms like "replaceable", "renewable", "obfuscated" and "tamperproof". In this session, these terms will be explained in the context of content security.

TERMINOLOGY AND TECHNOBABLE

Before embarking on comparing various security implementations and trade-offs, we will first define the terminology used. This step will help to overcome the ambiguous definitions and terminologies recently applied under the guise of marketing new security concepts.

HARDWARE VS. SOFTWARE SECURITY

"Overview"

The security system relies on a computation, or algorithm, to decode the protected content.

Most digital CA systems employ a unique key that enables a successful computation. The locations on the Set-Top Box (STB) for the decoding program and key are a the subject o the hardware and software security designations.

"Software Only" Security

This and other terms like "Hardware-less", "Downloadable" and even "Renewable" are used to describe security systems where the security solution supplied by a CA or DRM company does not include hardware. Here the product of the company may be limited to only software but the inference is sometimes wrongly made that no additional resources are required or costs are incurred.

Software Needs Hardware Too

Obvious, but software still requires hardware for execution. Conventionally this hardware is referred to as the CPU. In a security system, hardware security is a significant concern.

"Hardware" Security

As we are not discussing security using tumbler locks or brass keys, it should be of no surprise that the hardware referred to here is electronic in nature and it runs software! In many instances, the decoding algorithm is programmed into the "hardware" device.

Hardware is free?

A further misrepresentation is sometimes made that a so called "SW only" systems have zero hardware cost. While it may be true that there is no hardware cost in the STB to be passed on by the CA / DRM company, someone, usually the operator, pays.

The assertion that CPU cycles are incrementally free is also misleading. Any CPU on any STB is almost always maxed out. Indeed prior to launch of a new STB or service it is unusual to find less than 110% of CPU resources are already assigned. There is a significant opportunity cost and a real cost to freeing up cycles for security applications. Even if the security application takes only 20% of an existing CPU that can become several dollars in additional cost.

As the computational power of the STB increases and the sophistication of the CA system is enhanced to meet ever-growing threats, there is the risk that legacy STB's will be unable to support the CA system necessary to protect content.

To Summarize:

- Security software runs on hardware
- Security hardware runs software
- Hardware is always required
- Hardware can be secure or insecure
- You pay for security hardware even in a "SW" only system

Hardware can be secure or not. Likewise, security software can either be written without concern for potential reverse engineering or cleverly concealed attacks within the hardware/system. Efficient implementations are often a hybrid mix of several SW and HW security techniques.

SOFTWARE IN A SECURITY SYSTEM

Bug Free (and we really mean free)

A Bug in any software jeopardizes your hard-earned customer relationship and potentially your corporate reputation. While a bug in a security system can be catastrophic and possibly threaten your business. Even a single bug can quickly give hackers many more clues as to how a system operates compared with months or even years of analyzing a bug free system. This is further compounded in a security system where a bug can be classified as any unplanned operation regardless of the stimulus. For this reason the security of SW has much more stringent requirements including the need to do nothing when stimulated by any of an infinite array of malicious or accidental stimuli. Try requiring that of you web browser!

Beyond the impact on the CA system, any unplanned and unexpected operations will impact the viewer. This is likely to have a negative effect on the viewing experience.

Small is Good

For the reason above and for speed of operation, bug free security software is best written in very small kernels by very small teams with exhaustive regression testing.

Throwing tens or hundreds of staff at the problem will not help. Neither will bloating the code-base with huge code footprint. It is

in this area that a few very experienced programmers can easily out-perform the huge corporations who's bug ethics are driven by "good enough to ship" or "let the end customers find the bugs". It is easy to see that a large, complex, system presents many more opportunities for error and bugs.

Isolation

It is counter-productive to implement tight highly-secure bug-free security kernel only to have other applications sharing CPU or memory resources (planned or unplanned). That strategy will totally undermine security. Security will be at the mercy of the application suite du-jour. In-turn the QA issues of requiring that all the applications sharing the same code and memory space are totally bug free make this flawed implementation unworkable. Security software needs to be isolated in a protected environment isolated from a hostile memory and bugs. Denial of visibility and accessibility of the CA process is essential to protecting its secrets.

SW Obfuscation

This means that the operation or structure of the software is deliberately made non-obvious or non-intuitive to either human or machine. Although this term has recently gained some use in referring to automated transformations applied to pure software products like games, the technique has been in use for more than a decade in CA or DRM systems.

SW Tamper-proofing

Here the goal is to detect any abnormal operation in the SW due do any unwelcome external stimulus. When detected, the tampering will almost always result in either a temporary or permanent halt to the security processes.

HARDWARE IN A SECURITY SYSTEM

Having bug free security SW is useless if the operating states, registers etc, can easily be monitored. While it is well beyond the scope of this paper to discuss the security philosophies relating to hiding and keeping secrets, having a transparent hardware platform like a generic CPU, where the operation and architecture are well understood, fatally undermines almost any security scheme.

HW Obfuscation

Just as with SW, in the context of HW the structure of silicon or functional blocks are deliberately made non-obvious or non-intuitive to either human or machine. The term camouflage may also be used. Here the silicon structure is laid out in an apparently identical manner for many of the building blocks and the critical differences are hidden deep inside an obscure silicon structure.

HW Tamper-proofing

Here the goal is to detect any abnormal probing of the silicon or functional block. Numerous techniques are employed from detection layers to radiation detectors to produce electrical anomalies.

One common measure employs fuseable links that can be burned away or destroyed after the CA programming is loaded into memory. This makes reading that code and analyzing the memory structure far more difficult.

As with SW tamper proofing, when an attempted intrusion is detected, the tampering will almost always result in either a

temporary or permanent halt to the security processes.

RENEWABILITY – WHAT DOES IT REALLY MEAN?

Downloadable

The downloadable feature within the security system authenticates or identifies a network element i.e. STB, securely communicates a downloadable solution, and launches the solution into a secured environment.

Replaceable

This has two possible meanings:

1). Electronic - The replaceable feature within the security system revokes the current security solution, restores the secure environment, and securely enables the downloadable feature for the replacement security solution.

2). Physical – Here a physical device may be replaced. Replacement is based on proper authentication, binding/paring and secure provisioning. Note: this does not always infer that the removal of the previous device. Physically replaceable hardware cuts both ways. It permits total replacement of a compromised CA system but it also permits cloning of apparently legitimate hardware solutions to receive unauthorized service.

Renewable

The renewable feature within the security system suspends the current executing version of the security solution, maintains the secure environment, and securely enables the downloadable feature the a new or upgraded version of the security solution.

Countermeasures

The countermeasures feature within the security system allows for secure and validated updates to the current executing version of the security solution. It also assists the service provider in detecting and disabling compromised platforms.

PERFORMANCE CONSIDERATIONS

Latency

Providing an easy-to-use viewing experience is critical to keeping customers happy and giving them no reason to look at other methods of content delivery into the home. Although not often thought as a factor in subscriber retention or churn it is important to ensure that viewers are never annoyed by additional channel selection delays. In the new competitive video environment, channel change delays will become a differentiating factor for service providers.

Latency in a CA system can be broadly categorized as two issues:

1). The first is the time taken between a subscriber's request to view a channel or view a stored file and the proper permission communicated to the security process. This could be summarized as "checking the viewing rights". In any modestly sized broadcast system, there is not enough bandwidth available to broadcast all the rights for each viewer frequently enough to avoid annoying delays of seconds if not minutes. The CA system must provide specialized configurations, storage of permissions, and communication capabilities for the timely delivery of each customer's permissions.

A system without these specialized configurations and communication capabilities relying on a two-way out of band (OOB) network such as one with a dedicated DOCISIS / DSG return path, encounters processing and round trip delays such that the system cannot scale into tens of thousands or more subscribers and guarantee to operate with the required quality of service. In addition such an architecture would have to bear the cost of the BW and support costs for the burdensome continuous OOB two way traffic. The same system limitation applies to a pure IP network. Pure IP networks are often limited in the bandwidth dedicate to an individual STB limiting the number of communications carriers available to communicate with the STB

In order to achieve satisfactory low latencies it therefore becomes necessary to selectively stream and cache the subscriber rights.

2). The second relates to the real time cryptographic time base(s) used. When a subscriber requests to view a new stream. It is considered desirable that the channel change delay from security system is well under one second it is considered desirable to be under 100mS. A figure of 200 mS is generally accepted as the delay threshold that causes irritation.

Latency in a CA system is a complex subject with operational considerations including the likelihood of network outages, installation immediacies, warehouse support etc, but it is important that the base-line operation of any large deployment is non-immediate making available resources so that installations and customer support can be given high priority.

Rights Management Matrix

This is the operational heart (as opposed to the security heart) of a CA or DRM system. It is the complex alignment and communication of the various rights as mapped onto the marketing driven needs including packaging, floating previews, Push VOD rights etc. Much of this functionality must execute on millisecond boundaries, uniquely control individual subscribers in multimillion subscriber systems yet require very low communication bandwidth and minimum latency.

It is this complex functionality, tightly linked to cryptography that is often overlooked and misunderstood, particularly in the area of new and emerging IP network where it is naively thought the routers, CMTS', DSLAM's or other edge devices can execute this function.

CA HARDWARE IMPLEMENTATION EXAMPLES

Secure Microprocessor

A secure microprocessor is a specialized CPU with numerous enhancements. The principal enhancement is a hardened hardware and software environment to safeguard against security attacks. Hardening features can include intrusion detection, camouflage cell structures, encrypted communication, and cryptographically secured memory with randomized page, address, and value construction. Today's modern secure micros can include many security specific enhancements including true random number generators, public key generation, and several security algorithm accelerators. A secure micro is an optimized platform for implementing security solutions. A secure microprocessor is typically used in a BGA,

SIM, Smartcard, MCM, USB Key and potentially CableCARD (see below).

BGA

A BGA (Ball Grid Array) is a popular type of physical IC package that provides high I/O density, small footprint, and physical security features. Historically, IC I/O has been through pins on the perimeter of the IC package, the BGA provided I/O through an array of solder ball connections on the underneath side of the IC package. Since the I/O connections are physically sandwiched between the IC and PC board, this maximizes the I/O connections, minimizes the footprint, and physically restricts access. Secure microprocessors are often implemented in a BGA package

SIM

A SIM (Subscriber Identity Module) is the security module predominantly used in GSM mobile phones. For content security applications, a SIM can be considered to be identical to a smartcard in functionality, the main difference being a smaller physical size and different insertion requirements compared to a smartcard. Because a SIM is visibly smaller than a smartcard is sometime assumed that it must be less expensive. In fact the production process is nearly identical for both, and in some instances additional steps are required to punch the SIM form factor out of a larger smartcard carrier.

Smartcard

A Smart Card (SC) is a credit card size security card containing a secure microprocessor. A SC has a wide variety of applications ranging from phone cards, digital identification devices, and standards-based satellite / cable renewable security. There are two SC interface types: *Contact and Contactless*. The contact SC uses ISO-7816 standards pin connections to communicate via direct electrical contact. A contactless SC does not have contact pins but communicates via radio frequency (RF) using an embedded wire loop.

TV pass

A TvPass card is a proprietary security card from Motorola / General Instrument system providing a renewable security solution.

Proprietary solutions face cost challenges presented by non-standard manufacturing requirements and hardware with limited volume.

CableCARD

The CableCARD™ is a more substantial device, similar to a PCMCIA Type II- card designed for laptops, that slides into a slot on many newer high-end or high-definition television receivers and DVR's. The CableCARD™ eliminates the need for a cable STB (Set-Top Box) at least for the decoding function. The CableCARD™ contains a secure microprocessor, one or two-way data transceivers, and specialized circuitry to process security information and decrypt the digital content. It is, essentially, the entire CA system, hardware and software, on a removable device.

The current CableCARD being deployed has only one-way functionality. The standard for a two-way multi-stream CableCARD is in development.

MCM

A MCM (Multi-Chip Module) is structure consisting of two or more integrated circuits

interconnected within the same IC package. An MCM allows for high-density implementations with security provided by semiconductor-level integration. A typical configuration might be an audio-video decoder and descrambler packaged with a secure microprocessor. MCM technology attempts to keep critical interfaces within the chip structure away from attempts to compromise the security system

SOC

A SOC (System On Chip) is a general class of solutions allowing for high integration of many of the major subsystems within a digital STB (Set-Top Box) and likely including the security solution and the secure microprocessor. Today no realizable implementations are in current use.

General Purpose CPU

A general purpose CPU mentioned because of the compelling quantity of consumer PCs. Most if not all communications to and from the CPU are in the open and can be easily accessed. Monitoring, debug tools and expertise are widely available. Not an option to secure in a security system of high value content.

USB Key

The USB security key is used to combine a digital identity and security functions into an integrated security device. The USB security solution spans the following features: secure digital ID passwords and digital credentials securely stored on the key and automatically presented to applications as required, authentication for third party verification including multiple factor authentication utilizes a variety of authentication methods including biometrics, one-time-passwords, digital certificates, and traditional PINs and passwords. Again a secure microprocessor can be used.

CONCLUSION

Implementing a complete security system in a STB is always a complex blend of many hardware and software techniques. Securing the STB, while critical is only one aspect of an overall security solution perhaps just the most visible “tip of the iceberg”. Developing entirely bug free SW and systems is critical for security. Throwing huge resources or groups at the problem is a recipe for interminable security compromises. Likewise utopian “SW only” systems that claim near free functionality and perfect replaceability are the technological equivalent of diet pills.

Beyond the scope of this paper, there are many additional security issues in the supply, fulfillment, and support chains that have equally challenging solutions. Remember a security system is only as good as its weakest link and cryptography is only a small but important part of a security solution.