HARDWARE-BASED AND SOFTWARE-BASED SECURITY IN DIGITAL RIGHTS MANAGEMENT SOLUTIONS

Maria Nickolova, Eugene Nickolov

Abstract: *The main requirements to DRM platforms implementing effective user experience and strong security measures to prevent unauthorized use of content are discussed. Comparison of hardware-based and software-based platforms is made showing the general inherent advantages of hardware DRM solutions. Analysis and evaluation of the main flaws of hardware platforms are conducted, pointing out the possibilities to overcome them. The overview of the existing concepts for practical realization of hardware DRM protection reveals their advantages and disadvantages and the increasing demand for creation of multi-core architecture, which could assure an effective DRM protection without decreasing the user's freedom and importing risks for end system security.*

Keywords: Security, DRM protection.

ACM Classification Keywords: D.4.6 Security and Protection.

Introduction

Security design is one of the most challenging areas for system designers because it requires an extraordinary effort to build a system offering strong security features but not hindering the working process of users and being well accepted by them. This is particularly true as far as the compromise between the content owner's copyrights and the right of free access and exchange of information is concerned. The solution adopted in last decade is the digital rights management. Although most users don't agree with the use of DRM, it is of critical importance for authors, publishers and content providers - their business depends on the ability to control and to charge for access to their content.

Although the inherent insecurity of Internet, many upper-layer security protocols can be used to protect data during transmission but content is still at risk when it arrives at its destination. If the end device's boot process and critical information are not highly secure, the digital content can be stolen after the transmission and distributed without permission. This implies that end user devices must be built on a trusted platform and equipped with mechanisms for cryptographically validating the hardware environment and code signatures of downloaded software [1].

The DRM technologies allowing the protection of the content by access from unauthorized users could be divided into three groups: DRM implemented completely by software, DRM implemented completely by hardware, and the hybrid combinations of software and hardware. Certainly the most secure DRM is that which is implemented by hardware, the next most secure is the hybrid, and the least secure is via software.

Main requirements to DRM platforms

An effective DRM technology must provide a smooth and effective user experience for content use and in the same time must implement strong security measures to prevent unauthorized use of content [2]. The main requirements to it are:

1. It must ensure fully protected capabilities, which means the protection functions should be performed as part of the boot process. Otherwise during boot-up malicious software can easily hook the control functions and compromise system integrity. If end devices receive content over a network, such malicious software could be masked as a firmware upgrade or Trojan, or hidden using rootkits.

2. It must allow trusted integrity measurement and confirmation, that means the platform should own the capability to automatically check in real time during the boot all the new software and executable files in the system (certificates, digital signatures). Once this confirmation is done, the operating system loader can be started and the boot process proceeds as normal.

3. It must provide integrity reporting to notify the user about the results of the integrity measurement and possibly to prevent the user from playing back the DRM protected content in case of negative results from the integrity check.

Obviously these requirements could be implemented by hardware and/or software means.

Advantages of hardware-based DRM versus software-based

The analysis of the commercially available technologies for DRM protection shows two main reasons to use hardware-based security of the protected content: better overall robustness and improved user experience. The main benefits of the hardware-based security robustness are:

- Immunity from the inherent vulnerabilities and security holes of the used operating system. The security of all software applications is limited by the level of security provided by the underlying OS. Although the open and rich OS have bigger security challenges than a closed OS a hardware security module is an essential element to make the OS trustworthy.
- Impossibility to access, change or uninstall security features. Attacks to DRM protection often start by targeting the protection software - trying to uninstall it or stop its activity [3]. Obviously hardware-based DRM protection cannot be uninstalled as it is hard coded into the chips.
- Protected memory. Hardware-based DRM solutions manage the memory in a restricted manner and are able to prohibit access to it, providing better protection against attacks on the security mechanism. Software solutions use memory by the services of the operating system and several processes can access the same memory space simultaneously. Most OS provide some memory protection, but the safety of the memory space depends on the extent to which the operating system is robust and free of flaws. This is particularly important for the cryptographic algorithms which require the storage of the intermediate results during the execution of the cryptographic module. If the content of this temporary storage is exposed, the entire DRM system can be easily compromised.
- Better performance. The hardware DRM protection could be optimized for maximum security and operate independently, not degrading the performance of the computer or consuming its resources.
- Prevention of potential software conflicts. The software DRM protection is run on the same computer with many other security programs using together the same processor, memory, OS and other resources. This could provoke various conflicts resulting in poor performance and even in stopping the action of both DRM protection software and security programs.
- Secure Storage. Hardware-based DRM protection is able to better protect sensitive data, such as private keys. A software DRM implementation cannot prevent the exposure of keys and therefore they could be relatively easily compromised. Even very strong cryptographic algorithms could be easily compromised by an direct or indirect attack to their software implementation. Only a proper hardware implementation, to which countermeasures against known attacks are added, could protect the secrecy and the integrity of the DRM mechanism.
- True Random Number Generation. The software DRM technologies use pseudo-random numbers that decrease the security level of the DRM protection. As random numbers are used in DRM protection process for the creation of temporary and special values and are part of challenge response authentication, the better the random number generator, the more secure the DRM implementation.
- Easier, faster and cheaper attacks to software DRM solutions. This is related to the security vulnerabilities, which are inherent for software modules and to the presence of many hackers who have enough time, knowledge and wish for breaking the relevant protection.
- Quick dissemination. The compromising of software DRM solutions by only one hacker becomes quickly available for general use. The publishing in Internet of correspondent methodology allows it to be used by a lot of end-users before the manufacturer could take measures to remove the vulnerabilities in the protection, and to bring severe damage to operators, content providers and manufacturers.
- Less susceptibility to reverse engineering. Hardware-based platforms are able to apply special measures that hide the data-dependent fluctuations in power consumption while software-based DRM solutions are more vulnerable to attacks based on power analysis.

- Most content applications like music, video and games require efficient and effective user experience which is the key factor for the success of consumer electronic devices and therefore for the acceptance of DRM by users. The main benefits of the hardware-based improvement of user experience are:
- Superior performance in which user experience is prioritized without sacrificing security. Hardware solutions generally accelerate several times cryptographic functions (which are computation-intensive) in comparison with software solutions, making DRM security operations almost invisible for the end-user.
- Optimization of CPU power and memory use. Although the computing power of modern processors increases constantly and should allow relatively fast handling of cryptographic functions, processors are designed mainly for new demanding applications such as video rendering and high quality graphics. Therefore software-based cryptographic operations are able to overload them and to worsen the user experience. There are some cryptographic operations (exchange of protocols with long keys, for example) that affect inadmissibly user experience.
- Improved power consumption and memory use. The use of hardware-based DRM platforms allows the CPU to operate at a lower clock rate, saving power which is particularly important for battery powered mobile devices. Additionally, software-based solutions require more memory (code size needs large buffers) which affect the speed and the quality of other applications.

Disadvantages of hardware-based DRM platforms

- Software modification or creation of new software on a computer with hardware-based DRM technology may require hardware changes that could be slow and expensive.
- The simple replacing of a peripheral device running protected content could cause a hardware-based DRM system to refuse to run software.
- Network cards replacement could make a computer unusable until other necessary hardware modifications are done and passwords are reauthorized. This process may require the cooperation of several vendors.
- If DRM protection is compromised reinstalling is impossible.
- Manufacturers of hardware-based DRM are not able to warrant that DRM agents or their hardware assistants will not cause or help any safety or security failures.
- More difficult implementation of extended usability. Software DRM implementations facilitate the making of the licensed content usable by a user anywhere in his personal network (local hard drive, media center, iPOD, cell phone, home entertainment center or burning to a CD), for the hardware-based it’s more difficult and expensive.
- Higher security in hardware-based DRM solutions means higher costs, less interoperability, longer development cycles and potentially shorter market life.
- Limited flexibility. It’s difficult to make hardware DEM systems open to new uses, new business models, or new rights created by content owners.

It is clear that these flaws could be easily overcame and that only a hardware-based DRM implementation or a hybrid hardware/software solution could address all required security challenges while allowing seamless user experience [4].

Approaches for implementing hardware-based DRM

Two main concepts have been developed by now: trusted system concept and multi-core concept.

Trusted system concept

The Trusted Computing Group (TCG), successor of the Trusted Computing Platform Alliance (TCPA), is an initiative led by AMD, IBM, Intel, Hewlett-Packard, Microsoft, Sony, and Sun Microsystems. Its aim is to develop and promote open, vendor-neutral, industry standard specifications for trusted computing building blocks and software interfaces across multiple platforms [5]. The new principles in the TCG architecture expand the range of entities that are able to use TCG features as a trust basis. These entities could include not only the direct user of the platform and the owner but also some remote entities wishing to interact with this platform. The TCG architecture introduces the mechanism of remote attestation which allows remote third parties to ask a platform for details of its current software state. On the basis of the attestation made, third parties can decide to consider the platform's configuration as trustworthy or not. If correctly implemented, this kind of remote attestation could become an important feature for DRM clients on open platforms as it may help a content provider when he makes a decision about the reliability of the client before the content is actually provided. What makes TC technology especially attractive for implementing DRM is its ability to enforce usage policies. Once their security conditions are violated, TC systems stop working. Since their security conditions are built as a "chain of trust" [6] containing hardware-locked keys and certificates from trusted third parties, they are hard to modify, at least much harder than software-based systems. If a DRM solution relies on a trusted system, it is easy to implement a hard-to-break usage rights management chosen by content owners. TC technology is not necessary or sufficient to implement DRM but it can make implementing DRM easier and cheaper. An example of such a realization is the Intel Wireless Trusted Platform with the Certicom Security Architecture software. In this technology a special trusted platform module is built directly into the processor and provides secure key and password storage and protection. First, a secure boot process authenticates the hardware platform and the security architecture authentication module, then the module runs DRM applications and allows the users to access DRM protected content. The security architecture requires decoding the keys using information stored in secure hardware, to be able to access the content, after what these keys are used to decrypt and use the content, but only on the specific device. The encrypted content is not locked to this device, because another user is not able to use the content without paying to the content provider for having access to the rules for the content use [7].

Multi-core concept

Intel's Hyper-Threading technology allows parallel processing at thread-level on a single-core processor by sharing the processor's resources. In Intel multi-core processor, each thread is processed independently by a separate dedicated processor, which allows full parallel execution at hardware-level and software-level and is very suitable for DRM applications [8].

In 2005 Intel embedded DRM capabilities within its dual-core processor Pentium D and allowed (theoretically) copyright holders to prevent unauthorized use and distribution of DRM protected materials [9]. But some functional problems with the distribution of jobs in the cores and in the chip-set when both cores are enabled caused applications to crash or hang and finally made hardware DRM capabilities unusable for real protection of content. The next stage in the implementation of hardware-based DRM in Intel's products was Lenovo's ThinkPad model, launched in 2006. It uses a combination of fingerprint sensor, trusted platform module chip and special software from Microsoft and Adobe to control access and distribution only of PDF documents.

AMD also planned to incorporate DRM into future GPUs by blocking the access to the frame buffer and allowing access only to certain software from certain vendors but these plans didn't involve AMD multi-core processors because of the complexity of problems in sharing and synchronizing DRM-related actions.

In 2006 IBM announced the technology Secure Blue intended for use in digital media players, electronic organizers, mobile phones, computers and devices where data is encrypted and decrypted as it runs through the processor and maintained encrypted in the device's RAM. Secure Blue requires a few circuits to be added to any processor design in order to enforce strict access controls at the hardware level.

Conclusion

It is obvious that DRM is becoming an integrated part of any copyright protected intellectual product in digital form and therefore DRM protection should be implemented in hardware and/or software assuring highest stability and performance as well as the best copyright protection possible. Different adopted solutions have many advantages and disadvantages but clearly show that it is impossible to realize well working solutions based only on software tools. Hardware-based platforms, especially those using multi-core processors demonstrate really promising features by improving user experience along with the robustness of DRM protection. Technology from a hardware standpoint is already in place, thanks to the efforts of various chipset manufacturers who have driven an evolution to support the benefits of parallel processing. Now research must be conducted to develop suitable multi-CPU architectures and multithreaded software that will guarantee the building of the perfect DRM system – fast, flawless and cheap - that can be neither broken nor avoided.

Bibliography

[1] Biddle, P., England, P., Peinado, M. and Willman, B. (2003). The Darknet and the future of content protection. In Digital Rights Management-Technological, Economic, Legal and Political Aspects.LNCS 2770, Springer.

[2] CEN/ISSS, (2003). Digital Rights Management Report http://europa.eu.int/comm/enterprise/ict/policy/doc/drm.pdf.

]3] M. Nickolova, E. Nickolov, Verification and Application of Conceptual Model and Security Requirements on Practical DRM Systems in E-Learning. In: First International Workshop “Cyber Security” - CS 2006.

[4] Hauser, T. and Wenz, C. (2003): DRM Under Attack: Weaknesses in Existing Systems. In Digital Rights Management-Technological, Economic, Legal and Political Aspects. LNCS 2770, Springer.

[5] Peinado, M. Chen, Y. et al. (2004), NGSCB: A Trusted Open System. In Proceedings of 9th Australasian Conference on Information Security and Privacy ACISP, Sydney, Australia, July 13-15.

[6] Smith, S.W. (2005): Trusted computing platforms: Design and applications. Berlin, Heidelberg, New York: Springer.

[7] Dornan, A. (2006): Yes, trusted computing is used for DRM; Information Week, 17 February 2006.

[8] Rump, N. (2003): Digital rights management: Technological aspects. In: Becker et al. (2003).

[9] Pakman, D. (2005): Why DRM everything? A sensible approach to satisfying customers and selling more music in the digital age; Groklaw, 31 December 2005.

Authors' Information

Maria Nickolova – National Laboratory of Computer Virology, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; e-mail: maria@nlcv.bas.bg.

Eugene Nickolov - – National Laboratory of Computer Virology, BAS, Acad.G.Bonthev St., bl.8, Sofia-1113, Bulgaria; e-mail: eugene@nlcv.bas.bg.

MANAGEMENT OF INFORMATION ON PROGRAM FLOW ANALYSIS

Margarita Knyazeva, Dmitry Volkov

***Abstract**: The article proposes the model of management of information about program flow analysis for conducting computer experiments with program transformations. It considers the architecture and context of the flow analysis subsystem within the framework of Specialized Knowledge Bank on Program Transformations and describes the language for presenting flow analysis methods in the knowledge bank.*

***Keywords**: Knowledge bank; Ontology; Knowledge base; Ontology editor; Database editor; Flow analysis; Editor of flow analysis methods*

***ACM Classification Keywords**: I.2.5 Artificial intelligence: programming languages and software*

Introduction

The impossibility of carrying out computer experiments opportunely constitutes the main problem of program optimization science. Their goal is to determine how often transformations can be applied in real programs, what effect can be achieved, and what strategy is the best to be applied for the specified set of optimizing transformations. At present, optimizing compilers are the only means of conducting such experiments [Bacon, 1994] [GNU, 2007]. However, the period between the moment when a new transformation description is published and the moment when the realization of an optimizing compiler containing this transformation (if such a compiler is being developed) ends is so long that the results of computer experiments with this transformation appear to be out-of-date. Besides, an optimizing compiler usually contains a wide set of transformations and built-in strategy of their application so it is impossible to obtain reliable results of computer experiments related to a particular transformation (not to the whole set) or other strategy.

The absence of tools for conducting experiments results in transformations and transformation application strategies, whose characteristics are not known completely, being included in optimizing compilers. This adversely affects their making. Therefore to create a system for program transformation experiments aimed to