

A Foundation for Secure Mobile DRM Embedded Security

▶ The inherent weakness of software-only solutions is one of the most important factors limiting DRM-based distribution of premium content.

▶ *By Craig Heath & Alexander Klimov*

Symbian family of phones.



Digital Rights Management (DRM) is a compelling reason to enhance security of mobile phones against attacks by the phone holder. DRM systems allow content owners to specify and control the usage policy for their content; such systems are crucial for the entertainment industry and for secure information handling in corporate environments.

Modern Smartphone's have more memory and computing power than the desktop computers of the not-so-distant past — though they have yet to experience the same scale of security problems seen in the desktop world. The lack of security-centered design in the most widely-deployed desktop operating systems has severe consequences: a freshly installed system can be compromised

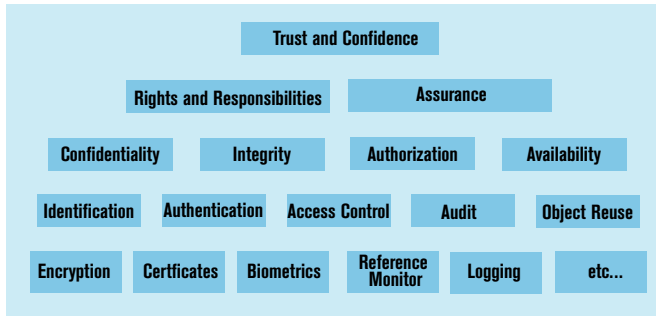
well before the user has time to download the latest security patches. As a

result, anti-virus and other malware prevention tools are necessary to safely use such desktop operating systems.

Symbian OS v9, the latest version of an advanced open mobile operating system, is a major enhancement that aims to significantly enhance protection against malicious code and social engineering attacks. In general, only verified applications from known sources are allowed to access protected resources, although the user may also grant some limited privileges. Even if a message containing malware tricks the user into installing that malware, its privileges are restricted so that it is not able to compromise the entire system.

Yet several DRM systems have come into use that do not provide protection against a well-motivated attacker. At least one well-known DRM system is widely used, despite the fact that step-by-step tutorials and special software that removes the DRM protection are readily available from the Internet.

Every widely-used DRM system that relies only on software suffers from the same weakness. Using cryptography, the system can protect content from users who should not be permitted to access the content at all; however, experience shows that it is impractical to build a software-only system that allows a limited number of legal accesses, but effectively prevents subsequent illegal accesses. This inherent weakness of software-only solutions is one of the



Symbian's view of mobile security as a holistic property.

most important factors limiting DRM-based distribution of premium content.

The situation is similar to low-tech newspaper vending machines that consist of a stack of newspapers and a box with a slot for coins. There are some dishonest customers who take the newspaper and do not put anything into the box but, on average, the loss is much smaller than the cost of theft-resistant vending machines. In some countries such low-tech devices are widely deployed to sell cheap goods, but they would never be used to sell expensive ones.

As a newer and better OS solution, Symbian OS v9 enables significant enhancements to software-only DRM systems. Once the operating system is running, it prevents unauthorized code from accessing DRM secrets and thus, malicious users cannot introduce a modified player that ignores the usage policy of the content owner.

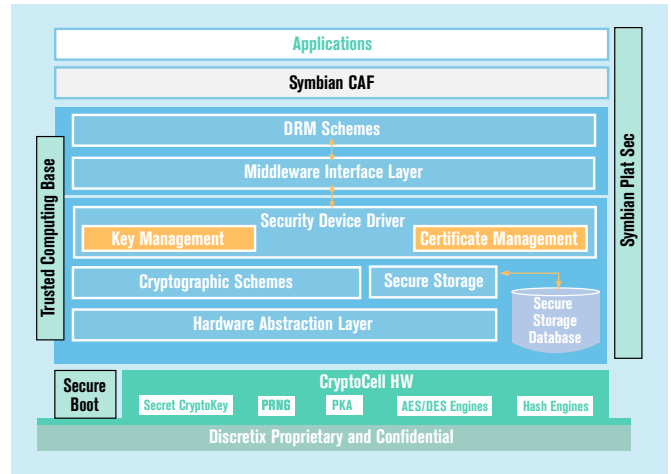
Unfortunately, some attacks still cannot be mitigated by a software-only solution. If an attacker has physical access to the phone's internal memory, then they could read the DRM secrets or even modify software that was supposed to enforce the access policy. It is with this situation in mind that we discuss here how to prevent attacks even from such a sophisticated adversary with a system that integrates Symbian OS v9 with hardware-based security.

Symbian Platform Security1 Overview

Each application executed by Symbian OS v9 may have some capabilities associated with it. To access a protected API, for example, the application must have the capability required by this API. To send an SMS, the application needs Network Services capability; to access DRM-protected content, the application needs DRM capability.

These capabilities are associated with each application during installation of the application. There are three broad classes of capabilities considered by Symbian Signed: the unsigned-sandboxed set includes all the capabilities which may be granted by the user to a non-signed application; the basic set of capabilities is granted to all certified Symbian Signed applications; lastly, selected capabilities from the extended set are granted to Symbian Signed applications if they undergo additional testing appropriate to those additional capabilities.

There are also capabilities which are typically not assigned to applications, but only to system software: TCB (Trusted Computing Base) capability provides unrestricted access to all the hardware and software on the platform and All Files capability provides read-only access to the entire file system and read-



DiscretixDRM Security Architecture for Symbian 9.x.

write access to the private directories of other processes.

The phone's internal file system contains a protected area (/sys/subtree), which is only accessible to applications with TCB capability. During installation of an application, executables, which include their assigned capabilities, are placed into /sys/bin/. It is also possible to install an application onto removable media — in this case, a cryptographic hash of the executable is stored in the internal memory and verified on each load.

The TCB, which is the OS kernel, the file server and the software installer, restricts execution to files from the internal path /sys/bin/ and those on removable media that were not altered since installation. Thus, as long as an attacker cannot alter the content of the /sys/ subtree (and the trusted code contains no critical bugs), no malicious code will be executed. Moreover, each installed application can access only those protected APIs that it was authorized to access.

Hardware-based Security

The security mechanisms of Symbian OS v9 provide strong

protection against software attacks; however, as a software-only solution they cannot protect against physical attacks. If an attacker has the necessary equipment and physical access to the phone, they can simply overwrite the content of internal memory when the TCB is not even loaded.

The purpose of DRM is to enforce the usage policy on the content according to the specification provided by the owner of the content and thus prevent the device user from using the content in an inappropriate manner. If the user performs an offline attack to introduce a malicious application with DRM capability into the system, then they will be able to defeat the DRM protection mechanisms. A software-based SIM-lock, a type of theft-prevention mechanism, is another application which requires resistance to someone who has physical access to the phone. Without such protection, a thief could modify the software so that it does not check if the SIM card is not authorized.

One might think that this simple attack could be counteracted by the TCB using the following strategy: instead of relying on the integrity of the internal storage, the TCB could save the certificate which comes with the software being installed and use it to check the integrity of the software and its associated capabilities before each execution. Unfortunately, this additional safeguard does not really enhance the security because the attacker can simply overwrite the part of the system which checks integrity of executables with the one which always says: "the signature is correct."

Using a hardware-based secure boot mechanism, however, it is possible to neutralize such "reflashing" attacks. Let us consider how this

works in detail. The secure boot loader stored in ROM has control of the CPU immediately after the reset of the device. The ROM also contains a public key which is used to verify signatures of all the binary code of the TCB. Before loading the TCB, the secure boot loader calculates cryptographic hashes of all the TCB components and checks their signatures. Therefore, if an attacker modifies any part of the TCB while the system is offline, they will not be able to boot the system. Note that the work done by the secure boot-loader does not add a significant delay into the boot sequence.

Secure boot protects the integrity of the software executed by the device, but it does not protect the confidentiality of the information in the secondary storage. Even if we encrypt all the information, we still need a location to store the encryption key. Hardware-based secure storage provides such a service, ensuring that the adversary cannot access (read or modify) protected information.

Recalling that the adversary cannot change the operating system because it is protected by the secure boot, let us consider how the secure storage works. Suppose that secure storage hardware receives requests from the operating system to store item X. The hardware uses an internal secret key to encrypt X, to calculate the Message Authentication Code (MAC) of X, and to calculate the MAC of the whole database. This information is then used to update secondary storage devoted to the secure area.

When the operating system requests a specific item, it is decrypted and returned; however, before any item is read, the secure storage checks that the database is

not corrupted. Thus an attacker cannot decrypt X because the key is stored inside secure hardware and cannot be extracted; cannot replace X with new data because it would corrupt the MAC; and cannot even replace X with one of its previous values because it would corrupt the MAC on the database as a whole.

If Non-Volatile Memory (NVM) is available, the hardware uses it to store the update counter: each time the database is updated, the counter is incremented and the new value is included into the database MAC calculation. Thus even if an attacker has saved the whole database and tries to revert it to the saved state, the NVM counter will be different and thus the database MAC verification will fail. This prevents an attack, for example, where a DRM system uses play counts to enforce the policy that some particular content shall not be rendered more than a predefined number of times, and an attacker tries to reset the play counters.

Conclusion

Although the platform security architecture of Symbian OS v9 provides protection against software attacks by malicious code, a hardware-based solution is also needed to protect against sophisticated attacks by someone in physical possession of the phone. Symbian OS v9 platform security and secure hardware together provide a firm foundation for development of DRM systems which are highly secure even against sophisticated adversaries.

About the Authors

Craig Heath is product manager, security and Alexander Klimov is Cryptography Architect for Discretix.