A virtual programmable trusted platform

Talha Tariq proposes an architecture for a TPM–SmartCard co-operative model that will allow developers to focus solely on the functionality and security of their own code.

Royal Holloway University of London
SearchSecurity.co.UK

DELIVERING THE PROMISE of Trusted Computing has been delayed by a number of problems. These include the relative unavailability of mainstream operating systems and hypervisors[1] with useful security properties, the difficulties in balancing the high levels of security provided by the TPM and ease of management, and issues with using the TPM to enhance existing security applications and scenarios. Although we believe that the various industry initiatives taken by the TCG and CPU manufacturers for hardware based platform security are a step in the right direction, the problem of secure isolated code execution and TCB minimization remains unsolved. This project discusses a new architecture for trusted computing in which an existing fixed-function Trusted Platform Module (TPM) is coupled with user application code running on a programmable smartcard. Hence, rather than proposing recommendations for hardware changes or building isolated execution environments inside a TPM, we use a platform that provides related, yet different services for secure/trusted execution, and couple this with the TPM. Although newer hardware platforms such as those incorporating Intel Trusted Execution Technology add support for virtualization and secure interfacing with the TPM, our solution assumes a highly untrusted environment and works on general purpose commodity hardware. Implementing a solution like this allows application developers to focus exclusively on the functionality and security of just their own code. This enables them to execute their applications in isolation from numerous potential shortcomings and vulnerabilities that exist in the form of both hardware and software attacks. Furthermore we provide an interface to extend the existing functionality of the TPM by implementing special purpose code modules inside a smartcard.

TRUSTED PLATFORMS

Trusted Platform Modules (TPMs) are secure cryptographic processors built into many computing platforms. When combined with Core- and Dynamic-Root-of-Trust-Measurement facilities (CRTM and DRTM) for reporting platform state, the TPM provides the basis for a secure and attestable execution environment for system

software and applications. Some of the most common TPM services that are used in this process include:

- **Attestation:** Reliable cryptographic reporting of the platform state to a remote challenger.

- **Sealing:** Protected storage/encryption of data that ensures release/decryption only to authorized software when it is in a particular configuration and state.

THE PROBLEM

The TPM bootstraps a rich and powerful trusted environment running on the main CPU from the small set of functions that it provides. However, even though the TPM provides many cryptographic capabilities and tamper resistance, it is not meant to perform general purpose program execution. The current mass market operating systems, hypervisors and general purpose applications only use the TPM services for platform integrity measurements, code measurements and data protection. The applications still run on the mainstream processor executing all code and the secrets held in primary or secondary memory storage. For many security sensitive applications this normal code execution environment (main-CPU, memory etc) is much less secure (to both hardware and software attacks) than that offered by the TPM, so in gaining flexibility much security is lost. This problem is evident from some of the recent attacks on applications utilizing TPMs such as Bitlocker. The problems of software robustness are even more challenging: mainstream operating systems have an ill-defined Trusted Computing Base (TCB) that is generally not secure enough for attestation to be meaningful.

An alternative to a host platform based secure environment is to build a secure execution environment inside a TPM such as that illustrated in FIGURE 1(A). Such devices have been studied by researchers, but unfortunately they do not yet exist.

A SOLUTION

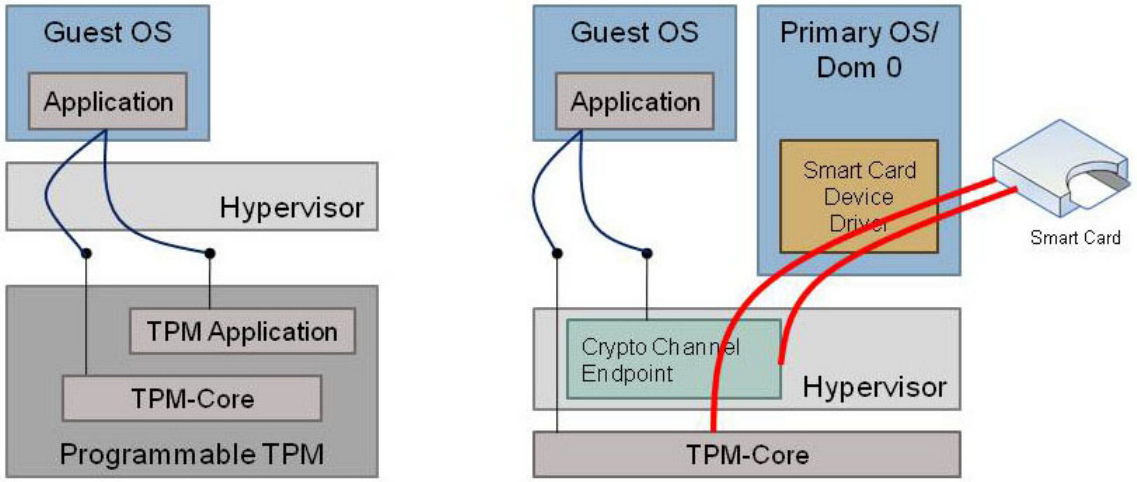
In this project we propose a different architecture. Instead of making expensive changes to the hardware and adding complex functionality to the TPM specifications, which would not only break the existing applications but would also increase the effort and cost of writing new ones, we evaluate the extent to which other platforms provide similar degree of hardware tamper resistance and

secure execution and couple them with a TPM to provide extended TPM services not possible with the current specifications either of TPM or smartcard alone—see FIGURE 1(B). For the purposes of this study we use multi-application programmable smartcards[2] that provide adequate tamper resistance, a programmable environment with application isolation, and crypto blocks for building confidentiality and integrity services. This enables greater levels of protection for information stored, processed and exchanged across different systems.

FIGURE 1

SCHEMATIC ILLUSTRATION OF A PROGRAMMABLE TPM AND ITS USE IN A HYPERVISOR SETTING

We assume that the TPM can load and run applications and the services implemented can be exposed to the hypervisor and guest operating systems. (b) Schematic of one instantiation of our coupled TPM smartcard architecture: The TCB and TPM are coupled to the smartcard using cryptographic keys. The cryptographic channels (thick lines) represent authenticated and secure connections from the smartcard to the TPM and the smartcard to the channel endpoints.



A VERY HIGH LEVEL ARCHITECTURE

Security Requirements: We address the lack of physical binding of the smartcard to the platform and the poor security of smartcard/TCB communications using cryptographic techniques. We seek to approximate an architecture in which the TPM contains a secure execution environment for user-extensible application programs. However in the proposed architecture the secure execution environment is external to the TPM, is independent of the host state, and can be freely roamed between machines. If the smartcard is to provide TPM-enhanced platform services, we need to couple the smartcard and the host platform more tightly using mutual end-point authentication and a secure channel.

In particular we identify the following security requirements:

1. The smartcard applications should be able to determine the host hardware and the host TCB;

2. The host TCB should be able to identify the identity of the smartcard and its applications (e.g. to ensure that confidential data is not communicated to an untrusted smartcard);

3. The applications should have a bidirectional confidential and integrity protected channel between the host platform and the smartcard.

Cryptographic Binding: A trusted authority determines the TPM-to-smartcard binding policy. In the case of an enterprise this might involve an IT department coupling an employee's smartcard with the TPM on her PC (either under conditions of physical security, or remotely given knowledge of keys in the devices to be coupled). In the case of end-user hardware or software manufacturers, this might involve shipping a pre-coupled TPM and smartcard together with an associated plat-

In the proposed architecture the secure execution environment is external to the TPM, independent of the host state, and can be freely roamed between machines with the TPM on them.

form certificate (see the application example of Mobile TPM bound with a SIM in the next section). In mobiles for instance the SIM card can be used to store a identification certificate.

We have implemented a system in which a unique TPM is coupled with a single smartcard, but generalizations are straightforward. We generate, store, and use the following cryptographic keys to identify the smartcard and associated TPM:

- The TPM generates an Attestation Identity Key (AIK) which we use to identify the TPM and the host. The public portion of this key is communicated to the smartcard under conditions of physical security and is stored in smartcard nonvolatile storage.

- The smartcard generates an RSA key-pair which we use to identify the card. The public portion of the key is communicated to the platform TCB under conditions of physical security and is secured in host platform secure storage.

The binding and initialization step need only be performed once. At run time, code in the TCB and in the smartcard builds a secure authenticated channel based on these keys using standard techniques (e.g. using an authenticated key exchange such as that provided by SSL).

Application Communication: The smartcard applications can be preloaded prior to the binding step or, in a more sophisticated version, we could provide a user-accessible smartcard execution environment and services that let the smartcard applications authenticate themselves. This might also involve a custom application loader in the smartcard.

EXTENDED SERVICES WITH TPM AND SMARTCARD COUPLING:

This architecture has many interesting characteristics: First it is a practical way of providing enhanced security functionality for existing TPMs. Second, it provides a way of prototyping new TPM functions to assess their usefulness before committing them to silicon. Finally it allows us to explore the design and assess the usefulness of a true "programmable TPM." We have built several advanced security services to help us understand this architecture and demonstrate its capabilities. These services include:

1. Count-Limited Objects

An implementation of TPM keys whose use is

count-limited. This capability is designed to simplify some aspects of key revocation and support rights-management.

2. Flexible Seal and Unseal
A smartcard implementation of the Seal, Unseal, and Unbind primitives that allow more complex policy expressions than the simple Platform Configuration Register (PCR) equality checks defined by TPM 1.2. One policy expression allows sealing to a software publisher identified by a public key (the publisher may later authorize any PCR configuration using a certificate signed with the associated private key). Another policy expression allows more complex authorized configurations (e.g. PCR configuration 1 or PCR configuration 2). Both of these enhancements are designed to make software updates and grouping of equivalent programs easier to manage.

3. Attestation Translation
A smartcard service that provides attestation using cryptography and signature formats is unavailable within a TPM. A simple proof of concept of attestation translation has been implemented; however a more sophisticated implementation could provide platform attestation in more widely used signature and certificate formats like X.509 in order to simplify the deployment of attestation using existing servers and protocols.

APPLICATIONS AND POTENTIAL USES

1. Roaming DRM (Digital Rights Management)
As described earlier for the Seal and Bind operation, data can be bound to a particular TPM, and the data will only be decrypted if the platform is in the particular expected state. These are powerful functionalities not offered by conventional platforms and we extend them further to provide

A more sophisticated implementation could provide platform attestation in more widely used signature and certificate formats like X.509 in order to simplify the deployment.

more secure and flexible services. We use these services to provide flexible seal or bind operations with count limited objects to provide a roaming digital rights management platform. The smartcard can either be used to encrypt (Bind) data to a particular TPM (or a set of TPMs) by encrypting data for each public key, or can be used to decrypt (unbind) data encrypted by the smartcard. For sealing operations, the smartcard can only seal data for a particular TPM and the TPM would be able to decrypt it. Hence data can be right protected within the secure storage of the smartcard and released only by policies of the trusted platform.

2. Enhanced digital signatures

We use the term enhanced, because this application enhances the conventional digital signature process by combining the best capabilities of smartcards, trusted computing and digital signatures, and solves a number of problems and issues with the digital signature systems present today. Instead of signing just the data/document, the smartcard also forces the platform to generate an attestation on its state and signs the data with it. The signature created on the data not only proves that a particular smartcard signed the data, but also proves that the document was signed on a particular machine and that the machine was in a particular configuration and state that was attested. Second, using a stronger smartcard and TPM coupling and trusted I/O, this method can be further strengthened, eliminating the problems where a person later disputes the digital signature on the basis that he/she didn't 'see' the document he/she was signing.

3. Smartcard SIM bound to a mobile TPM

The telecommunications industry faces numerous challenges in attempting to protect the state of their mobile hardware and software applications. With increasingly competitive and diverse markets, it is becoming very common (and with automated tools, very easy) to tamper with the state and protection of a handset. It has become quite common to see a partnership between an equipment manufacturer and the network operator to provide exclusive services with a new handset release. However, as soon as the security of a new phone is hacked it can cause severe damage both in terms of finance and credibility to both the network operator and the handset manufacturers.

Now consider a mobile phone equipped with a TPM. The network operator who controls the SIM and the equipment manufacturer who can preprogram a TPM can create a TPM/SIM binding

for stronger DRM and enhanced services. In the simplest scenario, a public key certificate could be loaded in the smart card for the TPM to identify it. The TPM's public key could also be stored or updated post issuance in the SIM to create a crypto-channel. Similarly, the TPM of the mobile phone would hold the public key of the SIM and hence the TPM and SIM can 'identify' each other. Hence, even if the handset's security or DRM protection is cracked and a SIM replaced with one from another network operator, the enhanced operator services will not be available since the TPM will not authenticate the SIM and would not disclose its secrets. This can be extended to even disable the phone completely depending upon the choice of application, regulations and other dynamics. This type of coupling can extend services from authenticated boot operations, secure services initialization, or creating specialized 'trusted' virtual machines for specific purposes.

4. E-cash Tokens that only work with authorized platforms

One of the major problems with payment tokens is that there is no trust relationship between the card and the reader. The Roaming DRM application can be simplified to make a smartcard/token only work with an authorized platform. This includes payment tokens, EMV based applications, transportation cards, identity cards etc. This is very simple as we can restrict applications inside the smartcard to talk only to authorized platforms, whose key or key hierarchy is already loaded in the smartcard. The smartcard can also refuse to disclose secrets or even its identity if it cannot verify or build a trust relationship with the TPM.

5. Ease of data migration between trusted platforms

As the smartcard is able to seal/bind data to the platforms it is bound to, and unbind data with migratable keys, we can bind data when exporting from trusted platform A and unbind it using a smartcard on another trusted platform B. This model can be extended to n platforms and an unlimited number of files which can be sealed from one platform to be unsealed on another trusted one.

6. Cryptographic schemes not supported by the TPM

One criticism of the current specifications of the TPM relates to the strict limitations on the cryptographic algorithms and primitives supported. Newer hashing algorithms like SHA-256, SHA-

512, etc., are not available (as of version 1.2 of TPM Specs). Furthermore, use of any other encryption / decryption algorithm, or newer signature formats are not supported too which might be needed by certain institutions or security sensitive applications. Since the smartcard can provide general purpose secure execution, any cryptographic primitive not supported natively by the TPM or the smartcard can be built as an applet/onCard application.

7. Flexible authorization applications
Flexible authorization can be extended beyond any two factor authentication where both the presence of a token and involvement of a platform needs to be assured for a process. The applications can include IPsec/VPN authentication where a user password is provided by the authorization of a smartcard, and keys for encryption can be released from a TPM by a TPM/SC collaborative model, hence giving a remote authentication of both a platform and the user.

CONCLUSION

We have shown that with appropriate coupling the resulting system approximates a "field-programmable TPM." A true field-programmable TPM would provide higher levels of security for functions that would otherwise need to execute in host software. Our coupling architecture supports many (but not all) of the security requirements and applications scenarios that you would expect of programmable TPM, but it has the advantage that it can be deployed using existing technology. Our work demonstrates that logic and cryptographic operations running on a smartcard coupled with the host platform and TPM can mitigate all of these issues, and is also an interesting prototyping environment for experimenting with new functionality that could be incorporated in future TPM designs. This coupling

A true field-programmable TPM would provide higher levels of security for functions that would otherwise need to execute in host software.

architecture strikes a useful balance between flexibility and deployability using today's generally available commodity hardware, but it is interesting to speculate on the design and improved functionality of a future programmable TPM

The applications implemented were chosen to exercise local- and remote-trust verification, and to mitigate some of the problems that we experienced in trying to apply trusted computing to real problems. Other candidate applications included keys with more sophisticated key management and migration functions, a software-TPM on the smartcard, a "roaming-TPM" for use in an enterprise, and general experimentation on the correct definition of security primitives for future inclusion in TPM designs. ■

ABOUT THE AUTHOR

Talha Tariq is a software engineer in the Systems Incubation Group at Microsoft Corporation, where he is currently working on high assurance virtualization and trusted platforms.

SOURCES:

[1] Though technically different; the terms Hypervisor and Virtual Machine Monitor (VMM) are used interchangeably here referring to hardware platform virtualization software that allows multiple operating systems to run concurrently on a host computer providing better security, partitioning, server consolidation, process isolation, disaster recovery and other virtualization services. For an introduction on hypervisor and its services please see http://www.microsoft.com/windowsserver2008/en/us/hyperv-overview.aspx

[2] We use Gemalto.NET cards with version 2 of .NETCF. Similar experimentation can also be done with JavaCards.