THE OPEN PLATFORM PROTECTION PROFILE (OP3) TAKING THE COMMON CRITERIA TO THE OUTER LIMITS

Marc Kekicheff, Forough Kashef

Visa International Services Association
Post Office Box 8299
San Francisco, CA 94128-8999

David Brewer

Gamma Secure Systems Limited
Diamond House, 149 Frimley Rd
Camberley, Surrey GU15 2PS, UK

Abstract

The Open Platform Specification sets a new standard for smart cards, governing the loading, installation and deletion of applications at any time that the card is on-line during the card lifecycle prior to card termination. The Open Platform Protection Profile (OP3) recasts the Open Platform (OP) security requirements into the language of the Common Criteria (CC) to facilitate the formal evaluation of OP smart cards. In doing so, OP3 stretches the CC to new limits. In particular solutions have had to be found to deal with the optional components within the OP Specification, and how to specify the requirements for the "Card/Chip Operating Environment (COE)" on which the OP software sits. In addition to providing the functionality to securely load, manage and delete applications, the OP software offers a variety of security services to applications on a "take it or leave it" basis. A third challenge has therefore been to determine how application protection profiles and security targets may be written to take advantage of these services, which are ostensibly provided via a "security API". The paper also explains how other protection profiles can be used to evaluate the result of integrating the COE together with the OP software. Moreover, the subject of platform independence is addressed, as it is of paramount importance not to have to evaluate every application in every combination with every other application on every platform. The paper also discusses other practical aspects of the application of the CC. It is the authors' intention that this paper will help people to apply the CC, particularly those who face similar challenges, and to create awareness of the security needs of smart card technology.

Key Words: Common Criteria, Java[1], Open Platform, Protection Profile, SCSUG-SCPP, Smart Card, Windows for Smart Cards[2].

Introduction

The paper examines the security requirements of the *OP Specification* [1] for reconfigurable smart cards, and how those requirements have been translated into the language of the *CC* [2] to produce the first public draft of the *Open Platform Protection Profile (OP3)* [3]. The work is part of an overall program to establish the trustworthiness of the OP technology and project that trustworthiness to the market place.

The development of the CC is deeply predicated, for historical reasons, on the *"Orange Book"* [4] specification of a "trusted operating system". It is therefore refreshing to apply an established methodology to a new paradigm. The recent publication of the *Smart Card Security Users' Group Smart Card Protection Profile (SCSUG-SCPP)* [5] has initiated interest in the application of the CC to smart cards, and in so doing explores the application of the CC to chip technology. The present work, OP3, further explores the relationship of protection profiles and, in particular, the use of security APIs.

[1] Sun ™, Sun Microsystems ™, Java ™ are trademarks of Sun Microsystems, Inc.
[2] Windows for Smart Cards ® is a registered trademark of Microsoft Corporation.

The Open Platform Specification

OP smart cards are reconfigurable multi-application smart cards intended for use by customers of card issuing institutions (referred to as Card Issuers) such as banks. The Card Issuer will issue cards most likely with at least one installed application, such as the Visa Smart Debit/Credit application. However, the Card Issuer may allow other organizations (referred to as Application Providers), such as retailers, to load and install their own applications (for example loyalty programs). Applications are written in a hardware independent programming language such as Java™. OP smart cards will therefore significantly shorten the "time-to-market", allowing new applications to be rapidly developed, piloted, rolled out in vast quantities and easily upgraded in response to market demands. The ability to house several applications on a single card and take advantage of new Internet and wireless technologies also has an enormous appeal to the merchant and consumer alike. Indeed, it is anticipated that the ability to reconfigure the application content of a smart card during its lifetime will be recognized as one of the greatest technological contributions to e-business.

The purpose of OP is simply to manage the loading, installation and deletion of applications. The primary components of the OP architecture (see Figure 1) are the *Card Manager*, the *Security Domains* and the *OP API*. The Card Manager is the on-card representative of the Card Issuer and is the central administrator of the entire card. Security Domains are the on-card representatives of Application Providers and may manage the loading and installation of applications pre-approved by the Card Issuer. The OP API provides to applications a programming interface to Open Platform system services supported by Card Manager and Security Domains. The OP shares a runtime environment and hardware neutral API (termed the *RTE API*) with the applications. Beneath this (not shown in the figure) is the runtime environment itself (e.g. JavaCard™ or Windows for Smart Cards®) and the integrated circuitry (collectively referred to in OP3 as the *Card/Chip Operating Environment (COE)*). Applications may invoke the OP services via the OP API and the services of the COE via the RTE API. The multiplicity of Security Domains allows each Application Provider's security data (such as cryptographic keys) to be kept separate and private from that of other Application Providers and the Card Issuer. The smart card takes power from *a Card Acceptance Device (CAD)* once inserted and is powered down when removed.

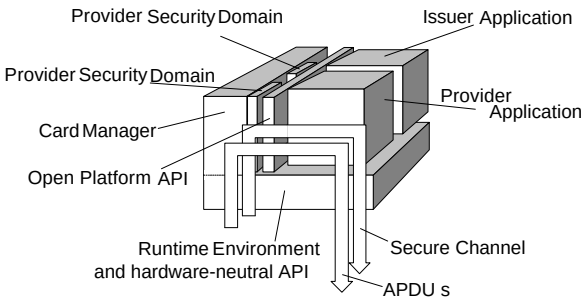

Figure 1: Architecture of an OP smart card

Communication with the card is via the CAD and an on-card *Application Protocol Data Unit (APDU) interface*. The OP authenticates itself to a host computer (e.g. the Card Issuer's card management system) via the CAD and vice versa. The users of OP are the on-card applications and the Card Issuer and Application Providers' host systems. If a Card Issuer offers the cardholder a choice of whether to load an application, the negotiation would be conducted via the CAD and the host system. If this results in a request to load an application, the host machine would establish a secure mutually authenticated communication channel (termed the *Secure Channel*) and command OP to load and install the application. OP does not require cardholder authentication. Applications may use the *Secure Channel Protocol* for their own purpose as a means for achieving mutual authentication and preserving the confidentiality and integrity of APDUs. OP also provides an optional authentication service to applications called the *Global PIN*.

Chip card technology is available today in a range of product and price configurations, based on the capabilities and security of card hardware and software. This is reflected in the OP Specification by a variety of options that allow Card Issuers to choose products that match their business and security

requirements. A minimal OP configuration (that omits the Security Domains) restricts the use of the smart card to the Card Issuer. At the other extreme, a different OP configuration allows Application Providers, with pre-authorization from the Card Issuer, to manage the loading and installation of their own applications. In another configuration, the Card Issuer manages the loading and installation of all applications and therefore acts on behalf of the Application Providers.

Security Requirements

Security Assumptions

It is important to recognize that the OP is merely a component of a much larger system that includes people, organizations and other computer systems. Some of these components are untrusted, whereas others form an integral part of overall system security. For example Card Acceptance Devices (CADs) are untrusted, whereas the buildings that house the back-end systems or hosts would be expected to be secure. In deriving the security requirements for the OP it was therefore necessary to make assumptions about the security that will be provided by the other components of the wider system. A general property of these assumptions is that their removal (e.g. from OP3) would expose an exploitable vulnerability that the OP could not possibly protect itself against. The assumptions are fully detailed in the OP3 and concern, for example, the security characteristics of the applications, the COE, the back-end systems and cryptographic key management.

In order to understand the OP security requirements it is particularly important to understand the assumptions that concern the other on-card entities, namely the applications and the COE.

The assumption that concerns applications asserts that all on-card applications must write to the RTE API and OP API so that these two APIs may mediate *all* application access to the OP and COE services. This means that the Card Issuer must check the validity of this assumption *before* authorizing an application to be loaded onto a card. This check may require the use of special tools to verify that the application cannot violate the defensive mechanisms of the COE. For example, in the case of JavaCard™, the traditional Java™ security model is split between the smart card of the off-card development system. A special byte-code checker is used to verify that the Java™ code, in which the applications are written, satisfies this assumption.

The assumption that concerns the COE asserts that the COE has the following properties:

- It is tamper resistant, making it practically very difficult for an attacker to extract any data directly from the chip by using techniques commonly employed in integrated circuit failure analysis and reverse engineering efforts.
- It is resistant to differential power analysis and other forms of sophisticated attack.
- Following power loss or smart card withdrawal prior to completion, it will allow the OP to eventually complete an interrupted operation successfully, or recover to a consistent and secure state.
- It will handle exceptions raised by applications and report the nature of the exception and the identity of the offending application to the OP.
- It prevents the OP security functions from being bypassed, deactivated, corrupted or otherwise circumvented.

- It enforces separation between applications (including OP components) so that one cannot interfere with another or even the COE itself.

- It prevents the contents of programmable non-volatile memory[3] from being accessed when that memory is reused (e.g. so that data belonging to a physically deleted application cannot be accessed).

These requirements share much in common with an Orange Book B3 operating system and, from the perspective of OP, provide a very similar secure operating environment.

Threats to Security

Figure 2 shows a categorization of the threats that are endemic to a smart card in general (see SCSUG-SCPP [5], for example) that are dealt with by the COE, and those that are specific to OP. Group I threats concern direct attacks on the chip circuitry using techniques that are usually reserved for testing and debugging chips. Group II threats concern more sophisticated attacks that monitor the external effects of the chip operation, such as power consumption. Group III concerns attacks using cards that have yet to be issued, cards from previous issue generations and clones of current cards. Group IV threats concern the card's usual interface to the outside world via the CAD and deals with problems such as those arising from the premature removal of the card from the CAD (often known as "tearing"). Group V deals with attacks on the RTE and OP that are made through the card's interface to the CAD. Group VI deals with the threats concerning the post-issuance loading of applications and their subsequent need to share resources.

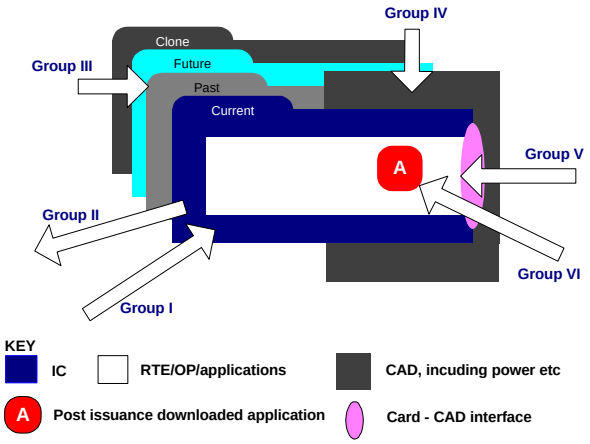

Figure 2: Attack path analysis for a smart card

OP has an interface to its users. The first point of an attack in Group V would therefore be an attempt to *impersonate an authorized user*. If the attack was successful then OP might be fooled into loading unauthorized applications, divulging security data and making unapproved management commands, such as terminating the card. Alternatively, a bone fide user may read, modify, execute or delete applications, information or other resources without having permission from the authority that owns or is responsible for the application, information or resources. For example, without proper security control, an Application Provider might accidentally delete the Card Issuer's applications. Thus the second type of attack is a user, even if properly authenticated, *may attempt to do things that are outside of their intended authorization*. The third form is that a user, even if properly authenticated and authorized, *may systematically experiment with different forms of input in attempt to violate OP security*. This attack is based on the "black box" software engineering technique of establishing the nature of algorithms and predicates ("IF" statements). If carried out exhaustively it could facilitate the reverse engineering of OP as well as the extraction of operational and security related information. A fourth type of attack is that *an attacker might eavesdrop on OP communications with a host*. They might do this as a precursor to masquerading as a Card Issuer or Application Provider, to steal confidential information (including application code) in transit or to record APDU sequences for a subsequent replay attack. The fifth type of attack is the *replay attack* itself. In this case intercepted APDU commands may be replayed,

[3] Smart cards will usually contain between 1K – 24K of programmable non-volatile memory (EEPROM) and a similarly limited amount of permanent memory (ROM).

possibly with modification or in a different sequence, to facilitate a successful attack. These five classes of attack are common to all forms of applications.

The first type of a Group VI attack of relevance to OP concerns the *malicious generation of errors in the set-up sequence* prior to card issue. During the stages of card issuance that involve loading the OP with cryptographic keys, lifecycle states etc, the data itself may be changed from the intended information or may be corrupted. Either event could be an attempt to penetrate the OP security functions or to expose the security in an unauthorized manner. Errors could be generated through simple errors, or through failure of some part of the transfer mechanisms. These errors could occur during loading of programs and/or loading of data. For example, memory usage limits could be exceeded, both at application load and when an application requests data memory. Similar attacks could be attempted when applications are loaded, installed and personalized post issuance. An attacker may utilize *unauthorized applications* to penetrate or modify the OP security functions. Such applications could include copies of authorized applications that had been infected with a virus or a Trojan horse. An attacker might attack an application based on an *analysis of the same application in a different environment*, such as a PC. An attacker might *delete applications without authorization*. Finally, an attacker may *force OP into a non-secure state* through inappropriate termination of selected operations, e.g. through premature termination of transactions or communications between OP and the CAD, insertion of interrupts, or by selecting related applications that may leave files open.

Normally, an adequate trust relationship will exist between the Application Providers and the Card Issuer, which will ensure that Application Providers are confident in the integrity of their applications when the Card Issuer loads them. Application Providers should also have confidence in the effectiveness of the Card Issuer's security systems that would prevent an attacker from modifying the application code prior to loading. Under certain circumstances, these assumptions may be invalid. In this case the ability of an attacker to *modify an Application Provider's application code prior to dynamic load* poses an addition threat.

Security Functions

The OP Specification details a wide variety of security functions to counter the aforementioned threats. Many of these are cryptographic in nature (see below). There are extensive access control rules, which serve to counter the threat that users might attempt to *delete applications without authorization* and that users *may attempt to do things that are outside of their intended authorization*. Some of these rules are discretionary in nature; the discretion being in the hands of the Card Issuer to approve access rights to Application Providers and applications. Other rules are mandatory in nature and are imposed by the OP Specification in the form of state transitions. Thus, the OP Specification describes a state machine. It refers to those states as the *Card Manager Lifecycle*, *Executable Load File Lifecycle* and *Application Lifecycle* states. A sample card configuration illustrating the possible lifecycle states and transitions of the Card Manager, Executable Load File and applications is illustrated in Figure 3 (see the OP Specification Chapter 5 for details). Note the sequence of Card Manager states, which also reflect the overall state of the card. The sequence starts with the OP_READY state. It is in this state that the OP is ready to accept the Card Issuer's instructions for loading cryptographic keys, and the pre-issuance loading and installation of applications. Applications can be loaded for future installation. The card must only be issued when it is in the SECURED state, a requirement that OP3 covers by an assumption. These discretionary and mandatory rules and state transitions serve to frustrate an attacker who *may systematically experiment with different forms of input in attempt to violate OP security*. This objective is further enhanced by rules that allow applications to block themselves and, given the right privileges, to even block the card itself. If an attacker finds a weakness in an application, by *analysis of the same application in a different environment*, the offending application can be locked and subsequently replaced

when a remedy has been found. Other security functions check the validity of OP security data and provide a form of intrusion detection. In the latter context the OP is empowered to lock applications if it perceives an internal security threat, for example because the application is raising too many runtime exceptions of a particular type. It is also necessary for the OP to de-allocate memory in the event of a power failure while loading an application. This provides a defense against an attacker who uses power failure to *force the OP into a non-secure state*.

Cryptographic Solutions

The remaining threats are addressed by cryptographic means. Mechanisms exist to load cryptographic keys both before and after issuance in a secure manner, to generate session keys and destroy keys. Mechanisms also exist to maintain independent key sets for the Card Manager and each Security Domain. Applications may therefore load their own keys encrypted by their Security Domain's key encrypting key.

The Secure Channel Protocol provides *host-card authentication*, *key confidentiality*, *message encryption*, *message authentication* and *MAC*[4] *chaining* using 3-DES and double length keys. Secure Channel mutual authentication is achieved through session key agreement by deriving a common set of session DES keys from static DES keys and subsequently using the session keys to calculate and to verify authentication data. Message authentication verifies the integrity as well as the authenticity of an APDU command sequence. The integrity of the sequence of commands being transmitted to the card is achieved by using the MAC from the current command as the Initial Chaining Vector (ICV) for the subsequent command. This ensures the card that all commands in a sequence have been received. These security functions counter the threats of *impersonating an authorized user*, *eavesdropping on OP communications*, *replay attack* and *malicious generation of errors in the set-up sequence*.

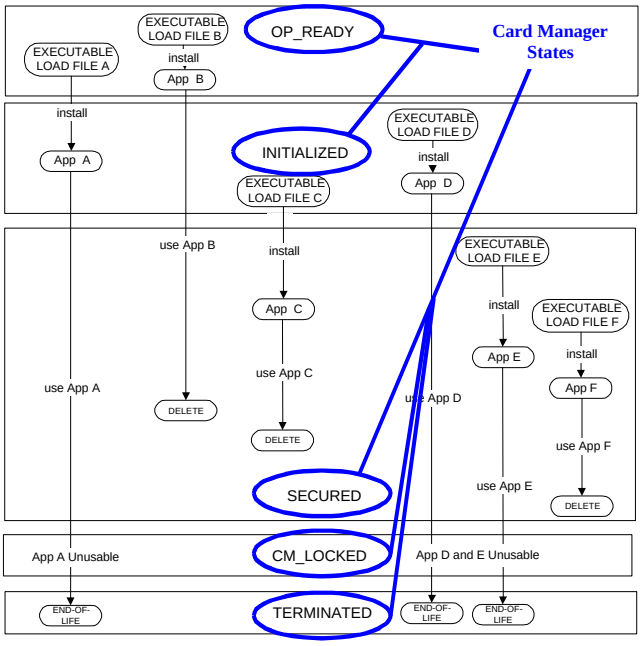

Figure 3: State transitions enforced by the Card Manager

Additional cryptographic functions using RSA with at least 1024 bit keys and the secure hash algorithm, SHA-1, are used to counter the threat of loading *unauthorized applications* when the task of loading and installing applications has been delegated to an Application Provider. There are two functions. In the first, the Card Issuer digitally signs the application to indicate that it is an authorized application. OP checks the validity of the signature prior to loading and installation. In the second, OP generates a cryptographic receipt to say that the application has been installed (or for that matter deleted) and returns that to the Card Issuer.

Another cryptographic function, which may use RSA/SHA-1 or DES to generate a data authentication pattern, is used to prove that *an Application Provider's application code has not been modified prior to dynamic load*.

[4] Message Authentication Code

The Open Platform Protection Profile (OP3)

The OP3 follows the usual structure with an *introduction*, a *description of the Target of Evaluation (TOE)*, a definition of the *security environment* (assumptions, policies and threats), *security objectives*, *IT security requirements* and *rationale*. To enhance readability rationale statements and application notes are also included "in-line" so that in defining a security objective, for example, the reader is immediately told which threats that security objective counters. The Evaluation Assurance Level (EAL) requirements are not repeated except where they have been refined. There are three appendices. The first lists the acronyms used. The second provides a specification for the COE and the third provides guidance to the authors of application protection profiles and security targets.

Almost every TOE Security Function (TSF) has been refined leaving the security target author little to do save mapping the TSFs onto the functional components that satisfy them. In this sense OP3 is quite unlike other protection profiles (see [5], [6] for example), and arises simply because the OP has a very specific job to do and in that sense is very "un-generic". In turn, this explains why certain TSFs are iterated several times. For example there are eight iterations of FCS_COP.1 (cryptographic operation), as there are eight distinct cryptographic operations.

OP3 either details the specific requirement in the OP Specification or refers to them by reference. The following extract serves as an example. Note the use of the + sign to indicate that the component is iterated.

> The TSF shall perform **delegated management receipt generation** in accordance with a specified cryptographic algorithm **(3-DES)** and cryptographic key **of double key length** that meet the following: **ANSI X9.52, FIPS 46/3 and OP Specification, paragraphs 7.9.2, 7.9.4, 7.9.5, 11.1, 11.1.3, 12.1.2.3, 13.9, 13.9.1, 13.9.2 and 13.9.3**. FCS_COP.1+7.1

The Challenges

Producing the OP3 presented a number of challenges.

Challenge No. 1 – Dealing with Optional Components

The first challenge concerned how to deal with the optional requirements in the OP Specification. These requirements fall into two categories: those concerning a choice of implementation and those concerning a choice of functionality. The former was easily dealt with by refining the TSF to offer OP3 specific selections to the ST author. FCS_COP.1+8 concerns load file verification, but the choice of whether to use DES or RSA/SHA-1 to implement the data authentication pattern rests with the Card Issuer and the security target author:

> The TSF shall perform **Load File verification** in accordance with a specified cryptographic algorithm **(Selection: *SHA-1, RSA or 3-DES in CBC mode*)** and cryptographic key sizes **(Selection: *not applicable*** (for SHA-1)**, *1024 bit*** (for RSA) ***or double key length*** (for 3-DES)**)** that meet the following: **(Selection: *FIPS 180-1*** (for SHA-1), ***ANSI X9.31*** (for RSA), ***ANSI X9.52***, ***FIPS 46/3*** (for 3-DES)**) and OP Specification, paragraphs 4.3.1, 11.2.2, 12.1.2.3, 12.2, 12.3, 13.7, 13.7.1 and 13.7.2**. FCS_COP.1+8.1

The latter proved more problematical. Nevertheless, the ability to group the optional components into TOE configurations (see Table 1) proved to be a simplifying strategy.

The first idea was to devise a family of protection profiles, one for each configuration. As the Global PIN is also an option, this meant that there would be ten protection profiles. These protection profiles would,

of course be very similar, differing only in the inclusion or exclusion of a few TFSs. However, ten protection profiles might mean ten registrations and ten evaluations. There was also considerable scope for introducing unwanted differences in the protection profiles over time. The authors therefore sought to combine these ten profiles into a single document, giving in-line instructions in the form of an "applicability" statement to say whether the assumption, policy, threat, security objective or TSF applied to a particular configuration or not. In practice, the authors felt that this was an inelegant solution. It effectively made every statement in the OP3 conditional on the OP configuration, making reviewing extremely difficult. An alternative solution was therefore attempted that proved far more satisfactory.

This alternative approach utilizes the CC concept of a package. The functionality identified in Table 1 was mapped onto four packages: *Basic*, *Delegated Management*, *DAP Verification* and *Global PIN* as shown in Table 2. The Basic Package must always be present. Including or excluding the other packages

	OP Functionality				Cryptographic Support	
Configuration/ Feature Set	Card Manager	Security Domains	Delegated Management	DAP Verification	DES	RSA
Configuration 1a	X				X	
Configuration 1b	X	X			X	
Configuration 1b*	X	X		X	X	
Configuration 2a	X	X	X		X	X
Configuration 2b	X	X	X	X	X	X

Table 1: OP configurations

serves to differentiate between the different OP configurations. The bulk of the assumptions, policies, threats, security objectives and TSFs are part of the Basic Package, only the differences belong in the other packages. The ability to iterate a component proved to be a useful feature in implementing these packages. For example, OP3 utilizes three iterations of FDP_IFC.1 [Information flow control policy] and FDP_IFF.1 [Information flow control functions]. Two respectively concern the mandatory rules for state transitions and use of the OP API, and form part of the Basic Package. The third concerns the mandatory rules associated with the Global PIN and belongs to the Global PIN Package.

	OP Functional Package			
Configuration	Basic	Delegated Management	DAP Verification	Global PIN
1a, 1b	✓			optional
1b*	✓		✓	optional
2a	✓	✓		optional
2b	✓	✓	✓	optional

Table 2: Functional packages

Challenge No. 2 – Specification of the COE

The first draft of the OP3 merely invoked the French Protection Profiles [6]. This was expedient, but the referenced profiles did not fully reflect the requirements of the COE. Moreover, their frequent upgrading warned of the peril in relying on a document that was not in the authors' control. The second draft made use of the SCSUG-SCPP. Rather than reference it, the second draft of OP3 embraced it. The scope of OP3 was therefore extended to include the COE. Certain SCSUG-SCPP requirements appeared

redundant, as, arguably, they were required by the applications, which were outside the scope of OP3, so they were discarded. The additional requirements, peculiar to the OP Specification were of course added, making the second draft of OP3 a complete stand alone protection profile for the OP and its COE. At first view this appeared to be a more satisfactory solution. However, the inclusion of the COE in OP3 increased the scope of the protection profile evaluation with a presumed increase in cost, timescales and risk. It also meant that changes to the SCSUG-SCPP, which (at the time of writing) is itself in a state of development, need be tracked in case there is some change to the SCSUG-SCPP that is relevant to the OP3. However, a far more important reason arose to abandon this approach. Figure 1 shows that the OP shares the RTE API with the applications. This meant that OP3 would have to refine the SCSUG-SCPP TSFs to describe not only the OP use of that API but also that of the applications. However, applications are outside the scope of the OP Specification. It was therefore decided to exclude the COE from the TOE (i.e. making the TOE once again just the OP) and provide just a specification for its security characteristics.

The starting point for the COE specification is the assumption described earlier in this paper (termed the *COE Assumption*). This was translated one-on-one into the security objectives for the COE. In turn, these were used to identify the security functions necessary for the COE to satisfy these objectives. Finally, the objectives were mapped onto the applicable security threats in the SCSUG-SCPP plus three others. First: an attacker may exploit the ability of one application to pass data to another to covertly leak sensitive data. Second: an attacker may exploit the ability of one application to share resources with another to modify the operation of that other application in an undesirable fashion. Third: an attacker may find ways to bypass, deactivate, corrupt or otherwise circumvent any additional levels of security provided by the applications within the COE's Scope of Control. These additional threats highlight the importance of not simply deferring to another protection profile. The applications in the third threat include the OP.

Security targets that are compliant with the OP3 would facilitate the evaluation of the OP software in hard or soft mask form in its operational environment. However, the CC requires that all assumptions are treated as axiomatic. The authors do not consider that such an important assumption should be taken on trust, even though the evaluation testing of OP would take place with the COE. To force the evaluation of the COE, particularly to ensure that it possesses the necessary security properties, the OP3 requires that security targets are prepared in compliance with OP3 and an "Integration Protection Profile". The OP3 cites the SCSUG-SCPP as a suitable protection profile. However, a question that remains to be answered is: is this approach sufficient to guarantee that the whole of the COE Assumption is validated, bearing in mind that the assumption implies the defense against threats which are not covered by the SCSUG-SCPP?

Challenge No. 3 – The OP API

From an application perspective the OP forms part of the application's COE. Hence the second challenge has to be revisited from the application perspective to ensure that OP3 does not inadvertently frustrate the development of applications for use with OP smart cards. It seemed sensible to follow the same approach. OP3 derives the *OP Assumption* by augmenting the COE Assumption as follows:

- It is impossible to load an application onto an OP smart card without the authorization of the Card Issuer.
- Card Issuers and Application Providers can check the authenticity and integrity of their applications when they are loaded and can ensure confidentiality of application code and data.
- It is only possible to remove applications from an OP smart card with the authority of its owner.

The application protection profile author is then invited to refer directly to OP3 as the means by which the OP Assumption is met, or alternatively use the assumption to generate a Secure Open Platform Specification, using the COE Specification given in the OP3 as a model. The requirement of the OP Specification, and hence OP3, are sufficiently detailed to allow the first of these approaches to work. The second approach facilitates the development of protection profiles for applications intended for use in a variety of environments (e.g. PCs as well as smart cards). In the former case, the OP Assumption (which embraces the COE Assumption defined earlier in this paper) ought to be sufficient to ensure platform independence. In other words, an application need not be evaluated on every platform in combination with every other possible application. The reasoning is as follows. The OP would have been evaluated in conjunction with the COE and the validity of both the COE and the OP Assumptions would have been established. The application is written in a hardware independent language.

In addition to providing a secure environment for applications, the OP offers the services of the Secure Channel Protocol, the Global PIN and the ability to control various state transitions via the OP API. The CC does not provide an elegant way to invoke such services; i.e. there are no CC components that are specifically aimed at defining security APIs. Instead, as currently recommended in OP3, the application author is invited to select those CC components that best describe the interface and use application notes to complete the specification. The OP3 recommends various CC components as a starting point, for example, FIA_UAU (User authentication) should be used to invoke the Global PIN. The OP3 notes that security targets should provide a mapping of the implementation of the TSFs used to invoke the OP services to the appropriate API calls defined in the OP Specification. Until the CC provides a means for defining security APIs, applications will have to follow this same, rather inelegant approach to invoke the RTE API.

Other Observations

Some CC components allow the initiation of a service to be defined but not its termination. There are two instances in OP3: FTP_ITC.1 [Inter-TSF trusted channel] and FPT_RVM.1 [Non-Bypassability of the TOE]. In the first case there is a variety of ways to close the Secure Channel, but they are the only ways defined in the OP Specification. In the second case the COE also needs to ensure that control is passed back to the TOE (specifically the Card Manager) as soon as an application completes. OP3 deals with these termination conditions and essential security requirements via application notes, but questions whether the CC should be extended for dealing with the termination conditions and security requirements.

The OP requirement to take action against applications that behave suspiciously is also problematic. OP3 addresses the requirement by specifying functionality within the COE to handle application exceptions and report the nature of the failure to the Card Manager. The OP3 then specifies functionality (using FAU_ARP.1 [Security alarms]) to take appropriate action on the COE furnished information. A similar approach is used to handle roll back. In this case, when the COE detects that there has been a power failure (which given the limitations of current smart card technology will usually be on the next power-up), the COE must inform OP of the failure so that it may take appropriate action. The OP3 links these related OP/COE functions via cross references in the in-line rationale statements and application notes.

Conclusions

The production of the OP3 has stretched the CC to its limits. The authors have no doubt that improvements to the CC, such as the introduction of components to deal with security APIs, would have made the task easier. Nevertheless, it has proved possible to use the CC in its current form to recast the OP Specification in the form of a protection profile.

The use of OP, the CC and the approaches taken to address the particular challenges in the production of the OP3 should have a number of business benefits:

- Users will have confidence that OP smart cards are implemented correctly in accordance with the OP Specification and relevant cryptographic standards and that there are no ways to bypass, corrupt, deactivate or otherwise circumvent its security features.
- Users and card vendors have a choice of OP configurations to meet different budget, market and security requirements.
- Card vendors can evaluate their cards in the country of their choice and have the certificate recognized in other countries worldwide under the Mutual Recognition Arrangement (MRA).
- Card vendors can minimize evaluation risk by evaluating their OP separately against OP3 first and subsequently with the COE against the Integration Profile.
- Applications can be developed independently of the internal OP/COE requirements, making use of the OP Assumption and the OP API (although an extension to the CC to utilize security APIs would make the latter easier).
- The ability to use hardware independent languages, such as Java™, allows the rapid development and rollout of new applications in response to market demands.
- The ability to house several applications on a single card and take advantage of new Internet and wireless technologies also has an enormous appeal to the merchant and consumer alike.

The OP Specification and OP3 are publicly available *now* at the visa website (www.visa.com). Work has already begun to develop products that meet the OP Specification and should soon be available as CC certified versions.

Acknowledgments

The authors would like to thank Dr. Ken Ayer for his enthusiastic support and most helpful comments and suggestions.

References

[1] *The Open Platform Specification, Version 2.0.1 issued May 2000, www.visa.com*

[2] *The Common Criteria for Information Technology Security Evaluation Version 2.1, August 1999 (ISO 15408:1999)*

[3] *The Open Platform Protection Profile, Version 0.5.0.1 issued May 2000 www.visa.com*

[4] *Department of Defense Trusted Computer System Evaluation Criteria (TCSEC), DOD 5200.28-STD, December 1985*.

[5] *The Smart Card Security User Group Smart Card Protection Profile, Draft Version 2.0, 1 May 2000 http://csrc.nist.gov/cc/sc/sclist.htm*

[6] *Protection Profile 9806 - Smartcard Integrated Circuit (revision of PP 9704 - Smartcard Integrated Circuit), Protection Profile 9810 - Smartcard Embedded Software, Protection Profile 9911 - Smart Card Integrated Circuit with Embedded Software (supersedes PP9809 - Smart Card Integrated Circuit with Embedded Software), http://www.eurosmart.com and http://www.scssi.gouv.fr*