GLOBALPLATFORM™

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

GlobalPlatform Value Proposition for Remote Post-Issuance Secure Access Modules (SAM) Management

White Paper

October 2011

secretariat@globalplatform.org | www.globalplatform.org |

Table of Contents

ABOUT GLOBALPLATFORM

GlobalPlatform is a cross industry, not-for-profit association which identifies, develops and publishes specifications which facilitate the secure and interoperable deployment and management of multiple embedded applications on secure chip technology. Its proven technical specifications are regarded as *the* international industry standard for building a trusted end-to-end solution which serves multiple actors and supports several business models.

The freely available specifications provide the foundation for market convergence and innovative new cross-sector partnerships. The technology has been adopted globally across finance, mobile/telecom, government, healthcare, retail and transit sectors. GlobalPlatform also supports an open compliance program ecosystem to ensure the long-term interoperability of secure chip technology.

As a member-driven association with cross-market representation from all world continents, GlobalPlatform membership is open to any organization operating within this landscape. Its 70+ members contribute to technical committees and market-led task forces.

For further information, visit www.globalplatform.org.

PUBLICATION ACKNOWLEDGEMENTS

GlobalPlatform wishes to thank all members of the Government Task Force and Transportation Sub Task Force. Special thanks go out to the following GlobalPlatform members and their respective organizations:

Full Members:

Jerome Becquart – ActivIdentity
Eric Le Saint – ActivIdentity

Participating Members:

Tim Richards – Aconite
Pier Aldershof – Collis
Stephanie El Rhomri – FIME

GlobalPlatform Team Members:

Alliances Management – Operations Secretariat
Gil Bernabeu – GlobalPlatform Technical Director
Kevin Gillick – Executive Director of GlobalPlatform

EXECUTIVE SUMMARY

This whitepaper explains the value that GlobalPlatform specifications bring to post-issuance management of Secure Access Modules (SAMs).

SAMs serve as the backbone of many electronic transaction systems, with one such example being Automated Fare Collection (AFC) in public transportation. As a smart card developed to securely store keys and perform cryptographic functions, SAMs can be managed by a central entity so that a system's security is guaranteed. This central management is possible, even if each of the components (and each of the SAMs) is manufactured by multiple vendors and managed by different parties. Whether deployed in point of sale terminals, smart meters, mobile handsets, or other, SAMs enable secure transactions when a Customer Medium is presented at a given terminal.

The benefits of SAMs become complicated, however, by several market dynamics. Because multiple applications can be present on a card—and these applications can be issued by different companies—there is a need to manage security both within and between these applications. Further, there is a need to manage the SAM itself. Traditionally, SAM Management has occurred at the moment the SAM is issued, and this is typically through a centralized issuance process. Without a way to remotely update SAMs after they have been installed, costly physical replacement is the only way to respond to new business needs, a discovered bug, or a need to incorporate new industry standards. And, while it has traditionally been possible to remotely update Application Personalization post-issuance, doing so is complicated by the fact that different parties need to update the keys, certificates, and ceiling counters for their own applications—*without interfering with the applications of other parties.*

As a result, SAM owners are confronted with the need to remotely and securely perform post-issuance SAM Management and Application Personalization, and this must be possible without running the risk of interference or leakage of one party's secrets to another. This requirement is further complicated by the lack of real-time connectivity in many networks, the EEPROM wear encountered in many systems, and the need to update blacklists for devices that should no longer be accepted.

The GlobalPlatform Card Specification v2.2.1, Ref. [1], provides several features that address these shortcomings with remote management of SAMs:

- A *Security Domain (SD)*, which acts as the on-card representative for an off-card entity and thus allows management of any application associated with the SD
- *Card Content Management*, which involves the creation, modification and deletion of Security Domains and applications on the card, privileges which are controlled by either Authorized Management or Delegated Management
- *Application Personalization*, which needs to happen without interfering with other parties' applications
- *Secure Channels*, which offer protection of the integrity, confidentiality and authenticity of all messages communicated

Such features resolve many of the issues with SAMs today, whether this involves issuance, reissuance, updating, activation, blacklisting, or status reporting of SAMs. The GlobalPlatform specification offers the means to remotely manage SAMs post-issuance, thus reducing cost and complication, and improving application flexibility—without compromising the security needs of the many parties and applications within a given card.

SECTION 1: INTRODUCTION

Prior to exploring the GlobalPlatform Card Specification and what it can bring to a variety of Secure Access Module (SAM) use cases, it is necessary to define the SAM and understand the role that SAMs play in electronic transaction systems. Further, it will be necessary to discuss the variety of applications in which SAMs are deployed today—from ticketing, to telecommunications, to financial transactions—so as to understand the pervasiveness of SAMs and, thus, the importance of identifying a solution for post-issuance remote management.

1.1. The Role of SAMs in Electronic Transaction Systems

Secure electronic transaction systems are systems that securely facilitate the exchange of large volumes of transaction data. Such systems have been broadly deployed for the purpose of electronic payments, electronic identification, mobile telecommunications, and Automated Fare Collection (AFC) in public transportation. Similar infrastructures are emerging in use cases as diverse as fee collection for electronic tolling or smart metering in electric power networks.

This kind of system consists of a number of central systems and many decentralized components; both are often manufactured and managed by multiple vendors and parties. In the course of a transaction (such as a check-in in public transportation or the passing of a toll gantry in an electronic tolling system), these components exchange information. This information is not only used for giving access to services, but also for billing and other purposes. Securing this information is therefore one of the highest priorities. Yet one of the factors complicating this is that the decentralized components do not have real-time connections with the central systems at all times. This means that these decentralized components must be able to function in standalone (offline) mode while keeping the level of security intact.

In many such electronic transaction systems, Secure Access Modules (SAMs) form the backbone of the security infrastructure. A SAM is a smart card developed to securely store keys and perform cryptographic operations. Every decentralized component of a secure electronic transaction system is equipped with such a SAM. And, all security-related operations carried out by such a component are delegated to its SAM. All SAMs in the system are in turn issued and managed by a central entity that is responsible for the security of the system. By employing SAMs, this entity is able to guarantee the security of the system—*even though the components making up the system are manufactured by multiple vendors and managed by different parties*.

Figure 1 shows an Automated Fare Collection system using SAMs. Please note that in this document, examples and illustrations will primarily be taken from AFC systems in public transportation. It is important to realize, however, that use of SAMs is not limited to this kind of system: a non-AFC example (in the financial industry) is discussed in Section 1.3.

Figure 1 – Automated Fare Collection System, showing the use of SAMs.

1.2. SAM Functionality

A SAM is a smart card developed for securely storing symmetric or asymmetric keys and sensitive data. Both hardware and software are engineered to prevent any secretive information from being leaked. SAM hardware, for example, contains countermeasures against leaking information via electromagnetic radiation, timing measurements, and other side channels. These properties mean that SAMs offer a much higher level of protection than the terminals into which they are inserted, which are often based on general-purpose computers.

Figure 1 above shows the typical use of SAMs in an AFC system. All terminals in the so-called Level-1 contain a SAM. These terminals interact directly with the Customer Media that

travelers use to enter the public transportation system. An AFC system may contain thousands or even tens of thousands of such Level-1 devices. Typically (as shown in Figure 1), these devices are property of and managed by several different Service Operators, each having a central system, which is represented in Level-3 of Figure 1. However, the overall security of the entire system is the responsibility of the Security Manager, which resides on one central system that connects all Service Operators. This is known as Level-4 of the system.

Note that Figure 1 concentrates on the management of the SAMs in the system, and therefore leaves out many other essential functions of the central systems, such as clearing and settlement of transactions, Customer Media management, equipment operating data management, float management, and more. This whitepaper will focus on the responsibilities and challenges of SAM Management and SAM application management.

Apart from employed in Level-1 equipment in AFC systems, SAMs could also be contained in the following:

- All Point of Sale terminals in electronic payment systems
- All Handsets in a mobile telecommunications system[1]
- All onboard units in vehicles and all roadside equipment (used for measuring or compliance checking) in electronic tolling systems
- All Smart Meters in electrical power grids
- All Vehicle Units in (a possible future version[2] of) the European Digital Tachograph system
- Access control systems (both logical and physical)

Although the exact functionality of SAMs will differ from one system type to the next, a number of common functions typically exist:

- A SAM usually contains the identifiers that identify the terminal into which it is inserted. Storing these identifiers in the SAM ensures that an attacker cannot change the identity of the terminal. Moreover, they allow the Registrar[3] to issue an identifier to a terminal without communicating with or relying on either the manufacturer of the terminal or the party managing it.
- A SAM usually contains the (symmetric or asymmetric) keys that enable the terminal into which it is inserted to communicate with other equipment. Some examples of this include the following:
    - In many AFC systems the SAM contains the Ticketing Keys. If a traveler presents a Customer Medium to a terminal, the terminal communicates an identifier for this Customer Medium to the SAM. Based upon this identifier and the Ticketing Keys, the SAM is able to generate the symmetric keys that the terminal must use to communicate with this particular Customer Medium.

[1] In fact, the SIM card of a GSM-based mobile telephone is functionally equivalent to a SAM.

[2] Note that these Vehicle Units currently do not contain a SAM. Instead, the private and public keys and certificates needed by each unit are stored on the Vehicle Unit itself, making a separate security evaluation of these units necessary.

[3] This whitepaper assumes the role model for AFC systems contained in ISO 24014-1, Ref [4]. In this model, the Registrar is responsible for issuing unique registration codes to all entities in an AFC system.

    - In AFC systems, the (symmetric or asymmetric) keys can enable the terminal in which it is inserted to digitally sign transaction data. Such transaction data includes, but is not limited to, the time, location, Customer Medium and validator identifiers, deducted e-purse value, and remaining e-purse value. These data will be created by a validator every time a passenger checks in. Based upon the transaction data, the AFC system decides how much the Service Operator gets paid. It is therefore of the utmost importance that the authenticity and integrity of such transactions is guaranteed. By signing the transaction on behalf of the terminal, the SAM creates this guarantee.
    - Conversely, a SAM contains the (symmetric or asymmetric) keys that enable the terminal in which it is inserted to verify signatures on incoming records from the central system. Typical records will include blacklists of Customer Media that have been reported as lost or stolen or updates to ceiling limits.
- A SAM can act as the secure repository of counters and cumulative values. In AFC systems, for example, it is important to keep track of the total number of top-up transactions performed by a certain terminal, as well as the total accumulated value of these transactions. This data is sent to the central system and can be used to cross-check the correctness and completeness of the transactions received by the central system. By storing these counters and cumulative values inside the SAM, their security can be protected.
- Finally, a SAM can keep track of the number of transactions a terminal is allowed to perform before it must contact the central system. In many AFC systems, for example, the terminals operate mainly in offline mode. This means that they perform transactions without the central systems explicitly giving permission for each transaction. This way of working increases transaction speed and diminishes communication overhead, but there are inherent, obvious risks:
    - If a terminal has not communicated with the central system for a long time, blacklists will become outdated. Customer Media that have only recently been put on the blacklist may still be accepted by the terminal.
    - If a terminal is stolen or manipulated, the central system has no way of preventing the terminal from executing transactions.

To mitigate the abovementioned risk, the SAM must stop functioning when a certain ceiling limit is reached. The only way to increase this ceiling limit is for the terminal to contact the central system.

In the next section, we explore a non-transit-related architecture. As previously noted, it is important to note that SAMs are used in other architectures, even though (as will become evident), the problems faced are fundamentally the same regardless of architecture.

1.3. Financial SAM Architectures

SAM Management requirements in financial products vary considerably. EMV transactions do not require a terminal-based SAM, although SAMs are often used to support terminal processing or to hold the scheme's public keys, blacklisted certificates, and general cryptographic support, such as Data Encryption Standard (DES) key management for secure communications. In other schemes, such as for e-purses, a SAM may be required to allow offline local transactions to be performed securely; in such an instance the requirements are very similar to those needed to support transit schemes. Similarly, if loyalty schemes are offered alongside financial applets, these may require SAM support.

There may be multiple schemes within such an architecture, and these schemes may feed new applets and personalization data directly to the SAMs through the acquirer networks or over dedicated merchant terminal management networks. Issuers within these schemes may have the option to personalize instances of the applets with their own keys and data.

As shown in Figure 2, Issuers or Merchants (or, indeed, any party with access) may provide applet updates to SAMs through acquirer or merchant networks. So, for instance, a large merchant chain may wish to offer its own store card, or a terminal network supplier may update applets on the SAM for security purposes. Network connectivity varies between merchants and merchant terminal management systems.

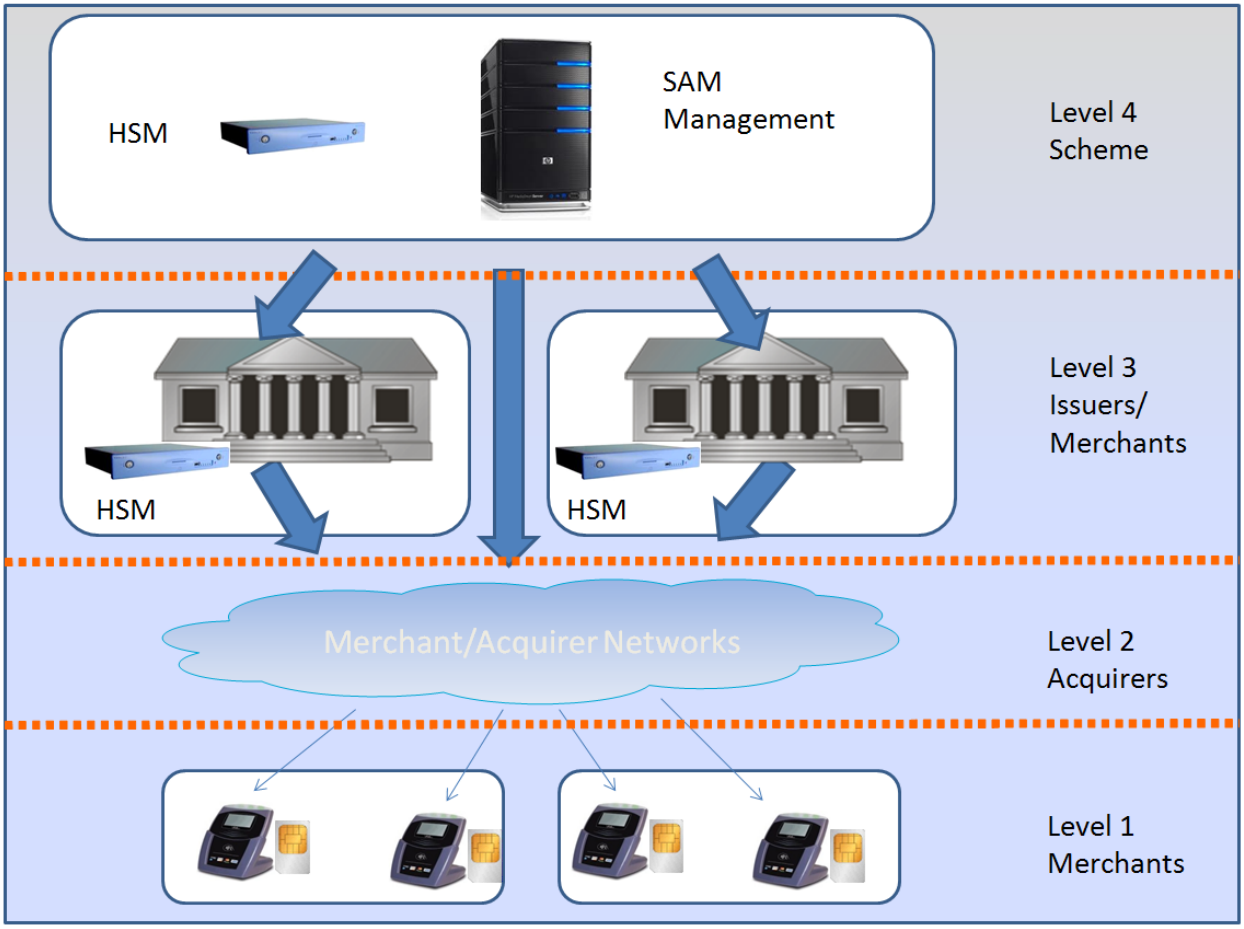

Figure 2 – SAMs within a Financial Network Architecture.

1.4. SAM Architecture

As shown in Figure 3, like any other smart card, a SAM consists of hardware and an operating system, provided as part of the chip firmware. Applications can be loaded and installed into the chip's memory and can then be personalized with data and cryptographic keys specific to an instance of that application. So, for instance, two transit service providers could use the same ticketing media application but have the application personalized differently with data and keys specific to their purposes.

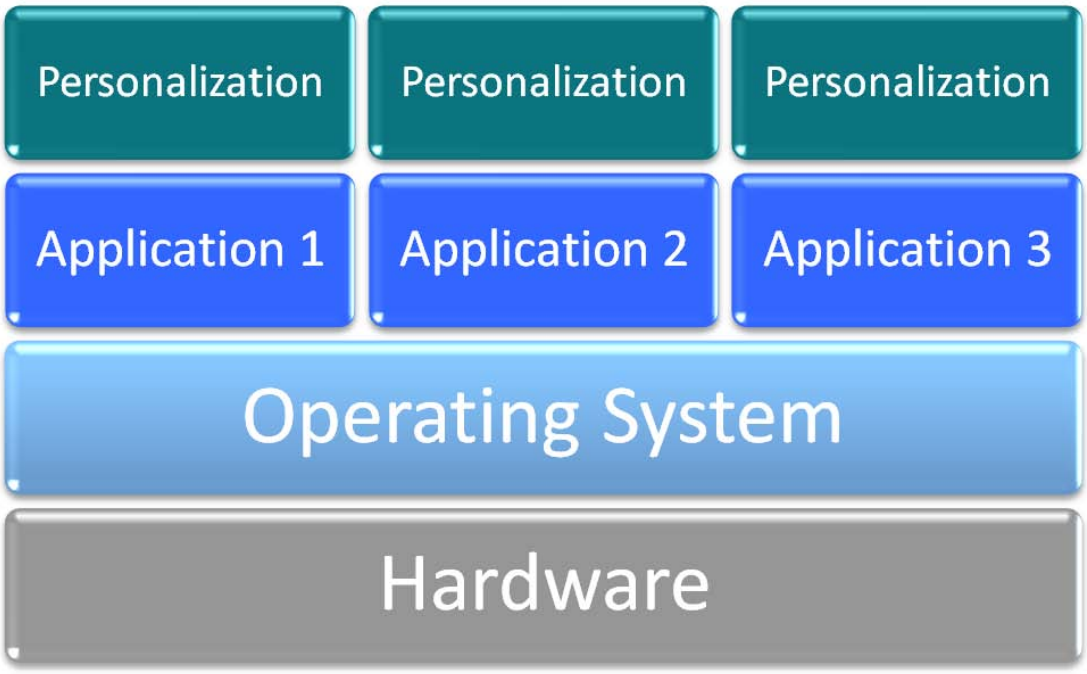

Figure 3 – Basic Smart Card Architecture.

In addition, multiple applications can simultaneously be present on a card. These applications might belong to the same off-card entity but have different functionality. For example, the ability to digitally sign transactions and the ability to derive the keys needed to communicate with a certain Customer Medium might for practical reasons be delegated to two different applications: a central clearing house for transactions and a local transit service provider.

The applications on a SAM might also belong to different off-card entities. For example, assume that a SAM is inserted into a point of sale terminal that accepts cards from different payment schemes. Each of the applications on the SAM would then correspond to one of these payment schemes and would contain the requisite keys and certificates needed to communicate with it.

Most of the applications on a SAM must be personalized. In the case of ticketing media applications, personalization means that the application is provisioned with the card-specific and application-specific keys, certificates, and ceiling counters that it needs to fulfill its tasks.

1.5. The Need for Remote Post-Issuance SAM Management

Like any other type of device, Secure Access Modules need to be managed. SAM Management involves two different processes. First, a number of applications have to be loaded and installed on the SAM. Next, each application on the SAM has to be personalized correctly. This whitepaper refers to the first process as 'SAM Management' and to the second as 'Application Personalization.'

Traditionally, SAM Management occurs only once: at the moment the SAM is issued, usually through a centralized issuance process wrapped in physical and logical security processes. Once a SAM is issued and out in the field, there is no way to install new applications, remove obsolete ones, or replace an application with a new version. In other words, remote post-issuance management of SAMs is not possible. This shortcoming can result in several problems:

- If new business functionality is required from the SAMs (including new types of Customer Media within an AFC system, such as identity applications or open payment tokens), the SAMs will have to be personalized with additional ticketing keys
- A bug might be detected in the functionality of one of applications on the SAMs
- Advances in cryptography lead to a desire to use a safer algorithm or longer keys for the cryptographic operations carried out by an application
- A new version of an industry standard emerges, necessitating an update of an application in order to be able to continue to communicate with terminals that comply to this new version
- Any other changes to industry requirements that were not anticipated at the time a particular SAM was issued

If there is no way to remotely update SAMs after they have been issued, the only alternative is to physically replace all SAMs in the system. With the number of SAMs in the field easily reaching into the tens of thousands, this is a logistical challenge. And as SAMs are generally based on higher-end microprocessors, this can be costly.

The second management process mentioned above is Application Personalization. Although post-issuance updating of SAM Application Personalization has been possible for some time, modern requirements are creating challenges on this level as well. As previously discussed, for some systems it is desirable that more than one party can have an application running on one specific SAM. Each of these parties should be able to update the keys, certificates and ceiling counters for its own application—*without interfering with the applications of other parties*. In such an environment, it is critical that one party is not able to get to know the keys and other sensitive data stored on the SAM by some other party.

SAM owners thus see themselves confronted with two major challenges:

1. The need to be able to remotely and securely perform post-issuance SAM Management and Application Personalization
2. The need to allow SAM Management and Application Personalization to be shared between multiple parties, without running the risk of interference or leakage of one party's secrets to another.

Section 2 discusses how the GlobalPlatform Card Specification offers the tools and techniques to solve these challenges.

1.6. Issues with Remote Post-Issuance SAM Management

Although the introduction of remote post-issuance SAM Management offers issuers significant business benefits, it also raises some new concerns that need to be resolved in order to offer issuers a comprehensive and viable solution. There are three types of issues that need to be addressed in the context of such a solution:

1. *Lack of real-time connectivity* – Many transit networks do not offer real-time connectivity, which means that the messaging interface between central systems and the remote SAMs needs to support a Store-and-Forward type environment.
2. *EEPROM wear* – Microprocessor EEPROM memory has limits on how often it can be written to before the memory is no longer guaranteed to work properly. This is normally managed by individual applications on the SAM, but in cases where the applications are being deleted and replaced, this functionality must be supported by central systems.
3. *Blacklisting* – Transit networks operate over wide areas, and the security of SAMs cannot always be guaranteed. Therefore, it is necessary to be able to block the use of SAMs in instances where they have been stolen. This is usually supported by way of a blacklist.

Section 2 explains the added value attained by using the GlobalPlatform specification to address these issues, along with a corresponding reduction in the number of supporting features.

SECTION 2: THE GLOBALPLATFORM VALUE PROPOSITION

Implementing a SAM on a smart card compliant to the GlobalPlatform Card Specification v2.2.1, Ref. [1], provides a solution for many of the issues outlined in Section 1. This chapter summarizes the properties of the architecture of such a card only insofar as they are relevant for the topic of this whitepaper. For details, the reader is referred to Ref. [1]. The last section of this chapter will come back to the problems with SAM Management identified in the previous chapter, and will show how the GlobalPlatform-defined features help to solve these problems.

Specifically, GlobalPlatform provides the following features that are relevant to the remote management of SAMs:

- Security Domains
- Card Content Management
- Application Personalization
- Secure Channels

Understanding each of these features will enable us to better understand how the issues discussed in Section 1 can be addressed by the GlobalPlatform Specification. The following four sections describes each of these features in turn, with Section 2.5 returning to, and providing answers for, the problems highlighted in Section 1.

2.1 Security Domains

One of the central ideas of the GlobalPlatform (GP) Card Specification is the concept of Security Domains. A Security Domain (SD) is a special type of application that acts as the on-card representative for an off-card entity. The Security Domain provides support for the control, security, and communication requirements of the off-card entity to which it belongs. Moreover, both normal applications and Security Domains can be associated with a particular Security Domain, thus forming a hierarchy as depicted in Figure 4. The holder of a particular Security Domain is able to manage and personalize any application that is associated with that SD.

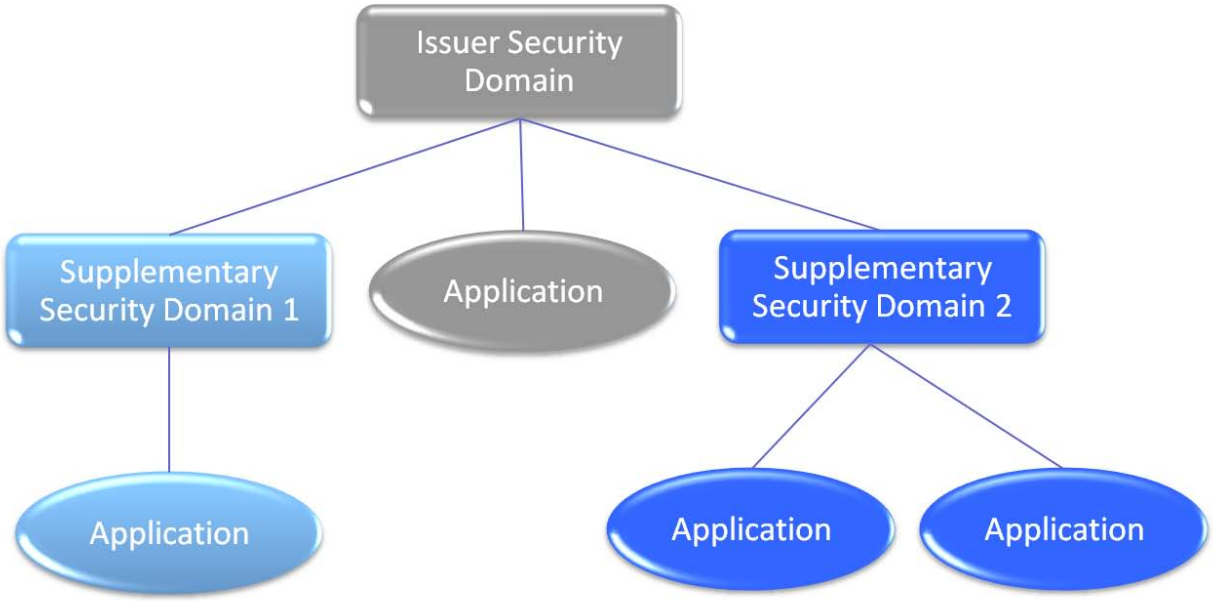

Figure 4 – GlobalPlatform Security Domain hierarchy: the lines in the figure show the association of applications to SSDs (and to the ISD), as well as SSDs to the ISD.

Every GlobalPlatform card has at least one Security Domain—the Issuer Security Domain (ISD), which represents the issuer of the card. By default, the ISD has a number of privileges, such as Card Reset, Global Lock, Global Delete, and Global Registry; these enable the Card Issuer (via its communication with the ISD) to retain overall control over the card. Moreover, as depicted in Figure 4, one or more applications can be directly associated with the ISD, meaning that the Card Issuer can offer services of its own to the users of the card.

A second type of Security Domain is the Supplementary Security Domain (SSD)[4]. This is the generic name for any SD other than the ISD. The off-card entity represented by a SSD is called an Application Provider. The main function of an SSD is to offer its Application Provider a way to confidentially manage and personalize the applications associated with the SSD. Figure 4 shows, in addition to the Card Issuer, the presence of two Application Providers on the card. The GlobalPlatform architecture ensures that the applications, keys, certificates and other sensitive information of these three parties are completely isolated from each other.

2.2 Card Content Management

Another important topic within the GP Card Specification is Card Content Management. Basically, Card Content Management is the creation, modification and deletion of Security

[4] A third type of SD is the Controlling Authority SD (CASD). Essentially, the CASD holds an on-card asymmetric key pair, which allows the owner of a SSD to confidentially personalize a secure channel key set into its SSD. Confidentially in this respects means that no other party, including the Card Issuer and other Application Providers, can get to know the key set. The CASD will not be further discussed in this whitepaper.

Domains and applications on the card. In order to clearly separate the responsibilities of Card Content Management between the different parties that (via their Security Domains) are represented on the card, GlobalPlatform has specified two main privileges for SDs:

- The ***Authorized Management*** privilege gives a Security Domain the power to perform Card Content Management operations within its hierarchy. Once an off-card entity has successfully set up a Secure Channel with an SD holding the Authorized Management privilege, it can order this SD to create, modify or delete any SD or application in its hierarchy. Note that it is the GlobalPlatform Environment ('the OPEN') [5] that in fact will carry out these tasks, after it has verified that the SD requesting them does in fact have the Authorized Management privilege.
- The ***Delegated Management*** privilege gives a Security Domain the same powers within its own hierarchy. However, the difference is that the GlobalPlatform Environment will ask for a Token before carrying out a certain content management operation at the request of an SD with the Delegated Management privilege. A Token is effectively the permission of the Card Issuer to carry out a specific operation. It carries a digital signature that is created by the off-card Card Issuer. The on-card Issuer Security Domain[6] contains the key that is needed to check the signature of the Token. Before carrying out the requested card content management operation, the OPEN will ask the ISD to verify the Token.

Suppose the ISD has the Authorized Management privilege. This means it can perform all Card Content Management operations within its own hierarchy—without further restrictions. Further, if an SSD within the ISD's hierarchy has the Delegated Management privilege, both the ISD and this SSD can perform SAM Management within the SSD's hierarchy. However, the SSD must have a valid Token in order to perform any operation and is not able to perform Card Content Management outside its hierarchy. If a third SD has no privileges for content management, it is not allowed to perform any Card Content Management operations at all.

All three privileges and their consequences are depicted in Figure 5, which shows the same on-card hierarchy as Figure 4. Figure 5 assumes that the ISD has the Authorized Management privilege and SSD2 holds the Delegated Management privilege, while SSD1 has no special privileges. Allowed Card Content Management operations are indicated with a continuous line; forbidden operations are indicated by a red cross.

[5] The GlobalPlatform Environment ('the OPEN') is a part of the operating system of a GlobalPlatform card.

[6] Note that technically, the ability to verify a Token is dependent on a separate Security Domain privilege. This whitepaper assumes that this privilege is given to the ISD, which by default is the case.

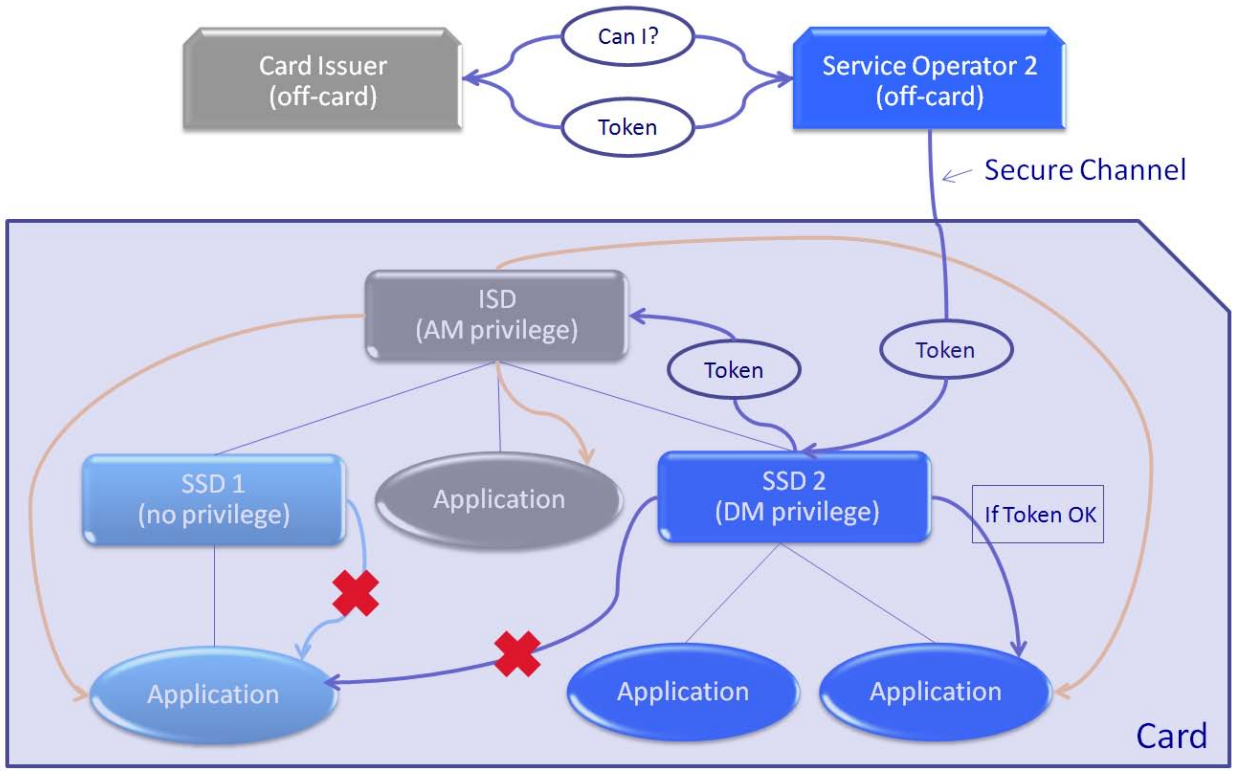

Figure 5 – Allowed Card Content Management operations for Security Domains with Authorized, Delegated and no Management privileges.

Let us apply these GlobalPlatform mechanisms to an AFC system as discussed in Section 1.1 and depicted in Figure 1. An attempt to understand an AFC system in these terms could result in the following Security Domain meanings:

- The ISD belongs to the Registrar / Security Manager at Level 4
- SSD1 belongs to one Service Operator (SO-1)
- SSD2 belongs to another Service Operator (SO-2)

This means that the Security Manager is able to load and delete applications for all Service Operators. By contrast, Service Operator 2 is able to load and delete applets in its own Security Domain, provided the Card Issuer gives its approval in the form of a Token. SO-2 could, for example, load onto the SAMs a specific loyalty applet, which contains the keys necessary to give his customers some kind of benefit. Since this application is specific to this Service Operator, the Security Manager will not want to be responsible for its management. So, after ensuring that the application will not interfere with the SAM's overall security, the Security Manager will give SO-2 permission to manage this specific application itself. On the other hand, the other Service Operator, SO-1, is not allowed to do any Card Content Management operations and is thus fully dependant on the Security Manager for this.

2.3 Application Personalization

The process of loading and installing applications on a SAM is the same, regardless the type of application. For example, there is no difference between loading a payment SAM application on a SAM in a payment terminal and loading a transit SAM application on a SAM in a validator within an AFC system. However, the personalization of SAM applications will differ from application to application. For example, the type and number of keys required in a payment SAM will be different from the keys required in an AFC SAM. These SAMs will have to keep track of different counters and will contain different pieces of sensitive data.

Fundamentally, there are three ways to do Application Personalization on a GlobalPlatform-compliant card:

- By sending a STORE DATA command to the associated Security Domain of the intended application. The Security Domain can (internally in the card) send the data in this command to the application[7].
- By communicating directly to the intended application, but using the keys of the associated Security Domain. Basically, this means that all commands are decrypted by the SD and that the SD encrypts and/or signs all responses on behalf of the application. This way, in principle the application does not need any personalization keys itself.
- By communicating directly to the intended application and using a method that is (from the viewpoint of the GP Card Specifications) proprietary. For example, for SAMs in a payment environment, it could be convenient to use EMV Scripting for SAM personalization, since this is a well-known method that is probably already implemented for the management of other cards.

For all three methods, however, the off-card entity needs a way to communicate to either the intended application or its associated Security Domain. This communication must moreover be confidential and authenticated. GlobalPlatform offers a way to achieve this in the form of so-called Secure Channels, which are discussed in the next section.

2.4 Secure Channels

2.4.1 Introduction

A Secure Channel is a temporary logical connection between an application on the card and an off-card entity. A Secure Channel can extend over several physical connections and devices, as illustrated in Figure 6. This figure shows a SAM, inserted in a Level-1 device, which belongs to Service Operator 2. However, other parties, such as the Security Manager on Level 4 or Service Operator 1, also have Security Domains on this SAM.

Each of these parties can set up a Secure Channel to his SD or the associated applications of that SD. Such a channel physically goes via the equipment of Service Operator 2. However, this does not mean that Service Provider 2 has knowledge of the content of the communication between, for example, Service Operator 1 and its Security Domain. This is because a Secure Channel offers protection of the integrity, confidentiality and authenticity of all messages communicated over the channel.

[7] Note that GlobalPlatform has specified the personalization process and messages in full detail in the 'Guide to Common Personalization' (Ref. [2]) and the accompanying 'Load and Personalization Interface Specification' (Ref [3]).

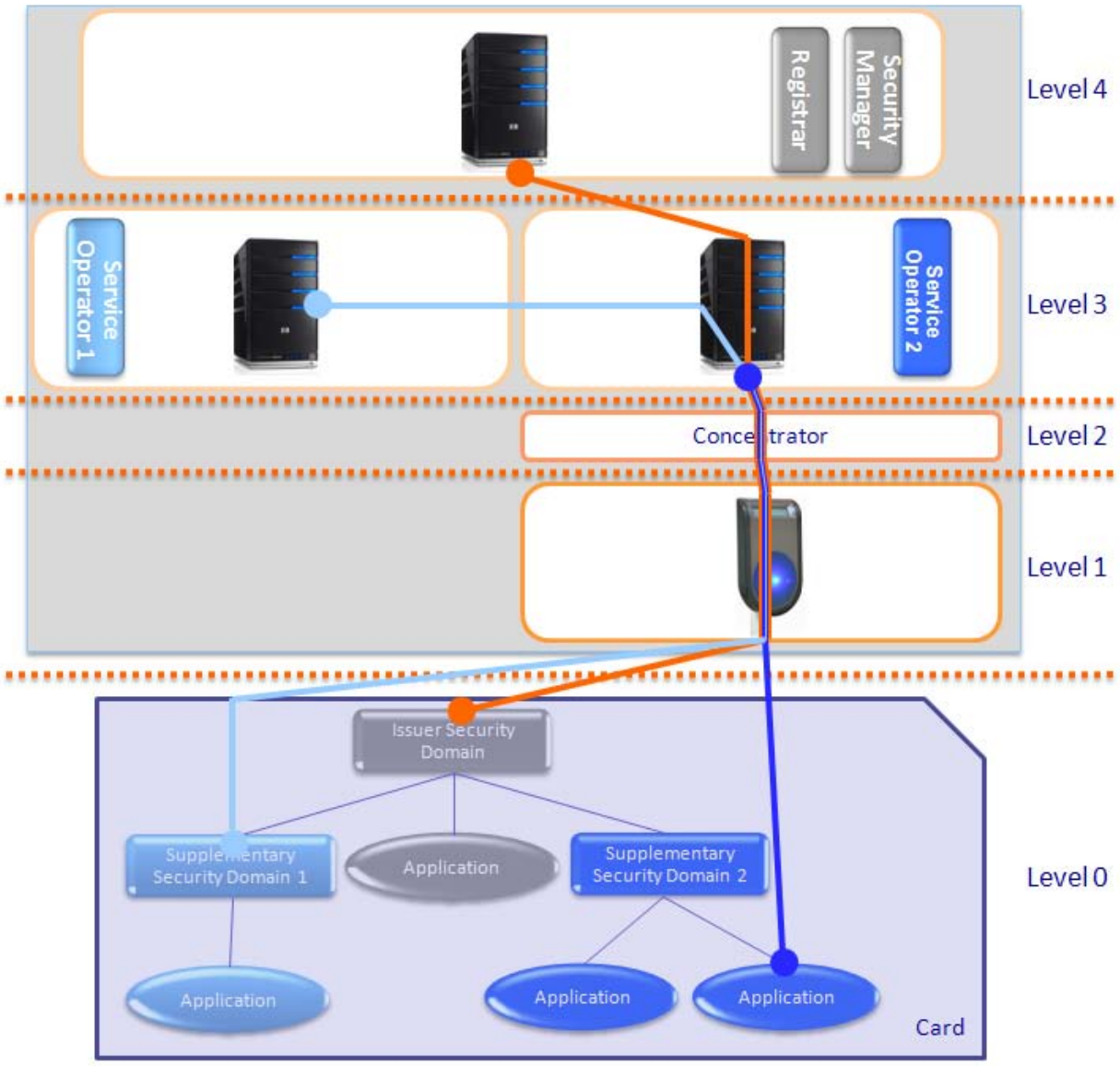

Figure 6 – Each off-card entity can have a secure channel to its own Security Domain or associated applications.

Technically speaking, this protection is brought about by the encryption of all messages and/or the addition of a Message Authentication Code (MAC). A MAC is a kind of digital signature over a message. For encryption and MAC calculation, the card and the off-card entity use session keys. These are keys that are derived from the keys in the Security Domain, but are used only for one session and are discarded afterwards. The application and the off-card entity agree on the session keys to be used for a specific session during the set up of the secure channel. It is for this reason that a Security Domain holds a set of symmetric or asymmetric keys.

2.4.2 Setting Up a Secure Channel

Setting up a Secure Channel between a smart card and an off-card entity is often done via a Mutual Authentication Procedure. In such a procedure, a few specific commands and responses are sent back and forth between the card and the off-card entity. These commands and responses contain two 'challenges,' one from each of the parties. The party that receives a challenge uses its keys to encrypt the challenge and returns it to the sender. The sender checks whether the encryption is correct, thereby checking that the other party is in possession of the correct keys and thus is trustworthy. Both parties therefore authenticate each other. This process, with some variations, is used widely in the world of smart cards, and GlobalPlatform has defined a similar process called 'explicit mode' secure channel initiation in its Card Specification.

A limitation of this process, however, is that the terminal that contains the card must be connected in near-real-time to the off-card entity. For some systems this is a big obstacle: Consider a smart card that is inserted into a mobile phone. Depending on factors outside the control of the off-card entity, the mobile phone might suddenly become unreachable because it is switched off, its battery dies, or its owner drives into a tunnel. This would unexpectedly close the Secure Channel, which probably means that all commands and responses sent thus far would need to be repeated after a new connection is established. Consider a second example: Some networks have connectivity that is far from real-time. Round-trip times of more than 15 minutes have been measured in some older networks. In Store-and-Forward based systems, round trip times can be as large as several days. This makes it difficult to set up and maintain a Secure Channel.

To enable the initiation of a Secure Channel for systems that cannot guarantee real-time connectivity, the GlobalPlatform Card Specification also defines an 'implicit mode' of Secure Channel initiation. Both the explicit and the implicit modes are schematically depicted in Figure 7. The session depicted in this figure consists of two commands, a LOAD and an INSTALL command, that together install a new application on the card. These commands are indicated by C2 and C3, and Figure 7 shows how the session proceeds in explicit mode versus implicit mode.

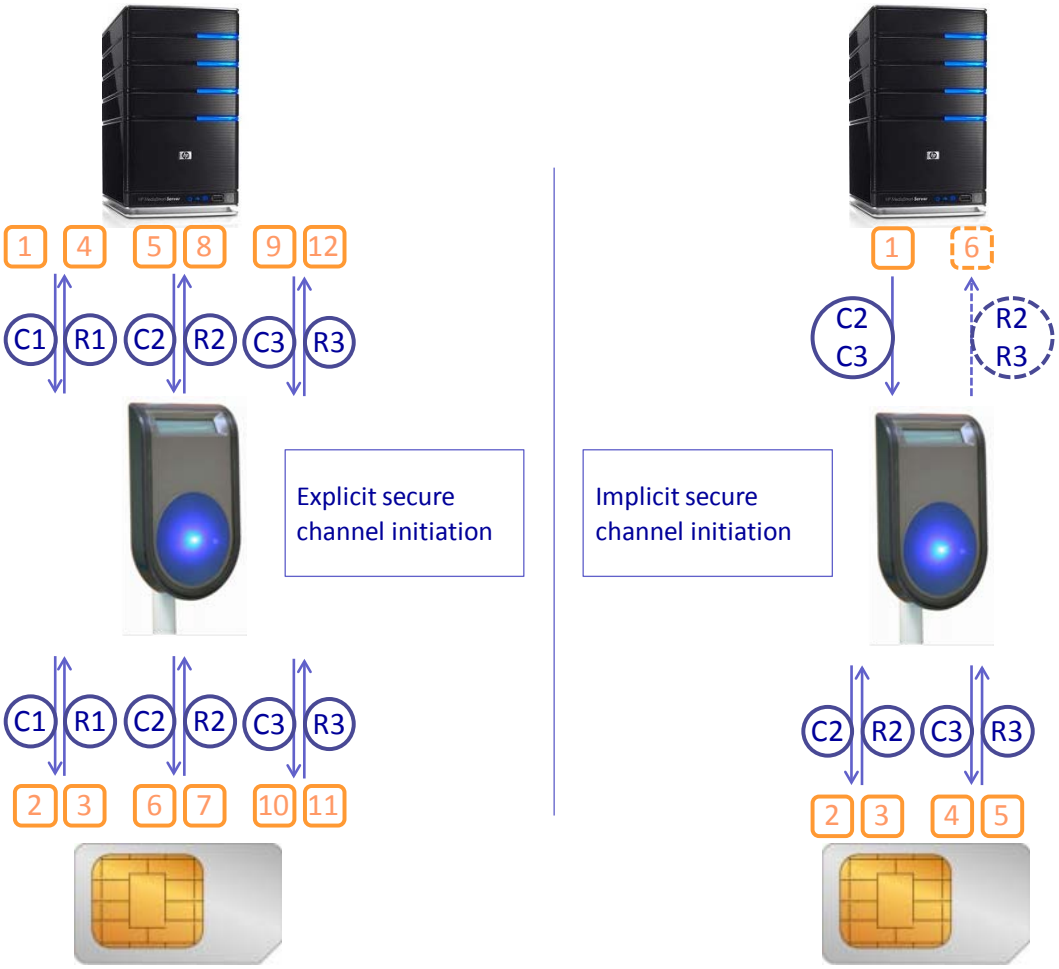

Figure 7 – Explicit and implicit secure channels: C# and R# denote individual commands and responses. The numbers in orange squares indicate the order in which commands and responses are sent.

There are two key differences between the two modes:

- *Sequence of commands and responses:* In explicit mode, each command is followed by a response. The terminal has no active role, but simply passes on each command to the card and each response to the off-card entity. Contrastingly, in implicit mode the off-card entity sends all commands destined for the card to the terminal in one script, without waiting for the card's replies. The terminal takes the individual commands in the package and sends them to the card one by one. When it has sent all commands and received the responses, it also sends the responses to the off-card in one script.
- *Presence of C1:* In explicit mode, a separate command C1 is used to set up the Secure Channel. In implicit mode, C1 is not present. Instead, the off-card entity just

sends the commands C2 and C3. From the way these commands are encrypted, the card can see that a Secure Channel was implicitly set up by the off-card entity[8].

The big advantage of the implicit mode is that no real-time connection between off-card entity and application is needed. Instead, the off-card entity just sends the script containing all commands to the terminal, without expecting an immediate answer. How long it takes before the script reaches the terminal is irrelevant, as long as the terminal is not completely unreachable for an extremely long period of time. Once the terminal has received the script, it can start sending the commands to the card.

The implicit mode also has a couple of disadvantages. First, as shown by the dotted lines in Figure 7, the off-card entity will receive the responses R2 and R3 to the commands only when the terminal is connected again. Depending on the network, this may take up to a couple of days. During that time, the off-card entity is not sure whether the card is updated or not. It therefore does not know the exact status of the card and cannot take any further action. It is important to note that this situation is a direct consequence of the nature of the network and cannot be mitigated.

A second disadvantage of implicit mode is that the terminal cannot authenticate the card before it sends the commands because no mutual authentication procedure is done and no feedback is received from the card. This may mean that the commands, including the data that they hold (such as new application code), are accidentally sent to the wrong card. However, if this were to happen, the receiving Security Domain would not be able to decrypt the commands or to check the MAC above it, since the off-card entity will have used what it perceives to be the 'wrong' session keys. Therefore, a kind of 'implicit authentication' occurs: If the receiving Security Domain is not the intended one, it will not be able to do anything with the data it receives. Moreover, the card will add a MAC to the responses R2 and R3 indicated in Figure 7. If the off-card entity checks these MACs and confirms them to be correct, it knows that the responses originated from the intended Security Domain.

2.5 GlobalPlatform and Remote SAM Management

Sections 1.5 and 1.6 identified a number of problems for remote SAM Management. The Section discusses a number of features of the GlobalPlatform Card Specification that are designed to overcome these problems. This section explicitly shows how the GlobalPlatform technology can be used for remote SAM Management.

Recall the problems identified in Section 1.5:

1. The need to be able to remotely and securely perform post-issuance SAM Management and Application Personalization

[8] Note that this means that the off-card entity has to decide which session keys must be used for the Secure Channel without receiving any input from the card. In order to do that, the off-card entity keeps track of a separate counter for each Security Domain it wants to communicate with. This counter is increased for every Secure Channel session that is set up between the off-card entity and the Security Domain. The off-card entity uses the current value of the counter for the target SD, plus the keys of that SD, to calculate the session keys that must be used for this session. Of course, the Security Domain itself has also to keep track of its own sequence counter.

2. The need to allow SAM Management and Application Personalization to be shared between multiple parties, without running the risk of interference or leakage of one party's secrets to another.

In order to solve the first problem above, GlobalPlatform introduced the concept of Security Domains on the card. As described in Section 2.1, each Security Domain acts as the on-card representative of an off-card entity. There can be multiple Security Domains on one card, allowing multiple parties to perform Card Content Management or personalization operations on the card (see Sections 2.2 and 2.3). Via an extensive system of Security Domain privileges, the roles, responsibilities and authorizations of each off-card entity can be fine-tuned. In particular, the Authorized Management and Delegated Management privileges allow a Security Domain to perform Card Content Management operations, such as adding or deleting an application. Moreover, the GlobalPlatform Environment on the card ensures a complete separation of all Security Domains.

To address the second problem above, in addition to using Security Domains, GlobalPlatform uses the concept of Secure Channels. As explained in Section 2.4, a Secure Channel is a temporary logical connection between a card and an off-card entity. In order to be able to set up a Secure Channel, a Security Domain contains a number of keys. The off-card entity is in possession of the Security Domain keys as well. Furthermore, the Secure Channel can run over a physical connection that is not owned by the party setting up the Secure Channel, which enables remote SAM Management. However, the Secure Channel nonetheless ensures that other parties are not able to read or change the communication over the Secure Channel.

Next, recall the problems identified in Section 1.6:

1. Lack of real-time connectivity
2. EEPROM wear
3. Blacklisting

To address connectivity concerns, the GlobalPlatform implicit secure channel initiation process discussed in section 2.4.2 enables the implementation of schemes in Store-and-Forward environments. Note, however, that in fully on-line environments, the explicit Secure Channel initiation process can also be used.

Minimizing EEPROM wear can be supported through careful interaction between SAMs and centralized system management. Applications must be designed to report back on their EEPROM wear, and this must be managed in the central systems so as to ensure that EEPROM usage is safe across multiple applications. However, it is also important to note that in the past two years, smart card technology has made considerable progress regarding EEPROM wear thanks to the growth of the M2M market. A requirement that smart cards guarantee an extended lifespan of ten years or more in harsh environments—such as extreme temperatures, humidity, or shocks—as led to deployment of smart cards that support anywhere from 100,000 to millions of write cycles. One can expect that SAM products will benefit from this NVM endurance improvement.

SAM blacklisting can be supported by the GlobalPlatform status management commands, which allow chips and/or applications to be blocked. Full management of lost and stolen SAMs is a function shared between central management systems and the chips themselves.

SECTION 3: USE CASES

Once we understand how the GlobalPlatform Specification can provide for remote post-issuance SAM management, it is illuminating to understand how many use cases are potentially impacted.

3.1 Primary Use Cases

The primary use cases associated with a remote SAM Management system using GlobalPlatform chips are as follows:

- *Issue SAMs*: The establishment of a secure initial state, ready for delivery and onsite installation
- *Re-issue SAMs:* The resetting of SAMs in a new initial state, ready for re-delivery and re-installation
- *Update SAM*: The post-issuance update of a SAM with a new application code, data, or keys
- *Activate SAM*: The activation of a SAM after issuance
- *Blacklist SAMs/Deactivate*: Deactivation of SAM applications after a SAM has been reported stolen
- *Report SAM Status*: This refers to the functionality responsible for reporting on SAM status for management purposes.

Each of these use cases is described in greater detail below.

3.2 Issue SAMs / Re-Issue SAMs

The goal of issuance and re-issuance is to ensure that SAMs are set in a secure issued state, so they can be distributed, installed and then activated. In such a state, the owner(s) of the initial security domain keys, application keys, or secrets stored in the SAM have approved the initial SAM configuration and security parameters, and they maintain control over this configuration and parameters.

Issuance and Re-Issuance should be conducted in a secure facility.

At issuance, the SAMs may be physically personalized and electronically provisioned with the initial application modules and any associated application keys.

GlobalPlatform specifications do not constrain the steps for SAM issuance but describe a SECURED state, with post-issuance update enabled and the appropriate Issuer Security Domain Keys installed.

It is expected that the SECURED issuance state of the SAM is operational, but deactivated state, i.e. the SAM, can be immediately operated after activation.

Records of the SAM, its unique identifier, its initial profile, and the Issuer Keyset being installed should be made available to the backend systems to support SAM remote management.

The applications and supplementary security domains loaded, installed and personalized onto the SAM are dependent on the requirements of the SAM Management scheme.

3.3 Remote SAMs update

In a SAM Management solution that supports remote post-issuance of SAMs, the SAM management system provides a wide range of new functionality:

- Updating existing applications; e.g. to introduce new application keys or secrets
- Updating existing or add new application modules; e.g. to introduce new versions of applet code
- Loading and installing new supplementary security domains; e.g. to allow new transit or payment providers to leverage the SAM capabilities
- Deleting existing applets or security domains; e.g. to remove unused applets from the SAM.

GlobalPlatform specifications allow both synchronous and asynchronous secure remote updates.

To ensure interoperability and security, SAM Management systems should provide this functionality in compliance with GlobalPlatform specifications.

3.4 Activate SAMs

SAM remote activation is the step to enable a SAM to become fully operational after it has already been issued. Typically, SAMs are not distributed in an activated state for security reasons and are only activated after they are installed in the network.
Note that remote SAM activation is normally a second step, after issuance of the applets: this is true for post-issuance update scenarios as well as issuance scenarios.

There are multiple ways to implement activation, which are determined by the requirements of the SAM management solution. However, GlobalPlatform offers two features that can support this process:

- The ability to modify the state of a SAM or an application on a GlobalPlatform SAM (e.g. to unlock an applet) in post-issuance
- The ability to set up a Secure Channel in remote issuance to ensure that only the authentic issuer of the SAM can activate the chip

3.5 Deactivate/Blacklist SAMs

Remote SAM blacklisting is the process of disabling a SAM after it has been issued in order to prevent it from being used. This is to stop issued SAMs from being used in the network, typically a security measure used to prevent the use of lost or stolen devices.

Blacklisting can be implemented in several ways, depending on the design of the solution. However, GlobalPlatform offers the following features that can support this process:

- The ability to modify the state of a SAM or an applet on a GlobalPlatform chip (e.g. to unlock an applet) in post-issuance
- The ability to set up a Secure Channel in remote issuance to ensure that only the authentic issuer of the SAM can blacklist the chip

Note that SAM blacklisting often involves the distribution of blacklists to transit network nodes, so that they can actively prevent the use of SAMs. This is necessary because attackers may attempt to use stolen SAMs in a variety of circumstances. Blacklisting is thus a combination of SAM management system and SAM behavior.

3.6 Report SAM Status

SAM remote management systems may maintain records of activity on SAMs. These systems must, at a minimum, be capable of identifying and maintaining a history of all content loaded and installed on a SAM.

The SAM management systems may be distributed between multiple parties, e.g. between the issuer of the SAM and one or more applet providers or transit operators that own independent security domains on the SAM.

At a minimum, the SAM management systems must be capable of the following:

- Uniquely identifying the SAM chips
- Uniquely identifying the SAM chip content, such as security domains and applets profiles, and maintain a history
- Uniquely identifying the SAM cryptographic keysets (to support post-issuance updates of security domains and chips)
- Recording the EEPROM usage associated with the SAM chip

Note that EEPROM usage is a chip-level function, although individual applets may be able to record, manage and report on their own use of EEPROM.

SECTION 4: CONCLUSION

With the ever-expanding market for SAMs in applications as far-reaching as financial applications, transportation systems, telecommunications, and even smart meters, it is critical that we understand the shortcomings that today's SAM implementations face.

These expanding market requirements are in one respect reducing the benefit of traditional SAMs: because traditional SAM management has occurred at the moment the SAM is issued, and because no provision has been made for remote updates, responding to changing market requirements often involves costly physical replacement of the SAM. Moving forward, the ability to remotely update SAMs post-issuance will be critical in enabling companies to evolve to changing market requirements, incorporate new industry standards, fix bugs, and more.

GlobalPlatform's Specification is ideally suited for remote post-issuance SAM management. By leveraging a Security Domain, which acts as the on-card representative for an off-card entity, it is possible to manage any particular application. Use of the Card Content Management functionality allows issuers to create, modify, and delete Security Domains and applications on the card. Application Personalization becomes possible without interfering with other parties' applications. And Secure Channels reinforce this by requiring keys and preventing information from being leaked or shared with other parties.

The result is that remote post-issuance SAM management is both possible and secure via the GlobalPlatform Specification, thus supporting an ever-growing—and no doubt set of future needs—for SAMs in today's market.

APPENDIX A: REFERENCES

Ref.	Title	Author	Version	Date
[1]	GlobalPlatform Card Specification	GlobalPlatform	2.2.1	January 2011
[2]	Guide to Common Personalization	GlobalPlatform	1.0	March 2003
[3]	Load and Personalization Interface Specification	GlobalPlatform	1.0	March 2003
[4]	EN-ISO 24014-1	CEN	First edition	2007-07-01

APPENDIX B: ABBREVIATIONS

Abbreviation	Meaning
AFC	Automated Fare Collection
DES	Data Encryption Standard
EEPROM	Electrically Erasable Programmable Read-Only Memory
ISD	Issuer Security Domain
M2M	Machine To Machine
MAC	Message Authentication Code
NVM	Non-volatile memories
SAM	Secure Access Module
SD	Security Domain
SSD	Supplemental Security Domain

APPENDIX C: DEFINITIONS

Application Personalization:

The process of correctly personalizing each application that is loaded onto a SAM. This is considered a subset of 'SAM Management,' which is defined below.

Application Provider:

The off-card entity represented by a Supplemental Security Domain (SSD).

Authorized Management:

A privilege setting specified by the GP Card Specification, Authorized Management gives a Security Domain the power to perform Card Content Management operations.

Automated Fare Collection (AFC):

For the purposes of this paper, a system in public transportation that securely facilitates the exchange of large volumes of electronic transactions.

Card Content Management:

The creation, modification, and deletion of Security Domains and the applications on the card. In order to separate the responsibilities of Card Content Management between different parties, GlobalPlatform specifies both Authorized Management and Delegated Management.

Customer Medium [Media]:

In an AFC system, that which a customer presents to a terminal for access.

Delegated Management:

A privilege setting specified by the GP Card Specification, Delegated Management indicates that the GlobalPlatform environment will ask for a 'Token' before carrying out a certain Content Management operation.

EMV:

A global standard, with contact and contactless specifications that facilitate worldwide interoperability and acceptance of integrated circuit (IC)-based payment instruments.

SAM Management:

The process of loading and installing a number of applications on a SAM. This precedes 'Application Personalization,' which is defined above.

Secure Access Module (SAM):

A smart card developed to securely store keys and perform cryptographic operations.

Secure Channel

A temporary logical connection between an application on the card and an off-card entity, a Secure Channel offers protection of the integrity, confidentiality, and authenticity of all messages communicated over the channel.

Security Domain:

Central to the GP Card Specification, a Security Domain is a special application that acts as the on-card representative for an off-card entity. It provides support for the control, security, and communication requirements of the off-card entity.

Security Manager:

In an AFC system, the Security Manager is responsible for overall security of the entire system. Residing in "Level-4" of the system, the Security Manager concentrates on the management of the SAMs in the system.

Store-and-Forward:

This method is used in networks that have intermittent connectivity. Any message sent to an unconnected device is stored and sent later when the device is connected.

Ticketing Keys:

Contained within the SAM, Ticketing Keys are used in combination with an identifier from a presented Customer Medium; the result is that the SAM is able to generate the symmetric keys that the terminal must use to communicate with a particular Customer Medium.

Token:

Required in Delegated Management, a Token is permission from the Card Issuer to carry out an operation.

Validator:

In the transit sector, this refers to the reader at the entry gate or in the bus—as opposed to a ticket vending machine or control device.

APPENDIX D: TABLE OF FIGURES