GLOBALPLATFORM™

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

GlobalPlatform Card

Remote Application Management over HTTP

Card Specification v2.2 – Amendment B

Version 1.1.1

Public Release

March 2012

Document Reference: GPC_SPE_011

This page intentionally left blank.

Contents

Figures

Tables

1 Introduction

This document defines a mechanism for an Application Provider to perform Remote Application Management (RAM) according to ETSI TS 102 226 [102 226] (i.e. loading, installation, and personalization) using the HTTP protocol (RFC 2616 [HTTP]) and PSK TLS security Over-The-Air. A third party communication network may be used if the Application Provider has no OTA capability. This third party shall not be able to access clear text of any confidential data and code belonging to the Application Provider. This document describes:

- How to open an Over-The-Air connection with a remote server, based on [HTTP] and PSK TLS security
- How commands are sent to a Security Domain
- How responses of these commands are returned to the remote server
- How this mechanism can be used over a third party communication network
- A new key type for PSK TLS keys

1.1 Audience

This amendment is intended primarily for card manufacturers and application developers developing GlobalPlatform card implementations.

It is assumed that the reader is familiar with smart cards and smart card production, and in particular familiar with the GlobalPlatform Card Specification [GPCS].

1.2 IPR Disclaimer

GlobalPlatform draws attention to the fact that claims that compliance with this specification may involve the use of a patent or other intellectual property right (collectively, “IPR”) concerning this specification may be published at https://www.globalplatform.org/specificationsipdisclaimers.asp. GlobalPlatform takes no position concerning the evidence, validity, and scope of these IPR claims.

1.3 References

Table 1-1: Normative References

Standard / Specification	Description	Ref
GlobalPlatform Card Specification	GlobalPlatform Card Specification v2.2	[GPCS]
ETSI TS 102 223	Smart Cards; Card Application Toolkit (CAT), Release 10	[102 223]
ETSI TS 102 226	Smart cards; Remote APDU structure for UICC based applications, European Telecommunications Standards Institute Project Smart Card Platform (EP SCP), Release 10	[102 226]
RFC 2616	Hypertext Transfer Protocol – HTTP/1.1	[HTTP]
RFC 2818	HTTP over TLS	[HTTPS]
RFC 2246	The TLS Protocol – Version 1.0	[TLS 1.0]
RFC 4346	The TLS Protocol – Version 1.1	[TLS 1.1]
RFC 5246	The TLS Protocol – Version 1.2	[TLS 1.2]
RFC 4366	Transport Layer Security (TLS) Extensions	[TLS Extns]
RFC 4279	Pre-Shared Key Cipher Suites for Transport Layer Security (TLS)	[PSK TLS]
RFC 5487	Pre-Shared Key Cipher Suites for TLS with SHA-256/384	[PSK 256]
RFC 4785	Pre-Shared Key (PSK) Cipher suites with NULL Encryption for Transport Layer Security (TLS)	[PSK NULL]
OMA SCWS	Smartcard Web Server V1.1, Open Mobile Alliance™	[OMA SCWS]
ISO/IEC 8825-1	Information technology — ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)	[8825-1]

1.4 Terminology and Definitions

Technical terms used in this document are defined in [GPCS].

1.5 Abbreviations and Notations

Table 1-2: Abbreviations

Abbreviation	Meaning
AID	Application Identifier
AP	Application Provider
API	Application Programming Interface
APDU	Application Protocol Data Unit
APSD	Security Domain of the Application Provider
BIP	Bearer Independent Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
OTA	Over-The-Air
OTASD	Security Domain of the Over-The-Air platform operator
RAM	Remote Applet Management
RID	Resource Identifier.
PIX	Proprietary Identifier extension
PSK TLS	Pre-Shared Key TLS
SCWS	Smart Card Web Server
TAR	Toolkit Application Reference
TLS	Transport Layer Security
URI	Uniform Resource Identifier

1.6 Revision History

Table 1-3: Revision History

Date	Version	List of Modifications
Nov 2008	1.0	Initial Release
June 2009	1.1	**HTTP Header modification** The "From" and "User-Agent" header fields are specified in the HTTP protocol ([HTTP]), but the content defined in version 1.0 for those headers were not compliant. o Prefixed proprietary headers The good practice of [HTTP] for custom headers is to prefix them by “X-“. All header names defined in this document (previously named Resume, Next-URI, Script-Status and Targeted-Application) are now prefixed by “X-Admin-“. o "From" Header Field [HTTP] specifies that the "From" request-header field, if given, shall contain an Internet e-mail address for the human user who controls the requesting user agent. Version 1.0 used the "From" header field in the HTTP post request to put the "Agent-ID" (identifier of the card). A custom “X-Admin-From” header field is now defined. o "User-Agent" Header Field [HTTP] specifies that the "User-Agent" request-header field contains information about the user agent originating the request. This is for statistical purposes, the tracing of protocol violations, and automated recognition of user agents for the sake of tailoring responses to avoid particular user agent limitations. In version 1.0, the "User-Agent" was used in the HTTP post request and in HTTP post response to identify the RAM over HTTP protocol. The "X-Admin-Protocol" header that will be used for the request and the response with the same value “globalplatform-remote-admin/1.0” is now defined. **Content-Type Value** [HTTP] only allows one slash in the value. Version 1.0 was inconsistent with this rule. A compliant Content-Type for POST request and response is now defined. o POST request: Content-Type: application/vnd.globalplatform.card-content-mgt-response;version=1.0 CRLF o POST response: Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0 CRLF **AID coding rules of the AID in the "X-Admin-Targeted-Application" header field is specified.** *(continues)*

Date	Version	List of Modifications
	1.1 *(continued)*	**Agent-ID definition** The value of the "Agent-ID" field is defined in the administration session triggering message or by the Security Domain parameters. In practice the remote admin server usually uses this field to identify the card instance (for example to keep an image of the card content) and not only the requesting application. **Support of TLS protocol v1.1 and v1.2.** **Support of Pre-Shared Key Cipher Suites for TLS with SHA-256.** **Connection Parameters to configure the point to point TCP connection** **Retry Policy** Report mechanism has been added to have a status on the HTTP Administration session request. o Report Failure Parameters. These parameters allow an application to request the system to send a report through another communication channel than the one defined in this document. o HTTPReportListener Interface. This interface is added to notify the applet whether the requested HTTPAdministrationSession has completed successfully. **HTTPAdministration Interface** The object implementing this interface shall belong to the JCRE to have access to any object. This avoids requesting Global Arrays that are not always available.
March 2012	1.1.1	• Added new section “Secure Channel Protocol Usage”. • Clarified the meaning of “administration session”, the meaning of “communication breakdown”, and when the Retry Policy shall be used. See sections “Administration Session Start” and “Administration Session End”. • Added precisions on the usage of PSK TLS keys. See section “Establishing a Secure Communication Channel”. • Clarified the meaning of “command session”, and its relation with Secure Channel tunneling. • Replaced section “PSK TLS key format” with section “Loading PSK TLS Keys” describing a suitable format for both PUT KEY and STORE DATA commands.

2 Use Cases and Requirements

OMA SCWS [OMA SCWS] defines a mechanism for securely uploading static SCWS content (HTML pages) from a remote entity to the card. It also defines a mechanism to map applications that generate dynamic SCWS content to a URL. These management actions use HTTPS for security.

This document specifies an extension to the SCWS mechanisms that allow loading and installation of applications via the same HTTPS channel. This enables the following additional use case:

- loading of static SCWS content as defined in [OMA SCWS], plus
- loading of dynamic SCWS content generating applications, plus
- mapping these applications to a SCWS URL as defined in [OMA SCWS],

within one session, all using the same HTTPS channel.

The mechanism defined in this document handles the Card Content Management as defined in [GPCS] and can also be used independently of the SCWS.

This document proposes a specification addendum to support the following requirements:

- It shall be possible to open a HTTPS connection between an Application Provider and its Security Domain (APSD).
- In this connection, the APSD acts as an HTTPS client, and the AP acts as an HTTPS server.
- This connection is used to send remote APDU format string as specified in [102 226], to the APSD. It may also be used to send other content types, handled by another application.
- The underlying transport protocol of this connection is out of scope of this specification.
- An intermediary OTA SD may be used.
- To ensure confidentiality, the targeted security domain may apply additional security to the remote APDU format string.

3 Specification Amendments

3.1 PSK TLS Key Type

Table 11-16 of [GPCS] is replaced by Table 3-1 in order to introduce PSK TLS key type:

Table 3-1: Key Type Coding

Value	Meaning
'00'-'7F'	Reserved for private use
'80'	DES – mode (ECB/CBC) implicitly known
'81'	Reserved (Triple DES)
'82'	Triple DES in CBC mode
'83'	DES in ECB mode
'84'	DES in CBC mode
'85'	Pre-Shared Key for Transport Layer Security
'86'-'8F'	RFU (symmetric algorithms)
'90'	HMAC-SHA1 – length of HMAC is implicitly known
'91'	HMAC-SHA1-160 – length of HMAC is 160 bits
'93'-'9F'	RFU (symmetric algorithms)
'A0'	RSA Public Key - public exponent e component (clear text)
'A1'	RSA Public Key - modulus N component (clear text)
'A2'	RSA Private Key - modulus N component
'A3'	RSA Private Key - private exponent d component
'A4'	RSA Private Key - Chinese Remainder P component
'A5'	RSA Private Key - Chinese Remainder Q component
'A6'	RSA Private Key - Chinese Remainder PQ component (q^{-1} mod p)
'A7'	RSA Private Key - Chinese Remainder DP1 component (d mod (p-1))
'A8'	RSA Private Key - Chinese Remainder DQ1 component (d mod (q-1))
'A9'-'FE'	RFU (asymmetric algorithms)
'FF'	Extended format

3.2 Security Domain and Remote Administration Server

A Security Domain is responsible for establishing a connection with an off-card entity, called Remote Administration Server, in order to start (or resume) an administration session. Such an administration session is used to receive a set of APDU commands from a Remote Administration Server and has the following characteristics:

- It is handled by the Security Domain.
- The physical link used for this connection is beyond the scope of the present document.
- It uses the industry standard security layer TLS protocol in order to secure communications (see RFC 2246 [TLS 1.0], RFC 4346 [TLS 1.1], and RFC 5246 [TLS 1.2]) and HTTPS (see RFC 2818 [HTTPS]). This specification references the TLS protocol as the GlobalPlatform Secure Channel Protocol '81' (SCP81). See section 3.3.2 for supported cipher suites.

This Security Domain:

- acts as an HTTP Client and is in charge of managing connection establishment to the Remote Administration Server
- is able to encapsulate and transparently transport any remote APDU format string (as defined in [102 226])
- is responsible for retry and reconnection management in case of communication breakdown
- can be triggered either by external events or by internal events (internally generated by the card) to initiate a connection to the Remote Administration Server
- according to the architecture decomposition of [OMA SCWS], the SD implements the SCWS (or card) administration agent

The Remote Administration Server is an HTTP server.

3.2.1 Secure Communication Configuration

For SCP81 the "i" parameter is formed as a bit map on one byte as defined in Table 3-2. A security domain may support one or multiple TLS versions.

Table 3-2: Values of Parameter "i"

b8	b7	b6	b5	b4	b3	b2	b1	Description
							1	[TLS 1.0] supported
						1		[TLS 1.1] supported
					1			[TLS 1.2] supported
	X	X	X	X				RFU (set to 0)
X								Reserved

Note: *"i" is a sub identifier within an object identifier, and bit b8 is reserved for use in the structure of the object identifier according to ISO/IEC 8825-1 [8825-1].*

3.3 Administration Protocol

3.3.1 Administration Session Start

An administration session starts when a Security Domain is triggered and the communication channel with the Remote Administration Server is set up.

The triggering of the Security Domain may result from:

- an external event, for example a message sent by a remote entity or by an off-card entity,
- an internal event, for example a timer,
- an application using a dedicated API method (see Chapter 4).

The Security Domain shall set up a communication channel with the Remote Administration Server, and then establish secure communications (see section 3.3.2) using its own PSK TLS key. It is assumed that the Security Domain knows all parameters needed to establish a connection and to handle its security. These parameters can be parameters of the triggering message or parameters of the Security Domain itself. See section 3.7.

In the remainder of this document, a communication breakdown means that a failure occurred over the communication channel, in which case the Retry Policy described in section 3.5 shall be used. A failure of the TLS protocol shall always be considered as a fatal error that shall terminate the administration session.

If an administration session triggering message is received while one administration session is being processed, the security domain shall stack this new administration session triggering until the end of the current one.

3.3.2 Establishing a Secure Communication Channel

Once the communication channel has been set up, the Security Domain shall establish a secure communication channel with the Remote Administration Server.

The Security Domain processes the PSK TLS over this communication channel to enable mutual authentication, integrity, and possibly confidentiality, using one of the following cipher suites:

For TLS 1.0 and TLS 1.1:

- TLS_PSK_WITH_3DES_EDE_CBC_SHA, as defined in RFC 4279 [PSK TLS]
- TLS_PSK_WITH_AES_128_CBC_SHA, as defined in RFC 4279 [PSK TLS]
- TLS_PSK_WITH_NULL_SHA, as defined in RFC 4785 [PSK NULL]

For TLS 1.2:

- TLS_PSK_WITH_AES_128_CBC_SHA256, as defined in RFC 5487 [PSK 256]
- TLS_PSK_WITH_NULL_SHA256, as defined in RFC 5487 [PSK 256]

The Key Version Number (KVN) and Key Identifier (KID) of the PSK TLS key, and the PSK Identity string that shall be used to initiate the PSK TLS session are read as part of Administration Session Security Parameters (see section 3.7).

The PSK TLS (SCP '81') key set consists of two kinds of keys: a PSK TLS key and a DEK (decryption/encryption) key. The DEK key may be used to decrypt or encrypt sensitive data using the `SecureChannel` interface (see section 3.3.3.1). It has the same KVN as the PSK TLS key, and a KID incremented by one. For example, when a PSK TLS session was opened using a PSK TLS key having a KVN of '40' and a KID of '01', then the DEK key that shall be used in this session is identified by a KVN of '40' and a KID of '02'.

The loading of new PSK TLS keys is described in section 3.8.

The Remote Administration Server shall support the Maximum Fragment Length Negotiation for TLS as defined in RFC 4366 [TLS Extns] and shall accept requests for a maximum fragment length down to 512 bytes. The Security Domain may use the Maximum Fragment Length Negotiation to request a maximum fragment length smaller than the default value of 16 Kbytes.

3.3.3 Fetching a Remote APDU Format String

Once the PSK TLS communication channel is established the Security Domain shall send an HTTP POST request in order to get a remote APDU format string. The targeted Security Domain shall verify the protection (if any) of each APDU read from the remote APDU format string as described in section 3.3.3.2.

When receiving the HTTP POST request from the Security Domain, the Remote Administration Server shall send an HTTP response which encapsulates a remote APDU format string dedicated to a Security Domain. This dedicated Security Domain is defined as follows:

- If no "`X-Admin-Targeted-Application`" header is present in the HTTP POST response, then the targeted Security Domain is the one which provides the PSK TLS security of the communication channel.
- If a "`X-Admin-Targeted-Application`" header is present in the HTTP POST response, the header value shall be read as the instance AID of the targeted Security Domain.

If requested, the Security Domain shall submit the remote APDU format string response in a new POST request to the Remote Administration Server over the PSK TLS secure channel.

The Remote Administration Server shall send the next remote APDU format string to the Security Domain over the PSK TLS channel, or send a final response requesting the end of the remote administration session in the POST response.

If the Security Domain receives a final response from the Remote Administration Server, it shall close the PSK TLS channel, and then close the underlying communication channel.

3.3.3.1 Usage of the `SecureChannel` Interface

If the targeted Security Domain is handling the PSK TLS (SCP '81') secure channel session, the security of the script is successful.

- `SecureChannel.getSecurityLevel()` is used to verify the secure channel security level
- `SecureChannel.processSecurity()` throws an ISO Exception with status code ISO7816.SW_INS_NOT_SUPPORTED.
- the `SecureChannel.unwrap()` method may be called and will not return an error, but will not perform any additional secure messaging processing.
- as the PSK TLS response will be secured implicitly according the PSK TLS security level, the `SecureChannel.wrap()` method may be called and will not return an error, but will not do any processing on the outgoing response message.
- `SecureChannel.encrypt()` and `SecureChannel.decrypt()` use the key having the same Key Version Number (KVN) and a Key Identifier (KID) incremented by one with respect to the key described in the Security Parameters of the current Administration Session (see section 3.7.3). The algorithm used is identified by the algorithm (3DES or AES) associated with the key. The CBC mode is always used (with null ICV).
- The security level reflects the PSK TLS cipher suite used during the session ;
  - TLS_PSK_WITH_3DES_EDE_CBC_SHA: AUTHENTICATED | C_MAC | C_DECRYPTION| R_MAC| R_ENCRYPTION.
  - TLS_PSK_WITH_AES_128_CBC_SHA(256): AUTHENTICATED | C_MAC | C_DECRYPTION| R_MAC| R_ENCRYPTION.
  - TLS_PSK_WITH_NULL_SHA(256): AUTHENTICATED | C_MAC | R_MAC.
  - SCP '81' not set up: NO_SECURITY_LEVEL.
  - `SecureChannel.resetSecurity()` throws an ISO Exception with status code ISO7816.SW_CONDITION_OF_USE_NOT_SATISFIED.

If the targeted security Domain is not handling the PSK TLS session, it shall apply its own secure channel to check the security of each command received in the remote APDU format string.

- In this case the `SecureChannel.processSecurity()` method is used to setup the secure channel session.
- `SecureChannel.unwrap()` secures each APDU command string.
- The security Domain shall explicitly wrap each command response of the remote APDU format string using its secure channel service `SecureChannel.wrap(byte[],short,short)`.

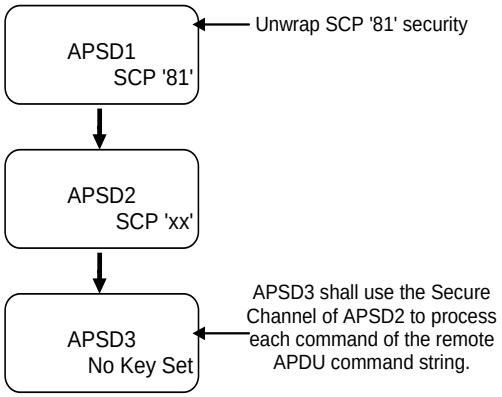
3.3.3.2 Secure Channel Protocol Usage

When the targeted Security Domain is the one unwrapping the remote APDU command string, then the remote APDU command string is trusted and processed. Any attempt to initiate a Secure Channel session (according to another Secure Channel Protocol) within the remote APDU command string shall be rejected.

When the targeted Security Domain is not the one unwrapping the remote APDU command string, then the following rules apply:

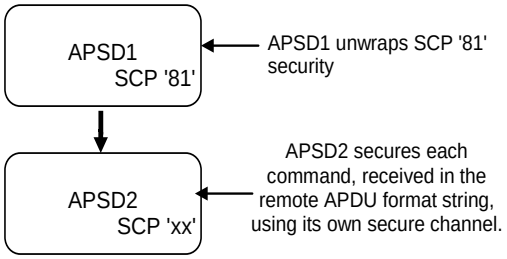
1. If the targeted Security Domain does not have any Secure Channel Key Set:
   a. If the Security Domain unwrapping the remote APDU command string is the Security Domain associated with the targeted Security Domain, then the remote APDU command string is trusted and processed by the targeted Security Domain.
   b. If the Security Domain unwrapping the remote APDU command string is not the Security Domain associated with the targeted Security Domain, then the remote APDU command string is not trusted and the targeted Security Domain shall request its associated Security Domain to verify the protection (SCP 'xx' with 'xx' in the range '01' to '7F') of the APDU commands received in the remote APDU command string.

Figure 4-1: Targeted Security Domain without any Secure Channel Key Set



2. If the targeted Security Domain has at least one complete Key Set for a Secure Channel Protocol, the remote APDU command string is not trusted, and:
   a. If the targeted Security Domain does not support SCP '81', then it shall use its own Secure Channel (SCP 'xx' with 'xx' in the range '01' to '7F') to verify the protection of the APDU commands received in the remote APDU command string.
   b. If the targeted Security Domain supports both SCP '81' and another Secure Channel Protocol (SCP 'xx' with 'xx' in the range '01' to '7F'), it shall use that other protocol to verify the protection of the APDU commands received within the remote APDU command string. Otherwise, the protection cannot be successfully verified and the APDU commands shall be rejected.

Figure 4-2: Targeted Security Domain without SCP '81' Capability



3.3.4 Administration Session End

An Administration Session ends when all the HTTP messages sent by the Remote Administration Server were received and processed. The Remote Administration Server explicitly ends the session by sending an HTTP response with no "`X-Admin-Next-URI`" header and with an empty body (see section 3.4.2). The Security Domain shall subsequently close the administration session and the communication channel.

In addition, an Administration Session will end upon failure of the Secure Communication Channel (see section 3.3.2).

If a communication breakdown occurs at some point during the administration session, the Security Domain shall try to resume the administration session as described in section 3.5.

3.4 Command Format

3.4.1 HTTP POST Request of Security Domain

The POST request is used by the Security Domain to fetch remote APDU format strings and to transmit response strings.

The POST request shall have the following format:

```
POST <URI> HTTP/1.1 CRLF
Host: <Administration Host> CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: <Agent ID> CRLF
[Content-Type: application/vnd.globalplatform.card-content-mgt-
response;version=1.0 CRLF]
[Content-Length: xxxx CRLF] or [Transfer-Encoding: chunked CRLF]
[X-Admin-Script-Status: <script-status> CRLF]
[X-Admin-Resume: true]
CRLF
[body-with-previous-response-string]
```

- The URI, the "`X-Admin-From`" value and the "`Host`" value to be used are defined in the administration session triggering message or by Security Domain parameters.
- The first POST request of a new administration session shall not contain any optional header field and no body.
- The "`X-Admin-Script-Status`" header value is used to return the delivery status of the previous remote APDU format string. The possible values are defined as follows:
  - "`ok`": this value is used if the previous remote APDU format string has been successfully delivered. A response string shall be sent.
  - "`unknown-application`": this value is used if the application targeted by the previous remote APDU format string could not be found. No response string shall be sent.
  - "`not-a-security-domain`": this value is used if the application targeted by the previous remote APDU format string is not a Security Domain. No response string shall be sent.
  - "`security-error`": this value is used if the Security Domain targeted by the previous secured remote APDU format string is not able to check its security. No response string shall be sent.
- If the administration session is resumed from a previous interrupted session, the Security Domain shall use the "`X-Admin-Resume`" header with the value “true” in the first POST request of the resumed session. The "`X-Admin-Resume`" header shall not be used in the following POST requests. See section 3.5, Retry Policy.
- If a response string is to be sent, the Security Domain shall use:
  - "`Content-Type`" header with the value "`application/vnd.globalplatform.card-content-mgt-response;version=1.0`"
  - "`Content-Length`" header with the exact length of the body in bytes or "`Transfer-Encoding`" header with the value "`chunked`".
  - A body with the complete response string of the previous remote APDU format string, in binary format. The chunked Transfer-Encoding may be used. Expanded Remote response structure format as defined in [102 226] shall be used.

3.4.2 HTTP POST Response of Remote Administration Server

The POST response is used by the Remote Administration Server to transmit the next remote APDU format string to a Security Domain and possibly to inform about the next URI that must be used to request the following admin command.

The POST response shall have the following format:

```
HTTP/1.1 200 OK CRLF [or HTTP/1.1 204 No Content CRLF]
X-Admin-Protocol: globalplatform-remote-admin/1.0  CRLF
[X-Admin-Next-URI: <next-URI> CRLF]
[Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0
CRLF]
[X-Admin-Targeted-Application: <security-domain-AID> CRLF]
[Content-Length: xxxx CRLF] or [Transfer-Encoding: chunked CRLF]
CRLF
[body-with-command-string]
```

- The Remote Administration Server shall use a successful status (200 OK) if the response contains a body else it shall use the 204 (No Content) if no body is sent.
- If Content-Type and X-Admin-Protocol are inconsistent, the session shall be closed.
- If the Remote Administration Server was not able to process the last HTTP POST request (unexpected URI, invalid header…) then it shall use an error status. The Security Domain shall close the administration session.
- If a "`X-Admin-Next-URI`" header is present in the response, the Security Domain shall use the given URI in the next POST request. The "`X-Admin-Next-URI`" header may be replaced by the "`SCWS-Next-URI`" header without any functional modification.
- If no "`X-Admin-Next-URI`" header is present in the response and if the body is empty, this is the final response of the Remote Administration Server and the administration session shall be closed.
- If no "`X-Admin-Next-URI`" header is present in the response and if the body is not empty, the remote APDU format string shall be handled as described above, but no response string shall be returned to the Remote Administration Server, and the administration session shall be closed.
- If the Remote Administration Server has remaining remote APDU format string to forward to a Security Domain it shall use a body with:
  - "`Content-Type`" header with the value "`application/vnd.globalplatform.card-content-mgt;version=1.0`"
  - "`Content-Length`" header with the exact length of the body in bytes or "`Transfer-Encoding`" header with the value "`chunked`".
  - A body with a remote APDU format string in binary format to be forwarded to a Security Domain. The chunked Transfer-Encoding may be used. Expanded Remote command structure format as defined in [102 226] shall be used.
- Optionally, "`X-Admin-Targeted-Application`" header field with the representation of the targeted Security Domain AID as header value, if the targeted Security Domain is not the one in charge of the PSK TLS security.
  - The AID shall be coded as follows; //aid/<RID>/<PIX>, where <RID> and <PIX> are the two components of the application AID. All the bytes of the RID and PIX including any leading 0 byte values shall be represented in the character string notation.

- A RID byte string is 5 bytes in length. Its character string equivalent shall be exactly 10 characters in length.
- A PIX byte string can be from 0 to 11 bytes in length. A PIX byte string of N bytes in length shall have an equivalent character string representation of exactly 2*N characters in length.

3.4.3 Interworking with the SCWS

If RAM over HTTP on a card is used together with SCWS administration as defined in [OMA SCWS], the following additional provisions shall apply:

- The PSK TLS secure channel to be used for RAM over HTTP may also be opened as defined in [OMA SCWS].
- Independent of how the PSK TLS channel was opened, sequential switching between RAM over HTTP and SCWS administration shall be supported as defined in the next two bullet points.
- To switch from SCWS management to RAM over HTTP, the empty response that ends SCWS management shall be replaced by a response from the Remote Administration Server having content as defined in this document. This shall start an administration session as defined in this document.
- To switch from RAM over HTTP to SCWS management, the final response from the Remote Administration Server defined in this document shall be replaced by a response from the SCWS Remote Administration Server having content as defined for the SCWS. This shall end an administration session as defined in this document.

3.5 Retry Policy

As soon as an administration session has been triggered and accepted by the Security Domain, it is responsible for the connection to the Remote Administration Server and for the accomplishment of the session.

This means that if a communication error occurs during the processing of the administration protocol, the Security Domain should try to reconnect according to a card issuer specific retry policy.

The retry policy may include the following:

- An end condition (e.g. number of retries) to be used to avoid network congestion by stale or inconsistent remote administration request.
- A time or counter or an event based retry policy if the connection attempts fails (like network congestion).

If the PSK TLS session establishment fails for security/authorization reason the administration session shall be immediately discarded.

If a communication breakdown occurs after valid requests have been exchanged between the Security Domain and the Remote Administration Server, the Security Domain shall always use the resume mode (see section 3.4.1).

The overall behavior shall be based on the following rules:

- The Security Domain will make several attempts for resuming the administration session. The waiting period between two attempts and the maximum number of attempts is specified by the retry policy. See section 3.7.4.
- If the communication is re-established and the Security Domain had received a complete script before the breakdown occurred, the Security Domain will process the script and try to resume the HTTP dialog with the next HTTP request with the "`X-Admin-Resume: true`" header present.
- If the communication is re-established after a breakdown at any other point in time, the Security Domain will try to resume the HTTP dialog by repeating the last HTTP request with the "`X-Admin-Resume: true`" header present.
- In both cases, the Remote Administration Server may continue the administration session from the given URL or restart it from its beginning.
- At the opposite, if a maximum number of attempts have been reached the administration session request is then abandoned.

If several administration requests are registered and need a retry, the Security Domain should handle these retries independently of each other's (e.g. not block the other retry attempts if the current one is not successful).

3.6 Command Session

A command session consists of one or several remote APDU format string(s) for a single targeted Security Domain. An administration session may transport several command sessions for several targeted Security Domains.

At the beginning of the Administration Session, a command session is implicitly started, targeting the triggered Security Domain. Subsequently, a new command session shall be started if the Security Domain targeted by the current HTTP POST response is not the same as the one targeted by the previous HTTP POST response. That means:

- the value of the header "`X-Admin-Targeted-Application`" has changed;
- or the value of the header "`Content-Type`" has changed;
- or the previous HTTP POST response contains a "`X-Admin-Targeted-Application`" header while the current one does not contain this header;
- or the current HTTP POST response contains a "`X-Admin-Targeted-Application`" header while the previous one does not contain this header.

During the command session, all APDU commands received in the APDU format string are forwarded to and processed by the targeted Security Domain.

If the targeted Security Domain is not the one unwrapping the remote APDU command string, and uses its own Secure Channel (SCP 'xx' with 'xx' in the range '01' to '7F') to verify the protection of the APDU commands received in the remote APDU command string, then that Secure Channel shall be terminated upon, and only upon, one of the following events: error detected in the protection of an APDU command, establishment of a new Secure Channel session, or end of the command session (as described hereafter).

A command session shall be closed if one of the following conditions occurs:

- The communication channel is closed.
- A new command session is started for another Targeted Application.
- A Card Reset occurs.

The internal notifications needed to implement the mechanisms described above are out of the scope of this document.

3.7 Administration Session Triggering Parameters

When starting an administration session, the triggered Security Domain shall use parameters to set up the connection, the security and the content of the first request. These parameters shall be retrieved from the message leading to this administration session ("administration session triggering parameters"). If parameters are missing in the triggering message, they shall be completed with the parameters, if available, of the triggered Security Domain. If parameters are still missing, they shall be completed with the parameters, if available, of the Issuer Security Domain. The parameters of the Issuer Security Domain, if available, are defined by the Card Issuer.

The administration session triggering parameters are TLV structured values. The following table identifies the possible tags for use in the administration session triggering message:

Table 3-3: Administration Session Triggering Parameters

Tag	Length	Name							Presence
'81'	0-n	Administration session triggering parameters							Mandatory
		Tag	**Length**	**Name**					
		'83'	1-n	Security Domain parameters value					Optional
				Tag	**Length**	**Name**			
				'84'	1-n	Connection parameters tag			Optional
				'85'	1-n	Security parameters			Optional
				'86'	1-n	Retry policy parameters			Optional
				'89'	1-n	HTTP POST parameters value			Optional
						Tag	**Length**	**Name**	
						'8A'	1-n	Administration Host parameter	Optional
						'8B'	1-n	Agent ID parameter	Optional
						‘8C'	1-n	Administration URI parameter	Optional

If a message containing the administration session triggering parameters is sent to the Security Domain, it may be sent to the TAR that processes the Expanded Remote Application data format according to [102 226].

3.7.1 TLV: Security Domain Administration Session Parameters

The administration parameters may be set, using tag '85', during Security Domain installation, using tag '85' inside the application specific parameters, or during Security Domain personalization using tag '85' with the Store Data command in TLV mode.

Note that tag '85' is a contextual tag and has no relation with the tag '85' defined in Table 3-3.

The Issuer Security Domain owns the default Administration session parameters.

Table 3-4: TLV Security Domain Administration Session Parameters

Tag	Length	Name					Presence
'85'	1-n	Security Domain Administration Session Parameters					Optional
		Tag	**Length**	**Name**			
		'84'	1-n	Connection parameters tag			Optional
		'85'	1-n	Security parameters value			Optional
		'86'	1-n	Retry policy parameters value			Optional
		'89'	1-n	HTTP POST parameters value			Optional
				Tag	**Length**	**Name**	
				'8A'	1-n	Administration Host parameter	Optional
				'8B'	1-n	Agent ID parameter	Optional
				'8C'	1-n	Administration URI parameter	Optional

3.7.2 Connection Parameters

The connection parameters TLV embed all the needed parameters to establish a point to point TCP connection between the Administration Agent and the Remote administration server.

Table 3-5: Connection Parameters

Description	Length
Connection parameters tag	1
Length (A)	1 or 2
Set of any comprehension TLV needed to open the TCP connection.	A

This parameter is typically used, if the connection between the Administration Agent and the Remote Administration Server is done over BIP, once merged with the configuration resource, the data shall contain all needed COMPREHENSION-TLV data objects that are defined for OPEN CHANNEL in ETSI TS 102 223 [102 223].

3.7.3 Security Parameters

The security parameters are defined as follows:

Table 3-6: Security Parameters

Description	Length
Security parameters tag	1
Length	1, 2 or 3
Length of PSK Identity	1
PSK Identity	1-n
Length of Key version/Key identifier	1
Key version/Key identifier	2

- PSK Identity is a string defined in [PSK TLS]. The administration agent shall support a PSK Identity length of at least 32 bytes.
- Key version and Key Identifier identify the PSK TLS key to be used for PSK TLS exchanges. It is as follows:
  - 1st byte is the key version number of the key.
  - 2nd byte is the key identifier of the key.

3.7.4 Retry Policy Parameters

The retry policy parameters are defined as follows:

Table 3-7: Retry Policy Parameters

Description	Length	Presence
Retry policy parameters tag	1	Mandatory
Length (2+5+A)	1	Mandatory
Retry counter	2	Mandatory
Retry waiting delay	5	Mandatory
Retry report failure	Var.	Optional

- Retry counter: value of the retry counter used by the retry policy
- Retry waiting delay: definition of the time to wait between two retries. This parameter is in the same format as the “timer” comprehension TLV defined in [102 223].
- Retry Report Failure is typically used to send a message using another communication channel in case of an abort of an administration request.

3.7.5 Administration Host Parameter

This TLV defines the "`Host`" header value to be used by the Security Domain when sending a POST request. It is defined as follows:

Table 3-8: Host Parameter

Description	Length
Administration Host parameter tag	1
Length	1, 2 or 3
"`Host`" header value	1-n

3.7.6 Agent Id Parameter

This TLV defines the "`X-Admin-From`" header value to be used by the Remote Administration Server to identify the requester when receiving a POST request. It is defined as follows:

Table 3-9: Agent Id Parameter

Description	Length
Agent Id parameter tag	1
Length	1, 2 or 3
"`X-Admin-From`" header value	1-n

3.7.7 Administration URI Parameter

This TLV defines the URI value to be used by the Security Domain when sending the first POST request of the administration session. It is defined as follows:

Table 3-10: Administration URI Parameter

Description	Length
Administration URI parameter tag	1
Length	1, 2 or 3
URI value	1-n

3.8 Loading PSK TLS Keys

PSK TLS keys shall be loaded using either a PUT KEY command or the STORE DATA command. The capability to load a PSK TLS key using the STORE DATA command remains optional.

When sending PSK TLS keys, the following rules apply:

- PSK TLS keys shall be sent encrypted:
  - Before ciphering a PSK TLS key, the PSK TLS key shall be padded with as few (if any) random bytes to fill the last block required by the ciphering algorithm.
  - The padded PSK TLS key shall be ciphered with the Data Encryption Key (DEK) and associated encryption algorithm as defined by the Secure Channel Protocol used to protect the PUT KEY command.
- A key check value shall be computed as the three most significant bytes of the SHA-1 digest of the PSK TLS Key.

3.8.1 PSK TLS Key Loading with the PUT KEY Command

The key data field of a PSK TLS key shall be coded as follows:

Table 3-11: PSK TLS Key Data Field

Name	Length	Value
Key type	1 byte	'85'
Length of PSK key data	Variable	'01' – '80', or '81 80' – '81 FF', or '82 01 00' – '82 FF FF'
Length of PSK key (in bytes)	Variable	'01' – '80', or '81 80' – '81 FF', or '82 01 00' – '82 FF FF'
Ciphered PSK key	Variable	'xxxx…'
Key Check Value length ('03')	1 byte	'03'
Key Check Value	3 bytes	'xxxx…'

3.8.2 PSK TLS Key Format for the STORE DATA Command

If the STORE DATA command is used to load PSK TLS keys, the CRT defined in the table below shall be used to describe the PSK TLS key sent to the Security Domain:

Table 3-12: Data Content for DGI '00B9' – PSK TLS Key

Tag	Length	Description	Presence
'B9'	Var	CRT tag (CT)	Mandatory
'80'	'01'	'85' (PSK TLS Key)	Mandatory
'81'	'01' or '02'	Key Length, in bytes (unsigned integer value)	Mandatory
'82'	'01'	Key Identifier	Mandatory
'83'	'01'	Key Version Number	Mandatory
'84'	'03'	Key check value	Mandatory

The DGI '8113' shall immediately follow DGI '00B9' and is used to populate the PSK TLS key:

Table 3-13: Data Content for DGI '8113' – PSK TLS Key Value

DGI	Length	Data Content	Encrypt
'8113'	Var – multiple of 8	PSK TLS key	Yes

4 API for Administration Session Triggering

This document introduces new services, available in the `org.globalplatform` package beginning with version 1.3, in order to:

- Request the triggering of an administration session. The HTTP Administration service is accessible as a uniquely registered Global Service and can be retrieved using the **`GPSystem.FAMILY_HTTP_ADMINISTRATION (0x84)`** constant identifying the HTTP Administration Service Family, and a Service ID of **`0x00`**, as shown in the following call to the API:

  ```
  GPSystem.getService(null,(short)(GPSystem.FAMILY_HTTP_ADMINISTRATION<<8))
  ```

- Be notified of the outcome of the administration session triggering request.

Annex A Examples

A.1 Nominal Case

First request sent by the Security Domain:

```
POST /server/adminagent?cmd=1 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: 0123456789 CRLF
CRLF
```

Command that shall be executed by the Security Domain in charge of the PSK TLS security:

```
HTTP/1.1 200 OK CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-Next-URI: /server/adminagent?cmd=2 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0
CRLF
Content-Length: xxxx CRLF
CRLF
[command-string]
```

Return of a command response:

```
POST /server/adminagent?cmd=2 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: 0123456789 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt-
response;version=1.0 CRLF
Content-Length: xxxx CRLF
X-Admin-Script-Status: ok CRLF
CRLF
[response-string]
```

Last response of Remote Administration Agent, communication shall be closed:

```
HTTP/1.1 204 No Content CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
CRLF
```

A.2 Nominal Case with an Intermediary Actor

First request sent by the OTA Security Domain:

```
POST /server/adminagent?cmd=1 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: 0123456789 CRLF
CRLF
```

Command that shall be executed by another Security Domain (Application Provider Security Domain):

```
HTTP/1.1 200 OK CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-Next-URI: /server/adminagent?cmd=2 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0
CRLF
X-Admin-Targeted-Application: //aid/A000000018/0001 CRLF
Content-Length: xxxx CRLF
CRLF
[secured-command-string]
```

Return of a command response:

```
POST /server/adminagent?cmd=2 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: 0123456789 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt-
response;version=1.0 CRLF
Content-Length: xxxx CRLF
X-Admin-Script-Status: ok CRLF
CRLF
[response-string]
```

Last response of Remote Administration Agent, communication shall be closed:

```
HTTP/1.1 204 No Content CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
CRLF
```

A.3 Error Case

First request sent by the OTA Security Domain:

```
POST /server/adminagent?cmd=1 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: 0123456789 CRLF
CRLF
```

Command that shall be executed by Application Provider Security Domain:

```
HTTP/1.1 200 OK CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-Next-URI: /server/adminagent?cmd=2 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt;version=1.0
CRLF
X-Admin-Targeted-Application: //aid/A000000018/0001 CRLF
Content-Length: xxxx CRLF
CRLF
[secured-command-string]
```

The previous message could not be processed due to security error on secured remote APDU format strings:

```
POST /server/adminagent?cmd=2 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-Script-Status: security-error CRLF
X-Admin-From: 0123456789 CRLF
CRLF
```

A.4 Communication Breakdown Case

Resume an administration session after a communication breakdown:

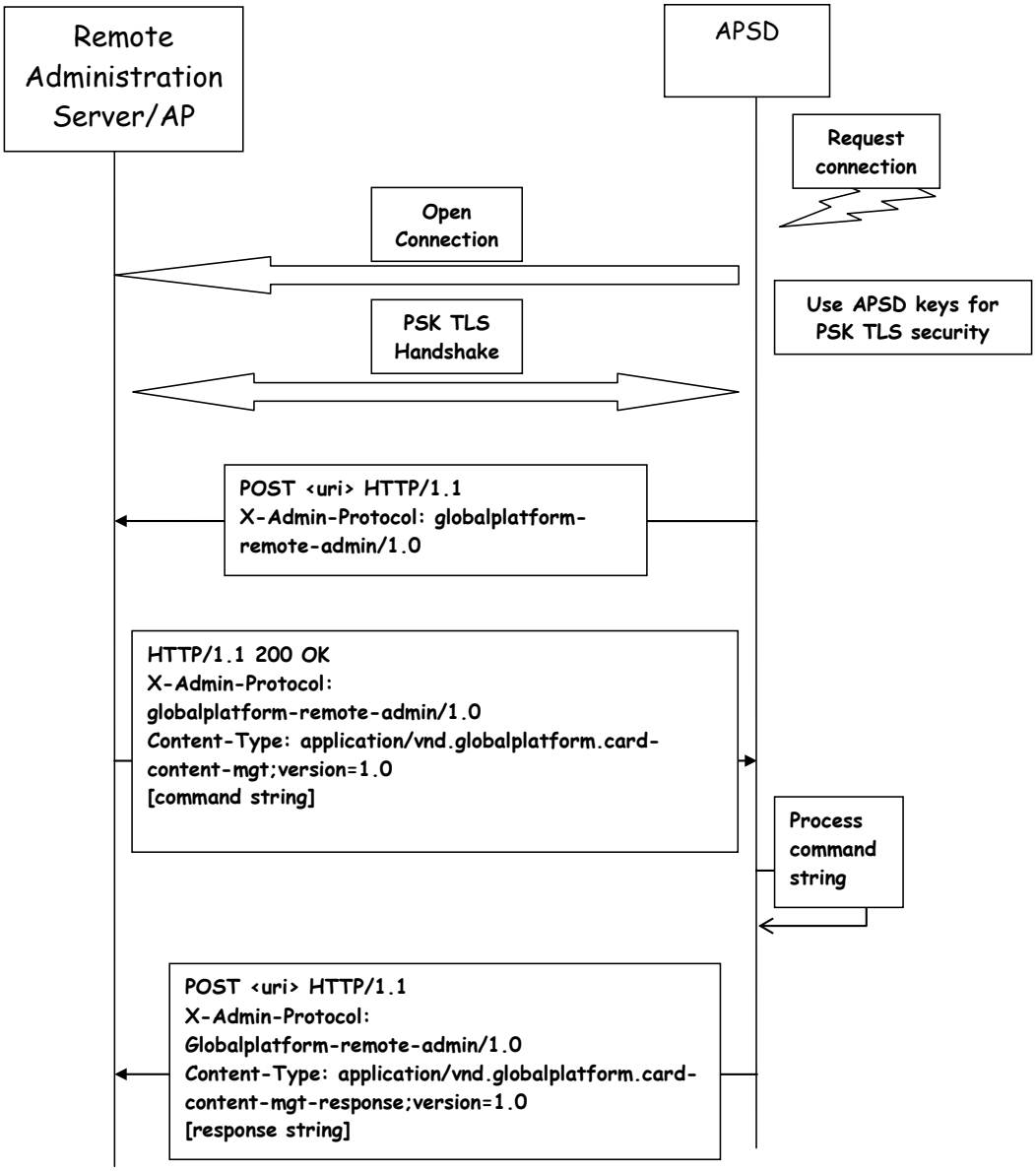
```
POST /server/adminagent?cmd=3 HTTP/1.1 CRLF
Host: 172.96.0.1 CRLF
X-Admin-Protocol: globalplatform-remote-admin/1.0 CRLF
X-Admin-From: 0123456789 CRLF
X-Admin-Resume: true
CRLF
```

A.5 Communication Flow

The actors and on-card components involved in this scenario are

- The Application Provider (AP) owning a Remote Administration Server
- The Security Domain of the Application Provider (APSD), compliant with [102 226], and having PSK TLS keys

Figure A-1: Communication Flow between an Application Provider Owning a Remote Administration Server and Its Security Domain



A.6 Communication Flow through an Intermediary Actor

The actors and on-card components involved in this scenario are

- The Application Provider (AP)
- The Remote Administration Server, owned by another entity
- The Security Domain in charge of the PSK TLS security, having PSK TLS keys (OTASD)
- The Security Domain of the Application Provider (APSD), compliant with [102 226], and if required supporting SCP02 for securing the APDUs

Figure A-2: Communication Flow between an Application Provider and Its Security Domain, through an Intermediary Actor