GLOBALPLATFORM
THE STANDARD FOR SMART CARD INFRASTRUCTURE

GlobalPlatform Card
Remote Application Management over HTTP
Card Specification v 2.2 - Amendment B
Version 0.5

Public Review

September 2008

Document Reference: GPC_SPE_011

Table of contents

1. Normative References

Standard / Specification	Description	Ref
GlobalPlatform Card v 2.2	Card specification from GlobalPlatform	[0]
ETSI TS 102 226	Smart cards; Remote APDU structure for UICC based applications, European Telecommunications Standards Institute Project Smart Card Platform (EP SCP), Release 7	[1]
RFC 2616	Hypertext Transfer Protocol – HTTP/1.1	[2]
RFC 2246	The TLS Protocol – Version 1.0	[3]
RFC 2818	HTTP over TLS	[4]
RFC 4279	Pre-Shared Key Cipher suites for Transport Layer Security (TLS)	[5]
RFC 4785	Pre-Shared Key (PSK) Cipher suites with NULL Encryption for Transport Layer Security (TLS)	[6]
ETSI TS 102 223	Smart Cards; Card Application Toolkit (CAT), Release 7	[7]
OMA SCWS	Smartcard Web Server V1.1, Open Mobile Alliance™	[8]
UICC configuration	GlobalPlatform Card configuration for Uicc Card	[9]

Table 1-1: Normative References

2. Abbreviations and Notations

Abbreviation	Meaning
AP	Application Provider
APDU	Application Protocol Data Unit
APSD	Security Domain of the Application Provider
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol over Secure Socket Layer
OTA	Over-The-Air
OTAPO	Over-The-Air Platform Operator
OTASD	Security Domain of the Over-The-Air platform operator
PSK TLS	Pre-Shared Key TLS
RAM	Remote Applet Management application compliant with ETSI TS 102 226
SCWS	Smart Card Web Server

Table 2-1: Abbreviations and Notations

3. Remote Application Management over HTTP

3.1. Scope of the document

This document defines a mechanism for an Application Provider to manage its application i.e. to load, install and personalize using the HTTP protocol 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.

3.2. Use Cases and Requirements

OMA SCWS [8] 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 [8], plus
- Loading of dynamic SCWS content generating applications, plus
- Mapping these applications to a SCWS URL as defined in [8],

within one session, all using the same HTTPS channel.

The mechanism defined in this document handles the Card Content Management as defined in [0] 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 (AP) 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 [1], 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.

4. Specification Amendments

4.1. PSK TLS key type

The Table 11-16 of the GlobalPlatform Card Specification v2.2 is replaced by Table 4-1: in order to introduce PSK TLS key type:

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 ($q^{-1} \bmod p$)
'A7'	RSA Private Key - Chinese Remainder DP1 component ($d \bmod (p-1)$)
'A8'	RSA Private Key - Chinese Remainder DQ1 component ($d \bmod (q-1)$)
'A9'-'FE'	RFU (asymmetric algorithms)
'FF'	Extended format

Table 4-1: New key type coding

4.2. Security Domain and Remote Administration Server

A Security Domain is responsible for establishing a connection with an off-card entity, called Remote Administration Server. This connection 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.
- The Secure channel protocol '81', option '00', is based on the industry standard security layer TLS (see [3]) and HTTPS (see [4])

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 [1]),
- Is responsible for retry and reconnection management in case of communication break down,
- 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 [8], the SD implements the SCWS (or card) administration agent.

The Remote Administration Server is an HTTP server.

4.3. Administration protocol

4.3.1. Administration Session triggering

An administration session starts when a Security Domain is triggered. 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 section 5 API functionality for administration session triggering).

The Security Domain shall receive a triggering message. The Security Domain will handle the administration session, using its own PSK-TLS keys for the communication security. It is assumed that the Security Domain knows all parameters needed to establish a connection or to handle its security. These parameters can be parameters of the triggering message or the parameters of the Security Domain itself. See section 4.7 Administration session triggering parameters.

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.

4.3.2. Communication channel setup

Once triggered, the Security Domain shall establish a communication channel with the Remote Administration Server.

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

- TLS_PSK_WITH_3DES_EDE_CBC_SHA, as defined in [5]
- TLS_PSK_WITH_AES_128_CBC_SHA, as defined in [5]
- TLS_PSK_WITH_NULL_SHA, as defined in [6]

The PSK TLS key version and key id to be used to initiate the PSK TLS session are read in the triggering parameters. See section 4.7 Administration session triggering parameters.

Provisioning of shared keys is described in section 4.1 PSK TLS key type.

The PSK TLS secure channel identifier is ‘81’ option is ‘00’.

4.3.3. Fetching a remote APDU format string

Once the PSK TLS communication channel is established the Security Domain shall send an HTTP POST command in order to get a remote APDU format string.

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 "`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 "`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.

The targeted Security Domain shall use its own secure channel to check the remote APDU format string.

- If the targeted security domain is handling the PSK TLS 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 TLS-PSK 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 found in the same Key Set version and the key Identifier incremented by one as identified in the Security Parameter (see 4.7.2). The algorithm used is identified by the algorithm [3DES or AES] associated to the key. The CBC mode is always used.
    - 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: AUTHENTICATED | C_MAC | C_DECRYPTION| R_MAC| R_ENCRYPTION R_MAC.
        - TLS_PSK_WITH_NULL_SHA: 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)`.

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 TLS secure channel.

The Remote Administration Server shall send the next remote APDU format string to the Security Domain over the 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.

4.4. Command format

4.4.1. 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
User-Agent: <card-user-agent> CRLF
From: <Agent ID> CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt-response/1.0
CRLF
[Content-Length: xxxx CRLF] or [Transfer-Encoding: chunked CRLF]
[Script-Status: <script-status> CRLF]
[Resume: true]
CRLF
[body-with-previous-response-string]
```

- The URI, the "`From`" value and the "`Host`" value to be used are defined in the administration session triggering message or by Security Domain parameters.
- The "`User-Agent`" header value shall be set to "`globalplatform/card-admin-agent/1.0`" if Content-Type is "`application/vnd.globalplatform.card-content-mgt-response/1.0`". The first request of a new administration session shall not contain any optional header field (except "`Resume`" header) and no body.
- The "`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 this administration session is resumed from a previous interrupted session, the Security Domain shall use the "`Resume`" header with the value "true" in the first POST request of the resume session. The "Resume" header shall not be used in the following POST requests. See section. 4.4.3 Interworking with the SCWS
- 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/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 [1] shall be used.

4.4.2. 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 administrative command.

The POST response shall have the following format:

```
HTTP/1.1 200 OK CRLF [or HTTP/1.1 204 No Content CRLF]
User-Agent: <remote-user-agent> CRLF
[Next-URI: <next-URI> CRLF]
Content-Type: application/vnd.globalplatform.card-content-mgt/1.0 CRLF
[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.
- The "`User-Agent`" header value shall be set to "`globalplatform/remote-admin-agent/1.0`" if Content-Type is "`application/vnd.globalplatform.card-content-mgt/1.0`". If another Content-Type is used, then the "User-Agent" header value shall be set accordingly, otherwise 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 "`Next-URI`" header is present in the response, the Security Domain shall use the given URI in the next POST request.
- If no "`Next-URI`" header is present in the response and if the body is empty, 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/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 [1] shall be used.
    - Optionally, "`Targeted-Application`" header with the hexadecimal 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. In that case, the APDUs in the body may be secured using SCP02 as defined in UICC configuration [9].
- If no "`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.
- The "`Next-URI`" header may be replaced by the "`SCWS-Next-URI`" header without any functional modification.

4.4.3. Interworking with the SCWS

If RAM over HTTP on a card is used together with SCWS administration as defined in OMA SCWS [8], the following additional provisions shall apply:

- The TLS secure channel to be used for RAM over HTTP may also be opened as defined in OMA SCWS [8].
- Independent of how the 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 message from the user agent and containing content type 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 message from the SCWS user agent and containing content type as defined for the SCWS. This shall end an administration session as defined in this document.

4.5. 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 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 4.4.1 HTTP POST request of Security Domain).

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 4.7.3 Retry policy parameters.
- If the communication is re-established, the Security Domain will try to resume the HTTP dialog by repeating the last HTTP request with the "`Resume: true`" header present. The Remote Administration Server may continue the administration session from this 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 others (e.g. not block the other retry attempts if the current one is not successful).

4.6. Command session

A command session consists in one or several remote APDU format string(s) for a single targeted Application. An administration session may transport several command sessions for several targeted Applications.

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

- The communication channel is opened.
- The Security Domain targeted by the current HTTP POST response is not the same than the one targeted by the previous HTTP POST response. That means:
    - The value of the header "`Targeted-Application`" has changed;
    - The value of the header "`Content-Type`" has changed;
    - Or the previous HTTP POST response contains a "`Targeted-Application`" header while the current one does not contain this header;
    - Or the current HTTP POST response contains a "`Targeted-Application`" header while the previous one does not contain this header.
- Before forwarding a remote APDU format string to a Security Domain, if no command session is currently opened.

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 targeted Security Domain shall be notified when a new command session starts. The notification internal processing is beyond the scope of the present document.

When a command session is closed, the relevant Security Domain shall be notified. The Security Domain may use this notification to clear its internal state. If the command session has been closed because a card reset has occurred, the Security Domain shall be notified at next card session.

4.7. Administration session triggering parameters

When starting an administration session, the targeted Security Domain shall use parameters to set up the connection, the security and the content of the first request. These parameters may be retrieved:

- From Security Domain parameters. The Issuer Security Domain owns the default card parameters,
- Or from the message leading to this administration session (the administration session triggering parameters)

If parameters are missing in the triggering message, they shall be completed with the targeted Security Domain's parameters or with the default card parameters. Default card parameters are predefined and chosen 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:

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**			
				'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

Table 4-2: Administration session triggering parameters

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 [1].

4.7.1. 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 the tag '85' is contextual tag and has no relation with the tag '85' defined in Table 4-2: Administration session triggering parameters. Issuer Security Domain owns the default Administration Session Parameters.

Tag	Length	Name					Presence
'85'	1-n	Security Domain Administration Session Parameters					Optional
		Tag	**Length**	**Name**			
		'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

Table 4-3: TLV Security Domain Administration Session Parameters

4.7.2. Security parameters

The security parameters are defined as follows:

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

Table 4-4: Security parameters

- PSK Identity is a string defined in [5]
- Key version/Key-Identifier identifies 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

4.7.3. Retry policy parameters

The security parameters are defined as follows:

Description	Length
Retry policy parameters tag	1
Length	1
Retry counter	2
Retry waiting delay	5

Table 4-5: Retry policy parameters

- 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 [7].

4.7.4. 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:

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

Table 4-6: Host parameter

4.7.5. Agent Id parameter

This TLV defines the “`From`” header value to be used by the Remote Administration Server to identify the requesting Application when receiving a POST request. It is defined as follows:

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

Table 4-7: Agent Id parameter

4.7.6. 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:

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

Table 4-8: Administration URI parameter

4.8. PSK –TLS key format

Name	Length
New Key Version Number	1 byte
Key type ('85')	1 byte
PSK key data length (n+1)	1 byte
Length of PSK key	1 byte
Ciphered PSK key	n bytes
Check value length ('03')	1 byte
Check value	3 bytes

Table 4-9: PSK TLS Key data field

Before ciphering, the PSK key shall be padded with as few (if any) random bytes to fill the last block required by the ciphering algorithm.

The padded PSK key shall be ciphered using CBC mode.

The key check value shall be the three most significant bytes of the SHA-1 digest of the PSK Key.

5. API functionality for administration session triggering

This section defines the API to be used by an on-card entity to request an administration session triggering. The HTTP Administration service is accessible as a uniquely registered Global Services. A reference on this service may be retrieved using the `GPSystem.getService(null, FAMILY_HTTP_ADMINISTRATION )` method.

5.1. GPSystem class

The annex A.1 of the GlobalPlatform Card Specification v2.2 is modified to introduce the family that identifies the Card Administration Server service and a new shareable interface in the GPSystem class.

FAMILY_HTTP_ADMINISTRATION

`public static final byte FAMILY_HTTP_ADMINISTRATION`

Indicates the family of the HTTP Administration Global Service Identifier (0x84)

public interface **HTTPAdministration** extends javacard.framework.Shareable

This interface handles an HTTP administration session triggering request.

Method Summary

Void	**requestHTTPAdministrationSession**(byte[] triggeringParameters, short offset, short length) Request an administration session

Method Detail

requestHTTPAdministrationSession

```
void requestHTTPAdministrationSession (
                byte[] triggeringParameters,
                short offset,
                short length)
```

Request an administration session. The Security Domain of the calling application will handle the PSK TLS security of the communication.

Parameters:

`triggeringParameters` – this buffer contains the administration session triggering parameters as defined in section 4.7 Administration session triggering parameters

`offset` – offset within `triggeringParameters`

`length` – length of parameters within `triggeringParameters`

Throws:

`java.lang.NullPointerException` - if `triggeringParameters` is equal to null.

`java.lang.ArrayIndexOutOfBoundsException` – if `offset` or `length` would lead to access outside array bounds

`javacard.framework.ISOException` – with the following reason codes:

- SW_WRONG_DATA if data within `triggeringParameters` are not correctly formatted
- SW_CONDITIONS_NOT_SATISFIED if operation could not be processed (for example if no SCP81 support is possible for the calling application)

A. Annex: 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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
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
Next-URI: /server/adminagent?cmd=2 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt/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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
From: 0123456789 CRLF
Content-Type: application/vnd.globalplatform.response-script/1.0 CRLF
Content-Length: xxxx CRLF
Script-Status: ok CRLF
CRLF
[response-string]
```

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

```
HTTP/1.1 204 No Content 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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
From: 0123456789 CRLF
CRLF
```

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

```
HTTP/1.1 200 OK CRLF
Next-URI: /server/adminagent?cmd=2 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt/1.0 CRLF
Targeted-Application: A0000000180001 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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
From: 0123456789 CRLF
Content-Type: application/vnd.globalplatform.response-script/1.0 CRLF
Content-Length: xxxx CRLF
Script-Status: ok CRLF
CRLF
[response-string]
```

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

```
HTTP/1.1 204 No Content 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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
From: 0123456789 CRLF
CRLF
```

Command that shall be executed by Application Provider Security Domain:

```
HTTP/1.1 200 OK CRLF
Next-URI: /server/adminagent?cmd=2 CRLF
Content-Type: application/vnd.globalplatform.card-content-mgt/1.0 CRLF
Targeted-Application: A0000000180001 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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
Script-Status: security-error 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
User-Agent: globalplatform/card-admin-agent/1.0 CRLF
From: 0123456789 CRLF
Resume: true
CRLF
```

A.5 Communication flow

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

- The Application Provider (AP)
- The Remote Administration Server
- The Security Domain of the Application Provider (APSD), compliant with [1], and having PSK TLS keys.

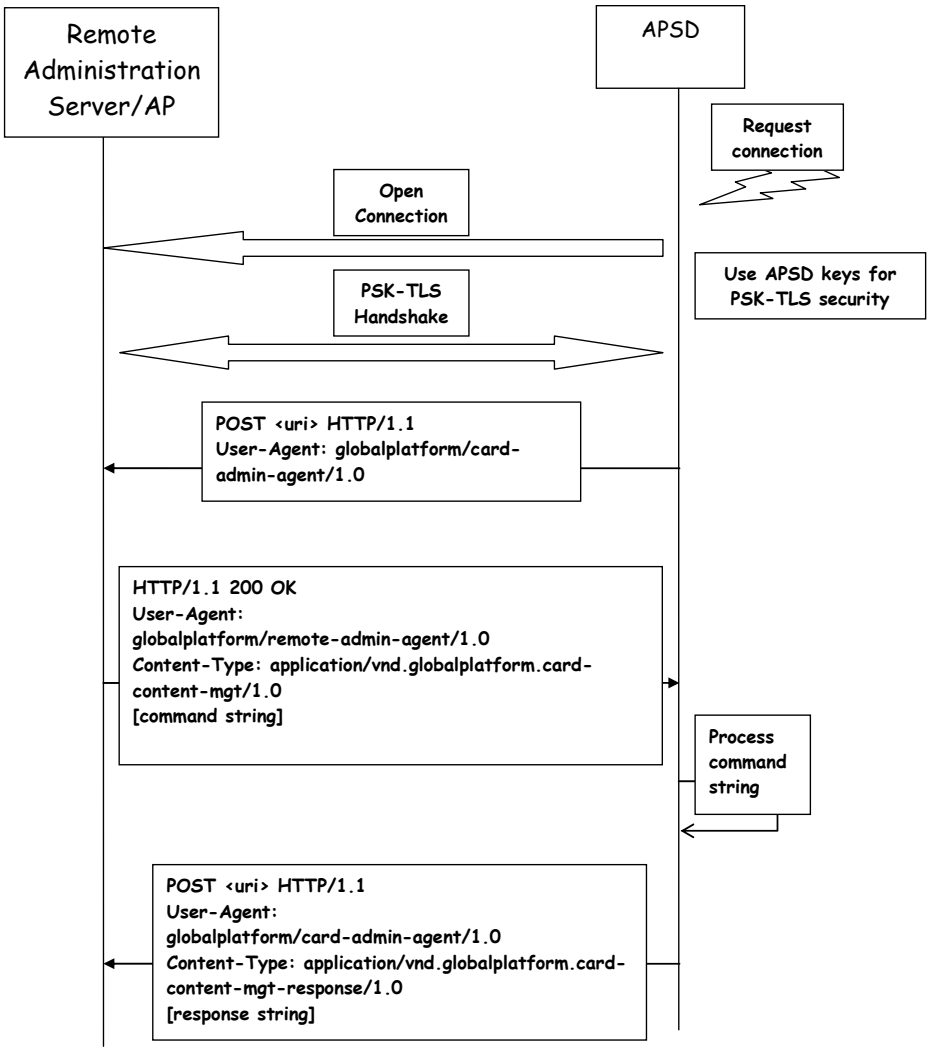

Figure A-1 : Communication flow between an Application Provider owning a Remote Administration Server and its Security Domain

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
- 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 [1], and if required supporting SCP02 for securing the APDUs.

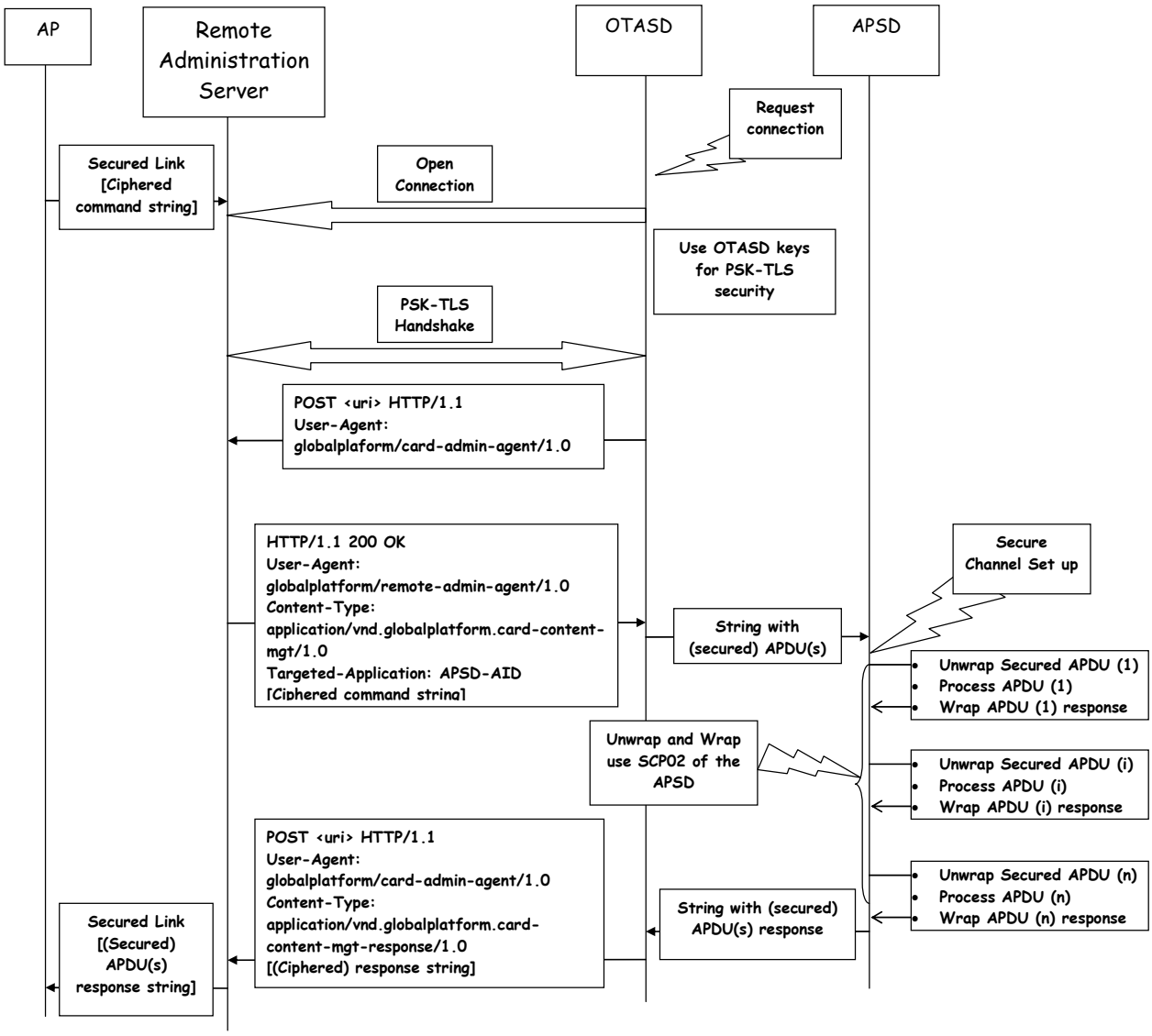

Figure A-2 : Communication flow between an Application Provider and its Security Domain, through an intermediary actor

6. Tables of Figures and Tables