GLOBALPLATFORM®
THE STANDARD FOR SECURE DIGITAL SERVICES AND DEVICES

GlobalPlatform Technology

Remote Application Management over CoAP

Card Specification v2.3 – Amendment M

Version 0.0.0.11

Public Review

December 2022

Document Reference: GPC_SPE_207

THIS SPECIFICATION OR OTHER WORK PRODUCT IS BEING OFFERED WITHOUT ANY WARRANTY WHATSOEVER, AND IN PARTICULAR, ANY WARRANTY OF NON-INFRINGEMENT IS EXPRESSLY DISCLAIMED. ANY IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT SHALL BE MADE ENTIRELY AT THE IMPLEMENTER'S OWN RISK, AND NEITHER THE COMPANY, NOR ANY OF ITS MEMBERS OR SUBMITTERS, SHALL HAVE ANY LIABILITY WHATSOEVER TO ANY IMPLEMENTER OR THIRD PARTY FOR ANY DAMAGES OF ANY NATURE WHATSOEVER DIRECTLY OR INDIRECTLY ARISING FROM THE IMPLEMENTATION OF THIS SPECIFICATION OR OTHER WORK PRODUCT.

Contents

Tables

1 Introduction

The SCP81 protocol described in GlobalPlatform Card Specification Amendment B ([GPCS-B]) provides secure communication to a device using an HTTP/TLS connection. SCP81 has become widely adopted because it is relatively simple for a server to implement. However, it is not very efficient for Low Power Wide Area Networks (LPWAN) such as NB-IoT ([NB-IoT]), where TCP may not be reliable because of the latency.

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 CoAP protocol (RFC 7252 [HTTP]) and PSK DTLS security Over-The-Air. More specifically, it describes how to adapt mechanisms described in [GPCS-B] in support of LPWAN by (1) Replacing TCP by UDP which is more adapted to LPWAN and (2) Replacing HTTP by CoAP to optimize the amount of exchanged data. As a consequence, the DTLS protocol is also used instead of TLS.

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 [GPCS] and [GPCS-B].

1.2 IPR Disclaimer

Attention is drawn to the possibility that some of the elements of this GlobalPlatform specification or other work product may be the subject of intellectual property rights (IPR) held by GlobalPlatform members or others. For additional information regarding any such IPR that have been brought to the attention of GlobalPlatform, please visit https://globalplatform.org/specifications/ip-disclaimers/. GlobalPlatform shall not be held responsible for identifying any or all such IPR, and takes no position concerning the possible existence or the evidence, validity, or scope of any such IPR.

1.3 References

Table 1-1: Normative References

Standard / Specification	Description	Ref
GlobalPlatform Card Specification GPC_SPE_034	GlobalPlatform Technology Card Specification v2.3.1	[GPCS]
GlobalPlatform Card Specification Amendment B GPC_SPE_011	GlobalPlatform Technology Card Specification v2.3 Amendment B v1.2 – RAM over HTTP	[GPCS-B]
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 6347	Datagram Transport Layer Security v1.2	[DTLS v1.2]
RFC 9147	Datagram Transport Layer Security v1.3	[DTLS v1.3]

Standard / Specification	Description	Ref
RFC 7252	The Constrained Application Protocol (CoAP)	[CoAP]
RFC 7959	Block-Wise Transfers in the Constrained Application Protocol (CoAP)	[Block CoAP]
RFC 9175	Constrained Application Protocol (CoAP): Echo, Request-Tag, and Token Processing	[CoAP Request-Tag]
3GPP Rel 16	3GPP Release 16	[NB-IoT]

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
APDU	Application Protocol Data Unit
API	Application Programming Interface
CoAP	Constrained Application Protocol
DTLS	Datagram Transport Layer Security
HTTP	Hypertext Transfer Protocol
LPWAN	Low Power Wide Area Network
NB-IoT	Narrowband – Internet of Things
OTA	Over-The-Air
PSK TLS	Pre-Shared Key TLS
RAM	Remote Application Management
RFM	Remote File Management
TLS	Transport Layer Security
URI	Uniform Resource Identifier

1.6 Revision History

GlobalPlatform technical documents numbered *n*.0 are major releases. Those numbered *n*.1, *n*.2, etc., are minor releases where changes typically introduce supplementary items that do not impact backward compatibility or interoperability of the specifications. Those numbered *n.n*.1, *n.n*.2, etc., are maintenance releases that incorporate errata and precisions; all non-trivial changes are indicated, often with revision marks.

Table 1-3: Revision History

Date	Version	List of Modifications
Jan 2021	0.0.0.1	Initial draft, developed from ARM contribution
Feb 2021	0.0.0.5	Committee Review
July 2022	0.0.0.10	Member Review
Dec 2022	0.0.0.11	Public Review Note that option numbers marked in Table 3-3: CoAP Options as X still need to be allocated through IANA. For initial testing/experimentation, numbers above 65000 have been assigned.
TBD	1.0	Public Release

2 Use Cases and Requirements

The use cases and requirements that apply to this specification are identical to those that apply to Amendment B, Remote Application Management over HTTP ([GPCS-B]). The only difference is the use of CoAP instead of HTTP and UDP instead of TCP.

3 Specification Amendments

This specification is derived from [GPCS-B] by replacing the HTTP protocol with the CoAP protocol (RFC 7252 [CoAP]), already widely used in IoT networks, and replacing the TLS protocol with the DTLS protocol (RFC 6347 [DTLS v1.2] or RFC 9147 [DTLS v1.3]), which is also used on top of UDP instead of TCP. This provides a very quick way to migrate servers and devices to a secure, standard protocol that is more efficient than HTTP/TLS.

This specification refers to the DTLS protocol as the GlobalPlatform Secure Channel Protocol '82' (SCP82). For a card implementation, although both protocols might be supported, supporting SCP82 does not imply any requirement to support SCP81 (as described in [GPCS-B]).

The following sections detail the necessary changes compared to [GPCS-B].

3.1 PSK TLS Key Type

DTLS uses the same cipher suites and keys as TLS, so the pre-shared key type can be re-used.

Table 3-1: Key Type Coding

Value	Meaning
'85'	Pre-shared Key for Transport Layer Security (TLS or DTLS)

3.2 Security Domain and Remote Administration Server

3.2.1 Secure Communication Configuration

For SCP82, the “i” parameter (implementation options) is formatted as a 1-byte bitmap as defined in Table 3-2, indicating all the DTLS versions supported by the Security Domain. A Security Domain may support one or multiple DTLS versions.

Table 3-2: Values of Parameter "i"

b8	b7	b6	b5	b4	b3	b2	b1	Description
							X	RFU (set to 0)
						X		RFU (set to 0)
					1			[DTLS v1.2] supported
				1				[DTLS v1.3] supported
	X	X	X					RFU (set to 0)
X								Reserved

3.3 Administration Protocol

The protocol remains essentially unchanged, although all the HTTP headers are converted to CoAP options as described below.

For CoAP, it is not normally recommended that the maximum fragment length be negotiated down to 512 bytes. The most straightforward implementation of CoAP requires that each CoAP message fits into a single UDP packet and a single DTLS fragment. In some contexts, CoAP messages may be longer than 1024 bytes, so it is recommended that in these contexts, a maximum fragment length of 2048 bytes is negotiated. Block-wise CoAP (RFC 7959 [Block CoAP]) may be used for longer messages, but it is recommended that block-wise transfers not be used for payloads of less than 1200 bytes.

Support of block-wise CoAP is optional, but if supported, both client and server SHALL implement the CoAP Request-Tag Option (see [CoAP Request-Tag]) to ensure the correct ordering of blocks from one or multiple requests.

All of the cipher suites listed in [GPCS-B] for TLS 1.2 and TLS 1.3 apply respectively to DTLS 1.2 and DTLS 1.3. It is not required for an implementation of this specification to support all the listed cipher suites. Supporting other cipher suites is allowed but out of scope of this specification.

3.4 Command Format

The primary difference between CoAP and HTTP is that HTTP headers are converted to CoAP options. Table 3-3 describes how to do such a conversion. The option numbers marked as X still need to be allocated through IANA, although for initial testing/experimentation it is possible to use numbers above 65000.

Editor's Note: Final option number values will be requested from IANA during Public Review and inserted in the document before publication. Meanwhile, the test values described below may be used for experimentation.

Content-Length or chunked transfer encoding is not required, as the length of the CoAP message data is determined from the CoAP packet. For simple CoAP messages, which fit in a single packet, the payload data is simply all the data from the end of the options to the end of the packet.

Table 3-3: CoAP Options

Option	Number	Type	Value
`Uri-Host`	3	string	Server host name corresponding to the `Host` header value. Configured using the "Administration Host" session triggering parameter or security domain parameter. May be omitted.
`Uri-Path`	11	string	Path element of URI. A URI may contain multiple `Uri-Path` options. The value for the initial POST request is configured by the "Administration URI" session triggering parameter or security domain parameter. May be empty if the path is "/". Values for subsequent POST requests are copied from the response message, representing the contents of the `X-Admin-Next-URI`.
`Uri-Query`	15	string	Query element of URI. As per the `Uri-Path` option above. May be omitted if no query elements are present in the URI.
`SCP82-Admin-From`	X / 65001	opaque	The agent ID of the triggered SD. This corresponds to the configured "Agent ID" session triggering parameter or security domain parameter, and the `X-Admin-From` header.
`SCP82-Targeted-Application`	X / 65002	opaque	The AID of the targeted application. Rather than the format specified in [GPCS-B], this shall be a simple binary value from 5-16 bytes in length. Omitted if the targeted Security Domain is the one in charge of the PSK DTLS security.
`SCP82-Content-Type`	X / 65003	opaque	Corresponds to the `Content-Type` header. Accepted values: `0x00`: RAM `0x01`: RAM response `0x02`: RFM `0x03`: RFM response <content-type>: Set by connection API in Open API call

`SCP82-Admin-Script-Status`	X / 65004	opaque	Corresponds to the `X-Admin-Script-Status` header. Accepted values:
			`0x01`: `ok` (default, so can be omitted)
			`0x02`: `unknown-application`
			`0x03`: `not-a-security-domain`
			`0x04`: `security-error`

`SCP82-Resume`	X / 65005	empty	If present, this POST request is sent by the card to resume a session. For CoAP this will only be used if the connection had to be closed and re-opened. The server may use the URI used in the POST request to identify how far the script has progressed and attempt to resume the session. This corresponds to the `X-Admin-Resume` header.

3.5 Retry Policy

The retry policy for CoAP/DTLS is identical to the retry policy described in [GPCS-B].

3.6 Command Session

Command sessions are handled in the same way as described in [GPCS-B] when using CoAP/DTLS.

3.7 Administration Session Triggering Message

Most of the parameters set during Security Domain installation or as triggering parameters remain unchanged for CoAP. The exceptions are:

- The UICC-terminal interface transport level parameter in the connection parameters TLV SHALL specify UDP instead of TCP.
- The CoAP POST Administration URI parameter is split into its component `Uri-Path` and `Uri-Query` parameters for simpler conversion to CoAP options. For reference, the entire CoAP POST parameters, replacing the HTTP POST parameters, is reproduced here:

Table 3-4: CoAP POST Parameters

<table>
<tr><th>Tag</th><th>Length</th><th colspan="4">Name</th></tr>
<tr><td rowspan="8">'89'</td><td rowspan="8">1-n</td><td colspan="4">CoAP POST parameters</td></tr>
<tr><td>Tag</td><td>Length</td><td colspan="2">Value</td></tr>
<tr><td>'8A'</td><td>1-n</td><td colspan="2">Administration Host parameter (identical to [GPCS-B])</td></tr>
<tr><td>'8B'</td><td>1-n</td><td colspan="2">Agent ID parameter (identical to [GPCS-B])</td></tr>
<tr><td>'AC'</td><td>1-n</td><td colspan="2">CoAP Administration URI parameter</td></tr>
<tr><td colspan="2"><table>
<tr><th>Tag</th><th>Length</th><th>Value</th></tr>
<tr><td>'8C'</td><td>1-n</td><td>Value of <code>Uri-Path</code></td></tr>
<tr><td colspan="3"><Repeat tag '8C' as required></td></tr>
<tr><td>'8D'</td><td>1-n</td><td>Value of <code>Uri-Query</code></td></tr>
<tr><td colspan="3"><Repeat tag '8D' as required></td></tr>
</table></td></tr>
</table>

All parameters are optional, as in [GPCS-B].

For example, a URI of "`/ram/admin?cmd=first`" would be encoded in the 'AC' TLV as follows:

```
'AC 17'
  '8C0372616D'          (Uri-Path 1 – "ram")
  '8C0561646D696E'        (Uri-Path 2 – "admin")
  '8D09636D643D6669727374' (Uri-Query 1 – "cmd=first")
```

Note that it is acceptable to have no `Uri-Path` or no `Uri-Query` values in the URI. For example, a URI of `/ram` would simply be encoded as:

```
'AC 05'
  '8C0372616D'          (Uri-Path 1 – "ram")
```

3.8 Security Domain Administration Session Parameters

These parameters are as per [GPCS-B], with the changes described above to the connection parameters and the HTTP POST parameters.

3.9 Loading PSK TLS keys

Pre-shared keys are loaded in the same way as described in [GPCS-B] when using CoAP/DTLS.

3.10 DNS Resolution

DNS resolution support is optional for this amendment. Since the use of DNS resolution is independent of the transport protocol, the descriptions in [GPCS-B] sections 3.10 and 3.11 apply to this specification as well.