GLOBALPLATFORM®

THE STANDARD FOR MANAGING APPLICATIONS ON SECURE CHIP TECHNOLOGY

GlobalPlatform Card Technology

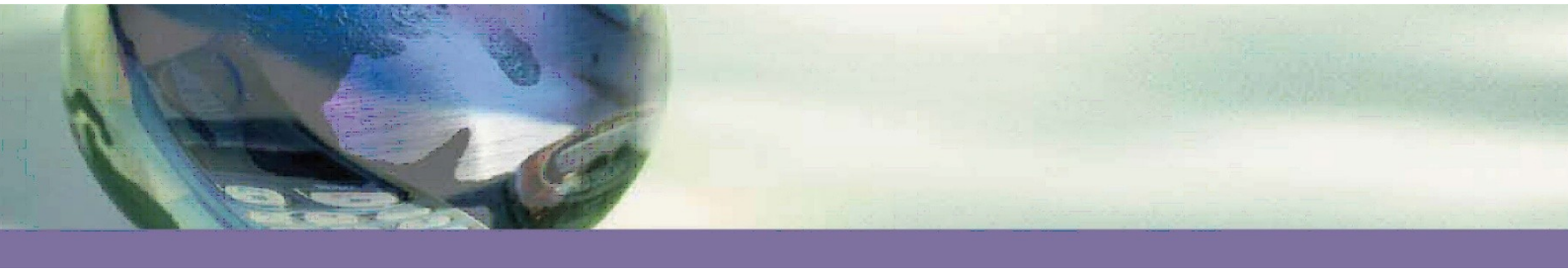
Card Specification – Privacy Framework

Version 1.0

Public Release

February 2017

Document Reference: GPC_SPE_100



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

Figures

Tables

1 Introduction

This document describes a framework based on the GlobalPlatform Card Specification [GPCS] that allows managing different privacy mechanisms (i.e. protection against traceability, protection of the communication, etc.) which remain under the control of the card issuer.

This specification describes the concepts of Global Privacy Protocol (GPP), Global Master File (GMF), and OPEN Privacy Extension (OPEX). The GPP is implemented by a GPP Application and registered on the card as a Global Service. The OPEX leverages this GPP service in order to establish authentication, secure communications, and restrict access to on-card applications, according to the privacy rules of the card issuer. It remains possible to install on-card applications whose access is not restricted by the OPEX and that are responsible for enforcing their own specific privacy mechanisms. Such applications are known as Privacy Trusted applications. The GMF is implemented by a GMF Application and registered on the card as a Global Service. The OPEX leverages this GMF service to provide the off-card entity with an access to MF files, which is required as a preliminary step by some privacy protocols.

This version of the specification assumes that the OPEX and GPP Application are able to manage a single GPP session on the basic logical channel (0). Support for more than one GPP session concurrently on several logical channels remains out of scope of this document.

Finally, this document promotes two privacy-enabled Secure Channel Protocols, namely SCP '21' (defined in this document) and SCP '22' (defined in [GPCS Amd G]).

1.1 Audience

This document is intended primarily for card manufacturers developing GlobalPlatform card implementations and application developers developing Applications for such cards. It is assumed that the reader is familiar with smart cards and smart card production, and in particular familiar with [GPCS].

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 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://www.globalplatform.org/specificationsipdisclaimers.asp. 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	GlobalPlatform Card Specification v2.3	[GPCS]
GlobalPlatform Card Specification – Amendment G	GlobalPlatform Card Specification v2.3 – Amendment G: Opacity Secure Channel, v1.0	[GPCS Amd G]
CEN/EN 419 212	Application Interface for smart cards used as Secure Signature Creation Devices, Part 1 (Basic services) & Part 2 (Additional services), 28/08/2014	[419 212]

Standard / Specification	Description	Ref
ISO/IEC 7816-4:2013	Identification cards – Integrated circuit cards – Part 4: Organization, security and commands for interchange	[7816-4]
BSI TR-03110	TR-03110 Technical Guideline Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token, v2.20	[TR 3110]
ICAO doc 9303	Machine Readable Travel Documents, 7th edition 2015	[ICAO 9303]
GlobalPlatform Privacy Framework Requirements	GlobalPlatform Government Task Force Privacy Framework Requirements	[PFR]

1.4 Terminology and Definitions

Selected technical terms used in this document are defined in [GPCS].

1.5 Abbreviations and Notations

Hexadecimal values are enclosed in straight single quotation marks (example: '0F').

Selected abbreviations used in this document are defined Table 1-2. Additional abbreviations are defined in [GPCS].

Table 1-2: Abbreviations and Notations

Abbreviation / Notation	Meaning
APDU	Application Protocol Data Unit
BAC	Basic Access Control
CHAT	Certificate Holder Authorization Template
DG	Data Group
EAC	Extended Access Control
eMRTD	Electronic Machine Readable Travel Document (e.g. ePassport)
GAP	General Authentication Procedure
GMF	Global Master File
GMFA	Global Master File Application
GPP	Global Privacy Protocol
GPPA	Global Privacy Protocol Application
mEAC	Modular Extended Access Control
MF	Master File
MSE	Manage Security Environment
OPEX	OPEN Privacy Extension
PACE	Password Authenticated Connection Establishment

Abbreviation / Notation	Meaning
SCP	Secure Channel Protocol
SD	Security Domain
SPP	Specific Privacy Protocol

1.6 Revision History

Table 1-3: Revision History

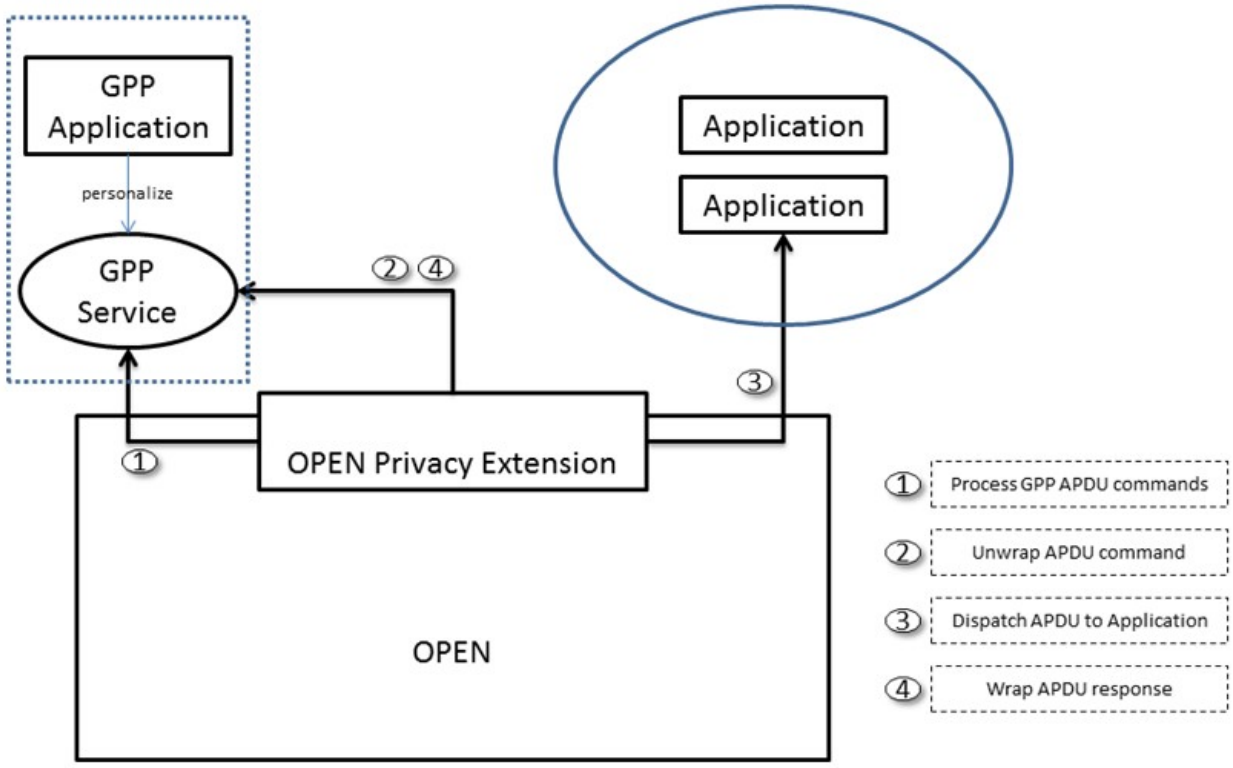
Date	Version	Description
February 2017	1.0	Initial Public Release

2 Global Privacy Protocol

2.1 Introduction

The Global Privacy Protocol (GPP) is a special kind of Secure Channel Protocol that applies globally to the entire card. The GPP implementation is personalized and made available by a Global Privacy Protocol Application (GPPA) as a Global Service. Once registered, this service becomes active and is ready for use by the OPEN Privacy Extension.

Figure 2-1: Global Privacy Protocol Overview



2.2 Current Privacy Status

The Current Privacy Status denotes a particular step reached in the execution of the Global Privacy Protocol, and is maintained by the GPP service (see section 2.3). Its value is made accessible to on-card Applications through a specific API (see Annex A).

Privacy Status values remain out of scope of this specification and shall be defined in specific configuration documents.

2.3 Global Privacy Protocol Service

The Global Privacy Protocol Service is a Global Service that implements the GPP.

Only a single GPP service may be registered on the card. The OPEN Privacy Extension shall enforce the GPP as soon as the GPP service has been successfully registered. Therefore, the GPP service should only be registered once it is fully personalized and ready for operation.

The following Global Service Identifier is reserved and shall be used to register the GPP service:

- '8600' is reserved for Global Privacy Protocol service.

It is strongly recommended that the GPP service implements Secure Channel Protocol '21' (see Annex D) that is known to enforce essential privacy properties. These privacy properties are described in [PFR].

The GPP service implements the `PrivacyProtocol`, the `GlobalPrivacyProtocol`, and the `SecureChannelProtocol` interfaces (see Annex A for more details), and is made available to on-card Applications and to the OPEN Privacy Extension through the `GPSystem.getService()` method. In particular, the Current Privacy Status (see section 2.2) established and maintained by the GPP service can be queried by on-card Applications using the `PrivacyProtocol` interface.

2.4 Global Privacy Protocol Application

The Global Privacy Protocol Application (GPPA) is the Application that registers the GPP service.

The GPPA is responsible for personalizing data and credentials required for the correct operation of the GPP. Personalization commands are application-specific and remain out of scope of this specification.

The GPPA should only register the GPP service once it is fully personalized and ready for operation.

3 Global Master File

3.1 Introduction

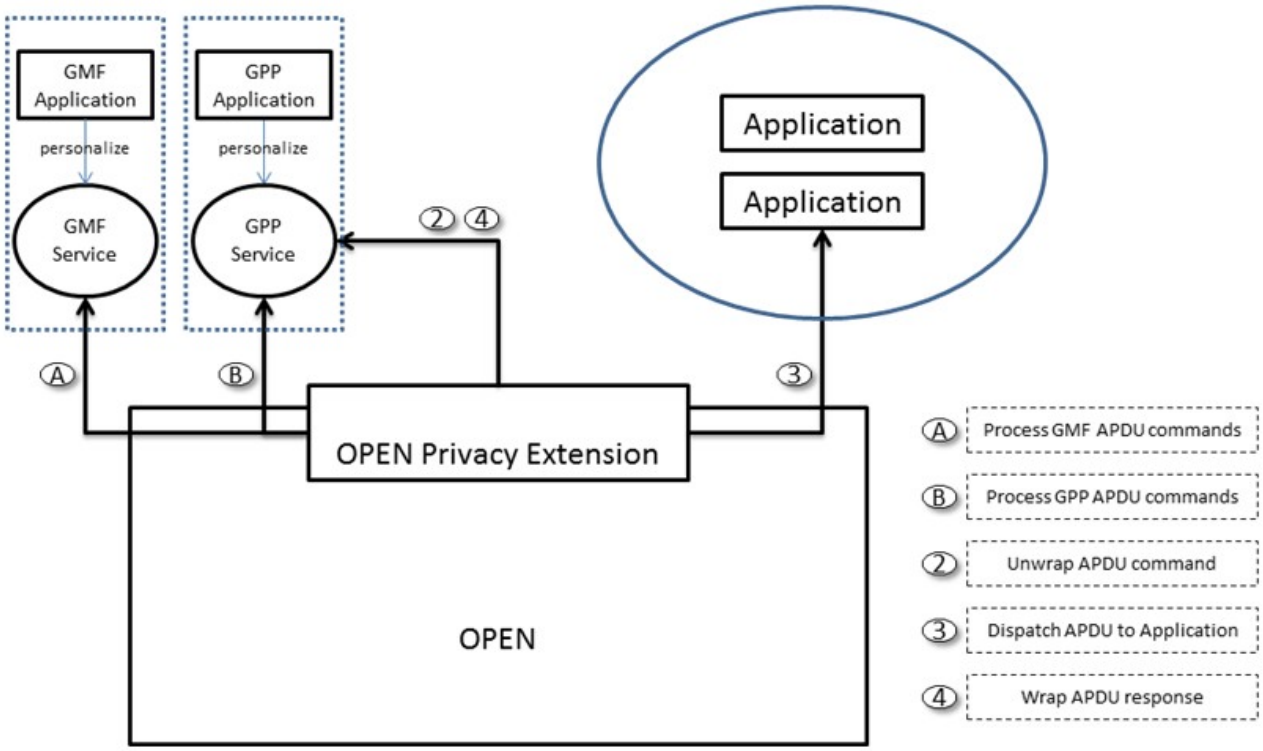
If the GPP requires accessing a Master File (MF) system, the GPP service may be complemented by a Global Master File (GMF) service personalized and registered by a Global Master File Application (GMFA).

The Global Master File includes everything that should be shared, such as files under the MF except application, ADF, and DF. A description of mandatory files for various use cases will be given in configuration documents.

That implies that all commands to access DFs, ADFs, and file structures under these (A)DFs are processed by the applications. Notice that applications for legacy reasons may include a Local Master File. The Local Master File is the MF managed by an application other than GMFA.

As an example, a GPP based on Secure Channel Protocol '21' will require files (i.e. EF.CardAccess) to be retrieved at the level of an MF. Once registered, the GMF service is used by the OPEN Privacy Extension to provide access to MF files.

Figure 3-1: Global Master File and Global Privacy Protocol



3.2 Global Master File Service

The Global Master File Service is a Global Service that implements access to MF files.

Only a single GMF service may be registered on the card. The OPEN Privacy Extension shall provide access to MF files as soon as the GMF service has been successfully registered.

The following Global Service Identifier is reserved and shall be used to register the GMF service:

- '8700' is reserved for Global Master File service.

The GMF service implements the interface `GlobalMasterFile` (see Annex A for more details), and is made available to the OPEN Privacy Extension through the `GPSystem.getService()` method.

The access control intended to be implemented by this specification is the ‘Read’ permission. Other permissions like ‘Write’, ‘Erase’, and ‘Delete’ are not part of this document and could be implementation specific.

Some use cases require “read” access conditions to be enforced for some files, hence the GMFA may enforce access conditions for some files accordingly; however, the way such access conditions would be set up remains out of scope.

The requirements regarding minimum capabilities shall be described in configuration documents.

3.3 Global Master File Application

The Global Master File Application (GMFA) is the Application that registers the GMF service.

The GMFA is responsible for personalizing the files accessible through the GMF service. Personalization commands are application-specific. Personalization may be described in a configuration document and remains out of scope of this specification.

Notice that the GMFA and GPPA may be the same Application. In such cases, the ‘Read’ permission can be configured by the GPPA.

4 OPEN Privacy Extension (OPEX)

4.1 Introduction

The OPEN Privacy Extension (OPEX) is responsible for:

- Requesting the GMF service to process APDU commands accessing the MF.
- Requesting the GPP service to process APDU commands relating to the GPP.
- Requesting the GPP service to unwrap APDU commands and wrap APDU responses.
- Comparing the Current Privacy Status and the Privacy Requirements of Applications.
- Allowing or rejecting the selection of Applications.

4.2 Enforcement of the Global Privacy Protocol

The OPEX shall enforce the GPP as soon as the GPP service has been successfully registered. However, currently selected Applications shall not be automatically deselected upon registration of the service, so that, e.g. it shall be possible to complete some personalization sequence.

If secure messaging is established by the GPP service, the OPEX shall request the GPP service to process secure messaging for all incoming APDU commands (i.e. unwrap) and outgoing APDU responses (i.e. wrap). This is the very first operation performed by the OPEX when a new command is received and the remaining of this section assumes that incoming commands are successfully unwrapped. See section 4.3 for details on the processing of secure messaging by the GPP service.

If there is no Application selected yet, the OPEX shall request the GPP service to process incoming APDU commands, except MF access commands (see section 4.5). If it is recognized as a GPP command, the incoming command shall be processed by the GPP service, which may result in an update of the Current Privacy Status (see section 2.2). Depending on the Current Privacy Status, commands unknown to the GPP are rejected or result in the successful selection of an Application (see section 4.4).

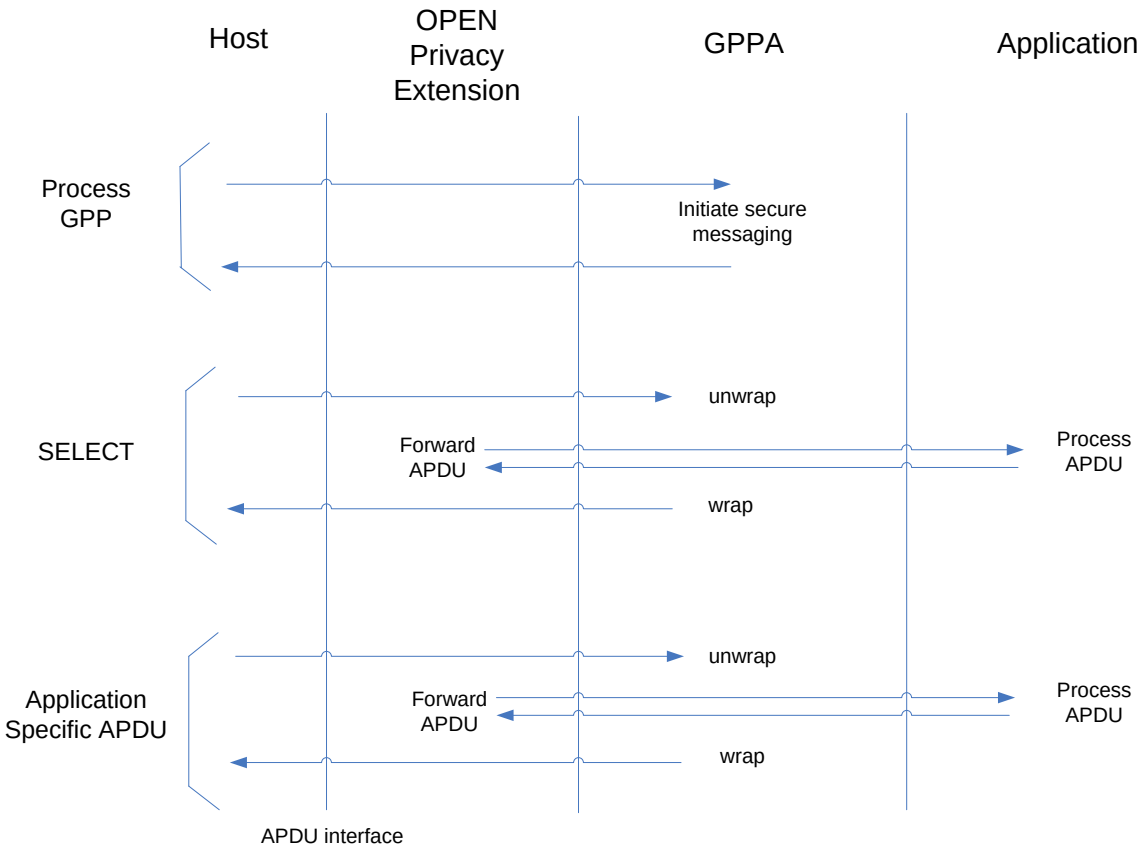
As soon as an Application becomes selected, the OPEX shall stop requesting the GPP service to process incoming APDU commands. The GPP service is said to be “on hold”. In this state:

- If secure messaging was previously established, the GPP service shall still be requested to process secure messaging on APDU commands and responses.
- The Current Privacy Status will not change until the GPP service has a chance to process APDU commands again, that is, until the currently selected Application is deselected. Applications for which the Privacy Requirements are not satisfied by the Current Privacy Status cannot be selected.
- If the MF is currently selected (as currently known by the OPEX; see section 4.5) and an MSE SET command is received, the OPEX shall request the GPP service to process the MSE SET command. If the MSE SET command is successfully processed by the GPP service, then the currently selected Application shall be deselected. As a consequence, the OPEX will again request the GPP service to process APDU commands, starting from the next command.

4.3 Handling of Secure Messaging

If and once established by the GPP service, secure messaging applies globally to all APDU commands, including SELECT [by AID] commands, GPP commands and MF access commands. The OPEX shall request the GPP service to process secure messaging for incoming APDU commands (i.e. unwrap) and outgoing APDU responses (i.e. wrap) as shown in the following diagram:

Figure 4-1: Runtime Secure Messaging Flow



4.4 Selection of Applications

When an APDU command is received, no Application is currently selected, and the APDU command is left unprocessed by both the GPP service and the GMF service (if any), the OPEX shall check whether this APDU command may result in the selection of an Application.

4.4.1 Implicit Selection

If there is no Application currently selected and the incoming APDU command was left unprocessed by both the GPP service and the GMF service (if any), and this APDU command is not a SELECT [by AID] command, then the OPEX shall:

1. Lookup for the “Default Selected” Application. The “Default Selected” Application is, by order of priority, the one specified using Implicit Selection Parameters (tag 'CF') then the one having the Card Reset privilege. If none is found, then the command shall be rejected and the procedure stops here.
2. Check that this Application is a Privacy Trusted Application (see section 4.7) or that the Current Privacy Status, as currently known by the GPP service, meets the Privacy Requirements of this Application (as specified by system install parameters; see section C.1). If not, the command shall be rejected and the procedure stops here.
3. Attempt to select the “Default Selected” Application. If selection fails, the command shall be rejected and the procedure stops here. Otherwise, the Application becomes the new currently selected Application.
4. Dispatch the APDU command to the currently selected Application.

4.4.2 Explicit Selection

If the incoming APDU command is a valid SELECT [by AID] command, then the OPEX shall:

1. Lookup for next candidate Application. If none can be found, the command shall be rejected and the procedure stops here.
2. Check that the candidate Application is a Privacy Trusted Application (see section 4.7) or that the Current Privacy Status, as currently known by the GPP service, meets the Privacy Requirements of this Application (as specified by system install parameters; see Annex C). If not, restart at step 1.
3. Attempt to select the candidate Application. If selection fails, the command shall be rejected and the procedure stops here. Otherwise, the Application becomes the new currently selected Application.
4. Dispatch the SELECT [by AID] command to the currently selected Application.

4.5 Access to the Global Master File

The OPEX shall manage accesses to MF files as soon as the GMF service has been successfully registered.

The OPEX shall always detect incoming SELECT MF commands (P1 = '00' and data field absent or set to '3F00') and:

- If there is no Application currently selected on the origin logical channel, the OPEX shall invoke the `GlobalMasterFile.processSelectFile(APDU)` method (of the GMF service) to process the SELECT MF command.
- If there is an Application currently selected on the origin logical channel, the OPEX shall dispatch the SELECT MF command to the currently selected Application.
- Upon successful processing of the SELECT MF command, either by the GMF service or the currently selected Application, the OPEX shall record the fact that the MF was successfully selected. This information shall be used by the OPEX to know how to process other file system related commands (see below) and to know whether the MSE SET command shall be processed by the GPP service (see section 4.2). If the command was processed by the currently selected Application, then the OPEX shall invoke the `GlobalMasterFile.selectMF()` method (of the GMF service) to notify the GMF service that the MF was successfully selected.

The OPEX shall discard information about MF selection after successful execution of a SELECT [by AID] command (i.e. successful selection of an Application).

If the MF is currently selected, then:

- The OPEX shall detect and request the GMF service to process the following APDU commands:
  - SELECT FILE [by FID] (other than SELECT MF)

    The OPEX shall invoke the `GlobalMasterFile.processSelectFile(APDU)` method (of the GMF service) to process the incoming command.
  - READ BINARY (odd or even INS byte)

    The OPEX shall invoke the `GlobalMasterFile.processReadBinary(APDU)` method (of the GMF service) to process the incoming command.
  - READ RECORD (odd or even INS byte)

    The OPEX shall invoke the `GlobalMasterFile.processReadRecord(APDU)` method (of the GMF service) to process the incoming command.
- If the GMF service indicates that the incoming command was not processed (e.g. because it does not manage the requested file) and there is an Application currently selected on the origin logical channel, then the OPEX shall dispatch the incoming command to the currently selected Application. This behavior potentially allows the currently selected Application to provide access to additional MF files not managed by the GMF service. This specific use case is supported to ensure interoperability with legacy Applications providing their own implementation of accesses to the MF.
- Otherwise, the incoming command was processed by the GMF service and the OPEX shall simply return the response and wait for the next incoming command.

If the MF is not currently selected, then:

- The OPEX shall dispatch the above commands to the currently selected Application as usual.

Note: All the APDU commands listed above are defined in [ICAO 9303] as subsets of [7816-4]. If needed, more specific choices will be made in configuration documents based on this framework regarding the options supported for file system related commands.

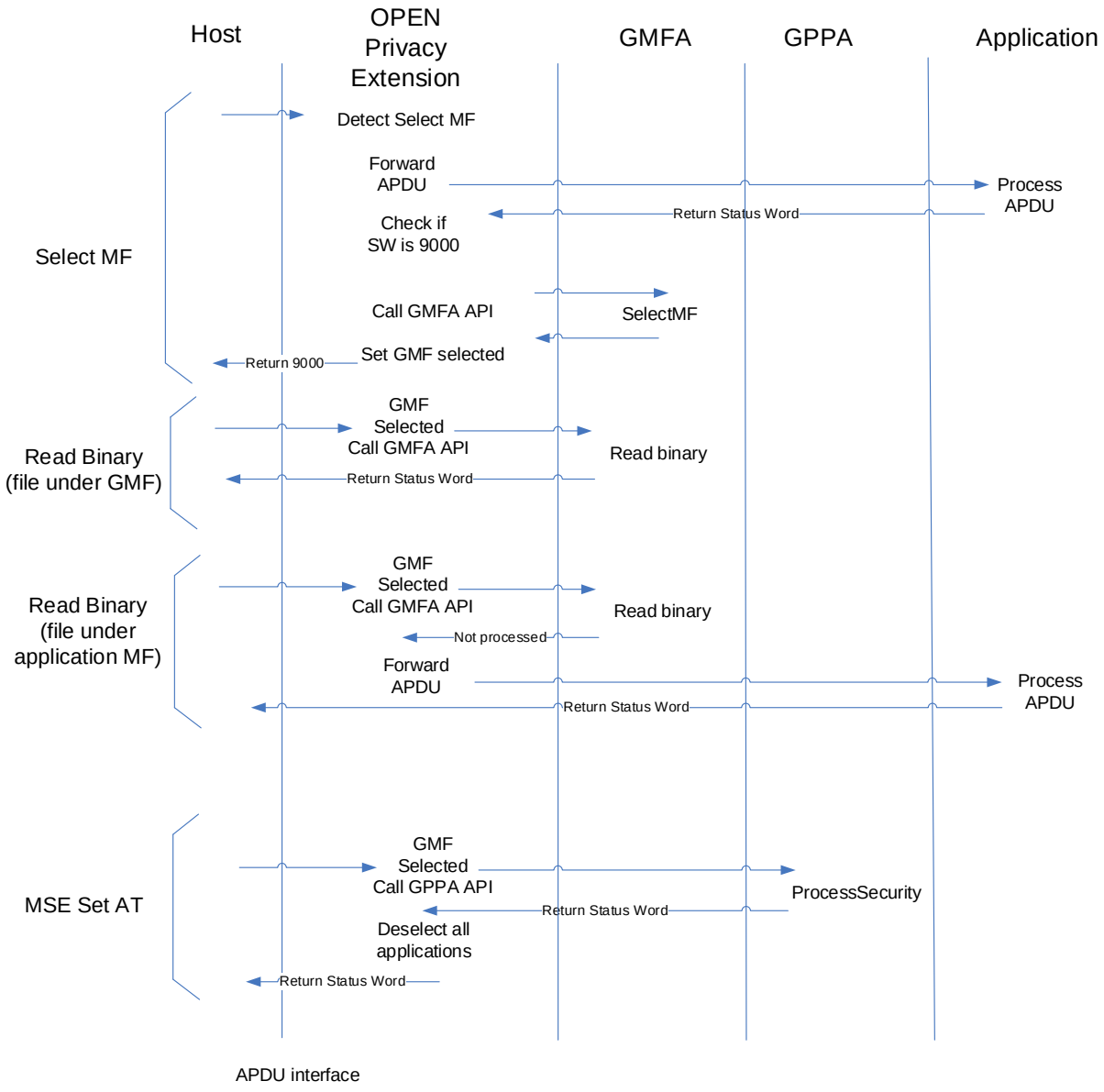
Note: The Read Record command is not mandatory. If not supported, the related API would throw an exception. If needed, the configuration documents would clarify the support of this command.

4.6 Interactions between OPEX, GMFA, GPPA, and Application

Some privacy protocols like Secure Channel Protocol '21' require reading files stored under the MF to be able to start the authentication protocol.

The following example illustrates an eMRTD application which has the Privacy Trusted privilege and potentially the Card Reset privilege (i.e. “default selected”). In the figure, it is assumed that the eMRTD application is already selected, and in the 2nd READ BINARY file, the eMRTD application serves a specific file.

Figure 4-2: Sample Interaction between OPEX, GMF, GPP, and Application



The following is a detailed explanation of the above diagram:

- **SELECT MF**
  - The OPEX detects a SELECT MF command.
  - The OPEX forwards the SELECT MF command to the currently selected Application which successfully processes the command.
  - The OPEX records the status “MF selected” and invokes `GlobalMasterFile.selectMF()` to the MF in the GMFA as well.
- **READ BINARY** (with file stored under the GMFA)
  - As the status “MF selected” is set, the OPEX detects the READ BINARY instruction.
  - The OPEX invokes `GlobalMasterFile.processReadBinary()` which returns `true` (command processed).
- **READ BINARY** (with file stored under currently selected Application)
  - As the status “MF selected” is set, the OPEX detects the READ BINARY instruction.
  - The OPEX invokes `GlobalMasterFile.processReadBinary()`which returns `false` (command not processed), meaning the file is not stored under the GMFA.
  - The OPEX then forwards the APDU command to the currently selected Application.
- **MSE SET**
  - As the status “MF selected” is set, the OPEX detects the MSE command.
  - The OPEX invokes `SecureChannelProtocol.ProcessSecurity()`
  - The process is successful and the OPEX deselects all Applications.
  - The OPEX will request the GPP service to process subsequent incoming commands (except MF access commands). (See section 4.2.)

4.7 Privacy Trusted Application

A Privacy Trusted Application is an Application that has been granted the Privacy Trusted privilege. Whereas the OPEX will prevent the selection of Applications for which the required Privacy Status has not been reached, it will not prevent the selection of Privacy Trusted Applications. Therefore, a Privacy Trusted Application is responsible for implementing and enforcing its own Specific Privacy Protocol.

It is strongly recommended for a Privacy Trusted Application to implement Secure Channel Protocol '21' Option '02' (see Annex D) or Secure Channel Protocol '22' (see Annex E).

See Annex B for the encoding of the Privacy Trusted privilege.

4.8 Application Specific Privacy Protocol

When installing an Application, one may set up as a requirement (see Annex C) that only some steps of the GPP be executed (i.e. conditions about the Current Privacy Status) before the OPEX allows the selection of that Application. In this case, the Application may be selected before all the steps of the GPP have been performed and may enforce an application specific privacy protocol by itself. It may check the Current Privacy Status using the API (see Annex A) before accepting its own selection or performing specific operations.

Such an Application may wish to use its own implementation of remaining GPP steps, or may wish to enforce other privacy mechanisms. However, it is strongly recommended that application specific privacy protocols enforce at least the remaining steps of the GPP in order to prevent any risk of privacy leakage.

Finally, notice that an Application that intends to completely override the GPP (i.e. no requirement for any of the GPP steps) shall be assigned the Privacy Trusted privilege.

Annex A Application Programming Interfaces

This document introduces a new Java Card API composed of the following packages:

- `org.globalplatform.securechannel` (version 1.0)

  This package defines new interfaces for managing and using Secure Channel Protocols. The `SecureChannelProtocol` interface extends the `org.globalplatform.SecureChannel` interface, answers a wider range of use cases, and offers more flexibility for the handling of protocol commands and the usage of secure messaging. When this package is available on a particular implementation, it shall be possible to cast `SecureChannel` instances returned by the `GPSystem.getSecureChannel()` method into `SecureChannelProtocol` instances. In addition, a new `SecureChannelKeys` interface allows managing the credentials used by a `SecureChannelProtocol` instance.

- `org.globalplatform.securechannel.provider` (version 1.0)

  This package defines means to retrieve instances of the `SecureChannelProtocol` interface (defined in the new `org.globalplatform.securechannel` package). Through this API, an Application is able to retrieve the raw implementation of some protocol (if available).

- `org.globalplatform.privacy` (version 1.0)

  This package defines additional interfaces for managing and using Privacy-enabled Secure Channel Protocols. The `PrivacyProtocol` interface shall be implemented by `SecureChannelProtocol` instances implementing a Privacy-enabled Secure Channel Protocol. The `GlobalPrivacyProtocol` interface shall be implemented by the `SecureChannelProtocol` instance registered as Global Privacy Protocol, which may be retrieved as described in section 2.3.

- `org.globalplatform.filesystem` (version 1.0)

  This package defines means to access the Global Master File. It defines the `GlobalMasterFile` interface that allows selecting and reading files stored under a master file. The unique `GlobalMasterFile` instance may be retrieved as described in section 3.2.

Annex B New Application Privileges

The following tables describe new Application privileges and how they are encoded.

Table B-1: New Application Privileges

Privilege Number	Privilege	Description	Notes
20	Privacy Trusted	An application having this privilege implements its own specific Privacy Protocol and is not impacted by the rules of the OPEN Privacy Extension.	For details see section 4.7

The new privileges are mapped on privileges byte number 3 as shown below:

Table B-2: Privileges Byte 3

b8	b7	b6	b5	b4	b3	b2	b1	Meaning	Privilege Number
1	-	-	-	-	-	-	-	Receipt Generation	16
-	1	-	-	-	-	-	-	Ciphered Load File Data Block	17
-	-	1	-	-	-	-	-	Contactless Activation	18
-	-	-	1	-	-	-	-	Contactless Self-Activation	19
-	-	-	-	1	-	-	-	Privacy Trusted	20
-	-	-	-	-	x	x	x	RFU	

Annex C New System Specific Install Parameters

C.1 Privacy Requirements

Privacy Requirements may be specified for an Application using new tag 'E0' as part of System Specific Install Parameters (tag 'EF'), as described in the following table.

Table C-1: System Specific Install Parameter for Privacy Requirements

Tag	Length	Value	Occurrence
'EF'	0-N	System Specific Install Parameters	0 to 1
…	…	…	…
'E0'	1-N	Privacy Requirements	0 to 1
'80' / 'A0'	1-N	Required Privacy Status	0 to N
'81' / 'A1'	1-N	Required Privacy Status Condition	0 to 1

The OPEN shall check the Privacy Requirements of an Application against the Current Privacy Status before authorizing the selection of that Application (see section 4.4).

Values used to specify Privacy Status and Privacy Status Conditions remain out of scope of this specification and shall be defined in specific configuration documents. Tags 'A0' and 'A1' shall be used instead of tags '80' and '81' if these values are constructed data objects (e.g. such as OIDs). Several occurrences of tag '80'/'A0' may be present. At most one occurrence of tag '81'/'A1' shall be present.

If no Privacy Requirements are specified for an Application, and unless this Application has been granted the Privacy Trusted privilege, then it will not be possible to select this Application until default Privacy Requirements have been reached. Such default Privacy Requirements remain out of scope of this specification and shall be defined in specific configuration documents.

Annex D Secure Channel Protocol '21'

Secure Channel Protocol '21' is introduced to enforce privacy requirements and refers to the mechanisms defined in [419 212] part 1. Two distinct protocol steps are defined:

- PACE (Password Authentication Connection Establishment) defined in [ICAO 9303].
  NOTE: The PACE algorithm is also defined in [419 212] part 1 section 9.
- mEAC (modular Extended Access Control) defined in [419 212] part 1 section 8.8, which uses EAC V1 or EAC V2.

Options of the protocol are defined by the “i” parameter which is a bit field encoded as described in the following table.

Table D-1: SCP '21' – Values of Parameter “i”

b8	b7	b6	b5	b4	b3	b2	b1	Description	GPP
							1	PACE	Yes
						1	0	EAC V1	No
					1	0	1	GAP (PACE and EAC V2)	Yes
x	x	x	x	x				RFU	

Options '01' (PACE) and '05' (GAP) are intended to be used as Global Privacy Protocols (GPP) but may also be used as Specific Privacy Protocol (SPP) by a Privacy Trusted Application.

Option '02' (EAC V1) cannot be used as a GPP and is only intended to be used as Specific Privacy Protocol (SPP) by Applications.

These protocol options are based on asymmetric cryptography. It is recommended to use ECDH instead of DH and ECDSA instead of RSA signature in order to obtain better performances. Notice that the usage of “PACE with Integrated Mapping and ECDH”, which is described in [419 212], is subject to patent restrictions and therefore remains out of scope of this document.

The following sections provide usage examples for the SCP '21' protocol.

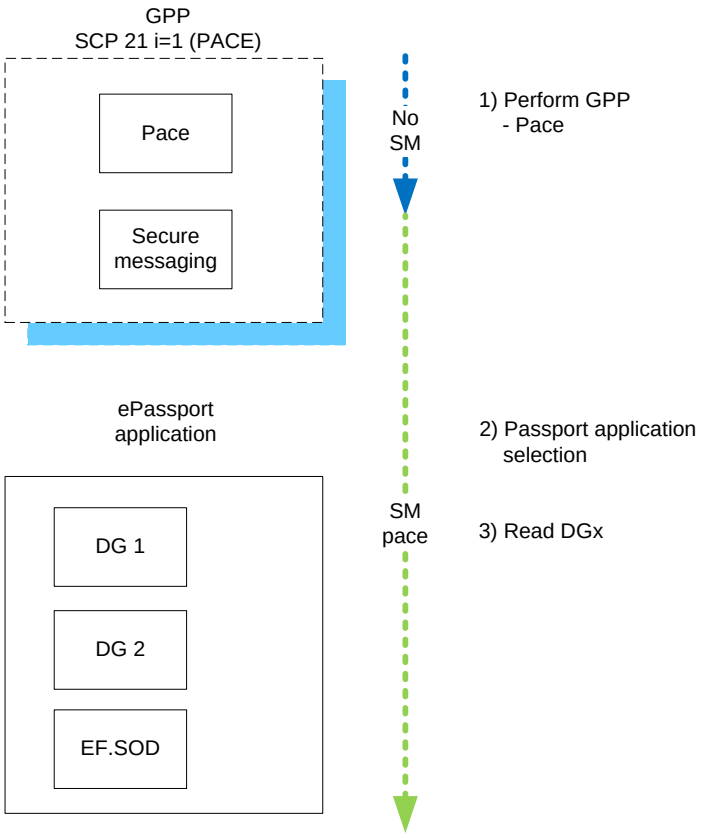
D.1 PACE Used as GPP for an eMRTD Application

This example describes the usage of the PACE protocol as GPP to protect the operation of an eMRTD application. The GPP Application registers both the GPP service and the GMF service. The GPP service implements SCP '21' with option “i” = '01' (PACE).

After card reset, the GPP is executed by the terminal (i.e. Inspection System Terminal) to establish secure messaging. The terminal selects the eMRTD application under secure messaging. The eMRTD application checks the Current Privacy Status before it allows reading DG files. The terminal reads DG files under secure messaging.

If the eMRTD application supports the BAC protocol, it should be granted the Privacy Trusted privilege (see Annex B) so that a terminal may also select it without executing the GPP and execute the BAC protocol.

Figure D-1: GPP “PACE” with eMRTD



D.2 PACE Used as GPP for an eMRTD Application Using EAC V1

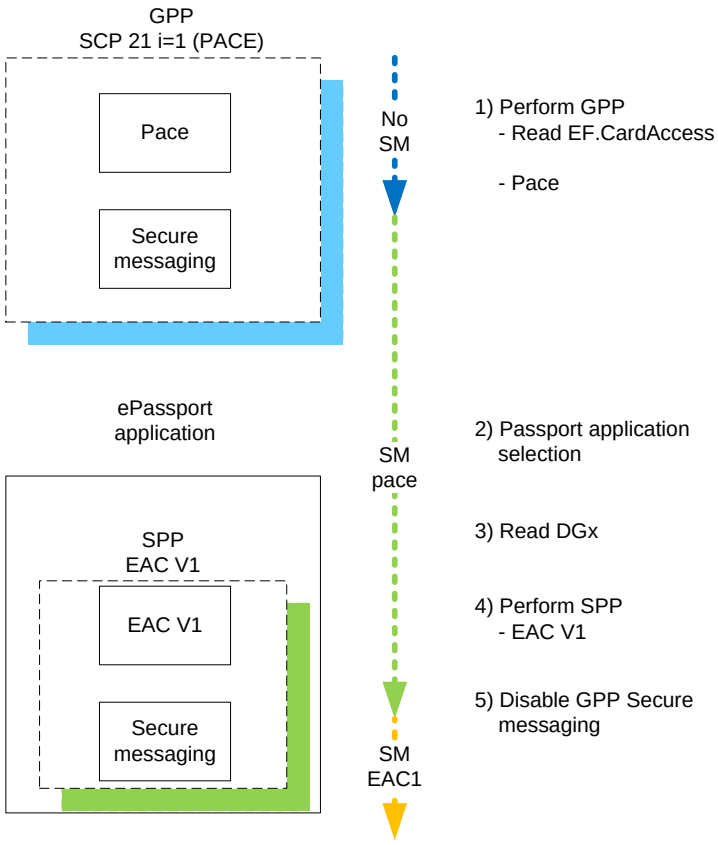
This example describes the usage of the PACE protocol as GPP to protect the Supplemental Access Control performed by an eMRTD application supporting EAC V1. The GPP Application registers both the GPP service and the GMF service. The GPP service implements SCP '21' with option “i” = '01' (PACE). The eMRTD application implements SCP '21' with option “i” = '02' (EAC V1) as an SPP.

After card reset, the GPP is executed by the terminal (i.e. Inspection System Terminal) to establish secure messaging. The terminal selects the eMRTD application under secure messaging. The eMRTD application checks the Current Privacy Status (using the `PrivacyProtocol` interface) before it allows reading DG files. The terminal reads DG files under secure messaging (i.e. DG14).

The eMRTD application then enforces its SPP. The eMRTD application deactivates the GPP secure messaging (on the origin logical channel) during the execution of the SPP using the `GlobalPrivacyProtocol` interface of the API.

If the eMRTD application supports the BAC protocol, it should be granted the Privacy Trusted privilege (see Annex B) so that a terminal may also select it without executing the GPP and execute the BAC protocol.

Figure D-2: GPP “PACE” with eMRTD EAC V1



D.3 GAP Used as GPP in a Multi-Applicative Context

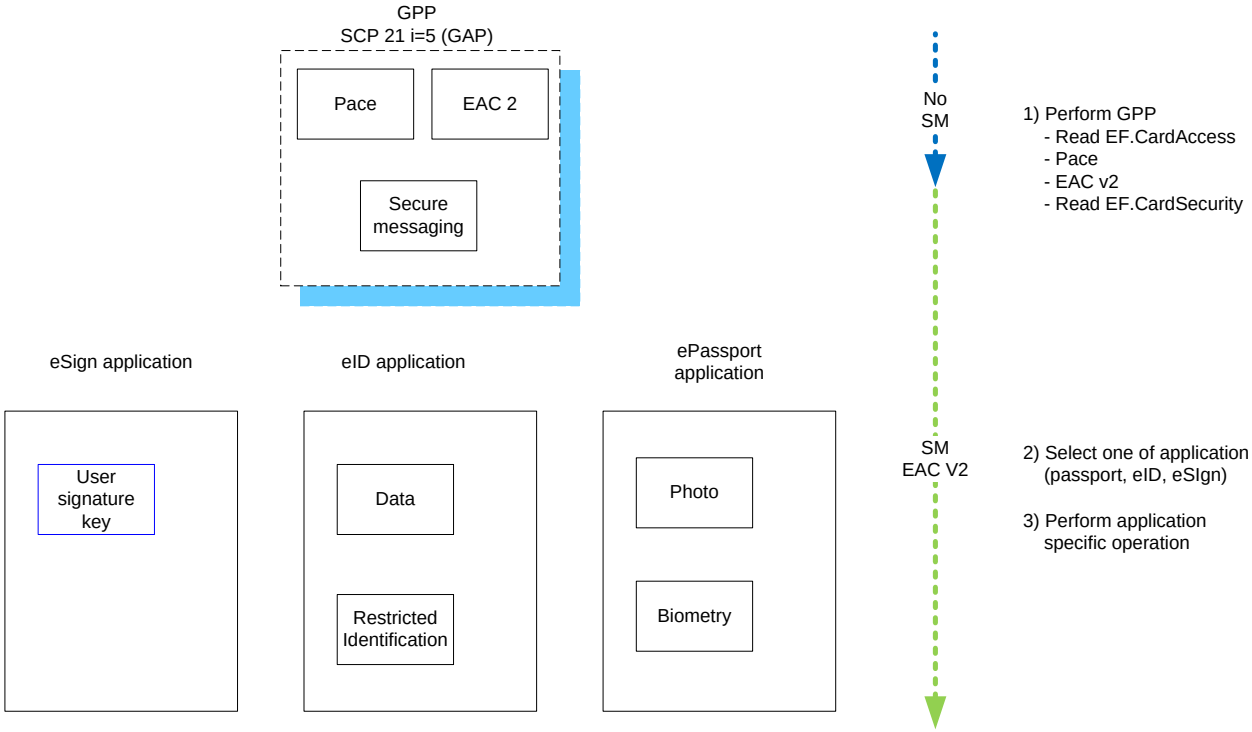
This example describes the usage of the GAP (PACE + EAC V2) protocol as GPP in order to enforce privacy requirements for several applications (e.g. eMRTD, eID, eSign, etc.) The GPP Application registers both the GPP service and the GMF service. The GPP service implements SCP '21' with option “i” = '05' (GAP).

After card reset, the GPP is executed by the terminal (i.e. Inspection System Terminal) to establish secure messaging. The terminal may select the eMRTD, eID or eSign application under secure messaging. All the commands subsequently sent to the application are protected by secure messaging.

To perform EAC V2 protocol, the terminal needs to read EF.CardSecurity and EF.ChipSecurity. These files are protected by a “read” permission (as defined in [TR 3110]), which could be enforced by the GMFA as defined in section 3.2.

Access conditions bound to the Certificate Holder Authorization Template (CHAT) can be verified using the `GlobalPrivacyProtocol` interface of the API.

Figure D-3: GPP “GAP” with eMRTD, eID, and eSign



Annex E Secure Channel Protocol '22'

Secure Channel Protocol '22' is defined in [GPCS Amd G].

This protocol shall be executed by the currently selected Application; hence, in the context of the GlobalPlatform Privacy Framework, this protocol is only intended to be implemented by Applications as a Specific Privacy Protocol (SPP). Such Applications shall be granted the Privacy Trusted privilege so that they can be selected prior to the execution of this protocol.