ACOSJ-G

User Manual • Uživatelský manuál • Užívateľský manuál • Használati utasítás • Benutzerhandbuch

Dear customer,

Thank you for purchasing our product. Please read the following instructions carefully before first use and keep this user manual for future reference. Pay particular attention to the safety instructions. If you have any questions or comments about the device, please contact the customer line.

✉ www.alza.co.uk/kontakt

✆ +44 (0)203 514 4411

Importer Alza.cz a.s., Jankovcova 1522/53, Holešovice, 170 00 Prague 7, www.alza.cz

Overview

The ACOSJ is a smart card operating system developed by Advanced Card Systems Ltd. It works based on the JAVA Card Virtual Machine and complies with GlobalPlatform Card Specification Version 2.2.1, JAVA Card Virtual Machine and complies with GlobalPlatform Card Specification Version 2.2.1, JAVA Card Specification Version 3.0.4 and Mapping Guidelines 1.0.1 on its functions and configurations.

The purpose of this document is to describe in detail the features and functions of the ACOSJ smart card operating system.

ACOSJ-G Versions

Version	Data Released	Modifications
ACOSJ v1.01	June 2015	• 40KB EEPROM • Operating Voltage: 2.7V to 5.5V
ACOSJ v2.00	May 2018	• 95KB EEPROM • Operating voltage: 2.1V to 5.5V • Changes in the configuration memory address
ACOSJ v2.04	April	• 95KB EEPROM • Operating voltage: 2.1V to 5.5V • Changes in the configuration memory address • Features enhancement

Symbols and Abbreviations

Abbreviation	Description
AES	Advanced Encryption Standard
AID	Application Identifier
APDU	Application Protocol Data Unit
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ATR	Answer-to-Reset
ATQ	Answer-to-Request (for contactless cards)
BCD	Binary Coded Decimal
BER	Basic Encoding Rules
CAT	Card Application Toolkit: or Cryptographic Authorization Template
CBC	Cipher Block Chaining
CCT	Control Reference Template for Cryptographic Checksum
CIN	Card Image Number/Card Identification Number
CLA	Class byte of the command message
CRT	Control Reference Template
CT	Control Reference Template for Confidentiality
CVM	Cardholder Verification Method
DAP	Data Authentication Pattern

DEK	Data Encryption Key
DER	Distinguished Encoding Rules
DES	Data Encryption Standard
DST	Control Reference Template for Digital Signature
ECB	Electronic Code Book
EMV	Europay, Mastercard and VISA: used to refer to the ICC Specifications for Payment Systems
ENC	Encryption
FCI	File Control Information
HEX	Hexadecimal
HMAC	Keyed-Hash Message Authentication Code
ICC	Integrated Circuit Card
ICV	Initial Chaining Vector
IIN	Issuer Identification Number
INS	Instruction byte of the command message
ISO	International Organization for Standardization
Lc	Exact length of data in a case 3 or case 4 command
Le	Maximum length of data expected in response to a case 2 or case 4 command
LV	Length Value
MAC	Message Authentication Code
MEL	MULTOS Executable Language. The instruction set of the MULTOS™ runtime environment
OID	Object Identifier
P1	Reference control parameter 1
P2	Reference control parameter 2
PIN	Personal Identification Number
PKI	Public Key Infrastructure
RAM	Random Access Memory
RFU	Reserved for Future Use
RID	Registered Application Provider Identifier
ROM	Read-only Memory
RSA	Rivest-only Memory
SCP	Secure Channel Protocol: or (ETSI) Smart Card Platform
SW	Status Word
SW1	Status Word One
SW2	Status Word Two
TLV	Tag Length Value
TP	Trust Point
“xx”	Hexadecimal values are expressed as hexadecimal digits between single quotation marks.
“X”	A value in a cell of a table whose purpose is described in the “Meaning” column of the table.
“-“	A value (0 or 1) in a cell of a table that does not affect the “Meaning” given for that row of the table.

Card Specifications

This section summarizes the features and functionalities of the ACOSJ-G.

Electrical Specifications

Operating Voltage (ACOSJ-G v1.01):	2.7 V to 5.5 V supply voltages
Operating Voltage (ACOSJ-G v2.04):	2.1 V to 5.5 V supply voltages
Maximum External Clock Frequency:	10 MHz
Maximum CPU Clock Frequency:	28 MHz
ESD protection	greater than 5 kV (HBM)

Environmental Specifications

Operating Temperature:	-25°C to +85°C

Communication Protocols

- T=CL protocol with baud up to 848 kbps

Memory

Capacity:	95 KB (ACOSJ-G 2.04), 40 KB (ACOSJ-G 1.01)
EEPROM Endurance:	500,000 erase/write cycles (25°C)
Data Retention:	30 years (25°C)

Cryptographic Functionalities

DES:	2K3DES,3K3DES (ECB and CBC)
AES:	128/192/256 bits (ECB and CBC)
RSA:	768 to 2048 bits
ECC:	Modulus 112/128/160/192/224/256/384 bits
Hash:	SHA1, SHA256, SHA384, SHA512
SEED:	128 bits

- SM2/SM3/SM4

Compliance to Standards

- Compliance with ISO 7816 Part 4
- Compliance with ISO 14443 (Type A and B)
- Compliance with JAVA Card Specification Version 3.0.4
- Compliance with Global Platform Specification Version 2.2.1
- Compliance with Mapping Guidelines 1.0.1

Answer to Select (ATS, Contactless Card)

After receiving a Request for Answer to Select (RATS) command from the card reading device, the card transmits and Answer to Select (ATS) in compliance with ISO 14443 Part 4.

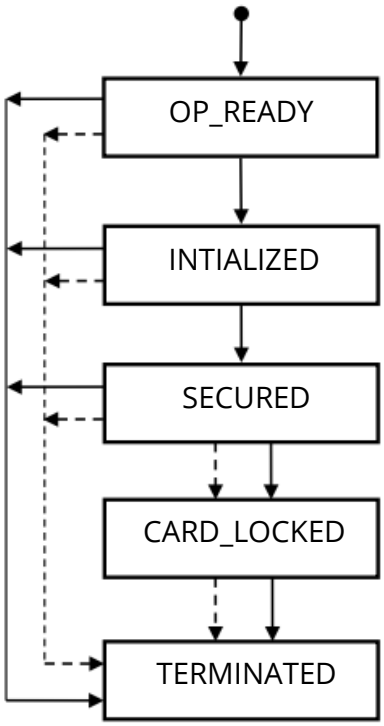
The following table shows the default ATS:

Parameter	ATS	Description
TL	0Eh	Length
T0	78h	Formal byte…codes Y(1) and FSCI
TA1	00h	Interface byte…codes DS and DR
TB1	71h	Codes FWI and SFGI
TC1	02h	Codes protocol options
T1	41h	Indicates “A”
T2	43h	Indicates “C”
T3	4Fh	Indicates “O”
T4	53h	Indicates “S”
T5	4Ah	Indicates “J”
T6	76h	Indicates “v”
T7~T9	31h 30h 31h or 32h 30h 34h	Indicates “101” Or Indicates “204”

Note: For full description of the ATS, kindly refer to ISO 14443 Part 4.

Card Life Cycle States

The ACOSJ has five card states: OP_READY, INTILIZED, SECURED, CARD_LOCKED and TEDMINATED. The figure below shows the card life cycle state transition:



Legend

Privileged Security Domain

Privileged Application

OP_READY

This state indicates that the runtime environment shall be available and the Issuer Security Domain, acting as the selected Application, shall be ready to receive, execute and respond to APDU commands.

The following functionalities shall be present when the card is in the state OP_READY:

- The runtime environment shall be ready for execution.
- The OPEN shall be ready for execution.
- The Issuer Security Domain shall be the implicitly selected Application for all card interfaces.
- Executable Load Files that were included in Immutable Persistent Memory shall be registered in the GlobalPlatform Registry.
- An initial key shall be available within the Issuer Security Domain.

The card shall be capable of Card Content changes, the loading of the Load Files containing applications not already present in the card may occur.

The installation, from Executable Load Files, of aby Application may occur.

Additionally, if any personalization information is available at this stage, Applications may be personalized.

The OP_READY state may be used by an off-card to perform the following actions:

- Supplementary Security Domains may be loaded and/or installed.
- The Security Domain keys may be inserted in order to maintain a cryptographic key separation from the Issuer Security Domain keys.

Initialized

This state is an administrative card production state. The state transition from OP_READY to INITIALIZED is irreversible. Its functionality is beyond the scope of this Specification. This state may be used to indicate that some initial data has been populated (e.g., Issuer Security Domain keys and/or data) but the card is not yet ready to be issued to the Cardholder.

Secured

This state is the intended operating card Life Cycle State in Post-Issuance. This state may be used by Security Domains and Applications to enforce their respective security policies. The state transition from INITIALIZED to SECURED is irreversible.

The SECURED state should be used to indicate to off-card entities that the Issuer Security Domain contains all necessary keys and security elements for full functionality.

Card_Locked

The card Life Cycle state CARD_LOCKED is present to provide the capability to disable the selection of Security Domain and Applications. The card Life Cycle state transition from SECURED to CARD_LOCKED is reversible.

Setting the card to the CARD_LOCKED state means that the card shall only allow selection of the application with the Final Application privilege.

Card Content changes, including any type of data management (specifically Security Domain keys and data), are not allowed in this state.

Either the OPEN or a Security Domain with Card Lock privilege, or an Application with Card Lock privilege, may initiate the transition from the state SECURED to the state CARD_LOCKED.

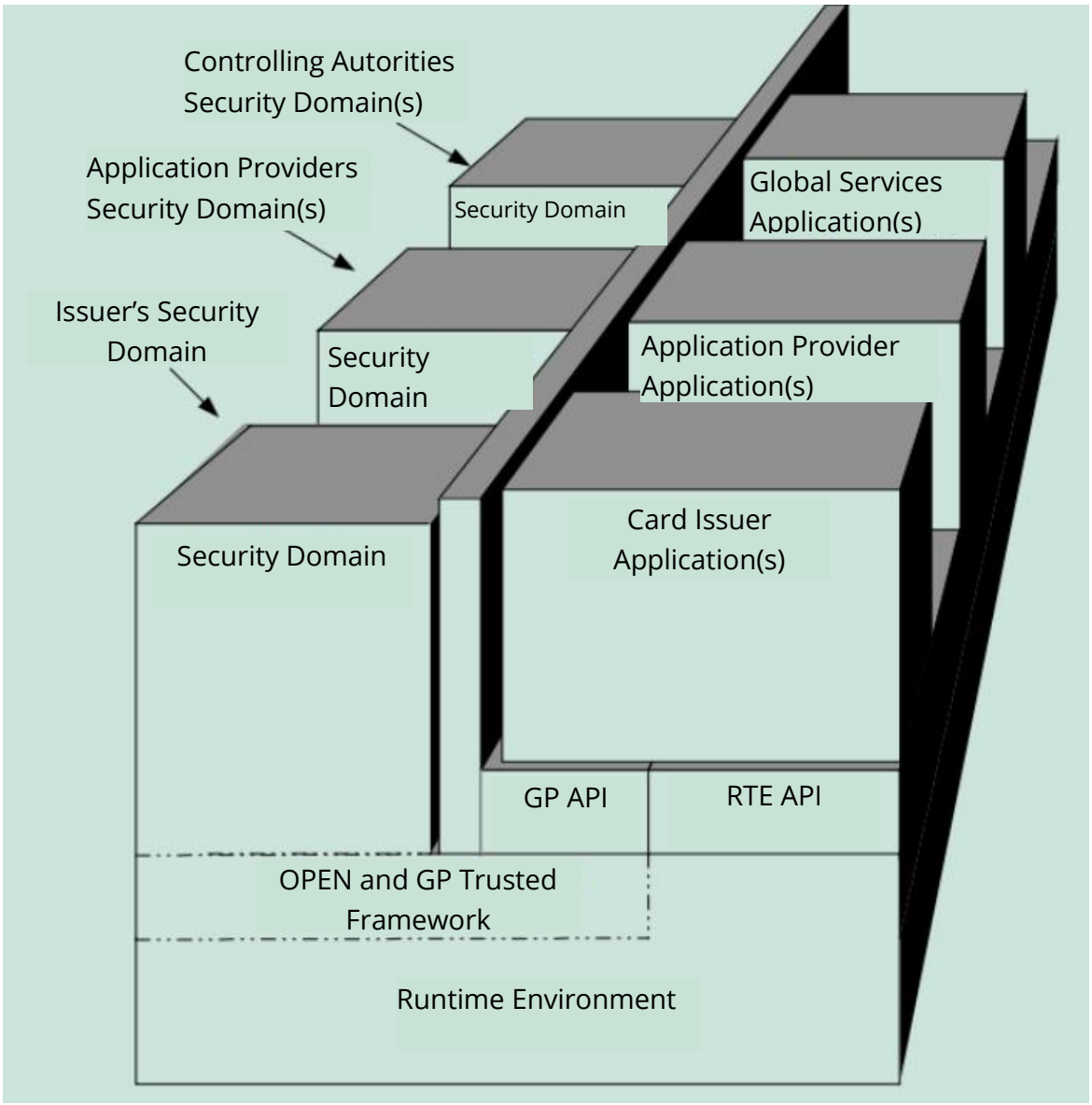
TERMINATED

This state signals the end of the card Life Cycle and the card. The state transition from any other state to TERMINATED is irreversible.

The state TERMINATED shall be used to permanently disable all card functionality with respect to any card content management and any life cycle changes. This card state is intended as a mechanism for an application to logically “destroy” the card for such reasons as the detection of severe security threat or expiration of the card. If a Security Domain has the Final Application privilege only the GET DATA command shall be processed, all other commands defined in this specification shall be disabled and shall return an error. Id an application has the Final Application privilege its command processing is subject to issuer policy.

The OPEN itself, or a Security Domain with Card Terminate privilege, or an Application with Card Terminate privilege, may initiate the transition from any of the previous states to the state TERMINATED.

Card Architecture



GP APDU Command Reference

This section details the GlobalPlatform APDU commands that may be implemented. The commands are listed alphabetically.

Summarizes the APDU commands that are supported by the Issuer Security Domains, and the requirements for support of these APDU commands by other Security Domains. When logical channels are supported, the MANAGE CHANNEL command is only processed by the OPEN and no Security Domain support is required for this command.

Command	OP_READY			INITIALIZED			SECURED			CARD_LOCKED		TERMINATED	
	AM SD	DM SD	SD	AM SD	DM SD	SD	AM SD	DM SD	SD	FASD	SD	FASD	SD
DELETE Executable Load File													
DELETE Executable Load File and related Application(s)													
DELETE Application	✓			✓			✓						
DELETE Key													
GET DATA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
GET STATUS	✓			✓			✓			✓			
INSTALL [for load]													
INSTALL [for install]													
INSTALL [for load, install and make selectable]													
INSTALL [for install and make selectable]	✓	✓		✓	✓		✓	✓					
INSTALL [for make selectable]													
INSTALL [for extradition]													
INSTALL [for registry update]													
INSTALL [for personalization]													
LOAD													
PUT KEY	✓			✓			✓						
SELECT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
SET STATUS	✓			✓			✓			✓			
STORE DATA	✓			✓			✓						

AM SD	Security Domain with Authorized Management privilege.
DM SD	Security Domain with Delegated Management privilege.
FA SD	Security Domain with Final Application privilege.

Note: Command support for and Application with Final Application Privilege which is not a Security Domain is subject to Issuer policy, within the constraints of what is permitted according to the Card Life Cycle State.

SD	Other Security Domain.
✓	Support required.
Blank Cell	Support optional.
Grey Cell	Support prohibited.

Summarizes the minimum-security requirements for the APDU commands.

Command	Minimum Security
DELETE	Secure Channel initiation or digital signature verification
GET DATA	None
GET STATUS	Secure Channel initiation
INSTALL	Secure Channel initiation or digital signature verification
LOAD	Secure Channel initiation or digital signature verification
MANAGE CHANNEL	Not applicable
PUT KEY	Secure Channel initiation
SELECT	Not applicable
SET STATUS	Secure Channel initiation
STORE DATA	Secure Channel initiation

General Coding Rules

Please refer to the ACOSJ Reference Manual.

DELETE Command

The DELETE command is used to delete a uniquely identifiable object such as an Executable Load File, an Application, and Executable Load File and its related Applications or a key. In order to delete and object, the object should be uniquely identifiable by the selected Application.

GET DATA Command

The GET DATA command is used to retrieve either a single data object, which may be constructed or a set of data objects. Reference control parameters P1 and P2 coding are used to define the specific data object tag. The data object may contain information pertaining to a key.

GET STATUS Command

The GET STATUS command is used to retrieve Issuer Security Domain, Executable Load File, Executable Module, Application or Security Domain Life Cycle status information according to a given match/search criteria.

INSTALL Command

The INSTALL command is issued to a Security Domain to initiate or perform the various steps required for Card Content management.

LOAD Command

This section defines the structure of the Load File transmitted in the LOAD command data field for loading a Load File. The ICC internal handling or storage of the Load File is beyond the scope of this Specification.

Multiple LOAD commands may be used to transfer a Load File to the card. The Load File is divided into smaller components for transmission. Each LOAD command shall be numbered starting at 00h. The LOAD command numbering shall be strictly sequential and increments by one. The card shall be informed of the last block of the Load File.

After receiving the last block of the Load File, the card shall execute the internal processes necessary for the Load File and any additional processes identified in the INSTALL [for load] command that preceded the LOAD commands.

MANAGE CHANNEL Command

The MANAGE CHANNEL command is processed by the open on cards that are aware of logical channels. It is used to open and close Supplementary Logical Channels. The Basic Logical Channel (channel number zero) can never be closed.

PUT KEY Command

The PUT KEY command is used to either:

- Replace an existing key with a new key: The new key has the same or a different Key Version Number but the same Key Identifier as the key being replaced.
- Replace multiple existing keys with new keys: The new keys have the same or a different Key Version Number (identical for all new keys) but the same Key Identifiers as the keys being replaced.
- Add a single new key: The new key has a different combination Key Identifier/Key Version Number than that of the existing keys.
- Add multiple new keys: The new keys have different combinations of Keys Identifiers/Key Version Number (identical to all new keys) than that of the existing keys.

When the key management operation requires multiple PUT KEY commands, chaining of the multiple PUT KEY commands is recommended to ensure integrity of the operation.

In this version of the Specification the public values of asymmetric keys are presented (transmitted) in clear text.

SELECT Command

The SELECT command is used for selecting an Application. The OPEN only processes SELECT commands indicating the SELECT [by name] shall be passed to the currently selected Security Domain or Application on the indicated logical channel.

SET STATUS Command

The SET STATUS command shall be used to modify the card Life Cycle State or the Application Life Cycle State.

STORE DATA Command

The STORE DATA command is used to transfer data to an Application or the Security Domain processing the command.

The Security Domain determines if the command is intended for itself or an application depending on a previously receiver command. If a preceding command was and INSTALL [for personalize] command, the STORE DATA command is destined for an application.

Multiple STORE DATA command are used to send data to the Application or Security Domain by breaking the data into smaller components for transmission. The Security Domain shall be informed of the last block.

A personalization session starts when a Security Domain receives a valid INSTALL [for personalize] command designating an Application (implementing either the Application or the Personalization interface) to which the Security Domain shall be forward subsequently received STORE DATA commands.

A personalization session ends when:

- The card is reset.
- The Security Domain is deselected (i.e., another, or the same, applet is selected on the same logical channel)
- The Security Domain is selected on the same or another logical channel.
- The Secure Channel session (if any) established by the Security Domain is reset, possibly by the targeted application.
- The Security Domain receives and INSTALL [for personalize] command (starting a new personalization session for another application)
- The Security Domain receives a STORE DATA command indicating P1. b8=1 (last block)

Any STORE DATA command received out of such a personalization session shall be processed by the Security Domain itself.

GlobalPlatform API

Global Platform on a JAVA Card

This section contains only the API required for Global Platform 2.2x Java Cards. Use of the Open Platform 2.0.1 API is still allowed for supporting older versions of Applications but is deprecated. It is defined in version 2.1.1 of this Specification.

The deprecated API and new API both access the same objects where applicable. While this may seem obvious for methods that have the same name across both classes (e.g. setATRHistBytes(), setCardContentState() and getCardContentSame()) it shall also be noted as for example that an application that uses the updated() method in the new API to change the value of the global PIN will affect the same global PIN of an application that uses the setPIN() method in the deprecated API to verify the global PIN.

GlobalPlatform Specific Requirements

In order to ensure the highest level of interoperability of GlobalPlatform implementations, GlobalPlatform also adopts the order defined in section 6.2 – Java Card™ 2.2x Virtual Machine Specification.

The following minor modifications to the standard functionality defined in the Java Card™ 2.1.1 Runtime Environment (JCRE) Specifications and the Java Card™ 2.1.1 Application Programming Interface are present in the implementation of GlobalPlatform. Some of the modifications are already consistent with the Java Card™ 2.2.x Specification and therefore not modifications anymore.

GPRegistryEntry objects shall be implanted as Shareable Interface Objects as defined in section 6.2.4 “Shareable Interfaces” of the Java Card™ 2.2.x Runtime Environment (JCRE) Specification in order to ensure shared access.

GlobalPlatform package AID

Each GlobalPlatfrom package AID will be concatenation of RID and a PIX. The AID value of the Java Card Export File for the new GlobalPlatform API (identical for both GlobalPlatform 2.1 and 2.1.1) based on the RID specified in Appendix H – GlobalPlatform Data Values and Card Recognition Data is “A00000015100”.

Installation

In section 3.1 – The Method install of the Java Card™ 2.2.x Runtime Environment (JCRE) Specifications, the parameters passed to the method are defined to be initialization parameters from the contents of the incoming byte array parameter.

This specification expands on this requirement and further defines the content of the Install Parameters. This expansion affects both the implementation of an OPEN and the behaviour of Java Card applet developed for a GlobalPlatform card. It does not affect the definition of the install () method of the Class Applet of the Java Card™ 2.2.x Application Programming Interface specification.

The Install Parameters shall identify the following data, present in the INSTALL [for install] command:

- The instance AID
- The Privileges
- The Application Specific Parameters. (While the APDU command contains Install Parameters representing TLV coded system and Application Specific Parameters, the application only required knowledge of the Application Specific Parameters, i.e. only LV of the TLV coded structure “C9” are present as application specific parameters.

The OPEN is responsible for ensuring that the parameters (bArray, bOffset and bLength) contain the following information:

The array, bArray, shall contain the following consecutive LV coded data:

- Length of the instance AID
- The instance AID
- Length of the Privileges
- The Privileges
- Length of the Application Specific Parameters
- The Application Specific Parameters

The byte, bOffset, shall contain an offset within the array pointing to the length of the instance AID.

The byte, bLength, shall contain a length indicating the total length of the above-defined data in the array.

The applet is required to utilize the instance AID as a parameter when invoking the register (byte [] bArray, short bOffset, byte bLength) method of the Class Applet of the Java Card 2.2.x Application Programming Interface specification.

T=0 Transmission Protocol

GlobalPlatform cards are intended to be functional in the widest range of environments (i.e., Card Acceptance Devices). Currently the Java Card 2.1.1 Runtime Environment (JCRE) Specifications and Java Card 2.2.x Runtime Environment (JCRE) Specifications describe the behaviour for case 2 commands (when suing the T=0 protocol) in contradiction to EMV 2000. GlobalPlatform mandates that the JCRE shall handle this case of command in accordance with ISO/IEC 7816: An applet receiving a case 2 command build the response and invokes the appropriate API to output the data. If the data is less than the data expected by the terminal, the OPEN will, store the data and output a “6Cxx” response code and wait for the CAD to re-issue the command with the correct length. When the re-issued command is received the JCRE will manage the outputting the stored data.

Atomicity

Unless otherwise specified all internal persistent objects of the GlobalPlatform API must be conform to a transaction in progress.

All operations performed by this API, expect the Application.processData() method shall be executed atomically. Objects used to enforce the implementation of velocity checking shall not conform to a transaction in progress.

Logical channels

The following logical channel restrictions apply to Java Card 2.2.x (see the Java Card 2.2.x Runtime Environment (JCRE) Specifications for more details):

Selection of an Application on a logical channel as defined in section 6.3 – Command Dispatch will be unsuccessful if this same Application, or any other Application instantiated from code in the same package from which the Application being selected was instantiated, is currently selected on another logical channel but the application code does not implement the MultiSelectable interface. Security Domains shall implement the MultiSelectable interface.

Changing context from a Security Domain to an Application as defined in section 7.3.3 Personalization Support will be unsuccessful if this same Application, or any other Application instantiated from code in the same package from which the Application being personalized was instantiated is currently selected on another logical channel, but the application code does not implement the MultiSelectable interface.

An Application that has the privilege of being selected by default and is intended for a card that supports Supplementary Logical Channels should implement the MultiSelectable interface.

GlobalPlarform only defines the assignment of logical channel numbers by the card. Optionally and as defined in Java Card 2.2, a card may also support assignment of logical channel numbers by the terminal.

Cryptographic Algorithms

GlobalPlatform cards supporting RSA cryptography should support key sizes not defined as constants in the Key Builder class. More specifically, support for key sizes being a multiple of 4 bytes (32 bits), and being within the allowed key lengths defined by the implementation should be available.

Level of trust

The Java Card 2.2.x specifications assume that the RID of the AID packages, applets and instances will be utilized to ensure a level of trust between these entities. In section 4.2.2-AID Usage of the Java Card 2.2.1 Application Programming Interface it is defined that the RID of an AID of a component must match the RID of the AID of the package, and in the definition of the register (byte [] bArray, short bOffset, byte bLength) method of the Java Card 2.2.x Application Programming Interface specification it is defined that

an exception must be thrown if the RID portion of the AID bytes in the bArray parameter does not match the RID portion of the Java Card name of the applet.

From a real-world implementation point of view, mandating that the RID of the instance AID must be the same as the RID of the component from which it was instantiated is not practical. GlobalPlatform implementations shall not mandate that there be any ling through the AID of an instance to its original package. (Requiring no connections between the AID of an instance and the AID of its original package). It does however assume that all applications in the same package share the same level of trust.

Invocation of GlobalPlatform methods

The Application Programming Interface defined herein is accessible to any Java Card applet developed with the intention of being present on a GlobalPlatform card. One limitation does not exist relating to the constructor of the applet and to install () method of the Class Applet of the Java Card 2.2.x Application Programming Interface. As this specification does not define exactly when the instance of an applet becomes an entry in the card's GlobalPlatform Registry, an applet developer can only assume that this has occurred following the successful completion of the install () method. To ensure interoperability, GlobalPlatform API methods that access to the GlobalPlatform Registry entry of the applet invoking the method, shall not be invoked from within the constructor or the install() method.

The following is a lost of methods that may be invoked from within the constructor or the install() method:

- getCardState;
- getCVM;
- getService;

The required behaviour of the card in the event that an application incorrectly invokes a method of the org.globalplatform.GPSystem class other than those listed above is undefined. For example, an exception may be thrown and the processing of the install() method may be aborted.

Selection

On GlobalPlatform cards, if an error occurs during the processing of the select() method or if the select() method returns False or if the Application cannot be selected because it does not implement the Multiselectable interface, the OPEN shall continue searching through the GlobalPlatform Registry for a subsequent full or partial match as defined in section 6.4.2.1.2. If no Application is selected, the corresponding logical channel remains open with no currently selected Application. Since no Application is currently selected, any subsequent command, other than a MANAGE CHANNEL or SELECT command, will be rejected. It is expected that the off-card entity will take appropriate action on such an error, e.g., select another Application, close the corresponding logical channel, reset or power off the card.

The Method Summary and Details for the GlobalPlatform Java Card API specification is now available in a separate document, which may be found on the GlobalPlatform website.

Pre-personalization

When the IC is powered for the first time it is in the pre-personalization state.

The INIT_CARD Command (see 8.2.2 INIT CARD Command) must be the first command at the pre-personalization state, the command initializes the card memory area with default values, and then a so-called ACOSJ ROOT Application (see 8.0 ACOSJ ROOT Application) is available to write critical system data (for example, “ATR”, “ATS” and so on) into the EEPROM. Once the data has been loaded and confirmed no further changes a command that disables the ACOSJ ROOT Application must be executed.

When the IC exits the pre-personalization state with the successful execution of the Active Card command (see 8-2.5 ACTIVATE Command), it is in the default GlobalPlatform card life cycle OP_READY state if it is not set to a different life cycle state.

If is not possible to return to the pre-personalization state once that active card command has been processed successfully.

Note: The Transport Key is the ACOSJ Root Application pre-personalization key, and the Transport Key must be requested from ACS.

ACOSJ ROOT Application

ACOSJ ROOT Application Description

The commands in the ROOT application can be used only after the ROOT application is correctly selected (using the select command the Transport Key). After entering into the ROOT Application, the user can read or configure card parameters though commands supported by the ROOT Application. To exit ROOT Application and select another application, the card must be reset.

Under the ROOT Application, parameters of the card can be read or configured.

The ROOT Application will become invalid once the card is activated.

ACOSJ ROOT Application Command Reference

SELECT Command

The SELECT command is used to select the ROOT application.

INIT_CARD Command

This command is used to reset the card. This command will initialize all the EEPROM.

READ Command

This command is used to read from the configuration area. Configuration parameters of the card can be read though this command.

WRITE Command

This command is used to write data to the configuration area. Configuration parameters of the card can be read though this command.

ACTIVE Command

This command is used to activate the card. Once this command is implemented successfully, the ROOT Application will become invalid, and the configuration data of the card cannot be read or set directly anymore.

ACOSJ IDENTIFY Application

ACOSJ IDENTIFY Application Description

After the IDENTIFY Application (AID: 6163732E636F732E61636F736A766572) is selected with SELECT Command, ACOSJ will return the version number of ACOSJ and indicates whether the car has been activated.

ACOSJ IDENTIFY Application Command Reference

SELECT Command

The SELECT command is used for selecting the INDENTIFY Application.

Status Bytes

SW1 SW2	Description
90 00h	Correct command
61 XX	SW2 more response bytes available
62 83h	Application permanently blocked
63 00h	Verification failed
63 CX	X retries left
67 00h	Wrong length of data
68 81h	Logical channel not supported
68 82h	Secure messaging not supported
69 84h	Invalid data
69 82h	Security status not satisfied
69 85h	Conditions of use not satisfied
69 88h	Wrong MAC
6A 86h	Incorrect P1 P2
6A 80h	Incorrect values in command data
6A 82h	File not found
6A 81h	Application locked
6A 83h	Record not found
6A 84h	Not enough memory space
6A 88h	Referenced data not found
6D 00h	Invalid INS, Command not existing
6E 00h	Invalid Class

References

The documents below served as references for the ACOSJ Functional Specification:

- GlobalPlatform Card Specification Version 2.2.1
- GlobalPlatform Card API Version 1.6
- Java Card 3 API, Classic Edition Version 3.0.4
- Java Card 3 Platform Runtime Environment Specification, Classic Edition Version 3.0.4 September 2011
- Java Card 3 Platform Virtual Machine Specification, Classic Edition Version 3.0.4 September 2011
- GlobalPlatform Card Mapping Guidelines of Existing GP v2.1.1 Implementation on v2.2.1 Version 1.0.1

Warranty Conditions

A new product purchased in the Alza.cz sales network is guaranteed for 2 years. If you need repair or other services during the warranty period, contact the product seller directly, you must provide the original proof of purchase with the date of purchase.

The following are considered to be a conflict with the warranty conditions, for which the claimed claim may not be recognized:

- Using the product for any purpose other than that for which the product is intended or failing to follow the instructions for maintenance, operation, and service of the product.
- Damage to the product by a natural disaster, the intervention of an unauthorized person or mechanically through the fault of the buyer (e.g., during transport, cleaning by inappropriate means, etc.).
- Natural wear and aging of consumables or components during use (such as batteries, etc.).
- Exposure to adverse external influences, such as sunlight and other radiation or electromagnetic fields, fluid intrusion, object intrusion, mains overvoltage, electrostatic discharge voltage (including lightning), faulty supply or input voltage and inappropriate polarity of this voltage, chemical processes such as used power supplies, etc.
- If anyone has made modifications, modifications, alterations to the design or adaptation to change or extend the functions of the product compared to the purchased design or use of non-original components.

Vážený zákazníku,

Děkujeme vám za zakoupení našeho produktu. Před prvním použitím si prosím pečlivě přečtěte následující pokyny a uschovejte si tento návod k použití pro budoucí použití. Zvláštní pozornost věnujte bezpečnostním pokynům. Pokud máte k přístroji jakékoli dotazy nebo připomínky, obraťte se na zákaznickou linku.

✉ www.alza.cz/kontakt

✆ +420 255 340 111

Dovozce Alza.cz a.s., Jankovcova 1522/53, Holešovice, 170 00 Praha 7, www.alza.cz

Přehled

ACOSJ je operační systém pro čipové karty vyvinutý společností Advanced Card Systems Ltd. Pracuje na základě virtuálního stroje karet JAVA a je v souladu se specifikací karet GlobalPlatform verze 2.2.1, virtuálním strojem karet JAVA a svými funkcemi a konfiguracemi je v souladu se specifikací karet GlobalPlatform verze 2.2.1, specifikací karet JAVA verze 3.0.4 a pokyny pro mapování 1.0.1.

Účelem tohoto dokumentu je podrobně popsat vlastnosti a funkce operačního systému čipových karet ACOSJ.

Verze ACOSJ-G

Verze	Zveřejněná data	Úpravy
ACOSJ v1.01	červen 2015	• 40KB EEPROM • Provozní napětí: 2,7 V až 5,5 V
ACOSJ v2.00	květen 2018	• 95KB EEPROM • Provozní napětí: 2,1 V až 5,5 V • Změny adresy konfigurační paměti
ACOSJ v2.04	duben	• 95KB EEPROM • Provozní napětí: 2,1 V až 5,5 V • Změny adresy konfigurační paměti • Vylepšení funkcí

Symboly a zkratky

Zkratka	Popis
AES	Pokročilý standard šifrování
AID	Identifikátor aplikace
APDU	Datová jednotka aplikačního protokolu
API	Rozhraní pro programování aplikací
ASCII	Americký standardní kód pro výměnu informací
ATR	Odpověď na obnovení
ATQ	Odpověď na dotaz (pro bezkontaktní karty)
BCD	Binárně kódované desetinné číslo
BER	Základní pravidla kódování
CAT	Sada nástrojů pro aplikaci karet: nebo Šablona kryptografické autorizace
CBC	Řazení šifrových bloků
CCT	Kontrolní referenční šablona pro kryptografický kontrolní součet
CIN	Číslo obrazu karty / identifikační číslo karty
CLA	Byte třídy příkazové zprávy
CRT	Kontrolní referenční šablona
CT	Kontrolní referenční šablona pro důvěrnost
CVM	Metoda ověření držitele karty
DAP	Vzor ověřování dat

DEK	Klíč pro šifrování dat
DER	Rozlišená pravidla kódování
DES	Standard šifrování dat
DST	Kontrolní referenční šablona pro digitální podpis
ECB	Elektronická kniha kódů
EMV	Europay, Mastercard a VISA: používá se pro odkaz na specifikace ICC pro platební systémy.
ENC	Šifrování
FCI	Informace o kontrole souborů
HEX	Hexadecimální
HMAC	Klíčový kód pro ověřování zpráv (Keyed-Hash Message Authentication Code)
ICC	Karta s integrovaným obvodem
ICV	Počáteční vektor řetězení
IIN	Identifikační číslo emitenta
INS	Instrukční bajt příkazové zprávy
ISO	Mezinárodní organizace pro normalizaci
Lc	Přesná délka dat v příkazu case 3 nebo case 4
Le	Maximální délka dat očekávaná v odpovědi na příkaz case 2 nebo case 4
LV	Délka Hodnota
MAC	Kód ověřování zpráv
MEL	Spustitelný jazyk MULTOS. Instrukční sada běhového prostředí MULTOS ™
OID	Identifikátor objektu
P1	Referenční řídicí parametr 1
P2	Referenční řídicí parametr 2
PIN	Osobní identifikační číslo
PKI	Infrastruktura veřejných klíčů
RAM	Paměť s náhodným přístupem
RFU	Rezervováno pro budoucí použití
RID	Identifikátor poskytovatele registrované aplikace
ROM	Paměť pouze pro čtení
RSA	Paměť pouze pro nýty
SCP	Secure Channel Protocol: nebo (ETSI) Smart Card Platform
SW	Stavové slovo
SW1	Stavové slovo jedna
SW2	Stavové slovo dvě
TLV	Tag Délka Hodnota
TP	Bod důvěry
"xx"	Šestnáctkové hodnoty jsou vyjádřeny jako šestnáctkové číslice mezi jednoduchými uvozovkami.
"X"	Hodnota v buňce tabulky, jejíž účel je popsán ve sloupci "Význam" tabulky.
"-"	Hodnota (0 nebo 1) v buňce tabulky, která nemá vliv na "Význam" uvedený pro daný řádek tabulky.

Specifikace karty

Tato část shrnuje vlastnosti a funkce ACOSJ-G.

Elektrické specifikace

Provozní napětí (ACOSJ-G v1.01):	Napájecí napětí 2,7 V až 5,5 V
Provozní napětí (ACOSJ-G v2.04):	Napájecí napětí 2,1 V až 5,5 V
Maximální frekvence externích hodin:	10 MHz
Maximální taktovací frekvence procesoru:	28 MHz
Ochrana ESD	vyšší než 5 kV (HBM)

Specifikace životního prostředí

Provozní teplota:	-25°C až +85°C

Komunikační protokoly

- Protokol T=CL s přenosovou rychlostí až 848 kb/s

Paměť

Kapacita:	95 KB (ACOSJ-G 2.04), 40 KB (ACOSJ-G 1.01)
Výdrž paměti EEPROM:	500 000 cyklů mazání/zápisu (25 °C)
Uchovávání dat:	30 let (25 °C)

Kryptografické funkce

DES:	2K3DES,3K3DES (ECB a CBC)
AES:	128/192/256 bitů (ECB a CBC)
RSA:	768 až 2048 bitů
ECC:	Modul 112/128/160/192/224/256/384 bitů
Hash:	SHA1, SHA256, SHA384, SHA512
SEED:	128 bitů

- SM2/SM3/SM4

Dodržování norem

- Shoda s normou ISO 7816 část 4
- Shoda s normou ISO 14443 (typ A a B)
- Soulad se specifikací karet JAVA verze 3.0.4
- Soulad s globální specifikací platformy verze 2.2.1
- Soulad s pokyny pro mapování 1.0.1

Odpověď pro výběr (ATS, bezkontaktní karta)

Po obdržení příkazu Request for Answer to Select (RATS) od čtecího zařízení karty vysílá karta příkaz Answer to Select (ATS) v souladu s normou ISO 14443, část 4.

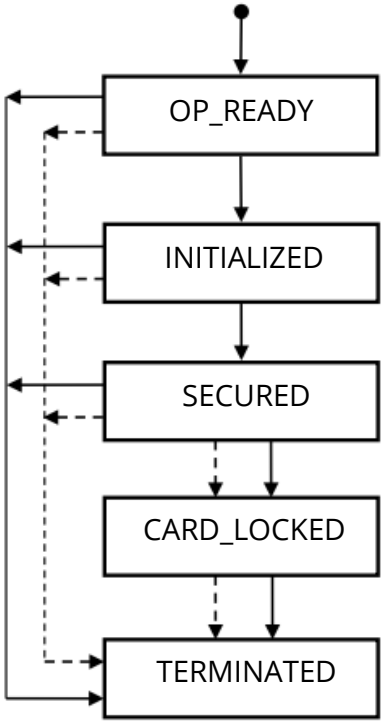
V následující tabulce jsou uvedeny výchozí systémy ATS:

Parametr	ATS	Popis
TL	0Eh	Délka
T0	78h	Formální bite...kódy Y(1) a FSCI
TA1	00h	Interface bite...kódy DS a DR
TB1	71h	Kódy FWI a SFGI
TC1	02h	Možnosti protokolu kódů
T1	41h	Označuje "A"
T2	43h	Označuje "C"
T3	4Fh	Označuje "O"
T4	53h	Označuje "S"
T5	4Ah	Označuje "J"
T6	76h	Označuje "v"
T7~T9	31h 30h 31h nebo 32h 30h 34h	Označuje "101" nebo označuje "204".

Poznámka: Úplný popis systému ATS naleznete v normě ISO 14443, část 4.

Stavy životního cyklu karty

ACOSJ má pět stavů karet: OP_READY, INTIALIZED, SECURED, CARD_LOCKED a TEDMINATED. Na obrázku níže je znázorněn přechod mezi jednotlivými stavy životního cyklu karty:



Legenda

Privilegovaná bezpečnostní doména

Privilegovaná aplikace

OP_READY

Tento stav znamená, že běhové prostředí (runtime) je k dispozici a bezpečnostní doména vydavatele, která vystupuje jako vybraná aplikace, je připravena přijímat, vykonávat a odpovídat na příkazy APDU.

Pokud je karta ve stavu OP_READY, musí být k dispozici následující funkce:

- Prostředí použití musí být připraveno k provedení.
- OPEN musí být připraven k provedení.
- Bezpečnostní doména vydavatele je implicitně vybraná aplikace pro všechna rozhraní karet.
- Spustitelné soubory pro načítání, které byly zahrnuty do neměnné trvalé paměti, se registrují v registru GlobalPlatform.
- Počáteční klíč musí být k dispozici v bezpečnostní doméně vydavatele.

Karta musí být schopna měnit obsah karty, může dojít k načtení souborů obsahujících aplikace, které na kartě ještě nejsou.

Může dojít k instalaci jakékoli aplikace ze spustitelných souborů načtení.

Pokud jsou v této fázi k dispozici personalizační informace, mohou být aplikace navíc personalizovány.

Stav OP_READY může být použit mimo kartu k provedení následujících akcí:

- Mohou být načteny a/nebo nainstalovány doplňkové domény zabezpečení.
- Klíče bezpečnostní domény mohou být vloženy za účelem zachování kryptografického oddělení klíčů od klíčů bezpečnostní domény vydavatele.

Initialized

Tento stav je stavem výroby administrativních karet. Přechod ze stavu OP_READY do stavu INITIALIZED je nevratný. Jeho funkčnost je mimo rozsah této specifikace. Tento stav lze použít k označení, že byla vyplněna některá počáteční data (např. klíče a/nebo data bezpečnostní domény vydavatele), ale karta ještě není připravena k vydání držiteli karty.

Secured

Tento stav je zamýšleným stavem životního cyklu provozní karty po vydání. Tento stav mohou používat bezpečnostní domény a aplikace k prosazování svých příslušných bezpečnostních politik. Přechod ze stavu INITIALIZED do stavu SECURED je nevratný.

Stav SECURED by se měl používat k označení subjektů mimo kartu, že bezpečnostní doména vydavatele obsahuje všechny potřebné klíče a bezpečnostní prvky pro plnou funkčnost.

Card_Locked

Stav životního cyklu karty CARD_LOCKED umožňuje zakázat výběr bezpečnostní domény a aplikací. Přechod ze stavu SECURED do stavu CARD_LOCKED je vratný.

Nastavení karty do stavu CARD_LOCKED znamená, že karta umožní výběr pouze aplikace s oprávněním Final Application.

V tomto stavu nejsou povoleny změny obsahu karty, včetně jakéhokoli typu správy dat (zejména klíčů a dat domény zabezpečení).

Přechod ze stavu SECURED do stavu CARD_LOCKED může iniciovat buď doména OPEN, nebo bezpečnostní doména s oprávněním Card Lock, nebo aplikace s oprávněním Card Lock.

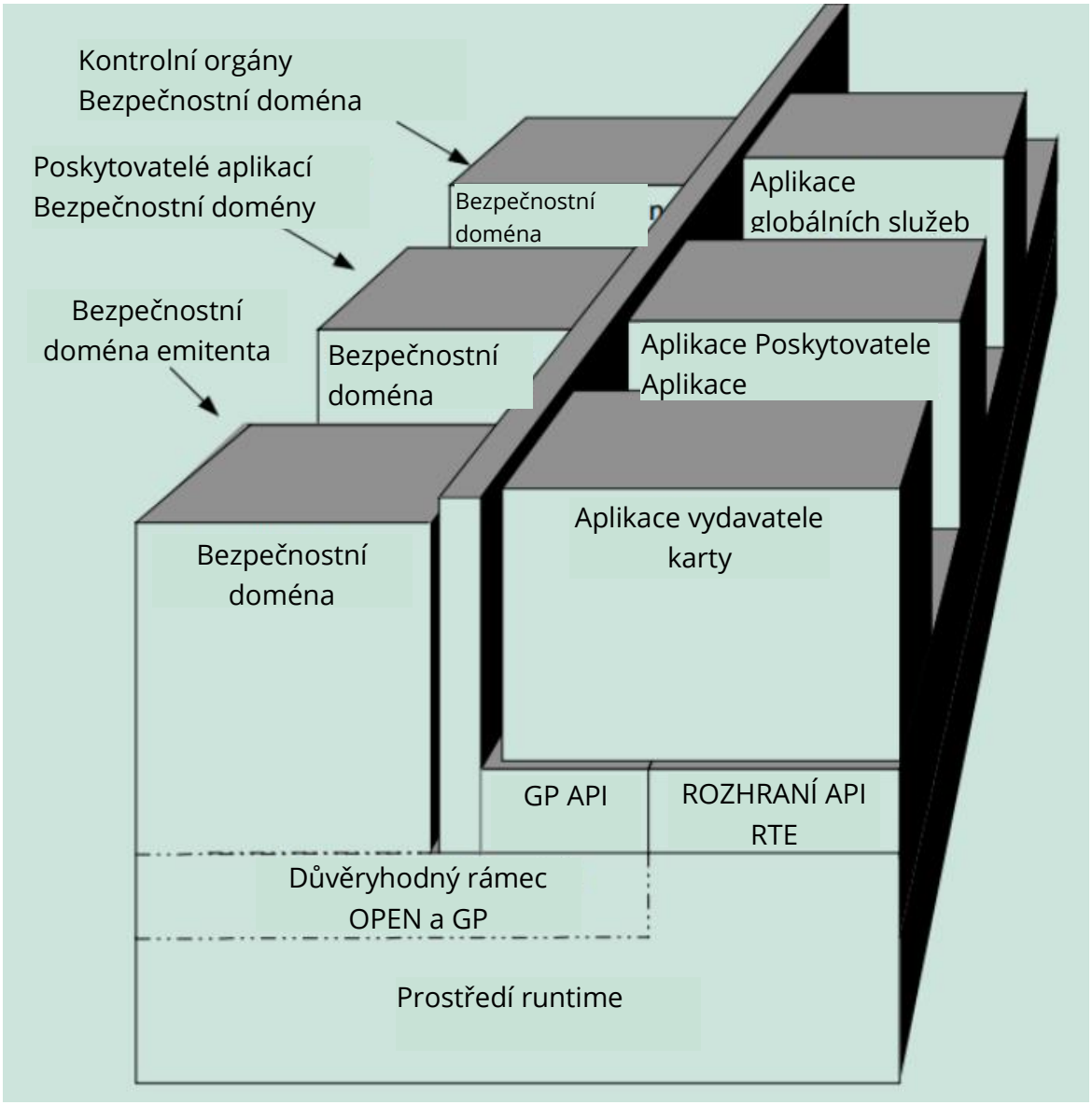
TERMINATED

Tento stav signalizuje konec životního cyklu karty a karty. Přechod z jakéhokoli jiného stavu do stavu TERMINATED je nevratný.

Stav TERMINATED se použije k trvalému vypnutí všech funkcí karty s ohledem na jakoukoli správu obsahu karty a jakékoli změny životního cyklu. Tento stav karty je určen jako mechanismus, kterým aplikace logicky "zničí" kartu například z důvodu zjištění závažného bezpečnostního ohrožení nebo vypršení platnosti karty. Pokud má bezpečnostní doména oprávnění Final Application, zpracovává se pouze příkaz GET DATA, všechny ostatní příkazy definované v této specifikaci jsou zakázány a vracejí chybu. Id aplikace má oprávnění Final Application, její zpracování příkazů podléhá zásadám vydavatele.

Přechod z některého z předchozích stavů do stavu TERMINATED může iniciovat samotné OPEN, bezpečnostní doména s oprávněním Ukončit kartu nebo aplikace s oprávněním Ukončit kartu.

Architektura karet



Odkaz na příkazy GP APDU

V této části jsou podrobně popsány příkazy APDU GlobalPlatform, které lze implementovat. Příkazy jsou uvedeny v abecedním pořadí.

Shrnuje příkazy APDU, které jsou podporovány doménami zabezpečení vydavatele, a požadavky na podporu těchto příkazů APDU jinými doménami zabezpečení. Pokud jsou podporovány logické kanály, příkaz MANAGE CHANNEL zpracovává pouze OPEN a pro tento příkaz není vyžadována podpora žádné bezpečnostní domény.

Příkaz	OP_READY			INICIÁLNÍ			ZABEZPEČENÝ			CARD_LOCKED		UKONČENO	
	AM SD	DM SD	SD	AM SD	DM SD	SD	AM SD	DM SD	SD	FASD	SD	FASD	SD
DELETE Spustitelný soubor pro načtení													
DELETE Spustitelný soubor načítání a související aplikace													
DELETE Application	✓			✓			✓						
DELETE Key													
GET DATA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
GET STATUS	✓			✓			✓			✓			
INSTALL [pro zatížení]													
INSTALL [pro instalaci]													
INSTALL [pro načtení, instalaci a volbu]													
INSTALL [pro instalaci a volbu]	✓	✓		✓	✓		✓	✓					
INSTALL [pro možnost volby]													
INSTALL [pro vydání]													
INSTALL [pro aktualizaci registru]													
INSTALE [pro personalizaci]													
LOAD													
PUT KEY	✓			✓			✓						
SELECT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
SET STATUS	✓			✓			✓			✓			
STORE DATA	✓			✓			✓						

AM SD	Doména zabezpečení s oprávněním Autorizovaná správa.
DM SD	Doména zabezpečení s oprávněním delegované správy.
FA SD	Doména zabezpečení s oprávněním Final Application.

Poznámka: Podpora příkazů a aplikací s oprávněním Final Application Privilege, které nejsou bezpečnostní doménou, podléhá zásadám vydavatele v rámci omezení povolených podle stavu životního cyklu karty.

SD	Ostatní bezpečnostní domény.
✓	Požadovaná podpora.
Prázdná buňka	Podpora je volitelná.
Šedá buňka	Podpora zakázána.

Shrnuje minimální požadavky na zabezpečení příkazů APDU.

Příkaz	Minimální zabezpečení
DELETE	Iniciace zabezpečeného kanálu nebo ověření digitálního podpisu
GET DATA	Žádné
GET STATUS	Iniciace zabezpečeného kanálu
INSTALL	Iniciace zabezpečeného kanálu nebo ověření digitálního podpisu
LOAD	Iniciace zabezpečeného kanálu nebo ověření digitálního podpisu
MANAGE CHANNEL	N/A
PUT KEY	Iniciace zabezpečeného kanálu
SELECT	N/A
SET STATUS	Iniciace zabezpečeného kanálu
STORE DATA	Iniciace zabezpečeného kanálu

Obecná pravidla kódování

Viz referenční příručka ACOSJ.

Příkaz DELETE

Příkaz DELETE se používá k odstranění jednoznačně identifikovatelného objektu, například spustitelného souboru s načítáním, aplikace, spustitelného souboru s načítáním a souvisejících aplikací nebo klíče. Aby bylo možné odstranit i objekt, měl by být objekt jednoznačně identifikovatelný vybranou Aplikací.

Příkaz GET DATA

Příkaz GET DATA slouží k získání buď jednoho datového objektu, který může být zkonstruován, nebo sady datových objektů. K definování konkrétní značky datového objektu se používají referenční řídicí parametry kódování P1 a P2. Datový objekt může obsahovat informace týkající se klíče.

Příkaz GET STATUS

Příkaz GET STATUS slouží k získání informací o stavu bezpečnostní domény vydavatele, spustitelného souboru načítání, spustitelného modulu, aplikace nebo životního cyklu bezpečnostní domény podle zadaných kritérií shody/vyhledávání.

Příkaz INSTALL

Příkaz INSTALL se vydává doméně zabezpečení za účelem zahájení nebo provedení různých kroků potřebných pro správu obsahu karty.

Příkaz LOAD

Tato část definuje strukturu souboru Load File přenášeného v datovém poli příkazu LOAD pro načtení souboru Load File. Vnitřní zpracování nebo uložení souboru Load File v ICC je mimo rozsah této specifikace.

K přenosu souboru Load na kartu lze použít více příkazů LOAD. Soubor Load File je pro přenos rozdělen na menší součásti. Každý příkaz LOAD se čísluje od 00h. Číslování příkazů LOAD je přísně sekvenční a zvyšuje se o jedničku. Karta musí být informována o posledním bloku souboru Load File.

Po přijetí posledního bloku souboru Load spustí karta interní procesy nezbytné pro soubor Load a všechny další procesy identifikované v příkazu INSTALL [pro načtení], který předcházel příkazům LOAD.

Příkaz MANAGE CHANNEL

Příkaz MANAGE CHANNEL se zpracovává otevřeně na kartách, které znají logické kanály. Slouží k otevírání a zavírání doplňkových logických kanálů. Základní logický kanál (kanál číslo nula) nelze nikdy uzavřít.

Příkaz PUT KEY

Příkaz PUT KEY se používá buď k:

- Nahrazení stávajícího klíče novým klíčem: Nový klíč má stejné nebo jiné číslo verze klíče, ale stejný identifikátor klíče jako nahrazovaný klíč.
- Nahrazení více stávajících klíčů novými klíči: Nové klíče mají stejné nebo jiné číslo verze klíče (stejné pro všechny nové klíče), ale stejné identifikátory klíčů jako nahrazované klíče.
- Přidání jednoho nového klíče: Nový klíč má jinou kombinaci identifikátor klíče/číslo verze klíče než stávající klíče.
- Přidání více nových klíčů: Nové klíče mají jiné kombinace identifikátorů klíčů/čísla verze klíčů (shodné pro všechny nové klíče) než stávající klíče.

Pokud operace správy klíčů vyžaduje více příkazů PUT KEY, doporučuje se řetězení více příkazů PUT KEY, aby byla zajištěna integrita operace.

V této verzi specifikace jsou veřejné hodnoty asymetrických klíčů prezentovány (přenášeny) v otevřeném textu.

Příkaz SELECT

Příkaz SELECT slouží k výběru aplikace. OPEN zpracovává pouze příkazy SELECT s uvedením SELECT [podle názvu] se předá aktuálně vybrané bezpečnostní doméně nebo Aplikaci na uvedeném logickém kanálu.

Příkaz SET STATUS

Příkaz SET STATUS se použije ke změně stavu životního cyklu karty nebo stavu životního cyklu aplikace.

Příkaz STORE DATA

Příkaz STORE DATA se používá k přenosu dat do aplikace nebo domény zabezpečení, která příkaz zpracovává.

Doména zabezpečení určí, zda je příkaz určen pro ni samotnou nebo pro aplikaci v závislosti na příkazu, který byl dříve přijat. Pokud byl předchozím příkazem a příkaz INSTALL [pro personalizaci], je příkaz STORE DATA určen pro aplikaci.

Vícenásobný příkaz STORE DATA se používá k odeslání dat do aplikace nebo bezpečnostní domény rozdělením dat na menší součásti pro přenos. Bezpečnostní doména je informována o posledním bloku.

Relace personalizace začíná, když bezpečnostní doména obdrží platný příkaz INSTALL [pro personalizaci], který určuje aplikaci (implementující buď aplikaci, nebo personalizační rozhraní), které má bezpečnostní doména předávat následně přijaté příkazy STORE DATA.

Relace personalizace končí, když:

- Karta je resetována.
- Doména zabezpečení je zrušena (tj. je vybrán jiný nebo stejný applet na stejném logickém kanálu).
- Doména zabezpečení je vybrána na stejném nebo jiném logickém kanálu.
- Relace zabezpečeného kanálu (pokud existuje) vytvořená doménou zabezpečení je resetována, případně cílovou aplikací.
- Doména zabezpečení obdrží příkaz INSTALL [pro personalizaci] (spuštění nové relace personalizace pro jinou aplikaci).
- Bezpečnostní doména obdrží příkaz STORE DATA s označením P1. b8=1 (poslední blok)

Každý příkaz STORE DATA přijatý z takové relace personalizace zpracuje samotná bezpečnostní doména.

API GlobalPlatform

Globální platforma na kartě JAVA

Tato část obsahuje pouze rozhraní API vyžadované pro karty Java Global Platform 2.2x. Použití rozhraní API otevřené platformy 2.0.1 je stále povoleno pro podporu starších verzí aplikací, ale je zastaralé. Je definováno ve verzi 2.1.1 této specifikace.

Zastaralé i nové rozhraní API přistupují ke stejným objektům, pokud je to možné. Ačkoli se to může zdát zřejmé u metod, které mají v obou třídách stejný název (např. setATRHistBytes(), setCardContentState() a getCardContentSame()), je třeba také poznamenat, že například aplikace, která použije metodu updated() v novém API ke změně hodnoty globálního PIN, ovlivní stejný globální PIN aplikace, která použije metodu setPIN() v zastaralém API k ověření globálního PIN.

GlobalPlatform Specifické požadavky

Aby byla zajištěna co nejvyšší úroveň interoperability implementací GlobalPlatform, přejímá GlobalPlatform také pořadí definované v části 6.2 - Specifikace virtuálního stroje Java Card™ 2.2x.

V implementaci GlobalPlatform jsou obsaženy následující drobné úpravy standardních funkcí definovaných ve specifikacích prostředí pro runtime prostředí Java Card™ 2.1.1 (JCRE) a rozhraní pro programování aplikací Java Card™ 2.1.1. Některé úpravy jsou již v souladu se specifikací Java Card™ 2.2.x, a proto se již nejedná o úpravy.

Objekty GPRegistryEntry se implantují jako objekty sdíleného rozhraní, jak je definováno v oddíle 6.2.4 "Sdílená rozhraní" specifikace prostředí Java Card™ 2.2.x Runtime Environment (JCRE), aby byl zajištěn sdílený přístup.

Balíček GlobalPlatform AID

Každé AID balíčku GlobalPlatfrom bude tvořit součet RID a PIX. Hodnota AID souboru pro export karet v jazyce Java pro nové rozhraní API GlobalPlatform (totožná pro GlobalPlatform 2.1 i 2.1.1) na základě RID uvedeného v dodatku H - Hodnoty dat GlobalPlatform a data pro rozpoznávání karet je "A00000015100".

Instalace

V části 3.1 - Instalace metody specifikace prostředí Java Card™ 2.2.x Runtime Environment (JCRE) jsou parametry předávané metodě definovány jako inicializační parametry z obsahu příchozího parametru pole bajtů.

Tato specifikace tento požadavek rozšiřuje a dále definuje obsah parametrů instalace. Toto rozšíření má vliv jak na implementaci OPEN, tak na chování appletu Java Card vytvořeného pro kartu GlobalPlatform. Nemá vliv na definici metody install () třídy Applet specifikace aplikačního programového rozhraní Java Card™ 2.2.x.

Parametry instalace identifikují následující údaje, které jsou obsaženy v příkazu INSTALL [pro instalaci]:

- Instance AID
- Privilegia
- Specifické parametry aplikace. (Zatímco příkaz APDU obsahuje parametry instalace představující systémové a aplikačně specifické parametry kódované TLV, aplikace vyžaduje pouze znalost aplikačně specifických parametrů, tj. jako aplikačně specifické parametry jsou přítomny pouze LV struktury TLV kódované "C9".

OPEN odpovídá za to, že parametry (bArray, bOffset a bLength) obsahují následující informace:

Pole bArray obsahuje následující po sobě jdoucí data s kódem LV:

- Délka instance AID
- Instance AID
- Délka platnosti oprávnění
- Privilegia
- Délka specifických parametrů aplikace
- Specifické parametry aplikace

Bajt bOffset obsahuje offset v rámci pole, který ukazuje na délku AID instance.

Bajt bLength obsahuje délku udávající celkovou délku výše definovaných dat v poli.

Při volání metody register (byte [] bArray, short bOffset, byte bLength) třídy Applet specifikace aplikačního programovacího rozhraní Java Card 2.2.x musí applet použít jako parametr identifikátor instance AID.

T=0 Protokol přenosu

Karty GlobalPlatform jsou určeny k tomu, aby fungovaly v co nejširším spektru prostředí (tj. zařízení pro akceptaci karet). V současné době specifikace prostředí Java Card 2.1.1 Runtime Environment (JCRE) a Java Card 2.2.x Runtime Environment (JCRE) popisují chování pro případ 2 příkazů (při použití protokolu T=0) v rozporu s EMV 2000. GlobalPlatform nařizuje, že JCRE musí tento případ příkazu zpracovat v souladu s ISO/IEC 7816: applet, který obdrží příkaz případu 2, sestaví odpověď a vyvolá příslušné API pro výstup dat. Pokud jsou data menší než data očekávaná terminálem, OPEN uloží data a vypíše kód odpovědi "6Cxx" a počká, až CAD znovu vydá příkaz se správnou délkou. Po přijetí opětovně vydaného příkazu bude JCRE řídit výstup uložených dat.

Atomicita

Není-li uvedeno jinak, musí být všechny interní perzistentní objekty rozhraní GlobalPlatform API v souladu s probíhající transakcí.

Všechny operace prováděné tímto rozhraním API, s výjimkou metody Application.processData(), se provádějí atomicky. Objekty používané k vynucení implementace kontroly rychlosti nesmí odpovídat probíhající transakci.

Logické kanály

Pro kartu Java Card 2.2.x platí následující omezení logických kanálů (více informací naleznete ve specifikacích prostředí Java Card 2.2.x Runtime Environment (JCRE)):

Výběr Aplikace na logickém kanálu, jak je definován v části 6.3 - Odesílání příkazů, bude neúspěšný, pokud je stejná Aplikace nebo jakákoli jiná Aplikace instancovaná z kódu ve stejném balíčku, ze kterého byla instancována vybíraná Aplikace, aktuálně vybrána na jiném logickém kanálu, ale kód aplikace neimplementuje rozhraní MultiSelectable. Bezpečnostní domény musí implementovat rozhraní MultiSelectable.

Změna kontextu z bezpečnostní domény na aplikaci podle definice v části 7.3.3 Podpora personalizace nebude úspěšná, pokud je stejná aplikace nebo jakákoli jiná aplikace instancovaná z kódu ve stejném balíčku, ze kterého byla personalizovaná aplikace instancována, aktuálně vybrána v jiném logickém kanálu, ale kód aplikace neimplementuje rozhraní MultiSelectable.

Aplikace, která má právo být ve výchozím nastavení vybrána a je určena pro kartu, která podporuje doplňkové logické kanály, by měla implementovat rozhraní MultiSelectable.

GlobalPlarform definuje pouze přiřazení čísel logických kanálů kartou. Volitelně a podle definice v Java Card 2.2 může karta podporovat také přiřazování čísel logických kanálů terminálem.

Kryptografické algoritmy

Karty GlobalPlatform podporující kryptografii RSA by měly podporovat velikosti klíčů, které nejsou definovány jako konstanty ve třídě Key Builder. Konkrétně by měla být k dispozici podpora pro velikosti klíčů, které jsou násobkem 4 bajtů (32 bitů) a které jsou v rámci povolených délek klíčů definovaných implementací.

Úroveň důvěry

Specifikace Java Card 2.2.x předpokládají, že RID balíčků AID, appletů a instancí budou použity k zajištění úrovně důvěryhodnosti mezi těmito entitami. V oddíle 4.2.2-AID Usage aplikačního programovacího rozhraní Java Card 2.2.1 je definováno, že RID AID komponenty musí odpovídat RID AID balíčku, a v definici metody register (byte [] bArray, short bOffset, byte bLength) Java Card 2.2.1 je definováno, že RID AID balíčku musí odpovídat RID AID komponenty.2.x Application Programming Interface je definováno, že musí být vyhozena výjimka, pokud se část RID bajtů AID v parametru bArray neshoduje s částí RID názvu appletu Java Card.

Z hlediska reálné implementace není praktické nařizovat, že RID AID instance musí být stejný jako RID komponenty, ze které byla instance vytvořena. Implementace GlobalPlatform nesmí nařizovat, aby existovala jakákoli vazba přes AID instance na její původní balíček. (Vyžadování žádných vazeb mezi AID instance a AID jejího původního balíčku). Předpokládá však, že všechny aplikace ve stejném balíku sdílejí stejnou úroveň důvěryhodnosti.

Volání metod GlobalPlatform

Zde definované rozhraní pro programování aplikací je přístupné pro jakýkoli applet Java Card vyvinutý s úmyslem být přítomen na kartě GlobalPlatform. Neexistuje jedno omezení týkající se konstruktoru appletu a metody install () třídy Applet aplikačního programovacího rozhraní Java Card 2.2.x. Protože tato specifikace přesně nedefinuje, kdy se instance appletu stane záznamem v registru GlobalPlatform karty, může vývojář appletu pouze předpokládat, že k tomu došlo po úspěšném dokončení metody install (). Aby byla zajištěna interoperabilita, nesmí být metody API GlobalPlatform, které přistupují k záznamu v registru GlobalPlatform appletu, který metodu volá, volány z konstruktoru nebo metody install().

Následující seznam metod, které lze vyvolat z konstruktoru nebo metody install(), je ztracený:

- getCardState;
- getCVM;
- getService;

Požadované chování karty v případě, že aplikace nesprávně vyvolá jinou metodu třídy org.globalplatform.GPSystem než výše uvedené, není definováno. Například může být vyhozena výjimka a zpracování metody install() může být přerušeno.

Výběr

Pokud na kartách GlobalPlatform dojde během zpracování metody select() k chybě nebo pokud metoda select() vrátí hodnotu False nebo pokud aplikaci nelze vybrat, protože neimplementuje rozhraní Multiselectable, bude OPEN pokračovat v hledání následné úplné nebo částečné shody v registru GlobalPlatform, jak je definováno v oddíle 6.4.2.1.2. Pokud není vybrána žádná aplikace, zůstane příslušný logický kanál otevřený bez aktuálně vybrané aplikace. Protože není aktuálně vybrána žádná Aplikace, bude odmítnut jakýkoli následný příkaz jiný než příkaz MANAGE CHANNEL nebo SELECT. Očekává se, že subjekt mimo kartu při takové chybě provede odpovídající akci, např. vybere jinou Aplikaci, uzavře příslušný logický kanál, resetuje kartu nebo ji vypne.

Shrnutí a podrobnosti o metodách pro specifikaci GlobalPlatform Java Card API jsou nyní k dispozici v samostatném dokumentu, který lze nalézt na webových stránkách GlobalPlatform.

Před personalizací

Při prvním zapnutí je IC ve stavu před personalizací.

Příkaz INIT_CARD (viz 8.2.2 Příkaz INIT CARD) musí být prvním příkazem ve stavu před personalizací, příkaz inicializuje oblast paměti karty výchozími hodnotami a poté je k dispozici tzv. aplikace ACOSJ ROOT (viz 8.0 Aplikace ACOSJ ROOT) pro zápis kritických systémových dat (například "ATR", "ATS" atd.) do paměti EEPROM. Po nahrání dat a potvrzení, že nedošlo k dalším změnám, musí být proveden příkaz, který aplikaci ACOSJ ROOT Application zakáže.

Když IC opustí stav před personalizací úspěšným provedením příkazu Active Card (viz 8-2.5 Příkaz ACTIVATE), nachází se ve výchozím stavu životního cyklu karty GlobalPlatform OP_READY, pokud není nastaven jiný stav životního cyklu.

Po úspěšném zpracování aktivního příkazu karty není možné vrátit se do stavu před personalizací.

Poznámka: Transportní klíč je klíč před personalizací kořenové aplikace ACOSJ a o transportní klíč je třeba požádat ACS.

Aplikace ACOSJ ROOT

ACOSJ ROOT Popis aplikace

Příkazy v aplikaci ROOT lze použít pouze po správném výběru aplikace ROOT (pomocí příkazu select Transport Key). Po vstupu do aplikace ROOT může uživatel číst nebo konfigurovat parametry karty prostřednictvím příkazů podporovaných aplikací ROOT. Pro ukončení aplikace ROOT a výběr jiné aplikace je nutné kartu resetovat.

V aplikaci ROOT lze číst nebo konfigurovat parametry karty.

Po aktivaci karty se aplikace ROOT stane neplatnou.

Příkazy aplikace ACOSJ ROOT

Příkaz SELECT

Příkaz SELECT slouží k výběru aplikace ROOT.

Příkaz INIT_CARD

Tento příkaz slouží k resetování karty. Tento příkaz inicializuje celou paměť EEPROM.

Příkaz READ

Tento příkaz slouží ke čtení z konfigurační oblasti. Tímto příkazem lze načíst konfigurační parametry karty.

Příkaz WRITE

Tento příkaz slouží k zápisu dat do konfigurační oblasti. Tímto příkazem lze načíst konfigurační parametry karty.

Příkaz ACTIVE

Tento příkaz slouží k aktivaci karty. Po úspěšném provedení tohoto příkazu se aplikace ROOT stane neplatnou a konfigurační data karty již nelze přímo číst ani nastavovat.

Aplikace ACOSJ IDENTIFY

ACOSJ IDENTIFY Popis aplikace

Po výběru aplikace IDENTIFY (AID: 6163732E636F732E61636F736A766572) příkazem SELECT vrátí ACOSJ číslo verze ACOSJ a uvede, zda byl vůz aktivován.

Odkaz na příkaz ACOSJ IDENTIFY aplikace

Příkaz SELECT

Příkaz SELECT slouží k výběru aplikace INDENTIFY.

Stavové bajty

SW1 SW2	Popis
90 00h	Správný příkaz
61 XX	SW2 k dispozici více bajtů odezvy
62 83h	Aplikace trvale zablokována
63 00h	Ověření se nezdařilo
63 CX	Zbývá X pokusů o opakování
67 00h	Nesprávná délka dat
68 81h	Logický kanál není podporován
68 82h	Zabezpečené zasílání zpráv není podporováno
69 84h	Neplatné údaje
69 82h	Stav zabezpečení není splněn
69 85h	Nesplněné podmínky použití
69 88h	Špatná MAC
6A 86h	Nesprávné P1 P2
6A 80h	Nesprávné hodnoty v příkazových datech
6A 82h	Soubor nebyl nalezen
6A 81h	Aplikace uzamčena
6A 83h	Záznam nebyl nalezen
6A 84h	Nedostatek místa v paměti
6A 88h	Odkazovaná data nebyla nalezena
6D 00h	Neplatný INS, Příkaz neexistuje
6E 00h	Neplatná třída

Odkazy

Níže uvedené dokumenty sloužily jako reference pro funkční specifikaci ACOSJ:

- Specifikace karty GlobalPlatform verze 2.2.1
- GlobalPlatform Card API verze 1.6
- Rozhraní Java Card 3 API, verze Classic Edition 3.0.4
- Specifikace prostředí Java Card 3 Platform Runtime Environment, Classic Edition Verze 3.0.4 září 2011
- Specifikace virtuálního stroje platformy Java Card 3, verze 3.0.4, verze Classic September 2011
- Pokyny pro mapování karet GlobalPlatform stávající implementace GP v2.1.1 na verzi v2.2.1 1.0.1

Záruční podmínky

Na nový výrobek zakoupený v prodejní síti Alza.cz se vztahuje záruka 2 roky. V případě potřeby opravy nebo jiného servisu v záruční době se obraťte přímo na prodejce výrobku, je nutné předložit originální doklad o koupi s datem nákupu.

Za rozpor se záručními podmínkami, pro který nelze reklamaci uznat, se považují následující skutečnosti:

- Používání výrobku k jinému účelu, než pro který je výrobek určen, nebo nedodržování pokynů pro údržbu, provoz a servis výrobku.
- Poškození výrobku živelnou pohromou, zásahem neoprávněné osoby nebo mechanicky vinou kupujícího (např. při přepravě, čištění nevhodnými prostředky apod.).
- přirozené opotřebení a stárnutí spotřebního materiálu nebo součástí během používání (např. baterií atd.).
- Působení nepříznivých vnějších vlivů, jako je sluneční záření a jiné záření nebo elektromagnetické pole, vniknutí kapaliny, vniknutí předmětu, přepětí v síti, elektrostatický výboj (včetně blesku), vadné napájecí nebo vstupní napětí a nevhodná polarita tohoto napětí, chemické procesy, např. použité zdroje atd.

Pokud někdo provedl úpravy, modifikace, změny konstrukce nebo adaptace za účelem změny nebo rozšíření funkcí výrobku oproti zakoupené konstrukci nebo použití neoriginálních součástí.

Vážený zákazník,

ďakujeme vám za zakúpenie nášho výrobku. Pred prvým použitím si pozorne prečítajte nasledujúce pokyny a uschovajte si tento návod na použitie. Venujte osobitnú pozornosť bezpečnostným pokynom. Ak máte akékoľvek otázky alebo pripomienky k prístroju, obráťte sa na linku služieb zákazníkom.

✉ www.alza.sk/kontakt

✆ +421 257 101 800

Dovozca Alza.cz a.s., Jankovcova 1522/53, Holešovice, 170 00 Praha 7, www.alza.cz

Prehľad

ACOSJ je operačný systém pre inteligentné karty vyvinutý spoločnosťou Advanced Card Systems Ltd. Funguje na základe virtuálneho stroja karty JAVA a je v súlade so špecifikáciou karty GlobalPlatform vo verzii 2.2.1. Virtuálny stroj karty JAVA a jeho funkcie a konfigurácie sú v súlade so špecifikáciou karty GlobalPlatform vo verzii 2.2.1, špecifikáciou karty JAVA vo verzii 3.0.4 a usmerneniami pre mapovanie 1.0.1.

Účelom tohto dokumentu je podrobne opísať vlastnosti a funkcie operačného systému inteligentných kariet ACOSJ.

Verzia ACOSJ-G

Verzia	Zverejnené údaje	Úpravy
ACOSJ v1.01	jún 2015	• 40KB EEPROM • Prevádzkové napätie: 2,7 V až 5,5 V
ACOSJ v2.00	máj 2018	• 95KB EEPROM • Prevádzkové napätie: 2,1 V až 5,5 V • Zmena adresy konfiguračnej pamäte
ACOSJ v2.04	apríl	• 95KB EEPROM • Prevádzkové napätie: 2,1 V až 5,5 V • Zmena adresy konfiguračnej pamäte • Vylepšenia funkcií

Symboly a skratky

Skratka	Popis
AES	Pokročilý štandard šifrovania
AID	Identifikátor aplikácie
APDU	Dátová jednotka aplikačného protokolu
API	Rozhranie na programovanie aplikácií
ASCII	Americký štandardný kódex pre výmenu informácií
ATR	Odpoveď na obnovenie
ATQ	Odpoveď na otázku (pre bezkontaktné karty)
BCD	Binárne kódované desatinné číslo
BER	Základné pravidlá kódovania
CAT	Súbor nástrojov na používanie kariet: alebo Šablóna kryptografickej autorizácie
CBC	Usporiadanie blokov šifier
CCT	Referenčná šablóna kryptografického kontrolného súčtu
CIN	Číslo obrazu karty / identifikačné číslo karty
CLA	Trieda príkazovej správy byte
CRT	Kontrolná referenčná šablóna
CT	Šablóna referencie na kontrolu dôvernosti

CVM	Metóda overenia držiteľa karty
DAP	Vzor overovania údajov
DEK	Kľúč na šifrovanie údajov
DER	Odlišné pravidlá kódovania
DES	Štandard šifrovania údajov
DST	Referenčná šablóna digitálneho podpisu
ECB	Elektronický číselník
EMV	Europay, Mastercard a VISA: používa sa na označenie špecifikácií ICC pre platobné systémy.
ENC	Šifrovanie
FCI	Informácie o kontrole súborov
HEX	Hexadecimálne
HMAC	Kód overovania správ s kľúčom-Hash
ICC	Karta s integrovaným obvodom
ICV	Počiatočný vektor reťazenia
IIN	Identifikačné číslo emitenta
INS	Inštrukčný bajt príkazovej správy
ISO	Medzinárodná organizácia pre normalizáciu
Lc	Presná dĺžka údajov v prípade 3 alebo 4
Le	Maximálna dĺžka údajov očakávaná v odpovedi na príkaz prípadu 2 alebo prípadu 4
SK	Dĺžka Hodnota
MAC	Kód na overenie správy
MEL	Spustiteľný jazyk MULTOS. Súbor inštrukcií pre beh systému MULTOS ™
OID	Identifikátor objektu
P1	Referenčný riadiaci parameter 1
P2	Referenčný kontrolný parameter 2
PIN	Osobné identifikačné číslo
PKI	Infraštruktúra verejného kľúča
RAM	Pamäť s náhodným prístupom
RFU	Vyhradené na budúce použitie
RID	Identifikátor registrovaného poskytovateľa aplikácie
ROM	Pamäť len na čítanie
RSA	Pamäť len pre nity
SCP	Secure Channel Protocol: alebo (ETSI) Smart Card Platform
SW	Stavové slovo
SW1	Štát Slovo jedna
SW2	Stavové slovo dva
TLV	Tag Dĺžka Hodnota
TP	Bod dôvery
"xx"	Hexadecimálne hodnoty sú vyjadrené ako hexadecimálne číslice medzi jednoduchými úvodzovkami.
"X"	Hodnota v bunke tabuľky, ktorej účel je opísaný v stĺpci "Význam" tabuľky.
"-"	Hodnota (0 alebo 1) v bunke tabuľky, ktorá nemá vplyv na "Význam" určený pre daný riadok tabuľky.

Špecifikácie karty

V tejto časti sú zhrnuté vlastnosti a funkcie ACOSJ-G.

Elektrické špecifikácie

Prevádzkové napätie (ACOSJ-G v1.01):	Napájacie napätie 2,7 V až 5,5 V
Prevádzkové napätie (ACOSJ-G v2.04):	Napájacie napätie 2,1 V až 5,5 V
Maximálna frekvencia externých hodín:	10 MHz
Maximálna taktovacia frekvencia procesora:	28 MHz
Ochrana pred ESD	vyššie ako 5 kV (HBM)

Špecifikácia životného prostredia

Prevádzková teplota:	-25°C až +85°C

Komunikačné protokoly

- Protokol T=CL s prenosovou rýchlosťou do 848 kb/s

Pamäť

Kapacita:	95 KB (ACOSJ-G 2.04), 40 KB (ACOSJ-G 1.01)
Životnosť pamäte EEPROM:	500 000 cyklov mazania/zápisu (25 °C)
Uchovávanie údajov:	30 rokov (25 °C)

Kryptografické funkcie

DES:	2K3DES,3K3DES (ECB a CBC)
AES:	128/192/256 bitov (ECB a CBC)
RSA:	768 až 2048 bitov
ECC:	Modul 112/128/160/192/224/256/384 bitov
Hash:	SHA1, SHA256, SHA384, SHA512
SEED:	128 bitov

- SM2/SM3/SM4

Dodržiavanie noriem

- Súlad s normou ISO 7816 časť 4
- Súlad s normou ISO 14443 (typ A a B)
- Súlad so špecifikáciou karty JAVA verzie 3.0.4
- Súlad s globálnou špecifikáciou platformy verzie 2.2.1
- Súlad s usmerneniami pre mapovanie 1.0.1

Odpoveď na výber (ATS, bezkontaktná karta)

Po prijatí príkazu Request for Answer to Select (RATS) od čítačky kariet karta vyšle príkaz Answer to Select (ATS) v súlade s normou ISO 14443, časť 4.

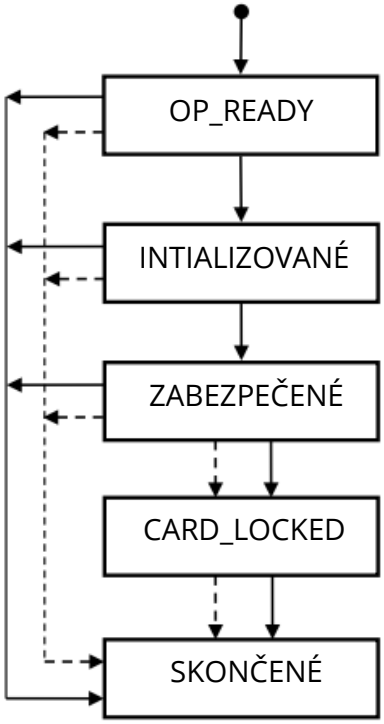
V nasledujúcej tabuľke sú uvedené predvolené systémy ATS:

Parameter	ATS	Popis
TL	0Eh	Dĺžka
T0	78h	Formálne byte...kódy Y(1) a FSCI
TA1	00h	Interface byte...DS a DR kódy
TB1	71h	Kódy FWI a SFGI
TC1	02h	Možnosti záznamu kódu
T1	41h	Označuje "A"
T2	43h	Označuje "C"
T3	4Fh	Označuje "O"
T4	53h	Označuje "S"
T5	4Ah	Označuje "J"
T6	76h	Označuje "in"
T7~T9	31h 30h 31h alebo 32h 30h 34h	Označuje "101" alebo označuje "204".

Poznámka: Úplný opis systému ATS nájdete v norme ISO 14443, časť 4.

Stavy životného cyklu karty

ACOSJ má päť stavov kariet: OP_READY, INTILIZED, SECURED, CARD_LOCKED a TEDMINATED. Na nasledujúcom obrázku je znázornený prechod medzi jednotlivými stavmi životného cyklu karty:



Legenda

Privilegovaná bezpečnostná doména

Privilegovaná aplikácia

OP_READY

Tento stav znamená, že prostredie behu je k dispozícii a doména zabezpečenia vydavateľa, ktorá vystupuje ako vybraná aplikácia, je pripravená prijímať, vykonávať a reagovať na príkazy APDU.

Ak je karta v stave OP_READY, musia byť k dispozícii nasledujúce funkcie:

- Bežecké prostredie musí byť pripravené na vykonávanie.
- OPEN musí byť pripravený na vykonanie.
- Bezpečnostná doména vydavateľa je predvolená aplikácia vybraná pre všetky rozhrania karty.
- Spustiteľné súbory načítania, ktoré boli zahrnuté do nemennej trvalej pamäte, sú zaregistrované v registri GlobalPlatform.
- Počiatočný kľúč musí byť k dispozícii v bezpečnostnej doméne vydavateľa.

Karta musí mať možnosť meniť obsah karty a Načítať súbory obsahujúce aplikácie, ktoré ešte nie sú na karte, sa môžu načítať.

Môže sa nainštalovať zo spustiteľných súborov, ktoré sa majú načítať, aby sa vytvorili aplikácie.

Ak sú v tejto fáze k dispozícii personalizačné informácie, žiadosti možno ďalej personalizovať.

Stav OP_READY možno použiť mimo karty na vykonanie nasledujúcich činností:

- Je možné načítať a/alebo nainštalovať ďalšie domény zabezpečenia.
- Kľúče bezpečnostnej domény môžu byť vložené, aby sa zachovalo kryptografické oddelenie kľúčov od kľúčov bezpečnostnej domény vydavateľa.

Inicializované

Toto je stav výroby administratívnych kariet. Prechod zo stavu OP_READY do stavu INITIALIZED je nevratný. Jeho funkčnosť je mimo rozsahu tejto špecifikácie. Tento stav sa môže použiť na označenie toho, že niektoré počiatočné údaje (napr. kľúče a/alebo údaje o bezpečnostnej doméne vydavateľa) boli vyplnené, ale karta ešte nie je pripravená na vydanie držiteľovi karty.

Zabezpečené

Tento stav je zamýšľaným stavom životného cyklu operačnej karty po jej vydaní. Tento stav môžu využívať bezpečnostné domény a aplikácie na presadzovanie svojich príslušných bezpečnostných politík. Prechod zo stavu INITIALIZED do stavu SECURED je nevratný.

Stav SECURED by sa mal používať na označenie subjektov, ktoré nie sú držiteľmi kariet, že bezpečnostná doména vydavateľa obsahuje všetky potrebné kľúče a bezpečnostné prvky pre plnú funkčnosť.

Card_Locked

Stav životného cyklu karty CARD_LOCKED umožňuje vypnúť výber bezpečnostnej domény a aplikácie. Prechod zo stavu SECURED do stavu CARD_LOCKED je vratný.

Nastavenie karty na CARD_LOCKED znamená, že karta umožní výber len aplikácií s povolením Final Application.

V tomto stave nie sú povolené zmeny obsahu karty vrátane akéhokoľvek typu správy údajov (najmä údajov o kľúčoch a bezpečnostných doménach).

Prechod zo stavu SECURED do stavu CARD_LOCKED môže byť iniciovaný buď doménou OPEN, bezpečnostnou doménou s povolením Card Lock alebo aplikáciou s povolením Card Lock.

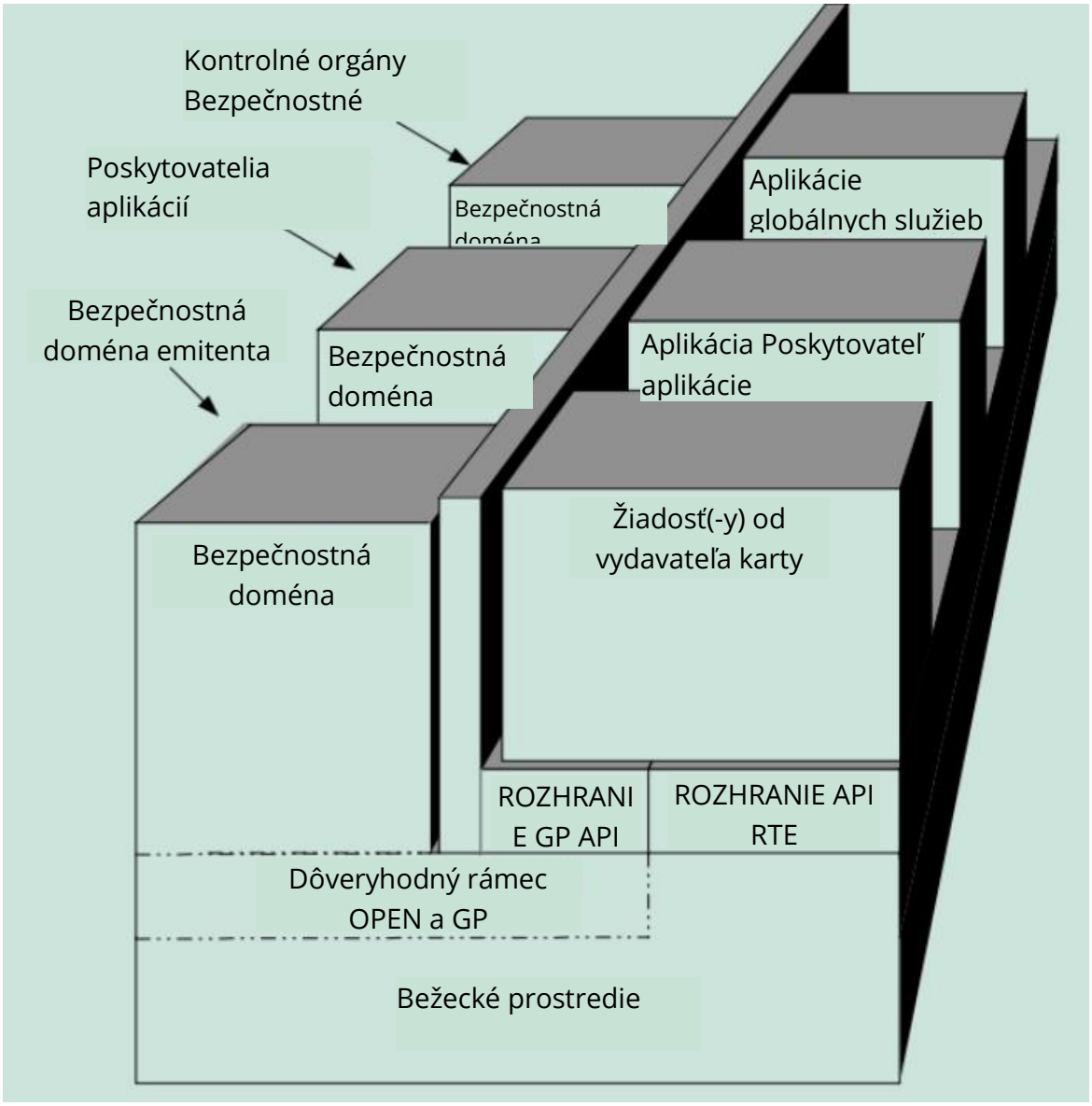
UKONČENÉ

Tento stav signalizuje koniec životného cyklu karty a karty. Prechod z akéhokoľvek iného stavu do stavu TERMINATED je nevratný.

Stav TERMINATED sa používa na trvalé vypnutie všetkých funkcií karty, pokiaľ ide o akúkoľvek správu obsahu karty a akékoľvek zmeny životného cyklu. Tento stav karty je určený ako mechanizmus, pomocou ktorého aplikácia logicky "zničí" kartu napríklad z dôvodu zistenia závažného ohrozenia bezpečnosti alebo uplynutia platnosti karty. Ak má bezpečnostná doména povolenie Final Application, spracuje sa iba príkaz GET DATA; všetky ostatné príkazy definované v tejto špecifikácii sú zakázané a vrátia chybu. Id aplikácia má povolenie na finálnu aplikáciu, jej spracovanie príkazov podlieha politike vydavateľa.

Prechod z ktoréhokoľvek z predchádzajúcich stavov do stavu Ukončené môže iniciovať samotný OPEN, bezpečnostná doména s oprávnením Ukončiť kartu alebo aplikácia s oprávnením Ukončiť kartu.

Architektúra karty



Odkaz na príkazy GP APDU

Táto časť podrobne opisuje príkazy APDU GlobalPlatform, ktoré možno implementovať. Príkazy sú uvedené v abecednom poradí.

Sumarizuje príkazy APDU, ktoré sú podporované vydavateľskými bezpečnostnými doménami, a požiadavky na podporu týchto príkazov APDU inými bezpečnostnými doménami. Ak sú podporované logické kanály, príkaz MANAGE CHANNEL spracúva iba príkaz OPEN a pre tento príkaz sa nevyžaduje podpora domény zabezpečenia.

Príkaz	OP_READY			INITIAL			ZABEZPEČENÉ			CARD_LOCKED		SKONČENÉ	
	AM SD	DM SD	SD	AM SD	DM SD	SD	AM SD	DM SD	SD	FASD	SD	FASD	SD
DELETE Spustiteľný súbor na načítanie													
DELETE Načítanie spustiteľných súborov a súvisiacich aplikácií													
DELETE Aplikácia	✓			✓			✓						
Kľúč DELETE													
ZÍSKAŤ DÁTA	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
ZÍSKAJTE STAV	✓			✓			✓			✓			
INSTALL [pre zaťaženie]													
INSTALL [nainštalovať]													
INSTALL [načítať, nainštalovať a vybrať]													
INSTALL [pre inštaláciu a výber]	✓	✓		✓	✓		✓	✓					
INSTALL [pre možnosť]													
INSTALL [pre vydanie]													
INSTALL [aktualizácia registra]													
INSTALL [pre personalizáciu]													
LOAD													
VLOŽIŤ KĽÚČ	✓			✓			✓						
VYBRAŤ	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
NASTAVIŤ STAV	✓			✓			✓			✓			
UKLADANIE ÚDAJOV	✓			✓			✓						

AM SD	Bezpečnostná doména s oprávnením na autorizovanú správu.
DM SD	Bezpečnostná doména s delegovanými oprávneniami na správu.
FA SD	Bezpečnostná doména s povolením Final Application.

Poznámka: Podpora príkazov a aplikácií s právami Final Application Privilege, ktoré nie sú bezpečnostnou doménou, podlieha politike vydavateľa v rámci obmedzení povolených stavom životného cyklu karty.

SD	Ostatné bezpečnostné domény.
✓	Potrebná podpora.
Prázdna bunka	Podpora je voliteľná.
Sivá bunka	Podpora je vypnutá.

Sumarizuje minimálne bezpečnostné požiadavky na príkazy APDU.

Príkaz	Minimálne zabezpečenie
DELETE	Iniciácia zabezpečeného kanála alebo overenie digitálneho podpisu
ZÍSKAŤ DÁTA	Nie
ZÍSKAŤ STAV	Iniciovanie zabezpečeného kanála
INŠTALÁCIA	Iniciácia zabezpečeného kanála alebo overenie digitálneho podpisu
LOAD	Iniciácia zabezpečeného kanála alebo overenie digitálneho podpisu
SPRÁVA KANÁLU	Neuplatňuje sa
VLOŽIŤ KĽÚČ	Iniciovanie zabezpečeného kanála
VYBRAŤ	Neuplatňuje sa
NASTAVIŤ STAV	Iniciovanie zabezpečeného kanála
ULOŽENIE DÁT	Iniciovanie zabezpečeného kanála

Všeobecné pravidlá kódovania

Pozrite si referenčnú príručku ACOSJ.

Príkaz DELETE

Príkaz DELETE sa používa na vymazanie jednoznačne identifikovateľného objektu, ako je napríklad načítateľný spustiteľný súbor, aplikácia, načítateľný spustiteľný súbor a súvisiace aplikácie alebo kľúč. Aby bolo možné objekt aj vymazať, mal by byť objekt jednoznačne identifikovateľný vybranou aplikáciou.

Príkaz GET DATA

Príkaz GET DATA sa používa na získanie buď jedného dátového objektu, ktorý možno skonštruovať, alebo množiny dátových objektov. Riadiace parametre referenčného kódovania P1 a P2 sa používajú na definovanie konkrétnej značky dátového objektu. Dátový objekt môže obsahovať informácie súvisiace s kľúčom.

Príkaz GET STATUS

Pomocou príkazu GET STATUS môžete získať informácie o stave vydavateľskej domény zabezpečenia, spustiteľného súboru načítania, spustiteľného modulu, aplikácie alebo životného cyklu domény zabezpečenia podľa zadaných kritérií zhody/vyhľadávania.

Príkaz INSTALL

Príkaz INSTALL sa vydáva bezpečnostnej doméne na spustenie alebo vykonanie rôznych krokov potrebných na správu obsahu karty.

Príkaz LOAD

Táto časť definuje štruktúru súboru Load File prenášaného v dátovom poli príkazu LOAD na načítanie súboru Load File. Interné spracovanie alebo ukladanie súboru Load File v ICC je mimo rozsahu tejto špecifikácie.

Na prenos súboru Load na kartu možno použiť viacero príkazov LOAD. Súbor Load File sa na účely prenosu rozdelí na menšie zložky. Každý príkaz LOAD je číslovaný od 00h. Číslovanie príkazov LOAD je prísne sekvenčné a zvyšuje sa o jednotku. Karta sa informuje o poslednom bloku súboru Load File.

Po prijatí posledného bloku súboru Load karta spustí vnútorné procesy potrebné pre súbor Load a všetky ostatné procesy identifikované v príkaze INSTALL [na načítanie], ktorý predchádzal príkazom LOAD.

Príkaz MANAGE CHANNEL

Príkaz MANAGE CHANNEL sa spracúva otvorene na kartách, ktoré poznajú logické kanály. Používa sa na otváranie a zatváranie ďalších logických kanálov. Základný logický kanál (kanál číslo nula) nemôže byť nikdy uzavretý.

Príkaz PUT KEY

Príkaz PUT KEY sa používa buď na:

- Výmena existujúceho kľúča za nový kľúč: Nový kľúč má rovnaké alebo iné číslo verzie kľúča, ale rovnaký identifikátor kľúča ako nahrádzaný kľúč.
- Nahradenie viacerých existujúcich kľúčov novými kľúčmi: Nové kľúče majú rovnaké alebo iné číslo verzie kľúča (rovnaké pre všetky nové kľúče), ale rovnaké identifikátory kľúča ako nahradené kľúče.
- Pridanie jedného nového kľúča: Nový kľúč má inú kombináciu identifikátora kľúča/čísla verzie kľúča ako existujúce kľúče.
- Pridanie viacerých nových kľúčov: nové kľúče majú odlišné kombinácie identifikátora kľúča a čísla verzie kľúča (rovnaké pre všetky nové kľúče) ako existujúce kľúče.

Ak si operácia správy kľúčov vyžaduje viacero príkazov PUT KEY, odporúča sa reťazenie viacerých príkazov PUT KEY, aby sa zabezpečila integrita operácie.

V tejto verzii špecifikácie sú verejné hodnoty asymetrických kľúčov prezentované (prenášané) v otvorenom texte.

Príkaz SELECT

Príkaz SELECT sa používa na výber aplikácie. OPEN spracúva iba príkazy SELECT s uvedením SELECT [podľa názvu], ktoré sa odovzdávajú aktuálne vybranej bezpečnostnej doméne alebo aplikácii na určenom logickom kanáli.

Príkaz SET STATUS

Príkaz SET STATUS sa používa na zmenu stavu životného cyklu karty alebo stavu životného cyklu aplikácie.

Príkaz STORE DATA

Príkaz STORE DATA sa používa na prenos údajov do aplikácie alebo domény zabezpečenia, ktorá príkaz spracúva.

Doména zabezpečenia určí, či je príkaz určený pre ňu alebo pre aplikáciu, v závislosti od príkazu, ktorý bol predtým prijatý. Ak bol predchádzajúci príkaz a príkaz INSTALL [pre personalizáciu] rovnaký, príkaz STORE DATA je určený pre aplikáciu.

Viacnásobný príkaz STORE DATA sa používa na odosielanie údajov do aplikácie alebo bezpečnostnej domény rozdelením údajov na menšie zložky na prenos. Bezpečnostná doména je informovaná o poslednom bloku.

Personalizačná relácia sa začína, keď bezpečnostná doména prijme platný príkaz INSTALL [pre personalizáciu], ktorý špecifikuje aplikáciu (implementujúcu buď aplikáciu, alebo personalizačné rozhranie), ktorej má bezpečnostná doména postúpiť následne prijaté príkazy STORE DATA.

Personalizácia sa skončí, keď:

- Karta sa resetuje.
- Bezpečnostná doména je vymazaná (t. j. v tom istom logickom kanáli je vybraný iný alebo rovnaký applet).
- Bezpečnostná doména je vybraná na rovnakom alebo inom logickom kanáli.
- Relácia zabezpečeného kanála (ak existuje) vytvorená bezpečnostnou doménou je resetovaná, prípadne cieľovou aplikáciou.
- Bezpečnostná doména dostane príkaz INSTALL [pre personalizáciu] (na spustenie novej relácie personalizácie pre inú aplikáciu).
- Bezpečnostná doména dostane príkaz STORE DATA s označením P1. b8=1 (posledný blok)

Každý príkaz STORE DATA prijatý z takejto personalizačnej relácie spracuje samotná bezpečnostná doména.

API GlobalPlatform

Globálna platforma na karte JAVA

Táto časť obsahuje len API potrebné pre karty Java Global Platform 2.2x. Používanie rozhrania API Open Platform 2.0.1 je stále povolené na podporu starších verzií aplikácií, ale je zastarané. Je definovaný vo verzii 2.1.1 tejto špecifikácie.

Ak je to možné, staršie aj nové rozhrania API pristupujú k rovnakým objektom. Hoci sa to môže zdať samozrejmé v prípade metód, ktoré majú rovnaký názov v oboch triedach (napr. setATRHistBytes(), setCardContentState() a getCardContentSame()), treba tiež poznamenať, že napríklad aplikácia, ktorá používa metódu updated() v novom rozhraní API na zmenu hodnoty globálneho kódu PIN, ovplyvní rovnaký globálny kód PIN aplikácie, ktorá používa metódu setPIN() v zastaranom rozhraní API na overenie globálneho kódu PIN.

Špecifické požiadavky GlobalPlatform

Aby sa zabezpečila najvyššia úroveň interoperability medzi implementáciami GlobalPlatform, GlobalPlatform prijíma aj usporiadanie definované v časti 6.2 - Špecifikácia virtuálneho stroja Java Card™ 2.2x.

Implementácia GlobalPlatform obsahuje nasledujúce drobné úpravy štandardných funkcií definovaných v špecifikáciách Java Card Runtime Environment (JCRE)™ 2.1.1 a Java Card Application Programming Interface (JAPI)™ 2.1.1. Niektoré úpravy sú už v súlade so špecifikáciou Java Card™ 2.2.x, a preto už nie sú úpravami.

Objekty GPRegistryEntry sú implantované ako objekty zdieľaného rozhrania, ako je definované v časti 6.2.4 "Zdieľané rozhrania" špecifikácie Java Card™ 2.2.x Runtime Environment (JCRE), aby sa zabezpečil zdieľaný prístup.

Balíček GlobalPlatform AID

Každý AID balíka GlobalPlatfrom bude súčtom RID a PIX. Hodnota súboru AID pre export karty Java pre nové rozhranie API GlobalPlatform (rovnaká pre GlobalPlatform 2.1 aj 2.1.1) na základe identifikátora RID uvedeného v dodatku H - Hodnoty údajov GlobalPlatform a údaje o rozpoznávaní karty je "A00000015100".

Inštalácia

V časti 3.1, Inštalácia metódy špecifikácie prostredia Java Card™ 2.2.x Runtime Environment (JCRE), sú parametre odovzdané metóde definované ako inicializačné parametre z obsahu parametra prichádzajúceho poľa bajtov.

Táto špecifikácia rozširuje túto požiadavku a ďalej definuje obsah parametrov inštalácie. Toto rozšírenie ovplyvňuje implementáciu OPEN aj správanie appletu Java Card vytvoreného pre kartu GlobalPlatform. Nemá vplyv na definíciu metódy install() triedy Applet špecifikácie aplikačného programového rozhrania Java Card™ 2.2.x.

Parametre inštalácie identifikujú nasledujúce informácie, ktoré sú obsiahnuté v príkaze INSTALL:

- Inštancia AID
- Privilégiá
- Parametre špecifické pre aplikáciu. (Zatiaľ čo príkaz APDU obsahuje inštalačné parametre predstavujúce systémové a aplikačné špecifické parametre zakódované TLV, aplikácia vyžaduje len znalosť aplikačných špecifických parametrov, t. j. ako aplikačné špecifické parametre sú prítomné len LV štruktúry TLV zakódované "C9".

OPEN je zodpovedný za to, aby parametre (bArray, bOffset a bLength) obsahovali tieto informácie:

Pole bArray obsahuje po sebe idúce údaje s kódom LV:

- Dĺžka inštancie AID
- Inštancia AID
- Trvanie povolenia
- Privilégiá
- Dĺžka špecifických parametrov aplikácie
- Špecifické parametre aplikácie

Bajt bOffset obsahuje offset v rámci poľa, ktorý udáva dĺžku inštancie AID.

Bajt bLength obsahuje dĺžku udávajúcu celkovú dĺžku vyššie definovaných údajov v poli.

Pri volaní metódy register (byte [] bArray, short bOffset, byte bLength) triedy Applet špecifikácie aplikačného programového rozhrania Java Card 2.2.x musí applet ako parameter použiť identifikátor inštancie AID.

T=0 Prenosový protokol

Karty GlobalPlatform sú navrhnuté tak, aby fungovali v čo najširšom rozsahu prostredí (t. j. v zariadeniach na prijímanie kariet). V súčasnosti špecifikácie Java Card 2.1.1 Runtime Environment (JCRE) a Java Card 2.2.x Runtime Environment (JCRE) opisujú správanie pre prípad 2 príkazov (pri použití protokolu T=0) v rozpore s EMV 2000. GlobalPlatform nariaďuje, že JCRE musí spracovať tento prípad príkazu v súlade s ISO/IEC 7816: applet, ktorý dostane príkaz prípadu 2, skonštruuje odpoveď a zavolá príslušné API na výstup údajov. Ak sú údaje menšie ako údaje očakávané terminálom, OPEN uloží údaje a vypíše kód odpovede "6Cxx" a počká, kým CAD znovu vydá príkaz so správnou dĺžkou. Po prijatí opätovne vydaného príkazu bude JCRE kontrolovať výstup uložených údajov.

Atomicita
Ak nie je uvedené inak, všetky interné perzistentné objekty API GlobalPlatform musia byť konzistentné s aktuálnou transakciou.

Všetky operácie vykonávané týmto rozhraním API, okrem metódy Application.processData(), sa vykonávajú atomicky. Objekty použité na vynútenie implementácie riadenia rýchlosti sa nesmú zhodovať s prebiehajúcou transakciou.

Logické kanály
Pre kartu Java Card 2.2.x platia nasledujúce obmedzenia logických kanálov (viac informácií nájdete v špecifikáciách prostredia Java Card 2.2.x Runtime Environment (JCRE)):

Výber aplikácie v logickom kanáli, ako je definovaný v časti 6.3, Odosielanie príkazov, zlyhá, ak je tá istá aplikácia alebo akákoľvek iná aplikácia inštanciovaná z kódu v tom istom balíku, z ktorého bola inštanciovaná vybraná aplikácia, aktuálne vybraná v inom logickom kanáli, ale kód aplikácie neimplementuje rozhranie MultiSelectable. Domény zabezpečenia musia implementovať rozhranie MultiSelectable.

Zmena kontextu z bezpečnostnej domény na aplikáciu podľa definície v časti 7.3.3 Podpora personalizácie zlyhá, ak je tá istá aplikácia alebo akákoľvek iná aplikácia inštanciovaná z kódu v tom istom balíku, z ktorého bola inštanciovaná personalizovaná aplikácia, aktuálne vybraná v inom logickom kanáli, ale kód aplikácie neimplementuje rozhranie MultiSelectable.

Aplikácia, ktorá má právo byť štandardne vybraná a je určená pre kartu, ktorá podporuje ďalšie logické kanály, by mala implementovať rozhranie MultiSelectable.

GlobalPlarform definuje iba priradenie logických čísel kanálov podľa karty. Voliteľne a podľa definície v Java Card 2.2 môže karta podporovať aj priradenie logických čísel kanálov terminálu.

Kryptografické algoritmy
Karty GlobalPlatform, ktoré podporujú kryptografiu RSA, by mali podporovať veľkosti kľúčov, ktoré nie sú definované ako konštanty v triede Key Builder. Konkrétne by sa mala poskytovať podpora pre veľkosti kľúčov, ktoré sú násobkom 4 bajtov (32 bitov) a ktoré sú v rámci povolených dĺžok kľúčov definovaných implementáciou.

Úroveň dôvery
Špecifikácie Java Card 2.2.x predpokladajú, že RID balíkov AID, appletov a inštancií sa budú používať na zabezpečenie úrovne dôveryhodnosti medzi týmito entitami. V časti 4.2.2 - Používanie AID aplikačného programového rozhrania Java Card 2.2.1 sa definuje, že RID komponentu AID sa musí zhodovať s RID balíka AID, a v definícii metódy register (byte [] bArray, short bOffset, byte bLength) Java Card 2.2.1 sa definuje, že RID balíka AID sa musí zhodovať s RID komponentu AID.2.x Application Programming Interface je definované, že výnimka musí byť vyhodená, ak sa časť RID bajtov AID v parametri bArray nezhoduje s časťou RID názvu appletu Java Card.

Z hľadiska reálnej implementácie nie je praktické nariadiť, aby RID inštancie AID bol rovnaký ako RID komponentu, z ktorého bola inštancia vytvorená. Implementácia GlobalPlatform nesmie vyžadovať žiadnu väzbu prostredníctvom inštancie AID na jej pôvodný balík. (Nevyžaduje sa väzba medzi inštanciou AID a AID jej pôvodného balíka). Predpokladá však, že všetky aplikácie v tom istom balíku majú rovnakú úroveň dôveryhodnosti.

Volanie metód GlobalPlatform

Tu definované aplikačné programovacie rozhranie je prístupné pre akýkoľvek applet Java Card vyvinutý so zámerom byť prítomný na karte GlobalPlatform. Na konštruktor appletu a metódu install() triedy Applet aplikačného programového rozhrania Java Card 2.2.x sa nevzťahuje žiadne obmedzenie. Keďže táto špecifikácia presne nedefinuje, kedy sa inštancia appletu stane záznamom v registri kariet GlobalPlatform, vývojár appletu môže len predpokladať, že sa tak stalo po úspešnom dokončení metódy install(). Na zabezpečenie interoperability sa metódy API GlobalPlatform, ktoré pristupujú k položke registra GlobalPlatform appletu, ktorý metódu volá, nesmú volať z konštruktora alebo metódy install().

Nasledujúci zoznam metód, ktoré možno zavolať z konštruktora alebo metódy install(), sa stratil:

- getCardState;
- getCVM;
- getService;

Nie je definované požadované správanie karty v prípade, že aplikácia nesprávne zavolá inú metódu triedy org.globalplatform.GPSystem, ako je uvedená vyššie. Môže sa napríklad vyhodiť výnimka a spracovanie metódy install() sa môže prerušiť.

Výber

Ak počas spracovania metódy select() dôjde k chybe na kartách GlobalPlatform alebo ak metóda select() vráti False, alebo ak aplikáciu nemožno vybrať, pretože neimplementuje rozhranie Multiselectable, OPEN bude pokračovať v hľadaní následnej úplnej alebo čiastočnej zhody v registri GlobalPlatform, ako je definované v časti 6.4.2.1.2. Keďže v súčasnosti nie je vybraná žiadna aplikácia, akýkoľvek ďalší príkaz okrem príkazu MANAGE CHANNEL alebo SELECT bude odmietnutý. Od subjektu mimo karty sa očakáva, že pri takejto chybe vykoná príslušné kroky, napr. vyberie inú aplikáciu, uzavrie príslušný logický kanál, resetuje kartu alebo ju vypne.

Zhrnutie a podrobnosti o metódach pre špecifikáciu GlobalPlatform Java Card API sú teraz k dispozícii v samostatnom dokumente, ktorý nájdete na webovej stránke GlobalPlatform.

Pred personalizáciou

Pri prvom zapnutí je IC v stave pred personalizáciou.

Príkaz INIT_CARD (pozri 8.2.2 Príkaz INIT CARD) musí byť prvým príkazom v stave pred personalizáciou, príkaz inicializuje oblasť pamäte karty predvolenými hodnotami a potom je k dispozícii tzv. aplikácia ACOSJ ROOT (pozri 8.0 Aplikácia ACOSJ ROOT) na zápis kritických systémových údajov (napr. "ATR", "ATS" atď.) do EEPROM. Po načítaní údajov a potvrdení, že neboli vykonané žiadne ďalšie zmeny, je potrebné vykonať príkaz na vypnutie aplikácie ACOSJ ROOT.

Keď IC opustí stav pred personalizáciou úspešným vykonaním príkazu Active Card (pozri 8-2.5 Príkaz ACTIVATE), nachádza sa v predvolenom stave životného cyklu karty GlobalPlatform OP_READY, pokiaľ nie je nastavený iný stav životného cyklu.

Po úspešnom spracovaní príkazu aktívnej karty nie je možné vrátiť sa do stavu pred personalizáciou.

Poznámka: Transportný kľúč je kľúč pred personalizáciou koreňovej aplikácie ACOSJ a o transportný kľúč musíte požiadať ACS.

Aplikácia ACOSJ ROOT

ACOSJ ROOT Popis aplikácie

Príkazy v aplikácii ROOT je možné použiť až po správnom výbere aplikácie ROOT (pomocou príkazu Select Transport Key). Po vstupe do aplikácie ROOT môže používateľ čítať alebo konfigurovať parametre karty pomocou príkazov podporovaných aplikáciou ROOT. Ak chcete ukončiť aplikáciu ROOT a vybrať inú aplikáciu, musíte kartu resetovať.

Parametre karty môžete čítať alebo konfigurovať v aplikácii ROOT.

Po aktivácii karty sa aplikácia ROOT stane neplatnou.

Príkazy aplikácie ACOSJ ROOT

Príkaz SELECT

Príkaz SELECT sa používa na výber aplikácie ROOT.

Príkaz INIT_CARD

Tento príkaz sa používa na resetovanie karty. Tento príkaz inicializuje celú pamäť EEPROM.

PREČÍTAŤ PRÍKAZ

Tento príkaz sa používa na čítanie z konfiguračnej oblasti. Tento príkaz možno použiť na načítanie konfiguračných parametrov karty.

Príkaz WRITE

Tento príkaz sa používa na zápis údajov do konfiguračnej oblasti. Tento príkaz možno použiť na načítanie konfiguračných parametrov karty.

Príkaz ACTIVE

Tento príkaz sa používa na aktiváciu karty. Po úspešnom vykonaní tohto príkazu sa aplikácia ROOT stane neplatnou a konfiguračné údaje karty už nie je možné priamo čítať ani nastavovať.

Aplikácia ACOSJ IDENTIFY

ACOSJ IDENTIFY Popis aplikácie

Po výbere aplikácie IDENTIFY (AID: 6163732E636F732E61636F736A766572) príkazom SELECT vráti ACOSJ číslo verzie ACOSJ a uvedie, či bol automobil aktivovaný.

Odkaz na príkaz aplikácie ACOSJ IDENTIFY

Príkaz SELECT

Príkaz SELECT sa používa na výber aplikácie INDENTIFY.

Stavové bajty

SW1 SW2	Popis
90 00h	Správny príkaz
61 XX	SW2 k dispozícii viac bajtov odpovede
62 83h	Aplikácia je trvalo zablokovaná
63 00h	Overenie sa nepodarilo
63 CX	X zostávajúcich opakovaných pokusov
67 00h	Nesprávna dĺžka údajov
68 81h	Logický kanál nie je podporovaný
68 82h	Zabezpečené zasielanie správ nie je podporované
69 84h	Neplatné údaje
69 82h	Stav zabezpečenia nie je splnený
69 85h	Nesplnené podmienky používania
69 88h	Zlá MAC
6A 86h	Nesprávne P1 P2
6A 80h	Nesprávne hodnoty v príkazových údajoch
6A 82h	Súbor nebol nájdený
6A 81h	Aplikácia je uzamknutá
6A 83h	Nenašiel sa žiadny záznam
6A 84h	Nedostatok miesta v pamäti
6A 88h	Odkazované údaje neboli nájdené
6D 00h	Neplatný INS, objednávka neexistuje
6E 00h	Neplatná trieda

Odkazy

Ako referenčné dokumenty pre funkčnú špecifikáciu ACOSJ boli použité tieto dokumenty:

- Špecifikácie karty GlobalPlatform verzia 2.2.1
- GlobalPlatform Card API verzia 1.6
- Rozhranie Java Card 3 API, verzia 3.0.4 Classic Edition
- Špecifikácia prostredia Java Card 3 Platform Runtime Environment, Classic Edition Verzia 3.0.4 September 2011
- Špecifikácia virtuálneho stroja platformy Java Card 3, verzia 3.0.4, klasická verzia september 2011
- Pokyny na mapovanie kariet GlobalPlatform z aktuálnej implementácie GP v2.1.1 na v2.2.1 1.0.1

Záručné podmienky

Na nový výrobok zakúpený v predajnej sieti Alza.sk sa vzťahuje záruka 2 roky. V prípade potreby opravy alebo iného servisu v záručnej dobe sa obráťte priamo na predajcu výrobku, je nutné predložiť originálny doklad o kúpe s dátumom nákupu.

Za rozpor so záručnými podmienkami, pre ktorý nemožno reklamáciu uznať, sa považujú nasledujúce skutočnosti:

- Používanie výrobku na iný účel, než na ktorý je výrobok určený alebo nedodržiavanie pokynov pre údržbu, prevádzku a servis výrobku.
- Poškodenie výrobku živelnou pohromou, zásahom neoprávnenej osoby alebo mechanicky vinou kupujúceho (napr. pri preprave, čistení nevhodnými prostriedkami a pod.).
- Prirodzené opotrebovanie a starnutie spotrebného materiálu alebo súčastí počas používania (napr. batérií atď.).
- Pôsobenie nepriaznivých vonkajších vplyvov, ako je slnečné žiarenie a iné žiarenie alebo elektromagnetické pole, vniknutie kvapaliny, vniknutie predmetu, prepätie v sieti, elektrostatický výboj (vrátane blesku), chybné napájacie alebo vstupné napätie a nevhodná polarita tohto napätia, chemické procesy, napr. použité zdroje atď.

Ak niekto vykonal úpravy, modifikácie, zmeny konštrukcie alebo adaptácie za účelom zmeny alebo rozšírenia funkcií výrobku oproti zakúpenej konštrukcii alebo použitie neoriginálnych súčastí.

Kedves vásárló,

Köszönjük, hogy megvásárolta termékünket. Kérjük, az első használat előtt figyelmesen olvassa el az alábbi utasításokat, és őrizze meg ezt a használati útmutatót a későbbi használatra. Fordítson különös figyelmet a biztonsági utasításokra. Ha bármilyen kérdése vagy észrevétele van a készülékkel kapcsolatban, kérjük, forduljon az ügyfélvonalhoz.

✉ www.alza.hu/kapcsolat

✆ +36-1-701-1111

Importőr Alza.cz a.s., Jankovcova 1522/53, Holešovice, 170 00 Prága 7, www.alza.cz

Áttekintés

Az ACOSJ az Advanced Card Systems Ltd. által kifejlesztett intelligens kártya operációs rendszer. A JAVA Card Virtual Machine alapján működik, és megfelel a GlobalPlatform Card Specification Version 2.2.1, JAVA Card Virtual Machine és megfelel a GlobalPlatform Card Specification Version 2.2.1, JAVA Card Specification Version 3.0.4 és Mapping Guidelines 1.0.1 funkciók és konfigurációk tekintetében.

E dokumentum célja, hogy részletesen ismertesse az ACOSJ intelligens kártya operációs rendszer jellemzőit és funkcióit.

ACOSJ-G verziók

Verzió	Közzétett adatok	Módosítások
ACOSJ v1.01	2015. június	• 40KB EEPROM • Működési feszültség: 2.7V és 5.5V között
ACOSJ v2.00	május 2018	• 95KB EEPROM • Működési feszültség: 2.1V és 5.5V között • A konfigurációs memória címének módosítása
ACOSJ v2.04	Április	• 95KB EEPROM • Működési feszültség: 2.1V és 5.5V között • A konfigurációs memória címének módosítása • Jellemzők javítása

Jelképek és rövidítések

Rövidítés	Leírás
AES	Fejlett titkosítási szabvány
AID	Alkalmazás azonosítója
APDU	Alkalmazási protokoll adategység
API	Alkalmazás programozási interfész
ASCII	Amerikai szabványos információcsere kód
ATR	Answer-to-Reset
ATQ	Answer-to-Request (érintés nélküli kártyák esetében)
BCD	Binárisan kódolt decimális
BER	Alapvető kódolási szabályok
CAT	Kártyaalkalmazási eszközkészlet: vagy kriptográfiai engedélyezési sablon
CBC	Cipher Block Chaining
CCT	A kriptográfiai ellenőrző összeg ellenőrzési referenciasablonja
CIN	Kártyaképszám/kártyaazonosító szám
CLA	A parancsüzenet osztálybájtja
CRT	Ellenőrzési referenciasablon
CT	Bizalmassági ellenőrzési referenciasablon

CVM	Kártyabirtokos-ellenőrzési módszer
DAP	Adathitelesítési minta
DEK	Adattitkosítási kulcs
DER	Megkülönböztetett kódolási szabályok
DES	Adattitkosítási szabvány
DST	Digitális aláírás ellenőrzési referenciasablonja
EKB	Elektronikus kódkönyv
EMV	Europay, Mastercard és VISA: az ICC fizetési rendszerekre vonatkozó specifikációira való hivatkozásként használatos.
ENC	Titkosítás
FCI	Fájlvezérlési információk
HEX	Hexadecimal
HMAC	Kulcs-Hash üzenet-hitelesítési kód
ICC	Integrált áramköri kártya
ICV	Kezdeti láncolási vektor
IIN	Kibocsátó azonosító száma
INS	A parancsüzenet utasításbájtja
ISO	Nemzetközi Szabványügyi Szervezet
Lc	Az adatok pontos hossza a 3. vagy 4. esetű parancsban
Le	A 2. vagy 4. esetű parancsra adott válaszként elvárt adatok maximális hossza
LV	Hossz Érték
MAC	Üzenet hitelesítési kód
MEL	MULTOS Végrehajtható nyelv. A MULTOS™ futtatási környezet utasításkészlete.
OID	Objektum azonosító
P1	Referencia vezérlő paraméter 1
P2	Referencia vezérlő paraméter 2
PIN-KÓD	Személyi azonosító szám
PKI	Nyilvános kulcsú infrastruktúra
RAM	Véletlen hozzáférésű memória
RFU	Jövőbeni felhasználásra fenntartva
RID	Regisztrált alkalmazásszolgáltató azonosítója
ROM	Csak olvasható memória
RSA	Kizárólag szegecselt memória
SCP	Biztonságos csatorna protokoll: vagy (ETSI) Smart Card platform
SW	Állapot szó
SW1	Status Word One
SW2	Második állapot szó
TLV	Címke hossza Érték
TP	Trust Point
"xx"	A hexadecimális értékek hexadecimális számjegyek formájában, szimpla idézőjelek között jelennek meg.
"X"	Egy táblázat cellájában lévő olyan érték, amelynek rendeltetését a táblázat "Jelentés" oszlopában írják le.
"-"	Egy táblázat cellájában lévő érték (0 vagy 1), amely nem befolyásolja a táblázat adott sorának "Jelentését".

Kártya specifikációk

Ez a szakasz összefoglalja az ACOSJ-G jellemzőit és funkcióit.

Elektromos specifikációk

Működési feszültség (ACOSJ-G v1.01):	2,7 V és 5,5 V közötti tápfeszültségek
Működési feszültség (ACOSJ-G v2.04):	2,1 V és 5,5 V közötti tápfeszültségek
Maximális külső órajel frekvencia:	10 MHz
Maximális CPU órajel frekvencia:	28 MHz
ESD védelem	5 kV-nál nagyobb (HBM)

Környezeti előírások

Üzemi hőmérséklet:	-25°C és +85°C között

Kommunikációs protokollok

- T=CL protokoll 848 kbps-ig terjedő bauddal

Memória

Kapacitás:	95 KB (ACOSJ-G 2.04), 40 KB (ACOSJ-G 1.01)
EEPROM tartósság:	500 000 törlési/írási ciklus (25°C)
Adatmegőrzés:	30 év (25°C)

Kriptográfiai funkciók

DES:	2K3DES,3K3DES (ECB és CBC)
AES:	128/192/256 bit (ECB és CBC)
RSA:	768-2048 bit
ECC:	Modulus 112/128/128/160/192/224/256/384 bitek
Hash:	SHA1, SHA256, SHA384, SHA512
SEED:	128 bit

- SM2/SM3/SM4

A szabványoknak való megfelelés

- Megfelelés az ISO 7816 4. részének
- Az ISO 14443 szabványnak való megfelelés (A és B típus)
- A JAVA Card Specification 3.0.4 verziójának való megfelelés
- Megfelelés a globális platform specifikáció 2.2.1. verziójának
- Megfelelés az 1.0.1. térképezési iránymutatásoknak

Válasz a kiválasztásra (ATS, érintésmentes kártya)

Miután a kártyaolvasó készüléktől megkapta a RATS (Request for Answer to Select) parancsot, a kártya az ISO 14443 4. részének megfelelően továbbítja a válasz a kiválasztásra (ATS) parancsot.

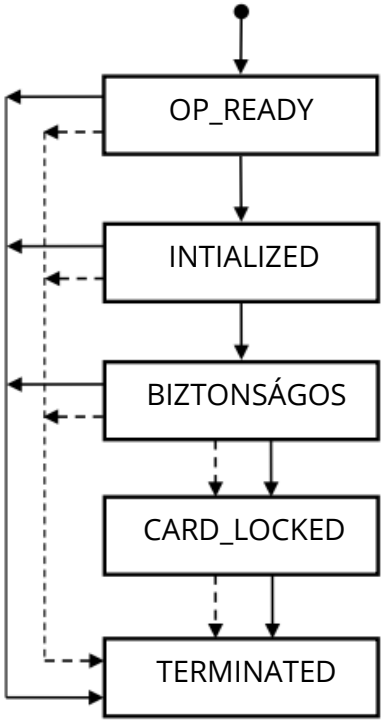
A következő táblázat az alapértelmezett ATS-t mutatja:

Paraméter	ATS	Leírás
TL	0Eh	Hosszúság
T0	78h	Y(1) és FSCI formális byte...kódok
TA1	00h	Interfészbájt...DS és DR kódok
TB1	71h	FWI és SFGI kódok
TC1	02h	Kódok protokoll opciók
T1	41h	"A" jelzi
T2	43h	"C" jelzi
T3	4Fh	"O" jelzi
T4	53h	"S" jelzi
T5	4Ah	Jelzi a "J" jelzést
T6	76h	Jelzi a "v" jelzést
T7~T9	31h 30h 31h vagy 32h 30h 34h	"101"-et vagy "204"-et jelez.

Megjegyzés: Az ATS teljes leírását lásd az ISO 14443 4. részében.

A kártya életciklusának állapotai

Az ACOSJ öt kártyaállománnyal rendelkezik: OP_READY, INTILIZED, SECURED, CARD_LOCKED és TEDMINATED. Az alábbi ábra a kártya életciklusának állapotátmenetét mutatja:



Legenda

Privilegizált biztonsági tartomány

Privilegizált alkalmazás

OP_READY

Ez az állapot azt jelzi, hogy a futásidejű környezet elérhető, és a kibocsátó biztonsági tartománya, amely a kiválasztott alkalmazásként működik, készen áll az APDU-parancsok fogadására, végrehajtására és megválaszolására.

A következő funkcióknak kell jelen lenniük, amikor a kártya OP_READY állapotban van:

- A futási környezetnek készen kell állnia a végrehajtásra.
- A Nyílt lap készen áll a végrehajtásra.
- A kibocsátó biztonsági tartománya az összes kártya interfészhez implicit módon kiválasztott alkalmazás.
- A megváltoztathatatlan tartós memóriában szereplő futtatható betöltési fájlokat regisztrálni kell a GlobalPlatform Registryben.
- A kibocsátó biztonsági tartományában rendelkezésre kell állnia egy kezdeti kulcsnak.

A kártyának képesnek kell lennie a kártya tartalmának módosítására, a kártyán még nem lévő alkalmazásokat tartalmazó Load Files betöltése előfordulhat.

Az aby Alkalmazás telepítése végrehajtható betöltési fájlokból történhet.

Továbbá, ha ebben a szakaszban bármilyen személyre szabott információ rendelkezésre áll, az Alkalmazások személyre szabhatók.

Az OP_READY állapotot a kártyán kívüli kártya a következő műveletek végrehajtására használhatja:

- Kiegészítő biztonsági tartományok tölthetők be és/vagy telepíthetők.
- A biztonsági tartomány kulcsai beilleszthetők annak érdekében, hogy a kriptográfiai kulcsok elkülönüljenek a kibocsátó biztonsági tartományának kulcsaitól.

Inicializált

Ez az állapot egy adminisztratív kártyagyártási állapot. Az OP_READY állapotból INITIALIZED állapotba való átmenet visszafordíthatatlan. Funkcionalitása meghaladja e specifikáció hatókörét. Ezt az állapotot arra lehet használni, hogy jelezze, hogy néhány kezdeti adatot feltöltöttek (pl. a kibocsátó biztonsági tartomány kulcsait és/vagy adatait), de a kártya még nem áll készen a kártyabirtokos számára történő kibocsátásra.

Biztosított

Ez az állapot a tervezett működési kártya életciklus-állapota a kibocsátás utáni időszakban. Ezt az állapotot a biztonsági tartományok és az alkalmazások használhatják saját biztonsági szabályzataik érvényesítésére. Az INITIALIZED állapotból a SECURED állapotba való átmenet visszafordíthatatlan.

A BIZTONSÁGOS állapotot arra kell használni, hogy jelezze a kártyán kívüli szervezetek számára, hogy a kibocsátó biztonsági tartománya tartalmazza a teljes funkcionalitáshoz szükséges összes kulcsot és biztonsági elemet.

Card_Locked

A kártya életciklusának CARD_LOCKED állapota azért van jelen, hogy a biztonsági tartomány és az alkalmazások kiválasztása letiltható legyen. A kártya életciklusának SECURED állapotából a CARD_LOCKED állapotba való átmenet visszafordítható.

A kártya CARD_LOCKED állapotba állítása azt jelenti, hogy a kártya csak a Végleges alkalmazás jogosultsággal rendelkező alkalmazás kiválasztását teszi lehetővé.

A kártya tartalmának módosítása, beleértve bármilyen típusú adatkezelést (különösen a biztonsági tartományok kulcsait és adatait), ebben az állapotban nem engedélyezett.

A Kártyazár jogosultsággal rendelkező NYITOTT vagy biztonsági tartomány, illetve a kártyazár jogosultsággal rendelkező alkalmazás kezdeményezheti az átmenetet a BIZTONSÁGOS állapotból a KÁRTYA_ZÁRVA állapotba.

TERMINATED

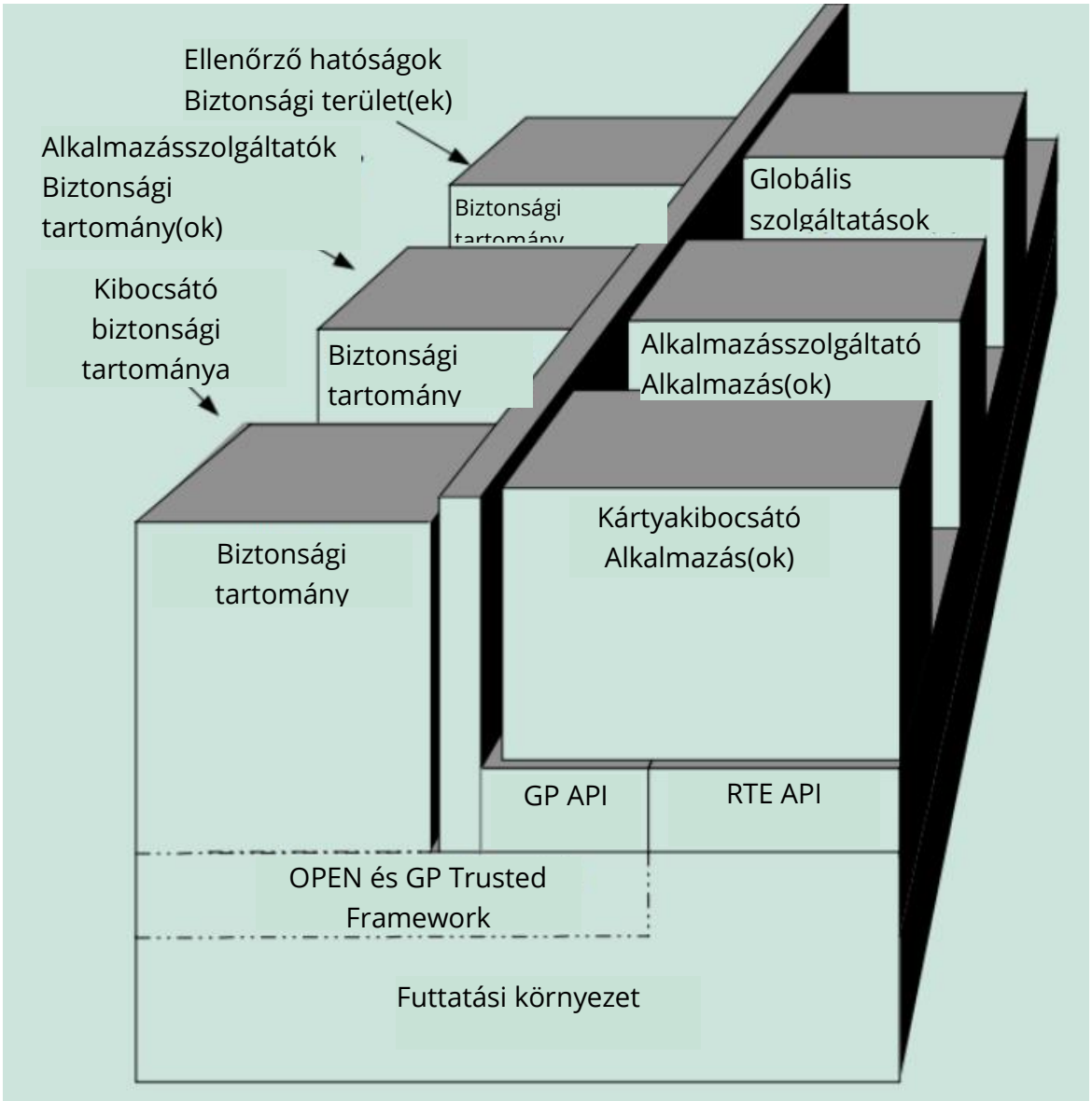
Ez az állapot jelzi a kártya életciklusának és a kártya végét. Bármely más állapotból a TERMINATED állapotba való átmenet visszafordíthatatlan.

A TERMINATED állapotot kell használni a kártya minden funkciójának végleges letiltására a kártya tartalmi kezelése és az életciklus módosításai tekintetében. Ez a kártyaállapot

olyan mechanizmus, amellyel az alkalmazás logikailag "megsemmisítheti" a kártyát olyan okokból, mint például súlyos biztonsági fenyegetés észlelése vagy a kártya lejárta. Ha egy biztonsági tartomány rendelkezik a végső alkalmazás jogosultsággal, csak a GET DATA parancsot kell feldolgozni, az ebben a specifikációban meghatározott összes többi parancsot le kell tiltani, és hibát kell visszaküldeni. Ha egy alkalmazás rendelkezik a végső alkalmazás jogosultsággal, a parancsfeldolgozás a kibocsátó politikájától függ.

Az OPEN maga, vagy a Kártya megszüntetése jogosultsággal rendelkező biztonsági tartomány, vagy a Kártya megszüntetése jogosultsággal rendelkező alkalmazás kezdeményezheti az előző állapotok bármelyikéből a TERMINATED állapotba való átmenetet.

Kártya architektúra



GP APDU parancshivatkozás

Ez a szakasz részletezi a GlobalPlatform APDU parancsokat, amelyek megvalósíthatók. A parancsok betűrendben vannak felsorolva.

Összefoglalja a kibocsátó biztonsági tartományai által támogatott APDU-parancsokat, valamint az ezen APDU-parancsok más biztonsági tartományok általi támogatására vonatkozó követelményeket. Ha a logikai csatornák támogatottak, a MANAGE CHANNEL parancsot csak az OPEN dolgozza fel, és ehhez a parancshoz nincs szükség biztonsági tartományi támogatásra.

Parancs	OP_READY			INITIALIZED			BIZTONSÁGOS			CARD_LOCKED		TERMINATED	
	AM SD	DM SD	SD	AM SD	DM SD	SD	AM SD	DM SD	SD	FASD	SD	FASD	SD
DELETE Végrehajtható betöltési fájl													
DELETE Futtatható betöltési fájl és a kapcsolódó alkalmazás(ok) törlése													
DELETE Alkalmazás	✓			✓			✓						
DELETE billentyű													
ADATOK MEGSZERZÉSE	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
ÁLLAPOT LEKÉRDEZÉSE	✓			✓			✓			✓			
INSTALL [a betöltéshez]													
INSTALL [a telepítéshez]													
INSTALL [a betöltés, telepítés és kiválasztás esetén]													
INSTALL [a telepítéshez és a választhatóvá tételhez]	✓	✓		✓	✓		✓	✓					
INSTALL [a választhatóvá tételhez]													
INSTALL [a kiadatáshoz]													
INSTALL [a rendszerleíró adatbázis frissítéséhez]													

INSTALL [a személyre szabáshoz]													
LOAD													
PUT KEY	✓			✓			✓						
SELECT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
ÁLLAPOT BEÁLLÍTÁSA	✓			✓			✓			✓			
ADATOK TÁROLÁSA	✓			✓			✓						

AM SD	Biztonsági tartomány engedélyezett kezelési jogosultsággal.
DM SD	Biztonsági tartomány delegált kezelési jogosultsággal.
FA SD	Biztonsági tartomány végső alkalmazási jogosultsággal.

Megjegyzés: A nem biztonsági tartománynak minősülő, végleges alkalmazási jogosultsággal rendelkező alkalmazások parancstámogatása a kibocsátó politikájától függ, a kártya életciklus-állapota szerint megengedett korlátokon belül.

SD	Egyéb biztonsági tartomány.
✓	Támogatás szükséges.
Üres cella	Támogatás opcionális.
Szürke sejt	Támogatás tilos.

Összefoglalja az APDU-parancsok minimális biztonsági követelményeit.

Parancs	Minimális biztonság
DELETE	Biztonságos csatorna kezdeményezése vagy digitális aláírás ellenőrzése
ADATOK MEGSZERZÉSE	Nincs
ÁLLAPOT LEKÉRDEZÉSE	Biztonságos csatorna kezdeményezése
INSTALLÁLÁS	Biztonságos csatorna kezdeményezése vagy digitális aláírás ellenőrzése
LOAD	Biztonságos csatorna kezdeményezése vagy digitális aláírás ellenőrzése
CSATORNA KEZELÉSE	Nem alkalmazható
PUT KEY	Biztonságos csatorna kezdeményezése
SELECT	Nem alkalmazható
ÁLLAPOT BEÁLLÍTÁSA	Biztonságos csatorna kezdeményezése
ADATOK TÁROLÁSA	Biztonságos csatorna kezdeményezése

Általános kódolási szabályok
Kérjük, olvassa el az ACOSJ referencia kézikönyvet.

DELETE parancs
A DELETE parancs egy egyedileg azonosítható objektum, például egy végrehajtható betöltési fájl, egy alkalmazás, egy végrehajtható betöltési fájl és a hozzá tartozó alkalmazások vagy egy kulcs törlésére szolgál. Az objektum törléséhez az objektumnak a kiválasztott Alkalmazás által egyedileg azonosíthatónak kell lennie.

GET DATA parancs
A GET DATA parancsot vagy egyetlen adatobjektum, amely lehet konstruált, vagy adatobjektumok halmazának lekérdezésére használják. A P1 és P2 kódolású referencia vezérlőparaméterek a konkrét adatobjektum-címke meghatározására szolgálnak. Az adatobjektum tartalmazhat egy kulcsra vonatkozó információt.

GET STATUS parancs
A GET STATUS parancs a kibocsátó biztonsági tartomány, a végrehajtható betöltési fájl, a végrehajtható modul, az alkalmazás vagy a biztonsági tartomány életciklusára vonatkozó állapotinformációk lekérdezésére szolgál egy adott találati/keresési feltételnek megfelelően.

INSTALL parancs
Az INSTALL parancsot egy biztonsági tartománynak kell kiadni a kártyatartalom kezeléséhez szükséges különböző lépések elindításához vagy végrehajtásához.

LOAD parancs
Ez a szakasz a LOAD parancs adatmezőjében a Load File betöltéséhez továbbított Load File szerkezetét határozza meg. A betöltési fájl ICC belső kezelése vagy tárolása túlmutat e specifikáció alkalmazási körén.

Több LOAD parancsot is lehet használni egy betöltési fájl kártyára történő átviteléhez. A Load File az átvitelhez kisebb összetevőkre van felosztva. Minden LOAD parancsot 00h-tól kezdődően kell számozni. A LOAD parancsok számozása szigorúan folyamatos és eggyel növekszik. A kártyát tájékoztatni kell a betöltési fájl utolsó blokkjáról.

A betöltési fájl utolsó blokkjának fogadása után a kártya végrehajtja a betöltési fájlhoz szükséges belső folyamatokat és a LOAD parancsokat megelőző INSTALL [betöltéshez] parancsban azonosított további folyamatokat.

MANAGE CHANNEL parancs
A MANAGE CHANNEL parancsot a logikai csatornákat ismerő kártyáknál a megnyitottak dolgozzák fel. Kiegészítő logikai csatornák nyitására és zárására szolgál. Az alap logikai csatorna (nulladik csatorna) soha nem zárható be.

PUT KEY parancs

A PUT KEY parancs a következőkre használható:

- A meglévő kulcsot egy új kulcsra cserélheti: Az új kulcsnak ugyanaz vagy más kulcsverziószáma van, de ugyanaz a kulcsazonosító, mint a lecserélendő kulcsnak.
- Több meglévő kulcs cseréje új kulcsokra: Az új kulcsok azonos vagy eltérő kulcsverziószámmal rendelkeznek (minden új kulcs esetében azonos), de ugyanazokkal a kulcsazonosítókkal, mint a lecserélt kulcsok.
- Egyetlen új kulcs hozzáadása: Az új kulcs a meglévő kulcsoktól eltérő kulcsazonosító/kulcsverziószám kombinációval rendelkezik.
- Több új kulcs hozzáadása: Az új kulcsok a Kulcsazonosítók/Kulcsverziószám (minden új kulcsnál azonos) kombinációjától eltérő kombinációval rendelkeznek, mint a meglévő kulcsok.

Ha a kulcskezelési művelet több PUT KEY parancsot igényel, a művelet integritásának biztosítása érdekében ajánlott a több PUT KEY parancs láncolása.

A specifikáció ezen változatában az aszimmetrikus kulcsok nyilvános értékei tiszta szövegben kerülnek bemutatásra (továbbításra).

SELECT parancs

A SELECT parancs egy Alkalmazás kiválasztására szolgál. Az OPEN csak azokat a SELECT parancsokat dolgozza fel, amelyek jelzik, hogy a SELECT [név szerint] az aktuálisan kiválasztott biztonsági tartománynak vagy alkalmazásnak kell átadni a megadott logikai csatornán.

SET STATUS parancs

A SET STATUS parancsot a kártya életciklus-állapotának vagy az alkalmazás életciklus-állapotának módosítására kell használni.

STORE DATA parancs

A STORE DATA parancs a parancsot feldolgozó alkalmazás vagy biztonsági tartomány számára történő adatátvitelre szolgál.

A biztonsági tartomány meghatározza, hogy a parancsot saját magának vagy egy alkalmazásnak szánja-e egy korábban fogadott parancs függvényében. Ha az előző parancs INSTALL [személyre szabáshoz] parancs volt, akkor a STORE DATA parancsot egy alkalmazásnak szánja.

A többszörös STORE DATA parancs az adatoknak az alkalmazás vagy a biztonsági tartomány számára történő elküldésére szolgál, az adatokat kisebb összetevőkre bontva az átvitelhez. A biztonsági tartományt tájékoztatni kell az utolsó blokkról.

A személyre szabási munkamenet akkor kezdődik, amikor a biztonsági tartomány egy érvényes INSTALL [személyre szabáshoz] parancsot kap, amely kijelöl egy alkalmazást (amely vagy az alkalmazást vagy a személyre szabási interfészt valósítja meg), amelyhez

a biztonsági tartománynak a későbbiekben kapott STORE DATA parancsokat továbbítania kell.
A személyre szabási munkamenet akkor ér véget, amikor:

- A kártya visszaáll.
- A biztonsági tartomány nincs kijelölve (azaz ugyanazon a logikai csatornán egy másik vagy ugyanaz az applet van kijelölve).
- A biztonsági tartomány ugyanazon vagy egy másik logikai csatornán van kiválasztva.
- A biztonsági tartomány által létrehozott Secure Channel munkamenet (ha van ilyen) visszaáll, esetleg a célzott alkalmazás által.
- A biztonsági tartomány INSTALL [for personalize] parancsot kap (új személyre szabási munkamenet indítása egy másik alkalmazáshoz).
- A biztonsági tartomány STORE DATA parancsot kap a P1 jelzésére. b8=1 (utolsó blokk)

Az ilyen megszemélyesítési munkamenetből kapott STORE DATA parancsot maga a biztonsági tartomány dolgozza fel.

GlobalPlatform API

Globális platform egy JAVA kártyán

Ez a szakasz csak a Global Platform 2.2x Java-kártyákhoz szükséges API-t tartalmazza. Az Open Platform 2.0.1 API használata továbbra is megengedett az alkalmazások régebbi verzióinak támogatásához, de már elavult. Ezt a specifikáció 2.1.1 verziója határozza meg.
Az elavult API és az új API adott esetben ugyanazokhoz az objektumokhoz fér hozzá. Bár ez nyilvánvalónak tűnhet a két osztályban azonos nevű módszerek esetében (pl. setATRHistBytes(), setCardContentState() és getCardContentSame()), meg kell jegyezni, hogy például egy olyan alkalmazás, amely az új API updated() módszerét használja a globális PIN értékének megváltoztatására, ugyanarra a globális PIN-re lesz hatással, mint az az alkalmazás, amely az elavult API setPIN() módszerét használja a globális PIN ellenőrzésére.

GlobalPlatform specifikus követelmények

A GlobalPlatform implementációk legmagasabb szintű interoperabilitásának biztosítása érdekében a GlobalPlatform a 6.2. szakaszban - Java Card™ 2.2x Virtual Machine Specification - meghatározott sorrendet is elfogadja.
A Java Card™ 2.1.1 Runtime Environment (JCRE) specifikációjában és a Java Card™ 2.1.1 Application Programming Interface-ben meghatározott szabványos funkciókhoz képest a GlobalPlatform implementációjában a következő kisebb módosítások vannak jelen. A módosítások egy része már összhangban van a Java Card™ 2.2.x specifikációval, ezért már nem módosítások.
A GPRegistryEntry objektumokat a Java Card™ 2.2.x Runtime Environment (JCRE) specifikációjának 6.2.4. "Shareable Interfaces" (Megosztható interfészek) szakaszában meghatározottak szerint megosztott interfészobjektumként kell beültetni a megosztott hozzáférés biztosítása érdekében.

GlobalPlatform csomag AID

Minden GlobalPlatfrom csomag AID-ja a RID és egy PIX összekapcsolása lesz. Az új GlobalPlatform API (a GlobalPlatform 2.1 és a 2.1.1.1 esetében is azonos) Java-kártya exportfájljának AID-értéke a H. függelékben - GlobalPlatform adatértékek és kártyafelismerési adatok - meghatározott RID alapján "A00000015100".

Telepítés

A Java Card™ 2.2.x Runtime Environment (JCRE) Specifications 3.1 - The Method install című szakaszában a metódusnak átadott paraméterek a bejövő bájttömb paraméter tartalmából inicializálási paraméterként vannak definiálva.

Ez a specifikáció kibővíti ezt a követelményt, és tovább határozza meg a Telepítési paraméterek tartalmát. Ez a bővítés mind az OPEN megvalósítását, mind a GlobalPlatform kártyához kifejlesztett Java Card applet viselkedését érinti. Nem érinti a Java Card™ 2.2.x alkalmazásprogramozási interfész specifikációban szereplő Class Applet install () metódusának meghatározását.

A Telepítési paraméterek az INSTALL [telepítéshez] parancsban szereplő következő adatokat azonosítják:

- A példány AID
- A kiváltságok
- Az alkalmazásspecifikus paraméterek. (Bár az APDU parancs tartalmazza a TLV-kódolt rendszer- és alkalmazásspecifikus paramétereket képviselő Telepítési paramétereket, az alkalmazásnak csak az alkalmazásspecifikus paraméterek ismeretére volt szüksége, azaz csak a TLV-kódolt "C9" szerkezet LV-je van jelen alkalmazásspecifikus paraméterként.

Az OPEN felelős azért, hogy a paraméterek (bArray, bOffset és bLength) a következő információkat tartalmazzák:

A bArray tömb a következő, egymást követő LV kódolt adatokat tartalmazza:

- A példány AID hossza
- A példány AID
- A kiváltságok hossza
- A kiváltságok
- Az alkalmazásspecifikus paraméterek hossza
- Az alkalmazásspecifikus paraméterek

A bOffset bájt a tömbön belüli eltolást tartalmazza, amely a példány AID hosszára mutat.

A bLength bájtnak tartalmaznia kell a tömbben lévő, fent meghatározott adatok teljes hosszát jelző hosszúságot.

Az appletnek a Java Card 2.2.x alkalmazásprogramozási interfész specifikációban szereplő Class Applet osztály register (byte [] bArray, short bOffset, byte bLength) metódusának meghívásakor paraméterként kell használnia az AID példányt.

T=0 Átviteli protokoll

A GlobalPlatform kártyák célja, hogy a környezetek legszélesebb körében működjenek (pl. kártyaelfogadó eszközök). Jelenleg a Java Card 2.1.1 Runtime Environment (JCRE) Specifications és a Java Card 2.2.x Runtime Environment (JCRE) Specifications az EMV 2000-nek ellentmondva írja le a 2. esetre vonatkozó parancsok viselkedését (amikor a T=0 protokollt használják). A GlobalPlatform előírja, hogy a JCRE-nek az ISO/IEC 7816 szabványnak megfelelően kell kezelnie ezt a parancsesetet: A 2. esetű parancsot fogadó applet felépíti a választ, és meghívja a megfelelő API-t az adatok kiadásához. Ha az adat kevesebb, mint a terminál által elvárt adat, az OPEN tárolja az adatokat, és egy "6Cxx" válaszkódot ad ki, majd megvárja, hogy a CAD a megfelelő hosszúságú parancsot újra kiadja. Amikor az újraküldött parancs megérkezik, a JCRE kezeli a tárolt adatok kiadását.

Atomicitás

Eltérő rendelkezés hiányában a GlobalPlatform API minden belső tartós objektumának meg kell felelnie egy folyamatban lévő tranzakciónak.

Az API által végrehajtott összes műveletnek, kivéve az Application.processData() metódust, atomikusan kell végrehajtódnia. A sebességellenőrzés végrehajtásához használt objektumok nem felelhetnek meg a folyamatban lévő tranzakciónak.

Logikai csatornák

A következő logikai csatornakorlátozások a Java Card 2.2.x-re vonatkoznak (további részletekért lásd a Java Card 2.2.x futási környezet (JCRE) specifikációit):

Egy Alkalmazás kiválasztása egy logikai csatornán a 6.3. szakaszban - Parancskiosztás meghatározottak szerint sikertelen lesz, ha ugyanez az Alkalmazás, vagy bármely más, ugyanabban a csomagban található kódból instanciált Alkalmazás, amelyből a kiválasztott Alkalmazás instanciálásra került, jelenleg egy másik logikai csatornán van kiválasztva, de az alkalmazás kódja nem valósítja meg a MultiSelectable interfészt. A biztonsági tartományoknak meg kell valósítaniuk a MultiSelectable interfészt.

A 7.3.3 Személyre szabás támogatása szakaszban meghatározott biztonsági tartományról egy Alkalmazásra történő kontextusváltás sikertelen lesz, ha ugyanez az Alkalmazás vagy bármely más, ugyanannak a csomagnak a kódjából, amelyből a személyre szabott Alkalmazás származik, létrehozott Alkalmazás jelenleg egy másik logikai csatornán van kiválasztva, de az alkalmazás kódja nem valósítja meg a MultiSelectable interfészt.

Egy olyan alkalmazásnak, amely alapértelmezés szerint kiváltható, és olyan kártyához készült, amely támogatja a kiegészítő logikai csatornákat, a MultiSelectable interfészt kell megvalósítania.

A GlobalPlarform csak a logikai csatornaszámok kártya általi hozzárendelését határozza meg. Opcionálisan és a Java Card 2.2-ben meghatározottak szerint a kártya támogathatja a logikai csatornaszámok terminál általi hozzárendelését is.

Kriptográfiai algoritmusok

Az RSA kriptográfiát támogató GlobalPlatform kártyáknak támogatniuk kell a Key Builder osztályban konstansként nem definiált kulcsméreteket. Konkrétabban, a kulcsméreteknek 4 bájt (32 bit) többszörösét kell támogatniuk, és a megvalósítás által meghatározott megengedett kulcshosszon belül kell lenniük.

A bizalom szintje

A Java Card 2.2.x specifikációk feltételezik, hogy az AID csomagok, appletek és példányok RID-jét használják az ezen entitások közötti bizalmi szint biztosítására. A Java Card 2.2.1 alkalmazásprogramozási interfész 4.2.2.2-AID-használat szakaszában az szerepel, hogy egy komponens AID-jének RID-jének meg kell egyeznie a csomag AID-jének RID-jével, valamint a Java Card 2.2.x szabvány register (byte [] bArray, short bOffset, byte bLength) metódusának definíciójában.2.x alkalmazásprogramozási interfész specifikációjában az szerepel, hogy kivételt kell dobni, ha a bArray paraméterben szereplő AID bájtok RID része nem egyezik meg az alkalmazás Java Card nevének RID részével.

A valós megvalósítás szempontjából nem célszerű előírni, hogy a példány AID RID-jének meg kell egyeznie annak a komponensnek a RID-jével, amelyből a példányt létrehozták. A GlobalPlatform implementációk nem írhatják elő, hogy a példány AID-jén keresztül bármilyen kapcsolat legyen az eredeti csomaggal. (A példány AID-je és az eredeti csomag AID-je közötti kapcsolat hiánya). Feltételezi azonban, hogy az ugyanabban a csomagban lévő összes alkalmazás azonos szintű bizalmat élvez.

A GlobalPlatform metódusok meghívása

Az itt meghatározott alkalmazásprogramozási interfész elérhető minden olyan Java Card applet számára, amelyet azzal a szándékkal fejlesztettek ki, hogy egy GlobalPlatform kártyán jelen legyen. Egy korlátozás nem áll fenn, amely a Java Card 2.2.x alkalmazásprogramozási interfész Applet osztályának konstruktorára és install () módszerére vonatkozik. Mivel ez a specifikáció nem határozza meg pontosan, hogy egy applet példánya mikor válik bejegyzéssé a kártya GlobalPlatform Registry-jében, az appletfejlesztő csak feltételezheti, hogy ez az install () metódus sikeres befejezését követően történt meg. Az interoperabilitás biztosítása érdekében a GlobalPlatform API azon metódusait, amelyek hozzáférnek a metódust meghívó applet GlobalPlatform Registry bejegyzéséhez, nem szabad a konstruktorból vagy az install() metódusból meghívni.
Az alábbiakban felsoroljuk azokat a metódusokat, amelyek a konstruktorból vagy az install() metódusból hívhatók meg:

- getCardState;
- getCVM;
- getService;

A kártya szükséges viselkedése abban az esetben, ha egy alkalmazás helytelenül hívja meg az org.globalplatform.GPSystem osztály egy, a fent felsoroltaktól eltérő metódusát, nem definiált. Például egy kivétel dobható, és a telepítés() metódus feldolgozása megszakadhat.

Kiválasztás

A GlobalPlatform kártyákon, ha a select() módszer feldolgozása során hiba lép fel, vagy ha a select() módszer False-t ad vissza, vagy ha az alkalmazás nem választható ki, mert nem valósítja meg a Multiselectable interfészt, a NYÍLT program folytatja a keresést a GlobalPlatform nyilvántartásban a 6.4.2.1.2. szakaszban meghatározottak szerint egy későbbi teljes vagy részleges egyezés után. Ha nincs kiválasztott alkalmazás, a megfelelő logikai csatorna nyitva marad, és nincs aktuálisan kiválasztott alkalmazás. Mivel jelenleg nincs kiválasztott alkalmazás, a MANAGE CHANNEL vagy SELECT parancson kívül minden további parancs elutasításra kerül. Elvárható, hogy a kártyán kívüli egység ilyen hiba esetén megfelelő lépéseket tegyen, pl. válasszon másik alkalmazást, zárja be a megfelelő logikai csatornát, állítsa vissza vagy kapcsolja ki a kártyát.

A GlobalPlatform Java Card API specifikációjának módszer-összefoglalója és részletei mostantól külön dokumentumban érhető el, amely a GlobalPlatform weboldalán található.

Előszemélyesítés

Az IC első bekapcsolásakor a személyre szabás előtti állapotban van.

Az INIT_CARD parancsnak (lásd 8.2.2 INIT CARD parancs) kell lennie az első parancsnak a megszemélyesítés előtti állapotban, a parancs inicializálja a kártya memóriaterületét alapértelmezett értékekkel, majd egy úgynevezett ACOSJ ROOT alkalmazás (lásd 8.0 ACOSJ ROOT alkalmazás) áll rendelkezésre a kritikus rendszeradatok (például "ATR", "ATS" stb.) EEPROM-ba írásához. Az adatok betöltése és a további változtatások elmaradásának megerősítése után egy olyan parancsot kell végrehajtani, amely letiltja az ACOSJ ROOT alkalmazást.

Amikor az IC az Aktív kártya parancs sikeres végrehajtásával kilép a megszemélyesítés előtti állapotból (lásd 8-2.5 ACTIVATE parancs), az alapértelmezett GlobalPlatform kártya életciklus OP_READY állapotába kerül, ha nincs más életciklus állapotba állítva.

Ha az aktív kártyaparancs sikeres feldolgozása után nem lehet visszatérni a megszemélyesítés előtti állapotba.

Megjegyzés: A szállítási kulcs az ACOSJ gyökéralkalmazás megszemélyesítés előtti kulcsa, és a szállítási kulcsot az ACS-től kell kérni.

ACOSJ ROOT alkalmazás

ACOSJ ROOT alkalmazás leírása

A ROOT alkalmazásban lévő parancsok csak akkor használhatók, ha a ROOT alkalmazás megfelelően ki van választva (a Transport Key kiválasztási parancs segítségével). A ROOT alkalmazásba való belépést követően a felhasználó a ROOT alkalmazás által támogatott parancsok segítségével olvashatja vagy konfigurálhatja a kártya paramétereit. A ROOT alkalmazásból való kilépéshez és egy másik alkalmazás kiválasztásához a kártyát vissza kell állítani.

A ROOT alkalmazás alatt a kártya paraméterei olvashatók vagy konfigurálhatók.

A ROOT alkalmazás érvénytelenné válik, amint a kártya aktiválásra kerül.

ACOSJ ROOT alkalmazásparancs-referencia

SELECT parancs
A SELECT parancs a ROOT alkalmazás kiválasztására szolgál.

INIT_CARD parancs
Ez a parancs a kártya alaphelyzetbe állítására szolgál. Ez a parancs inicializálja az összes EEPROM-ot.

READ Command
Ez a parancs a konfigurációs területről való olvasásra szolgál. A kártya konfigurációs paraméterei ezzel a paranccsal olvashatók.

WRITE parancs
Ez a parancs adatok írására szolgál a konfigurációs területre. A kártya konfigurációs paraméterei ezzel a paranccsal olvashatók.

ACTIVE parancs
Ez a parancs a kártya aktiválására szolgál. A parancs sikeres végrehajtása után a ROOT alkalmazás érvénytelenné válik, és a kártya konfigurációs adatai már nem olvashatók vagy állíthatók be közvetlenül.

ACOSJ IDENTIFY alkalmazás

ACOSJ IDENTIFY Alkalmazás leírása
Az IDENTIFY alkalmazás (AID: 6163732E636F732E61636F736A766572) SELECT paranccsal történő kiválasztása után az ACOSJ visszaadja az ACOSJ verziószámát, és jelzi, hogy a kocsi aktiválva van-e már.

ACOSJ IDENTIFY alkalmazásparancs-hivatkozás

SELECT parancs
A SELECT parancs az INDENTIFY alkalmazás kiválasztására szolgál.

Állapot bájtok

SW1 SW2	Leírás
90 00h	Helyes parancs
61 XX	SW2 több válaszbájt áll rendelkezésre
62 83h	Alkalmazás véglegesen blokkolva
63 00h	Az ellenőrzés sikertelen
63 CX	X megismétlési kísérlet maradt
67 00h	Az adatok helytelen hossza
68 81h	Logikai csatorna nem támogatott
68 82h	Biztonságos üzenetküldés nem támogatott
69 84h	Érvénytelen adatok
69 82h	A biztonsági státusz nem teljesül
69 85h	A felhasználási feltételek nem teljesülnek
69 88h	Rossz MAC
6A 86h	Helytelen P1 P2
6A 80h	Hibás értékek a parancsadatokban
6A 82h	Fájl nem található
6A 81h	Alkalmazás zárolva
6A 83h	Nem talált rekord
6A 84h	Nincs elég memóriahely
6A 88h	A hivatkozott adat nem található
6D 00h	Érvénytelen INS, nem létező parancs
6E 00h	Érvénytelen osztály

Hivatkozások

Az alábbi dokumentumok az ACOSJ funkcionális specifikációjának referenciájaként szolgáltak:

- GlobalPlatform kártya specifikáció 2.2.1. verziója
- GlobalPlatform kártya API 1.6 verzió
- Java Card 3 API, klasszikus kiadás 3.0.4 verziója
- Java Card 3 Platform Runtime Environment Specification, Classic Edition 3.0.4 verzió 2011. szeptember 2011. szeptember
- Java Card 3 Platform Virtual Machine Specification, Classic Edition 3.0.4 verzió 2011. szeptember 2011.
- A GlobalPlatform kártya leképezési irányelvei a meglévő GP v2.1.1.1 implementáció v2.2.1 1.0.1 verzióra történő átültetéséhez

Jótállási feltételek

Az Alza.cz értékesítési hálózatában vásárolt új termékre 2 év garancia vonatkozik. Ha a garanciális időszak alatt javításra vagy egyéb szolgáltatásra van szüksége, forduljon közvetlenül a termék eladójához, a vásárlás dátumával ellátott eredeti vásárlási bizonylatot kell bemutatnia.

Az alábbiak a jótállási feltételekkel való ellentétnek minősülnek, amelyek miatt az igényelt követelés nem ismerhető el:

- A terméknek a termék rendeltetésétől eltérő célra történő használata, vagy a termék karbantartására, üzemeltetésére és szervizelésére vonatkozó utasítások be nem tartása.
- A termék természeti katasztrófa, illetéktelen személy beavatkozása vagy a vevő hibájából bekövetkezett mechanikai sérülés (pl. szállítás, nem megfelelő eszközökkel történő tisztítás stb. során).
- A fogyóeszközök vagy alkatrészek természetes elhasználódása és öregedése a használat során (pl. akkumulátorok stb.).
- Káros külső hatásoknak való kitettség, például napfény és egyéb sugárzás vagy elektromágneses mezők, folyadék behatolása, tárgyak behatolása, hálózati túlfeszültség, elektrosztatikus kisülési feszültség (beleértve a villámlást), hibás táp- vagy bemeneti feszültség és e feszültség nem megfelelő polaritása, kémiai folyamatok, például használt tápegységek stb.
- Ha valaki a termék funkcióinak megváltoztatása vagy bővítése érdekében a megvásárolt konstrukcióhoz képest módosításokat, átalakításokat, változtatásokat végzett a konstrukción vagy adaptációt végzett, vagy nem eredeti alkatrészeket használt.

Sehr geehrter Kunde,

vielen Dank für den Kauf unseres Produkts. Bitte lesen Sie die folgenden Anweisungen vor dem ersten Gebrauch sorgfältig durch und bewahren Sie diese Bedienungsanleitung zum späteren Nachschlagen auf. Beachten Sie insbesondere die Sicherheitshinweise. Wenn Sie Fragen oder Kommentare zum Gerät haben, wenden Sie sich bitte an den Kundenservice.

✉ www.alza.de/kontakt

✆ 0800 181 45 44

✉ www.alza.at/kontakt

✆ +43 720 815 999

Lieferant Alza.cz a.s., Jankovcova 1522/53, Holešovice, 170 00 Prag 7, www.alza.cz

Übersicht

ACOSJ ist ein von Advanced Card Systems Ltd. entwickeltes Chipkartenbetriebssystem. Es basiert auf der JAVA Card Virtual Machine und entspricht in seinen Funktionen und Konfigurationen der GlobalPlatform Card Specification Version 2.2.1, der JAVA Card Virtual Machine und der GlobalPlatform Card Specification Version 2.2.1, der JAVA Card Specification Version 3.0.4 und Mapping Guidelines 1.0.1.

Der Zweck dieses Dokuments ist es, die Eigenschaften und Funktionen des ACOSJ Chipkarten-Betriebssystems im Detail zu beschreiben.

ACOSJ-G-Versionen

Version	Freigegebene Daten	Änderungen
ACOSJ v1.01	Juni 2015	• 40KB EEPROM • Betriebsspannung: 2,7V bis 5,5V
ACOSJ v2.00	Mai 2018	• 95KB EEPROM • Betriebsspannung: 2,1V bis 5,5V • Änderungen der Konfigurationsspeicheradresse
ACOSJ v2.04	April	• 95KB EEPROM • Betriebsspannung: 2,1V bis 5,5V • Änderungen der Konfigurationsspeicheradresse • Erweiterung der Funktionen

Symbole und Abkürzungen

Abkürzung	Beschreibung
AES	Erweiterter Verschlüsselungsstandard
AID	Kennung der Anwendung
APDU	Anwendungsprotokoll-Dateneinheit
API	Schnittstelle zur Anwendungsprogrammierung
ASCII	American Standard Code for Information Interchange
ATR	Antwort-zu-Rückruf
ATQ	Answer-to-Request (für kontaktlose Karten)
BCD	Binär codiert Dezimal
BER	Grundlegende Kodierungsregeln
CAT	Kartenanwendungs-Toolkit: oder Kryptographische Autorisierungsvorlage
CBC	Cipher Block Chaining
CCT	Kontrollreferenzvorlage für kryptografische Prüfsumme
CIN	Kartenbildnummer/Kartenidentifikationsnummer
CLA	Klassenbyte der Befehlsnachricht
CRT	Kontroll-Referenzvorlage
CT	Kontrollreferenzvorlage für Vertraulichkeit
CVM	Methode zur Überprüfung des Karteninhabers

DAP	Muster für die Datenauthentifizierung
DEK	Schlüssel zur Datenverschlüsselung
DER	Unterschiedene Kodierungsregeln
DES	Datenverschlüsselungsstandard
DST	Kontrollreferenzvorlage für die digitale Signatur
EZB	Elektronisches Codebuch
EMV	Europay, Mastercard und VISA: bezieht sich auf die ICC-Spezifikationen für Zahlungssysteme
ENC	Verschlüsselung
FCI	Informationen zur Dateisteuerung
HEX	Hexadezimal
HMAC	Keyed-Hash Message Authentication Code
ICC	Integrierte Schaltkreiskarte
ICV	Ursprünglicher Verkettungsvektor
IIN	Emittenten-Identifikationsnummer
INS	Befehlsbyte der Befehlsnachricht
ISO	Internationale Organisation für Normung
Lc	Genaue Länge der Daten in einem Case-3- oder Case-4-Befehl
Le	Maximale Länge der Daten, die als Antwort auf einen Case-2- oder Case-4-Befehl erwartet werden
LV	Länge Wert
MAC	Authentifizierungscode der Nachricht
MEL	MULTOS Ausführbare Sprache. Der Befehlssatz der MULTOS-Laufzeitumgebung™
OID	Objekt-Identifikator
P1	Referenzkontrollparameter 1
P2	Referenzsteuerungsparameter 2
PIN	Persönliche Identifikationsnummer
PKI	Infrastruktur für öffentliche Schlüssel
RAM	Speicher mit wahlfreiem Zugriff
RFU	Reserviert für zukünftige Verwendung
RID	Kennung des registrierten Anwendungsanbieters
ROM	Nur-Lese-Speicher
RSA	Nur-Rivest-Speicher
SCP	Secure Channel Protocol: oder (ETSI) Smart Card Plattform
SW	Statuswort
SW1	Statuswort Eins
SW2	Statuswort Zwei
TLV	Tag Länge Wert
TP	Vertrauenspunkt
"xx"	Hexadezimale Werte werden als hexadezimale Ziffern zwischen einfachen Anführungszeichen ausgedrückt.
"X"	Ein Wert in einer Zelle einer Tabelle, dessen Zweck in der Spalte "Bedeutung" der Tabelle beschrieben ist.
"-"	Ein Wert (0 oder 1) in einer Zelle einer Tabelle, der sich nicht auf die für diese Zeile der Tabelle angegebene "Bedeutung" auswirkt.

Spezifikationen der Karte

Dieser Abschnitt fasst die Merkmale und Funktionalitäten des ACOSJ-G zusammen.

Elektrische Spezifikationen

Betriebsspannung (ACOSJ-G v1.01):	2,7 V bis 5,5 V Versorgungsspannungen
Betriebsspannung (ACOSJ-G v2.04):	2,1 V bis 5,5 V Versorgungsspannungen
Maximale externe Taktfrequenz:	10 MHz
Maximale CPU-Taktfrequenz:	28 MHz
ESD-Schutz	größer als 5 kV (HBM)

Spezifikationen

Betriebstemperatur:	-25°C bis +85°C

Kommunikationsprotokolle

- T=CL-Protokoll mit einer Baudrate von bis zu 848 kbps

Speicher

Kapazität:	95 KB (ACOSJ-G 2.04), 40 KB (ACOSJ-G 1.01)
EEPROM-Ausdauer:	500.000 Schreib-/Löschzyklen (25°C)
Aufbewahrung von Daten:	30 Jahre (25°C)

Kryptographische Funktionalitäten

DES:	2K3DES,3K3DES (ECB und CBC)
AES:	128/192/256 Bits (ECB und CBC)
RSA:	768 bis 2048 Bits
ECC:	Modulus 112/128/160/192/224/256/384 Bits
Hash:	SHA1, SHA256, SHA384, SHA512
SEED:	128 Bits

- SM2/SM3/SM4

Einhaltung von Normen

- Übereinstimmung mit ISO 7816 Teil 4
- Übereinstimmung mit ISO 14443 (Typ A und B)
- Übereinstimmung mit der JAVA-Kartenspezifikation Version 3.0.4
- Übereinstimmung mit der globalen Plattformspezifikation Version 2.2.1
- Einhaltung der Kartierungsrichtlinien 1.0.1

Antwort auf Auswahl (ATS, kontaktlose Karte)

Nach Erhalt eines RATS-Befehls (Request for Answer to Select) vom Kartenlesegerät sendet die Karte eine ATS (Answer to Select) gemäß ISO 14443 Teil 4.

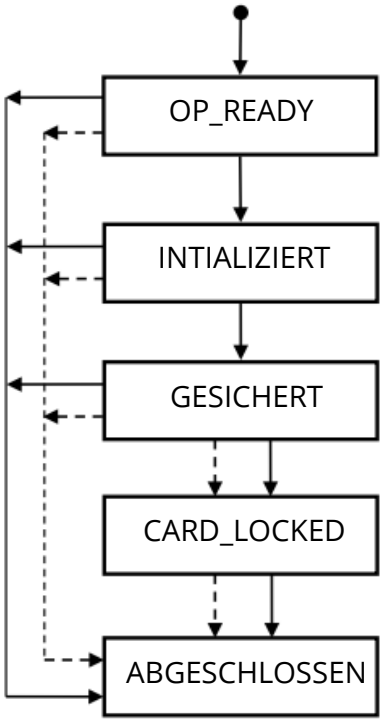
Die folgende Tabelle zeigt die Standard-ATS:

Parameter	ATS	Beschreibung
TL	0Eh	Länge
T0	78h	Formale Byte...codes Y(1) und FSCI
TA1	00h	Schnittstellenbyte...Codes DS und DR
TB1	71h	Codes FWI und SFGI
TC1	02h	Codes Protokolloptionen
T1	41h	Zeigt "A" an
T2	43h	Zeigt "C" an
T3	4Fh	Zeigt "O" an
T4	53h	Zeigt "S" an
T5	4Ah	Zeigt "J" an
T6	76h	Zeigt "v" an
T7~T9	31h 30h 31h oder 32h 30h 34h	Zeigt "101" an oder zeigt "204" an

Anmerkung: Eine vollständige Beschreibung des ATS findet sich in ISO 14443 Teil 4.

Zustände im Lebenszyklus der Karte

Der ACOSJ hat fünf Kartenzustände: OP_READY, INTILIZED, SECURED, CARD_LOCKED und TEDMINATED. Die folgende Abbildung zeigt die Zustandsübergänge im Lebenszyklus der Karte:



Legende

Privilegierte Sicherheitsdomäne

Privilegierte Anwendung

OP_READY

Dieser Status zeigt an, dass die Laufzeitumgebung verfügbar ist und die Issuer Security Domain als ausgewählte Anwendung bereit ist, APDU-Befehle zu empfangen, auszuführen und zu beantworten.

Die folgenden Funktionen müssen vorhanden sein, wenn sich die Karte im Zustand OP_READY befindet:

- Die Laufzeitumgebung muss zur Ausführung bereit sein.
- Das OPEN ist bereit für die Ausführung.
- Die Sicherheitsdomäne des Ausstellers ist die implizit ausgewählte Anwendung für alle Kartenschnittstellen.
- Ausführbare Ladedateien, die in den unveränderlichen persistenten Speicher aufgenommen wurden, müssen im GlobalPlatform-Register registriert werden.
- Innerhalb der Sicherheitsdomäne des Emittenten muss ein Anfangsschlüssel verfügbar sein.

Die Karte muss in der Lage sein, den Karteninhalt zu ändern, d. h., es können Ladedateien geladen werden, die Anwendungen enthalten, die noch nicht auf der Karte vorhanden sind.

Die Installation, von ausführbaren Ladedateien, einer anderen Anwendung kann auftreten.

Wenn in dieser Phase Personalisierungsinformationen verfügbar sind, können die Anwendungen zusätzlich personalisiert werden.

Der Zustand OP_READY kann von einer Off-Card für die folgenden Aktionen verwendet werden:

- Ergänzende Sicherheitsdomänen können geladen und/oder installiert werden.
- Die Schlüssel des Sicherheitsbereichs können eingefügt werden, um eine kryptografische Trennung von den Schlüsseln des Sicherheitsbereichs des Ausstellers zu gewährleisten.

Initialisiert

Dieser Zustand ist ein administrativer Kartenproduktionszustand. Der Zustandsübergang von OP_READY zu INITIALIZED ist unumkehrbar. Seine Funktionalität liegt außerhalb des Anwendungsbereichs dieser Spezifikation. Dieser Status kann verwendet werden, um anzuzeigen, dass einige anfängliche Daten eingegeben wurden (z. B. Schlüssel und/oder Daten der Issuer Security Domain), die Karte aber noch nicht bereit ist, an den Karteninhaber ausgegeben zu werden.

Gesichert

Dieser Zustand ist der vorgesehene Lebenszykluszustand der Betriebskarte nach der Einführung (Post-Issuance). Dieser Zustand kann von Sicherheitsdomänen und Anwendungen zur Durchsetzung ihrer jeweiligen Sicherheitsrichtlinien verwendet werden. Der Zustandsübergang von INITIALIZED zu SECURED ist unumkehrbar.

Der Status SECURED sollte verwendet werden, um kartenfremden Stellen anzuzeigen, dass der Sicherheitsbereich des Ausstellers alle erforderlichen Schlüssel und Sicherheitselemente für die volle Funktionalität enthält.

Karte_gesperrt

Der Kartenlebenszyklus-Status CARD_LOCKED ist vorhanden, um die Auswahl des Sicherheitsbereichs und der Anwendungen zu deaktivieren. Der Übergang des Kartenlebenszyklus-Zustands von SECURED zu CARD_LOCKED ist reversibel.

Wird die Karte in den Zustand CARD_LOCKED versetzt, bedeutet dies, dass die Karte nur die Auswahl der Anwendung mit dem Recht "Endgültige Anwendung" zulässt.

Änderungen am Karteninhalt, einschließlich jeglicher Art von Datenverwaltung (insbesondere Sicherheitsdomänenschlüssel und -daten), sind in diesem Zustand nicht zulässig.

Der Übergang vom Zustand SECURED in den Zustand CARD_LOCKED kann entweder vom OPEN oder von einer Sicherheitsdomäne mit dem Privileg Card Lock oder von einer Anwendung mit dem Privileg Card Lock eingeleitet werden.

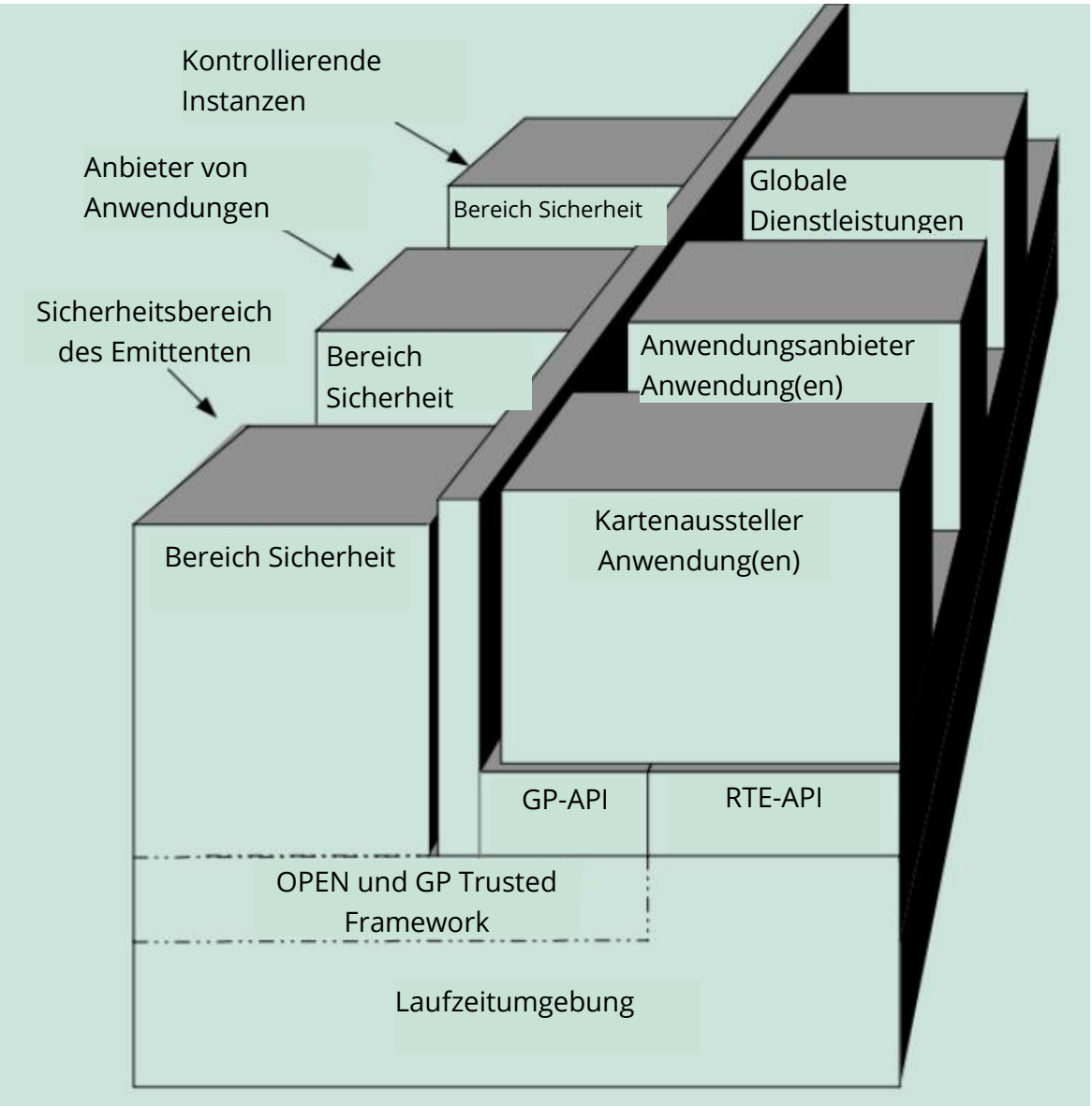
ABGESCHLOSSEN

Dieser Zustand signalisiert das Ende des Kartenlebenszyklus und der Karte. Der Zustandsübergang von jedem anderen Zustand zu TERMINATED ist irreversibel.

Der Status TERMINATED wird verwendet, um die gesamte Kartenfunktionalität in Bezug auf die Verwaltung des Karteninhalts und alle Änderungen im Lebenszyklus dauerhaft zu deaktivieren. Dieser Kartenstatus ist als Mechanismus gedacht, mit dem eine Anwendung die Karte logisch "zerstören" kann, z. B. wenn eine schwerwiegende Sicherheitsbedrohung festgestellt wurde oder die Karte abgelaufen ist. Verfügt eine Sicherheitsdomäne über das Recht "Endgültige Anwendung", so wird nur der Befehl GET DATA verarbeitet; alle anderen in dieser Spezifikation definierten Befehle sind deaktiviert und geben einen Fehler zurück. Verfügt eine Anwendung über das Recht der endgültigen Anwendung, so unterliegt ihre Befehlsverarbeitung den Richtlinien des Ausstellers.

Das OPEN selbst oder eine Sicherheitsdomäne mit dem Privileg "Karte beenden" oder eine Anwendung mit dem Privileg "Karte beenden" kann den Übergang von einem der vorherigen Zustände zum Zustand TERMINATED einleiten.

Architektur der Karten



GP APDU-Befehlsreferenz

In diesem Abschnitt werden die GlobalPlatform APDU-Befehle beschrieben, die implementiert werden können. Die Befehle sind alphabetisch geordnet.

Fasst die APDU-Befehle zusammen, die von den ausstellenden Sicherheitsdomänen unterstützt werden, sowie die Anforderungen für die Unterstützung dieser APDU-Befehle durch andere Sicherheitsdomänen. Wenn logische Kanäle unterstützt werden, wird der Befehl MANAGE CHANNEL nur vom OPEN verarbeitet, und für diesen Befehl ist keine Unterstützung durch eine Sicherheitsdomäne erforderlich.

Befehl	OP_READY			INITIALISIERT			GESICHERT			CARD_LOCKED		ABGESCHLOSSEN	
	AMSD	DMSD	SD	AMSD	DMSD	SD	AMSD	DMSD	SD	FASD	SD	FASD	SD
DELETE Ausführbare Ladedatei													
DELETE Ausführbare Ladedatei und zugehörige Anwendung(en)													
LÖSCHEN Anwendung	✓			✓			✓						
DELETE-Taste													
DATEN ABRUFEN	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	
STATUS ERHALTEN	✓			✓			✓			✓			
INSTALL [für Last]													
INSTALL [zur Installation]													
INSTALL [für Laden, Installieren und wählbar machen]													
INSTALL [für installieren und wählbar machen]	✓	✓		✓	✓		✓	✓					
INSTALL [für wählbar machen]													
INSTALL [zur Auslieferung]													
INSTALL [für die Aktualisierung der Registrierung]													
INSTALL [zur Personalisierung]													
LOAD													
PUT SCHLÜSSEL	✓			✓			✓						
SELECT	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓			
STATUS SETZEN	✓			✓			✓			✓			
DATEN SPEICHERN	✓			✓			✓						

AM SD	Sicherheitsdomäne mit der Berechtigung "Autorisierte Verwaltung".
DM SD	Sicherheitsdomäne mit der Berechtigung "Delegierte Verwaltung".
FA SD	Sicherheitsbereich mit dem Recht der endgültigen Anwendung.

Hinweis: Die Befehlsunterstützung für eine Anwendung mit endgültigen Anwendungsrechten, die keine Sicherheitsdomäne ist, unterliegt den Richtlinien des Ausstellers innerhalb der Grenzen dessen, was gemäß dem Kartenlebenszyklusstatus zulässig ist.

SD	Anderer Sicherheitsbereich.
✓	Unterstützung erforderlich.
Leere Zelle	Unterstützung optional.
Graue Zelle	Unterstützung verboten.

Fasst die Mindestsicherheitsanforderungen für die APDU-Befehle zusammen.

Befehl	Minimale Sicherheit
DELETE	Initiierung eines sicheren Kanals oder Überprüfung der digitalen Signatur
DATEN ABRUFEN	Keine
STATUS ERHALTEN	Initiierung eines sicheren Kanals
INSTALLIEREN	Initiierung eines sicheren Kanals oder Überprüfung der digitalen Signatur
LOAD	Initiierung eines sicheren Kanals oder Überprüfung der digitalen Signatur
KANAL VERWALTEN	Nicht anwendbar
PUT SCHLÜSSEL	Initiierung eines sicheren Kanals
SELECT	Nicht anwendbar
STATUS SETZEN	Initiierung eines sicheren Kanals
DATEN SPEICHERN	Initiierung eines sicheren Kanals

Allgemeine Kodierungsregeln

Bitte beachten Sie das ACOSJ Referenzhandbuch.

DELETE-Befehl

Der Befehl DELETE wird verwendet, um ein eindeutig identifizierbares Objekt wie eine ausführbare Ladedatei, eine Anwendung, eine ausführbare Ladedatei und die damit verbundenen Anwendungen oder einen Schlüssel zu löschen. Um ein Objekt zu löschen, muss das Objekt durch die ausgewählte Anwendung eindeutig identifizierbar sein.

GET DATA-Befehl

Der GET DATA-Befehl wird verwendet, um entweder ein einzelnes Datenobjekt, das konstruiert sein kann, oder einen Satz von Datenobjekten abzurufen. Die Referenzkontrollparameter P1 und P2 werden verwendet, um das spezifische Datenobjekt-Tag zu definieren. Das Datenobjekt kann Informationen enthalten, die sich auf einen Schlüssel beziehen.

GET STATUS Befehl

Der Befehl GET STATUS dient zum Abrufen von Informationen über den Status der Sicherheitsdomäne des Ausstellers, der ausführbaren Ladedatei, des ausführbaren Moduls, der Anwendung oder des Lebenszyklus der Sicherheitsdomäne entsprechend den angegebenen Suchkriterien.

INSTALL-Befehl

Der INSTALL-Befehl wird an eine Sicherheitsdomäne ausgegeben, um die verschiedenen für die Verwaltung von Karteninhalten erforderlichen Schritte einzuleiten oder durchzuführen.

LOAD-Befehl

Dieser Abschnitt definiert den Aufbau der im Datenfeld des LOAD-Kommandos zum Laden einer Lastdatei übertragenen Lastdatei. Die ICC-interne Handhabung oder Speicherung der Ladedatei liegt außerhalb des Rahmens dieser Spezifikation.

Mehrere LOAD-Befehle können verwendet werden, um eine Lastdatei auf die Karte zu übertragen. Die Lastdatei wird für die Übertragung in kleinere Komponenten aufgeteilt. Jeder LOAD-Befehl ist mit 00h beginnend zu nummerieren. Die Nummerierung der LOAD-Befehle ist streng fortlaufend und wird um eins erhöht. Die Karte wird über den letzten Block der Ladedatei informiert.

Nach dem Empfang des letzten Blocks der Ladedatei führt die Karte die für die Ladedatei erforderlichen internen Prozesse und alle zusätzlichen Prozesse aus, die im Befehl INSTALL [for load], der den LOAD-Befehlen vorausging, angegeben wurden.

MANAGE CHANNEL Befehl

Der MANAGE CHANNEL-Befehl wird von der Open on-Karte verarbeitet, die logische Kanäle kennt. Er wird zum Öffnen und Schließen von zusätzlichen logischen Kanälen verwendet. Der grundlegende logische Kanal (Kanalnummer Null) kann niemals geschlossen werden.

PUT KEY Befehl

Der Befehl PUT KEY wird verwendet, um entweder:

- Ersetzen Sie einen vorhandenen Schlüssel durch einen neuen Schlüssel: Der neue Schlüssel hat die gleiche oder eine andere Schlüsselversionsnummer, aber den gleichen Schlüsselidentifikator wie der zu ersetzende Schlüssel.
- Ersetzen Sie mehrere vorhandene Schlüssel durch neue Schlüssel: Die neuen Schlüssel haben die gleiche oder eine andere Schlüsselversionsnummer (identisch für alle neuen Schlüssel), aber die gleichen Schlüsselkennungen wie die zu ersetzenden Schlüssel.
- Fügen Sie einen einzelnen neuen Schlüssel hinzu: Der neue Schlüssel hat eine andere Kombination aus Schlüsselidentifikator und Schlüsselversionsnummer als die vorhandenen Schlüssel.
- Fügen Sie mehrere neue Schlüssel hinzu: Die neuen Schlüssel haben andere Kombinationen von Schlüsselbezeichnern/Schlüsselversionsnummern (identisch für alle neuen Schlüssel) als die vorhandenen Schlüssel.

Wenn für die Schlüsselverwaltung mehrere PUT KEY-Befehle erforderlich sind, wird die Verkettung mehrerer PUT KEY-Befehle empfohlen, um die Integrität des Vorgangs zu gewährleisten.

In dieser Version der Spezifikation werden die öffentlichen Werte der asymmetrischen Schlüssel im Klartext dargestellt (übertragen).

SELECT-Befehl

Der SELECT-Befehl wird für die Auswahl einer Anwendung verwendet. Das OPEN verarbeitet nur SELECT-Befehle, die angeben, dass SELECT [nach Name] an die aktuell ausgewählte Sicherheitsdomäne oder Anwendung auf dem angegebenen logischen Kanal weitergeleitet werden soll.

SET STATUS Befehl

Der Befehl SET STATUS wird verwendet, um den Lebenszyklusstatus der Karte oder den Lebenszyklusstatus der Anwendung zu ändern.

STORE DATA-Befehl

Der Befehl STORE DATA wird verwendet, um Daten an eine Anwendung oder die Sicherheitsdomäne, die den Befehl verarbeitet, zu übertragen.

Die Sicherheitsdomäne bestimmt anhand eines zuvor empfangenen Befehls, ob der Befehl für sie selbst oder für eine Anwendung bestimmt ist. Wenn ein vorhergehender Befehl ein INSTALL-Befehl [zur Personalisierung] war, ist der Befehl STORE DATA für eine Anwendung bestimmt.

Mehrere STORE DATA-Befehle werden verwendet, um Daten an die Anwendung oder den Sicherheitsbereich zu senden, indem die Daten für die Übertragung in kleinere Komponenten aufgeteilt werden. Der Sicherheitsbereich muss über den letzten Block informiert werden.

Eine Personalisierungssitzung beginnt, wenn eine Sicherheitsdomäne einen gültigen INSTALL [for personalize]-Befehl empfängt, der eine Anwendung bestimmt (die entweder die Anwendungs- oder die Personalisierungsschnittstelle implementiert), an die die Sicherheitsdomäne nachfolgend empfangene STORE DATA-Befehle weiterleiten soll.

Eine Personalisierungssitzung endet, wenn:

- Die Karte wird zurückgesetzt.
- Der Sicherheitsbereich ist abgewählt (d. h. ein anderes oder dasselbe Applet ist auf demselben logischen Kanal ausgewählt)
- Die Sicherheitsdomäne wird auf demselben oder einem anderen logischen Kanal ausgewählt.
- Die von der Sicherheitsdomäne eingerichtete Secure-Channel-Sitzung (falls vorhanden) wird zurückgesetzt, möglicherweise von der Zielanwendung.
- Die Sicherheitsdomäne empfängt den Befehl INSTALL [for personalize] (Start einer neuen Personalisierungssitzung für eine andere Anwendung)
- Der Sicherheitsbereich erhält einen STORE DATA-Befehl mit der Angabe P1. b8=1 (letzter Block)

Jeder STORE DATA-Befehl, der aus einer solchen Personalisierungssitzung heraus empfangen wird, muss von der Sicherheitsdomäne selbst verarbeitet werden.

GlobalPlatform API

Globale Plattform auf einer JAVA-Karte

Dieser Abschnitt enthält nur die für Global Platform 2.2x Java Cards erforderliche API. Die Verwendung der Open Platform 2.0.1 API ist für die Unterstützung älterer Versionen von Anwendungen weiterhin zulässig, wird aber nicht mehr empfohlen. Sie ist in Version 2.1.1 dieser Spezifikation definiert.

Die veraltete API und die neue API greifen beide auf die gleichen Objekte zu, wo dies möglich ist. Während dies bei Methoden, die in beiden Klassen denselben Namen haben (z. B. setATRHistBytes(), setCardContentState() und getCardContentSame()), offensichtlich erscheint, ist auch zu beachten, dass beispielsweise eine Anwendung, die die Methode updated() in der neuen API verwendet, um den Wert der globalen PIN zu ändern, dieselbe globale PIN einer Anwendung beeinflusst, die die Methode setPIN() in der veralteten API verwendet, um die globale PIN zu überprüfen.

GlobalPlatform-spezifische Anforderungen

Um ein Höchstmaß an Interoperabilität der GlobalPlatform-Implementierungen zu gewährleisten, übernimmt GlobalPlatform auch die in Abschnitt 6.2 - Java Card™ 2.2x Virtual Machine Specification - definierte Reihenfolge.

Die folgenden geringfügigen Änderungen an der Standardfunktionalität, die in den Java Card™ 2.1.1 Runtime Environment (JCRE) Spezifikationen und der Java Card™ 2.1.1 Application Programming Interface definiert sind, sind in der Implementierung von

GlobalPlatform enthalten. Einige der Änderungen sind bereits mit der Java Card™ 2.2.x Spezifikation vereinbar und daher keine Änderungen mehr.

GPRegistryEntry-Objekte müssen als Shareable Interface Objects gemäß Abschnitt 6.2.4 "Shareable Interfaces" der Java Card™ 2.2.x Runtime Environment (JCRE) Specification implantiert werden, um einen gemeinsamen Zugriff zu gewährleisten.

GlobalPlatform-Paket AID

Die AID jedes GlobalPlatfrom-Pakets ist eine Verkettung von RID und einem PIX. Der AID-Wert der Java-Kartenexportdatei für die neue GlobalPlatform-API (identisch für GlobalPlatform 2.1 und 2.1.1) auf der Grundlage der in Anhang H - GlobalPlatform-Datenwerte und Kartenerkennungsdaten angegebenen RID ist "A00000015100".

Einrichtung

In Abschnitt 3.1 - The Method install der Java Card™ 2.2.x Runtime Environment (JCRE) Specifications sind die an die Methode übergebenen Parameter als Initialisierungsparameter aus dem Inhalt des eingehenden Byte-Array-Parameters definiert.

Die vorliegende Spezifikation erweitert diese Anforderung und definiert den Inhalt der Installationsparameter weiter. Diese Erweiterung betrifft sowohl die Implementierung eines OPEN als auch das Verhalten eines Java Card-Applets, das für eine GlobalPlatform-Karte entwickelt wurde. Sie wirkt sich nicht auf die Definition der install ()-Methode der Klasse Applet der Java Card™ 2.2.x Application Programming Interface Spezifikation aus.

Die Installationsparameter müssen die folgenden Daten identifizieren, die im Befehl INSTALL [for install] enthalten sind:

- Die Instanz AID
- Die Privilegien
- Die anwendungsspezifischen Parameter. (Während der APDU-Befehl Install-Parameter enthält, die TLV-kodierte System- und anwendungsspezifische Parameter darstellen, benötigt die Anwendung nur Kenntnisse über die anwendungsspezifischen Parameter, d. h. nur LV der TLV-kodierten Struktur "C9" sind als anwendungsspezifische Parameter vorhanden.

Der OPEN ist dafür verantwortlich, dass die Parameter (bArray, bOffset und bLength) die folgenden Informationen enthalten:

Das Array, bArray, muss die folgenden aufeinanderfolgenden LV-codierten Daten enthalten:

- Länge der Instanz AID
- Die Instanz AID
- Dauer der Privilegien
- Die Privilegien
- Länge der anwendungsspezifischen Parameter
- Die anwendungsspezifischen Parameter

Das Byte bOffset muss einen Offset innerhalb des Arrays enthalten, der auf die Länge der Instanz AID zeigt.

Das Byte bLength muss eine Länge enthalten, die die Gesamtlänge der oben definierten Daten im Array angibt.

Das Applet muss die Instanz AID als Parameter verwenden, wenn es die Methode register (byte [] bArray, short bOffset, byte bLength) der Klasse Applet der Java Card 2.2.x Application Programming Interface Spezifikation aufruft.

T=0 Übertragungsprotokoll

GlobalPlatform-Karten sollen in den unterschiedlichsten Umgebungen (d. h. Kartenakzeptanzgeräten) funktionieren. Derzeit beschreiben die Java Card 2.1.1 Runtime Environment (JCRE)-Spezifikationen und die Java Card 2.2.x Runtime Environment (JCRE)-Spezifikationen das Verhalten für Fall-2-Befehle (bei Verwendung des T=0-Protokolls) im Widerspruch zu EMV 2000. GlobalPlatform schreibt vor, dass JCRE diesen Befehlsfall in Übereinstimmung mit ISO/IEC 7816 behandelt: Ein Applet, das einen Befehl des Falles 2 empfängt, baut die Antwort auf und ruft die entsprechende API auf, um die Daten auszugeben. Sind die Daten kleiner als die vom Terminal erwarteten, speichert das OPEN die Daten, gibt einen "6Cxx"-Antwortcode aus und wartet darauf, dass das CAD den Befehl mit der richtigen Länge erneut ausgibt. Wenn der neu erteilte Befehl empfangen wird, verwaltet das JCRE die Ausgabe der gespeicherten Daten.

Atomarität

Sofern nicht anders angegeben, müssen alle internen persistenten Objekte der GlobalPlatform-API mit einer laufenden Transaktion konform sein.

Alle von dieser API durchgeführten Operationen, mit Ausnahme der Methode Application.processData(), müssen atomar ausgeführt werden. Objekte, die verwendet werden, um die Implementierung der Geschwindigkeitsprüfung zu erzwingen, dürfen nicht mit einer laufenden Transaktion übereinstimmen.

Logische Kanäle

Die folgenden Einschränkungen für logische Kanäle gelten für Java Card 2.2.x (weitere Einzelheiten finden Sie in den Spezifikationen der Java Card 2.2.x Runtime Environment (JCRE)):

Die Auswahl einer Anwendung auf einem logischen Kanal gemäß Abschnitt 6.3 - Befehlsversand ist nicht erfolgreich, wenn dieselbe Anwendung oder eine andere Anwendung, die aus dem Code desselben Pakets instanziiert wurde, aus dem die auszuwählende Anwendung instanziiert wurde, derzeit auf einem anderen logischen Kanal ausgewählt ist, der Anwendungscode jedoch nicht die Schnittstelle MultiSelectable implementiert. Sicherheitsdomänen müssen die Schnittstelle MultiSelectable implementieren.

Ein Kontextwechsel von einer Sicherheitsdomäne zu einer Anwendung, wie in Abschnitt 7.3.3 Personalisierungsunterstützung definiert, ist nicht erfolgreich, wenn dieselbe

Anwendung oder eine andere Anwendung, die aus dem Code desselben Pakets instanziiert wurde, aus dem die zu personalisierende Anwendung instanziiert wurde, derzeit auf einem anderen logischen Kanal ausgewählt ist, der Anwendungscode jedoch nicht die Schnittstelle MultiSelectable implementiert.

Eine Anwendung, die das Privileg hat, standardmäßig ausgewählt zu werden, und für eine Karte bestimmt ist, die zusätzliche logische Kanäle unterstützt, sollte die Schnittstelle MultiSelectable implementieren.

GlobalPlarform definiert nur die Zuweisung von logischen Kanalnummern durch die Karte. Optional und wie in Java Card 2.2 definiert, kann eine Karte auch die Zuweisung von logischen Kanalnummern durch das Terminal unterstützen.

Kryptographische Algorithmen

GlobalPlatform-Karten, die RSA-Kryptografie unterstützen, sollten Schlüsselgrößen unterstützen, die nicht als Konstanten in der Klasse Key Builder definiert sind. Insbesondere sollten Schlüsselgrößen unterstützt werden, die ein Vielfaches von 4 Byte (32 Bit) sind und innerhalb der von der Implementierung festgelegten zulässigen Schlüssellängen liegen.

Grad des Vertrauens

Die Java Card 2.2.x-Spezifikationen gehen davon aus, dass die RID der AID-Pakete, Applets und Instanzen verwendet wird, um ein gewisses Maß an Vertrauen zwischen diesen Entitäten zu gewährleisten. In Abschnitt 4.2.2-AID Usage der Java Card 2.2.1 Application Programming Interface ist definiert, dass die RID einer AID einer Komponente mit der RID der AID des Pakets übereinstimmen muss, und in der Definition der Methode register (byte [] bArray, short bOffset, byte bLength) der Java Card 2.2.x Application Programming Interface Spezifikation ist festgelegt, dass eine Ausnahme ausgelöst werden muss, wenn der RID-Anteil der AID-Bytes im bArray-Parameter nicht mit dem RID-Anteil des Java Card-Namens des Applets übereinstimmt.

In der Praxis ist es nicht praktikabel, vorzuschreiben, dass der RID der Instanz-AID mit dem RID der Komponente übereinstimmen muss, aus der sie instanziiert wurde. GlobalPlatform-Implementierungen dürfen nicht vorschreiben, dass es eine Verbindung zwischen der AID einer Instanz und ihrem ursprünglichen Paket geben muss. (Es werden keine Verbindungen zwischen der AID einer Instanz und der AID ihres ursprünglichen Pakets gefordert). Es wird jedoch davon ausgegangen, dass alle Anwendungen im gleichen Paket den gleichen Grad an Vertrauen genießen.

Aufruf von GlobalPlatform-Methoden

Die hier definierte Anwendungsprogrammierschnittstelle ist für jedes Java Card-Applet zugänglich, das mit der Absicht entwickelt wurde, auf einer GlobalPlatform-Karte vorhanden zu sein. Eine Einschränkung besteht nicht in Bezug auf den Konstruktor des Applets und die Methode install () der Klasse Applet der Java Card 2.2.x Application Programming Interface. Da in dieser Spezifikation nicht genau festgelegt ist, wann die Instanz eines Applets zu einem Eintrag im GlobalPlatform-Register der Karte wird, kann

ein Applet-Entwickler nur davon ausgehen, dass dies nach dem erfolgreichen Abschluss der install ()-Methode geschehen ist. Um die Interoperabilität zu gewährleisten, dürfen GlobalPlatform-API-Methoden, die auf den GlobalPlatform-Registry-Eintrag des die Methode aufrufenden Applets zugreifen, nicht innerhalb des Konstruktors oder der install()-Methode aufgerufen werden.

Es folgt eine Auflistung der Methoden, die im Konstruktor oder in der install()-Methode aufgerufen werden können:

- getCardState;
- getCVM;
- getService;

Das erforderliche Verhalten der Karte für den Fall, dass eine Anwendung fälschlicherweise eine andere als die oben aufgeführten Methoden der Klasse org.globalplatform.GPSystem aufruft, ist nicht definiert. So kann beispielsweise eine Ausnahme ausgelöst und die Verarbeitung der Methode install() abgebrochen werden.

Auswahl

Tritt bei GlobalPlatform-Karten während der Verarbeitung der select()-Methode ein Fehler auf oder gibt die select()-Methode den Wert "False" zurück oder kann die Anwendung nicht ausgewählt werden, weil sie die Schnittstelle "Multiselectable" nicht implementiert, setzt das OPEN die Suche im GlobalPlatform-Register nach einer nachfolgenden vollständigen oder teilweisen Übereinstimmung gemäß Abschnitt 6.4.2.1.2 fort. Wird keine Anwendung ausgewählt, bleibt der entsprechende logische Kanal offen, ohne dass derzeit eine Anwendung ausgewählt ist. Da derzeit keine Anwendung ausgewählt ist, wird jeder nachfolgende Befehl außer einem MANAGE CHANNEL- oder SELECT-Befehl zurückgewiesen. Es wird erwartet, dass die kartenexterne Stelle bei einem solchen Fehler geeignete Maßnahmen ergreift, z. B. eine andere Anwendung auswählt, den entsprechenden logischen Kanal schließt, die Karte zurücksetzt oder ausschaltet.

Die Methodenzusammenfassung und -details für die GlobalPlatform Java Card API-Spezifikation sind jetzt in einem separaten Dokument verfügbar, das auf der GlobalPlatform-Website zu finden ist.

Vor-Personalisierung

Wenn der IC zum ersten Mal eingeschaltet wird, befindet er sich im Zustand vor der Personalisierung.

Der INIT_CARD-Befehl (siehe 8.2.2 INIT CARD-Befehl) muss der erste Befehl im Vorpersonalisierungszustand sein. Der Befehl initialisiert den Kartenspeicherbereich mit Standardwerten, und anschließend steht eine so genannte ACOSJ ROOT-Anwendung (siehe 8.0 ACOSJ ROOT-Anwendung) zur Verfügung, um kritische Systemdaten (z. B. "ATR", "ATS" usw.) in das EEPROM zu schreiben. Sobald die Daten geladen sind und

keine weiteren Änderungen mehr bestätigt werden, muss ein Befehl ausgeführt werden, der die ACOSJ ROOT Application deaktiviert.

Wenn der IC den Vorpersonalisierungszustand mit der erfolgreichen Ausführung des Befehls "Active Card" (siehe 8-2.5 ACTIVATE-Befehl) verlässt, befindet er sich im Standardzustand des GlobalPlatform-Kartenlebenszyklus OP_READY, wenn er nicht auf einen anderen Lebenszykluszustand eingestellt ist.

Es ist nicht möglich, in den Zustand vor der Personalisierung zurückzukehren, wenn der aktive Kartenbefehl erfolgreich verarbeitet wurde.

Hinweis: Der Transportschlüssel ist der Vorpersonalisierungsschlüssel der ACOSJ-Stammanwendung und muss von ACS angefordert werden.

ACOSJ ROOT Anwendung

ACOSJ ROOT Anwendungsbeschreibung

Die Befehle in der ROOT-Anwendung können nur verwendet werden, wenn die ROOT-Anwendung korrekt ausgewählt wurde (mit dem Auswahlbefehl der Transporttaste). Nach dem Eintritt in die ROOT-Anwendung kann der Benutzer die Kartenparameter über die von der ROOT-Anwendung unterstützten Befehle lesen oder konfigurieren. Um die ROOT-Anwendung zu verlassen und eine andere Anwendung auszuwählen, muss die Karte zurückgesetzt werden.

Unter der ROOT-Anwendung können die Parameter der Karte gelesen oder konfiguriert werden.

Der ROOT-Antrag wird ungültig, sobald die Karte aktiviert ist.

ACOSJ ROOT Anwendungsbefehlsreferenz

SELECT-Befehl

Der SELECT-Befehl wird zur Auswahl der ROOT-Anwendung verwendet.

INIT_CARD Befehl

Dieser Befehl wird zum Zurücksetzen der Karte verwendet. Mit diesem Befehl wird das gesamte EEPROM initialisiert.

READ-Befehl

Dieser Befehl dient zum Auslesen des Konfigurationsbereichs. Die Konfigurationsparameter der Karte können mit diesem Befehl gelesen werden.

WRITE-Befehl

Dieser Befehl wird verwendet, um Daten in den Konfigurationsbereich zu schreiben. Die Konfigurationsparameter der Karte können mit diesem Befehl gelesen werden.

ACTIVE-Befehl

Dieser Befehl wird zur Aktivierung der Karte verwendet. Sobald dieser Befehl erfolgreich ausgeführt wurde, wird die ROOT-Anwendung ungültig, und die Konfigurationsdaten der Karte können nicht mehr direkt gelesen oder eingestellt werden.

ACOSJ IDENTIFY Anwendung

ACOSJ IDENTIFY Anwendungsbeschreibung

Nachdem die IDENTIFY-Anwendung (AID: 6163732E636F732E61636F736A766572) mit dem SELECT-Befehl ausgewählt wurde, gibt ACOSJ die Versionsnummer von ACOSJ zurück und zeigt an, ob das Fahrzeug aktiviert wurde.

ACOSJ IDENTIFY Anwendungsbefehlsreferenz

SELECT-Befehl

Der Befehl SELECT dient zur Auswahl der Anwendung INDENTIFY.

Status-Bytes

SW1 SW2	Beschreibung
90 00h	Richtiges Kommando
61 XX	SW2 mehr Antwortbytes verfügbar
62 83h	Anwendung dauerhaft blockiert
63 00h	Verifizierung fehlgeschlagen
63 CX	X Wiederholungsversuche übrig
67 00h	Falsche Länge der Daten
68 81h	Logischer Kanal wird nicht unterstützt
68 82h	Secure Messaging wird nicht unterstützt
69 84h	Ungültige Daten
69 82h	Sicherheitsstatus nicht erfüllt
69 85h	Verwendungsbedingungen nicht erfüllt
69 88h	Falscher MAC
6A 86h	Falsch P1 P2
6A 80h	Falsche Werte in den Befehlsdaten
6A 82h	Datei nicht gefunden
6A 81h	Anwendung gesperrt
6A 83h	Datensatz nicht gefunden
6A 84h	Nicht genügend Speicherplatz
6A 88h	Referenzierte Daten nicht gefunden
6D 00h	Ungültiger INS, Befehl nicht vorhanden
6E 00h	Ungültige Klasse

Referenzen

Die folgenden Dokumente dienten als Referenzen für die ACOSJ Funktionsspezifikation:

- GlobalPlatform-Kartenspezifikation Version 2.2.1
- GlobalPlatform Card API Version 1.6
- Java Card 3 API, klassische Ausgabe Version 3.0.4
- Spezifikation der Java Card 3 Platform Runtime Environment, Classic Edition Version 3.0.4 September 2011
- Spezifikation der Java Card 3 Platform Virtual Machine, Classic Edition Version 3.0.4 September 2011
- GlobalPlatform Card Mapping Guidelines der bestehenden GP v2.1.1 Implementierung auf v2.2.1 Version 1.0.1

Garantiebedingungen

Auf ein neues Produkt, das im Vertriebsnetz von Alza gekauft wurde, wird eine Garantie von 2 Jahren gewährt. Wenn Sie während der Garantiezeit eine Reparatur oder andere Dienstleistungen benötigen, wenden Sie sich direkt an den Produktverkäufer. Sie müssen den Originalkaufbeleg mit dem Kaufdatum vorlegen.

Als Widerspruch zu den Garantiebedingungen, für die der geltend gemachte Anspruch nicht anerkannt werden kann, gelten:

- Verwendung des Produkts für einen anderen Zweck als den, für den das Produkt bestimmt ist, oder Nichtbeachtung der Anweisungen für Wartung, Betrieb und Service des Produkts.
- Beschädigung des Produkts durch Naturkatastrophe, Eingriff einer unbefugten Person oder mechanisch durch Verschulden des Käufers (z.B. beim Transport, Reinigung mit unsachgemäßen Mitteln usw.).
- Natürlicher Verschleiß und Alterung von Verbrauchsmaterialien oder Komponenten während des Gebrauchs (wie Batterien usw.).
- Exposition gegenüber nachteiligen äußeren Einflüssen wie Sonnenlicht und anderen Strahlungen oder elektromagnetischen Feldern, Eindringen von Flüssigkeiten, Eindringen von Gegenständen, Netzüberspannung, elektrostatische Entladungsspannung (einschließlich Blitzschlag), fehlerhafte Versorgungs- oder Eingangsspannung und falsche Polarität dieser Spannung, chemische Prozesse wie verwendet Netzteile usw.
- Wenn jemand Änderungen, Modifikationen, Konstruktionsänderungen oder Anpassungen vorgenommen hat, um die Funktionen des Produkts gegenüber der gekauften Konstruktion zu ändern oder zu erweitern oder nicht originale Komponenten zu verwenden.