Project – “IS 16695 (Part 3)”

Embedded System
CDAC Noida
(21st Dec 2023)

About – "IS 16695"

- SCOSTA is BIS standard IS 16695 Part 1 and Part 2.
  - Part 1: Basic Command Set
  - Part 2: Public Key Infrastructure –
    - This standard is a superset of Part1 of this standard. It describes specifications for public key infrastructure support in an operating system for smart cards.

Revised and Published in 2022

Bureau of Indian Standards
The National Standards Body of India

- ***This Project (IS 16695 Part 3) provides***
  - ***The support for smart card operating system implementers as well as testers or validators to verify compliance with IS 16695 Part 1.***

IS 16695 Part 3 – "Components"

- **Test Scripts Architecture**
  - Syntax
  - Notations
  - Commands (File, Security, Secure Messaging, etc.)
  - Configurable for Cryptographic algorithms
- **Test Tool Platform**
  - Script-based Testing, easy-to-understand language
  - Support Contact [T0, T1 - ISO-7816-3] and Contactless(ISO-14443)
  - Secure Messaging
  - Error Condition Check
  - Intersperse C/C++ code
- **Test Case(s)**

IS 16695 Part 3 –"Components"

✓ **Test Scripts Architecture -Syntax**

- White Spaces do not matter and Each command is terminated by ';' character.
- A general syntax of the command is the following.

  CommandAPDU expected_status;
- Here "CommandAPDU" is an ISO command. Upon its execution, the received status is checked for the "expected_status". "expected_status" is optional and when not specified, it defaults to checking for no errors (i.e. OK).
- "expected_status" is the string of kind "SW?=<value>". The "expected_status" is optional

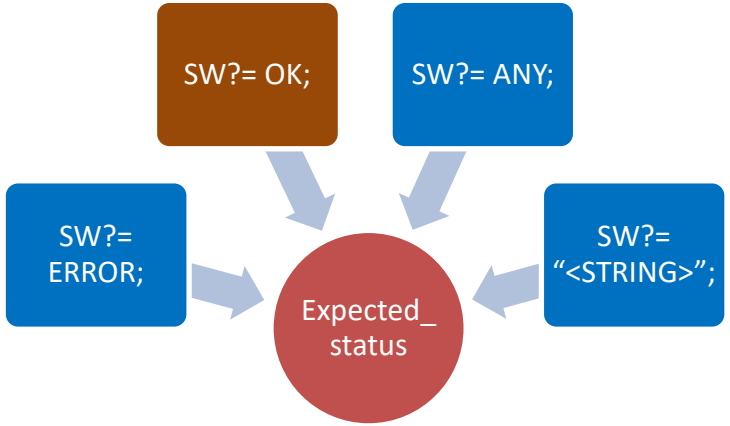
CommandAPDU :

✓ *READBINARY*

{RB || READBINARY} [SFI =<sfid>] <offset> <numbytes> [EXPVAL{?= || !=} {<hexstring> || <string>}]

✓ *GETCHALLENGE*

{GETC || GETCHALLENGE} [ALGO=<algo>]<numbytes> [EXPVAL{?= || !=} {<hexstring> || <string>}]



IS 16695 Part 3 – "Components"

- ✓ Test Scripts Architecture –Syntax
- ✓ **Test Scripts Architecture -Notations**

- ***Bold Italic Text*** provides the syntax of the command language.
- **UPPERCASE** words in bold and italic indicate keywords.
- Constructs enclosed inside **[square brackets]** are optional.
- Constructs enclosed inside **{curly brackets}** are mutually exclusive – that is only one of them can exist.
- Constructs enclosed between **<angular brackets>** are the values – typically integers and strings.
- Constructs separated by a ‖ represent any one of the list.

IS 16695 Part 3 –"Components"

- ✓ Test Scripts Architecture –Syntax
- ✓ Test Scripts Architecture – Notations
- ✓ **Test Scripts Architecture - Commands**

- **File-related operation**
  - Create File
    - CREATEFILE FILEID=<fileid> {DF [DFNAME=<dfname>] [SE=<se>] [SEFILE=<sefile>] || {INTERNAL || WORKING} {TRANSPARENT FILESIZE=<filesize> || {FIXEDLENGTH || VARLENGTH || CYCLIC} MNR=<mnr> MRL=<mrl> [SIMPLETLV] } [SFI=<sfid>] [DATACODING={WRITE_OR || WRITE_AND || WRITE_ONCE}] } [LCSI=<lcsi>][COMPACT_ATTR=<compact_access_rule>][EXPANDED_ATTR=<expanded_access_rule>]
  - READBINARY
    - {RB || READBINARY} [SFI =<sfid>] <offset> <numbytes> [EXPVAL{?= || !=} {<hexstring> || <string>}]
  - DELETEFILE
    - {DF || DELETEFILE} {DFNAME || CDF || CEF || PDF || MF || MFPATH || CDFPATH} [= {<string> || <id>}] {FIRST || LAST || NEXT || PREVIOUS || TOLAST || FROMLAST}
- **Security-related operation**
  - External Authenticate
    - {EAUTH || EXTERNALAUTHENTICATE} ALGO=<algo> {GRD || SRD}=<refdatano>[RESPONSE =] <response>
  - VERIFY
    - VERIFY {GRD || SRD} =<refdata> [<passwd>]
  - GETCHALLENGE
    - {GETC || GETCHALLENGE} [ALGO=<algo>]<numbytes> [EXPVAL{?= || !=} {<hexstring> || <string>}]

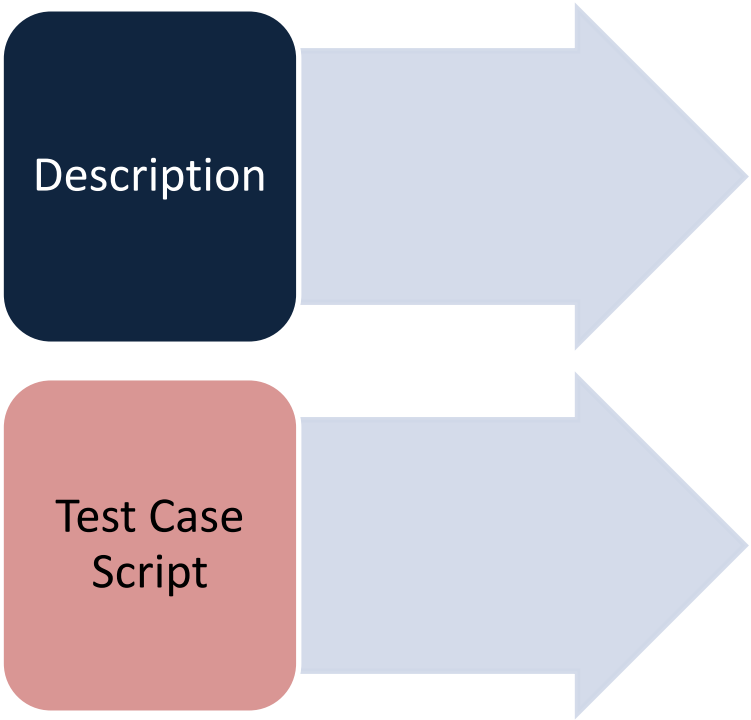
IS 16695 Part 3 –"Components"

- ✓ Test Scripts Architecture –Syntax
- ✓ Test Scripts Architecture – Notations
- ✓ Test Scripts Architecture – Commands
- ✓ **Test Tool Platform**

- Supports the transport layer in T=0, T=1 and T=CL protocol. The APDUs can be specified in a well understood scripting language.
- The tool supports secure messaging. In this mode, the APDUs can be specified in usual manner but are transported to the smart card using the secure messaging.
  - The script writer has absolute control on the format of the SM on the command and can use it to verify the format on the received response APDUs.
- The tool also supports error condition checking. It is possible for the script writer to intentionally provide wrong APDUs or wrong SM specifications and check for the correct behavior of the card (which in this case must be a return SW1 and SW2 to indicate errors).
- The tool can also check for the expected values and can report mismatch in case the expected values are not returned by the card.
- The tool also handles automatically the warning conditions returned by the card and can handle getting data using GET RESPONSE command if needed.

IS 16695 Part 3 –"Components"

- ✓ Test Scripts Architecture –Syntax
- ✓ Test Scripts Architecture – Notations
- ✓ Test Scripts Architecture – Commands
- ✓ Test Tool Platform
- ✓ **Test Case(s)**



Thank You