INTERNATIONAL STANDARD ISO/IEC 18013-3:2009

TECHNICAL CORRIGENDUM 2

Published 2013-11-15

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION • МЕЖДУНАРОДНАЯ ОРГАНИЗАЦИЯ ПО СТАНДАРТИЗАЦИИ • ORGANISATION INTERNATIONALE DE NORMALISATION
INTERNATIONAL ELECTROTECHNICAL COMMISSION • МЕЖДУНАРОДНАЯ ЭЛЕКТРОТЕХНИЧЕСКАЯ КОМИССИЯ • COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE

Information technology — Personal identification — ISO-compliant driving licence —

Part 3: Access control, authentication and integrity validation

TECHNICAL CORRIGENDUM 2

Technologies de l'information — Identification des personnes — Permis de conduire conforme à l'ISO —

Partie 3: Contrôle d'accès, authentification et validation d'intégrité

RECTIFICATIF TECHNIQUE 2

Technical Corrigendum 1 to ISO/IEC 18013-3:2009 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 17, *Cards and personal identification*.

ICS 35.240.15 **Ref. No. ISO/IEC 18013-3:2009/Cor.2:2013(E)**

Published in Switzerland

Page 34, Annex B

Replace clause B.10.4 with the following:

B.10.4 Example Using Configuration 4

Static document keying material:

K_{doc} = ‘348D2F25C266CC8068F99391BF0F5CCB87
6B5F5DDB004D0E5C8BCD1D3ACF2FDADA’

Compute Basic Access Keys:

Input: $K_{seed} = H_{SHA\text{-}256}(K_{doc})$
K_{seed} = ’ 2E3AB26DC47C4BA6724E58514492ABF3
B2B92BD21A40BEBFAF0D7A52D291EA98’

Encryption Key (K_{enc}) computation:

1. Concatenate K_{seed} and c (c = 1):
D = ‘2E3AB26DC47C4BA6724E58514492ABF3
00000001’

2. Calculate the hash of D:
$H_{SHA\text{-}256}(D)$ = ‘0AFD72514422FD43622BB3F1680F6243
5A6F9B8E83C92A299D3B89124D89B611’

3. Form key:
K_{enc} = ‘0AFD72514422FD43622BB3F1680F6243
5A6F9B8E83C92A299D3B89124D89B611’

Message Authentication Key (K_{mac}) computation:

4. Concatenate K_{seed} and c (c = 2):
D = ‘2E3AB26DC47C4BA6724E58514492ABF3
00000002’

5. Calculate the hash of D:
$H_{SHA\text{-}256}(D)$ = ‘F3BC7313E7D34BB3BE0EB07B4DF9DE6A
E73A4CA604FE1516AEBFB4140115A5A6’

6. Form key:
K_{mac} = ‘F3BC7313E7D34BB3BE0EB07B4DF9DE6A
E73A4CA604FE1516AEBFB4140115A5A6’

Authentication and Establishment of Session Keys:

IS:

1. Request an 8 byte random challenge from the document’s SIC:

Command APDU:

CLA	INS	P1	P2	Le
‘00’	‘84’	‘00’	‘00’	‘08’

Document SIC:

2. Generate random challenge and return it to IS:
RND.ICC = 'E880AAE12EB3A5FB'

Response APDU:

Response Data Field	SW1	SW2
RND.ICC	'90'	'00'

IS:

3. Generate an 8-byte random challenge and 24-byte random keying material:
RND.IFD = 'B962840EFBFE80C9'
K.IFD = '1D05B3E621AC7BB4786AC1657D0C4C11
58875525EB21659D905674FCAFF94421'

4. Concatenate RND.IFD, RND.ICC and K.IFD:
S = 'B962840EFBFE80C9E880AAE12EB3A5FB
1D05B3E621AC7BB4786AC1657D0C4C11
58875525EB21659D905674FCAFF94421'

5. Encrypt S using AES with key K_{enc}:
E_IFD = 'DA020143D3816ACB4EF104FDAAFA30A7
BC49BFE6B616D9D061F728EB063362A1
C435F95DDACBE36C37A09472BBCD464B'

6. Compute CMAC over E_IFD with key K_{mac}:
M_IFD = '4F3B9205ADB2DD20'

7. Construct command data for MUTUAL AUTHENTICATE and send command to the document's SIC:
cmd_data = 'DA020143D3816ACB4EF104FDAAFA30A7
BC49BFE6B616D9D061F728EB063362A
1C435F95DDACBE36C37A09472BBCD464B
4F3B9205ADB2DD20'

Command APDU:

CLA	INS	P1	P2	Lc	Command Data Field	Le
'00'	'82'	'00'	'00'	'38'	cmd_data	'38'

Document SIC:

8. Generate 16-byte random keying material:
K.ICC = '56F1510FDCC2B01787E80D2D5E340840
20C93698AF4599C9B9B7D68EB2E958B7'

9. Calculate XOR of K.IFD and K.ICC:
K_{seed} = '4BF4E2E9FD6ECBA3FF82CC4823384451
784E63BD4464FC5429E1A2721D101C96'

10. Derive session keys:
KS_{enc} = ‘60BDD38EE1B27EEAC7AF9907889F2E04
74C7AF231C71705BB2A84BF87BA825FF’
KS_{mac} = ‘978E2D4BFC62716966B215A28980ED04
1756A53EBC56AE7CE9F8341167210C33’

11. Initialize send sequence counter:
SSC = ‘2EB3A5FBFBFE80C9’

12. Concatenate RND.ICC, RND.IFD and K.ICC; and add padding:
R = ‘E880AAE12EB3A5FBB962840EFBFE80C9
56F1510FDCC2B01787E80D2D5E340840
20C93698AF4599C9B9B7D68EB2E958B7’

13. Encrypt R using AES with key K_{enc}:
E_ICC = ‘2918E899CF1B797F5F869521B1B942B7
8F72C19AA8162C82BA5295733D33C2F7
2BABED4C7687E8D2A58E9C4F109F92A2’

14. Compute CMAC over E_ICC with key K_{mac}:
M_ICC = ‘2FDBF985C7DA7CCF’

15. Construct response data and send response APDU to the IS:
resp_data = ‘2918E899CF1B797F5F869521B1B942B7
8F72C19AA8162C82BA5295733D33C2F7
2BABED4C7687E8D2A58E9C4F109F92A2
2FDBF985C7DA7CCF’

Response APDU:

Response Data Field	SW1	SW2
resp_data	‘90’	‘00’

IS:

16. Calculate XOR of K.IFD and K.ICC:
K_{seed} = ‘4BF4E2E9FD6ECBA3FF82CC4823384451
784E63BD4464FC5429E1A2721D101C96’

17. Derive session keys:
KS_{enc} = ‘60BDD38EE1B27EEAC7AF9907889F2E04
74C7AF231C71705BB2A84BF87BA825FF’
KS_{mac} = ‘978E2D4BFC62716966B215A28980ED04
1756A53EBC56AE7CE9F8341167210C33’

18. Initialize send sequence counter:
SSC = ‘2EB3A5FBFBFE80C9’

Secure Messaging:

IS:

1. SELECT EF.COM (file identifier = ’01 1E’):

Unprotected command APDU:

CLA	INS	P1	P2	Lc	Command Data Field
‘00’	‘A4’	‘02’	‘00’	‘02’	’01 1E’

a) Mask class byte and pad command header:
cmd_header = ‘0CA4020C800000000000000000000000’

b) Pad data:
p_data = ‘011E800000000000000000000000000’

c) Encrypt p_data using AES with KS_{enc}:
enc_data = ‘C74A8B66F7EA68098B8B4F1E51F9BE58’

d) Build DO‘87’:
DO87 = ‘871101C74A8B66F7EA68098B8B4F1E51
F9BE58’

e) Concatenate cmd_header and DO87:
M = ‘0CA4020C800000000000000000000000
871101C74A8B66F7EA68098B8B4F1E51
F9BE58’

f) Compute CMAC of M with KS_{mac}:
- Increment SSC:
SSC = ‘2EB3A5FBFBFE80CA’
- Concatenate padded SSC and M:
N = ‘00000000000000002EB3A5FBFBFE80CA
0CA4020C800000000000000000000000
871101C74A8B66F7EA68098B8B4F1E51
F9BE58’
- Compute MAC:
CC = ‘EC6B4CF08A7206D8’

g) Build DO‘8E’:
DO8E = ‘8E08EC6B4CF08A7206D8’

h) Construct command data:
cmd_data = ‘871101C74A8B66F7EA68098B8B4F1E51
F9BE588E08EC6B4CF08A7206D8’

Protected command APDU:

CLA	INS	P1	P2	Lc	Command Data Field	Le
‘0C’	‘A4’	‘02’	‘0C’	‘1D’	cmd_data	‘00’

Document SIC:

2. Set EF.COM as the currently selected file and send affirmative response to IS:

Unprotected response APDU:

SW1	SW2
'90'	'00'

a) Build DO'99':
DO99 = '99029000'

b) Compute CMAC of DO99 with KS_{mac}:
- Increment SSC:
SSC = '2EB3A5FBFBFE80CB'
- Concatenate padded SSC and DO99:
N = '00000000000000002EB3A5FBFBFE80CB
99029000'
- Compute MAC:
CC = '22CC755FA2A7973B'

c) Build DO'8E':
DO8E = '8E0822CC755FA2A7973B'

d) Construct response data:
resp_data = '990290008E0822CC755FA2A7973B'

Protected response APDU:

Response Data Field	SW1	SW2
resp_data	'90'	'00'

IS:

3. READ BINARY of the first 4 bytes:

Unprotected command APDU:

CLA	INS	P1	P2	Le
'00'	'B0'	'00'	'00'	'04'

a) Mask class byte and pad command header:
cmd_header = '0CB000008000000000000000000000000'

b) Build DO '97':
DO97 = '970104'

c) Concatenate cmd_header and DO97:
M = '0CB000008000000000000000000000000
970104'

d) Compute CMAC of M with KS_{mac}:
- Increment SSC:
  SSC = '2EB3A5FBFBFE80CC'
- Concatenate padded SSC and M:
  N = '0000000000000002EB3A5FBFBFE80CC
  0CB000008000000000000000000000000
  970104'
- Compute MAC:
  CC = '7C564CD2EC22E606'

e) Build DO'8E':
DO8E = '8E087C564CD2EC22E606'

f) Construct command data:
cmd_data = '9701048E087C564CD2EC22E606'

Protected command APDU:

CLA	INS	P1	P2	Lc	Command Data Field	Le
'0C'	'B0'	'00'	'00'	'0D'	cmd_data	'00'

Document SIC:

4. Return 4 bytes of EF.COM starting at offset 0:

data = '600D5F01'

Unprotected response APDU:

Response Data Field	SW1	SW2
data	'90'	'00'

a) Pad data:
p_data = '600D5F0180000000000000000000000'

b) Encrypt p_data using AES with KS_{enc}:
enc_data = 'DBBA6E8C7C837A22FD94F7F3455A64AE'

c) Build DO'87':
DO87 = '871101DBBA6E8C7C837A22FD94F7F345
5A64AE'

d) Build DO'99':
DO99 = '99029000'

e) Concatenate DO'87' and DO'99':
M = '871101DBBA6E8C7C837A22FD94F7F345
5A64AE99029000'

f) Compute CMAC of M with KS_{mac}:
- Increment SSC:
  SSC = '2EB3A5FBFBFE80CD'
- Concatenate padded SSC and M:
  N = '00000000000000002EB3A5FBFBFE80CD
  871101DBBA6E8C7C837A22FD94F7F345
  5A64AE99029000'
- Compute MAC:
  CC = 'CB87EE6B23392361'

g) Build DO'8E':
DO8E = '8E08CB87EE6B23392361'

h) Construct response data:
resp_data = '871101DBBA6E8C7C837A22FD94F7F345
5A64AE990290008E08CB87EE6B233923
61'

Protected response APDU:

Response Data Field	SW1	SW2
resp_data	'90'	'00'

IS:

5. READ BINARY of the remaining 11 bytes:

Unprotected command APDU:

CLA	INS	P1	P2	Le
'00'	'B0'	'00'	'04'	'0B'

a) Mask class byte and pad command header:
cmd_header = '0CB000048000000000000000000000000'

b) Build DO '97':
DO97 = '97010B'

c) Concatenate cmd_header and DO97:
M = '0CB000048000000000000000000000000
97010B'

d) Compute CMAC of M with KS_{mac}:
- Increment SSC:
  SSC = '2EB3A5FBFBFE80CE'
- Concatenate padded SSC and M:
  N = '00000000000000002EB3A5FBFBFE80CE
  0CB000048000000000000000000000000
  97010B'
- Compute MAC:
  CC = '98EC6D1082ECDF5F'

e) Build DO‘8E’:
DO8E = ‘8E0898EC6D1082ECDF5F’

f) Construct command data:
cmd_data = ‘97010B8E0898EC6D1082ECDF5F’

Protected command APDU:

CLA	INS	P1	P2	Lc	Command Data Field	Le
‘0C’	‘B0’	‘00’	‘04’	‘0D’	cmd_data	‘00’

Document SIC:

6. Return 11 bytes of EF.COM starting at offset 4:

data = ‘04303130305C04616B6567’

Unprotected response APDU:

Response Data Field	SW1	SW2
data	‘90’	‘00’

a) Pad data:
p_data = ‘04303130305C04616B65678000000000’

b) Encrypt p_data using AES with KS_{enc}:
enc_data = ‘9D4B6092AEEC6868505D1CFDC112EA0D’

c) Build DO‘87’:
DO87 = ‘8711019D4B6092AEEC6868505D1CFDC1
12EA0D’

d) Build DO‘99’:
DO99 = ‘99029000’

e) Concatenate DO‘87’ and DO‘99’:
M = ‘8711019D4B6092AEEC6868505D1CFDC1
12EA0D99029000’

f) Compute CMAC of M with KS_{mac}:
- Increment SSC:
SSC = ‘2EB3A5FBFBFE80CF’
- Concatenate padded SSC and M:
N = ‘00000000000000002EB3A5FBFBFE80CF
8711019D4B6092AEEC6868505D1CFDC1
12EA0D99029000’
- Compute MAC:
CC = ‘7A8EA0EDBEA375DA’

g) Build DO‘8E’:
DO8E = ‘8E087A8EA0EDBEA375DA’

h) Construct response data:
resp_data = ‘8711019D4B6092AEEC6868505D1CFDC1
12EA0D990290008E087A8EA0EDBEA375
DA’

Protected response APDU:

Response Data Field	SW1	SW2
resp_data	‘90’	‘00’

Page 86, C.2.1.7

Delete “0:00”

Page 86, C.2.2.2

In Table C.5 insert the following normative statement for bits 7 and 8 in the "Meaning" column after the sentence "Reserved for future use (set to zero).":

“The IDL application shall ignore the value of these bits.”

After Table C.6 insert the following normative statement:

“The IDL application shall ignore the access right bits of non-existing data groups.”

Page 87, C.2.2.3

Replace

“The effective authorization of the application’s trust root certificate is equal to its relative authorization”

by

“The effective authorization of a trust root certificate is equal to its relative authorization”.

Page 90, C.4.3.1

In the heading, replace “Trust root certificate” by “Current trust root certificate”

In the first sentence replace “The SIC has a single trust root certificate…” by “The SIC has a single current trust root certificate…”

Page 90, C.4.3.3

Replace

“On application selection, this key is set to the trust root certificate’s public key”

by

“On application selection, this key is set to the current trust root certificate’s public key”

Insert the following at the end of the section:

"The public key shall be available for external authentication only in the case where its certificate has a path length constraint of 0."

Page 90, C.4.4.1

Replace

“The MSE:SET DST command can optionally be used to select a specific certificate verification key before using the PSO: VERIFY CERTIFICATE command”

by

“The MSE:SET DST command shall be used to select an alternate trust root public key as currently active certificate verification key. The MSE:SET DST command can optionally be used to select any other specific certificate verification key as currently active certificate verification key before using the PSO: VERIFY CERTIFICATE command.”

Page 91, C.4.4.2

Replace

“Verification is successful if and only if:

a) the effective authorization of the parent certificate indicates a path length constraint value that is > 0,

b) the certificate’s expiration date is after the current on-card date, and

c) the certificate’s signature verifies using the currently active certificate verification key.”

by

“Verification is successful if the following conditions are fulfilled:

a) the syntax of the certificate is in conformance with annex C.2 of this standard,

b) the certificate’s signature verifies using the currently active certificate verification key,

c) the effective authorization of the parent certificate indicates a path length constraint value that is > 0 (this check does not apply if a trust root public key that has not been imported by means of a certificate is used for the certificate verification),

d) the certificate's expiration date is after the current on-card date or equals the current on-card date,

e) the certificate effective date is after the effective date of the currently active certificate verification key or equals the effective date of the currently active certificate verification key,

f) for certificates which are not trust root certificates the expiration date is before the expiration date of the currently active certificate verification key or equals the expiration date of the currently active certificate verification key, and

g) the certificate's expiration date is after its effective date or equals its effective date."