ICAO

MACHINE READABLE TRAVEL DOCUMENTS TECHNICAL REPORT

CLARIFICATION ON THE CHIP AUTHENTICATION WITH 3DES

Version – 1.0 | September 2024

ISO/IEC JTC1 SC17 WG3

Clarification on Chip Authentication with 3DES

Issue 1: Implementation of supported APDUs for Chip-Authentication with 3DES

Issue: Multiple eMRTDs have been observed in the field that support CA with 3DES but do not implement the “MSE:Set KAT” command. In consequence, if an Inspection System would try to perform Chip Authentication using the “MSE:Set KAT” command, the inspection procedure would fail.

ICAO Doc 9303 Part 11 Section 6.2.4 states:

Depending on the symmetric algorithm to be used, two implementations of Chip Authentication are available.

- The following command SHALL be used to implement Chip Authentication with 3DES Secure Messaging:
    1. MSE:Set KAT
- The following sequence of commands SHALL be used to implement Chip Authentication with AES Secure Messaging and MAY be used to implement Chip Authentication with 3DES Secure Messaging:
    1. MSE:Set AT
    2. GENERAL AUTHENTICATE

Clarification: The specification requires any eMRTD supporting CA with 3DES to implement the “MSE:Set KAT” command. Consequently, any eMRTD supporting CA with 3DES that additionally implements the “MSE:Set AT” always needs to implement “MSE:Set KAT” too. A Configuration that implements CA with 3DES without implementing the “MSE:Set KAT” command is not compliant to Doc 9303 Part 11.

Recommendation for Issuers: Any issuer currently issuing eMRTDs supporting CA with 3DES but not implementing the “MSE:Set KAT” command should update their product to support the respective command at the earliest time possible.

Alternatively, the deprecation of CA with 3DES in favor of CA with AES could be envisaged, whereby the requirement to implement the “MSE:Set KAT” command would be eliminated.

Recommendation for Inspection Systems: Any eMRTD implementing only the “MSE:Set AT” command supports CA with AES in addition to CA with 3DES. Consequently, if an Inspection Systems inspects an eMRTD that supports both flavours of CA it could always prefer to perform CA with AES, which would circumvent the issue of the missing “MSE:Set KAT” command.

Issue 2: Encoding of Public Key for Chip Authentication with DH within ChipAuthenticationPublicKeyInfo

Issue: The Public Key of an eMRTD's Chip Authentication Key Pair must be provided to an Inspection System in order to perform the Chip Authentication. The Public Key SHALL be provided as `SubjectPublicKeyInfo` in the `ChipAuthenticationPublicKeyInfo` structure (c.f. Doc 9303 Part 11 sec. 6.2, 9.1 and 9.2.6).

```
ChipAuthenticationPublicKeyInfo ::= SEQUENCE {
      protocol OBJECT IDENTIFIER(id-PK-DH | id-PK-ECDH),
      chipAuthenticationPublicKey SubjectPublicKeyInfo,
      keyId INTEGER OPTIONAL
}

SubjectPublicKeyInfo ::= SEQUENCE {
      algorithm AlgorithmIdentifier,
      subjectPublicKey BIT STRING
}

AlgorithmIdentifier ::= SEQUENCE {
      algorithm OBJECT IDENTIFIER,
      parameters ANY DEFINED BY algorithm OPTIONAL
}
```

Doc 9303 Part 11 Sec 9.6 specifies the permitted encodings for Public Keys for Chip Authentication with DH. The Table provided there indicates that X9.42 MUST be used for the encoding of Chip Authentication Public Keys.

However this is overruled by Doc 9303 Part 11 Section 6.2.3.1, which states:
"For Chip Authentication with DH the respective algorithms and formats from Section 9.6 and Table 5 MUST be used. For Public Keys, PKCS#3 [PKCS#3] MUST be used instead of X9.42 [X9.42]."

The two statements resulted in an ambiguous interpretation of the specification. In result, passports implementing Chip Authentication with DH with either of the two Public Key Encoding have been observed in the field.

Hereby X9.42 mandates using the OID of "1.2.840.10046.2.1" while PKCS#3 mandates using the OID of "1.2.840.113549.1.3.1". The encoding of the key parameters differs for PKCS#3 and X9.42 (specification also provided further below).

Clarification: eMRTDs implementing Chip Authentication with DH MUST use the Public Key encoding specified in PKCS#3 and therefore must use OID "1.2.840.113549.1.3.1" within `AlgorithmIdentifier`.

Recommendation for Issuers: Any issuer currently issuing eMRTDs supporting CA with DH that encode the CA Public Key according to X9.42 should update their product at the earliest time possible and change the Public Key encoding to PKCS#3.

Recommendation for Inspection Systems: Inspection Systems supporting Chip Authentication with DH should be able to read Public Keys encoded according to PKCS#3 and X9.42. Both PKCS#3 and X9.42 specify a SEQUENCE structure for the `parameters` field within `AlgorithmIdentifier` (see below), whereby in both cases the first two entries in the Sequence are the `INTEGER` p and g. These both parameters are sufficient to perform Chip Authentication. The additional parameter q

from the X9.42 encoding is only required to validate the Public Key, in order to prevent a small subgroup attack on the key pair. However, in case of CA Passive Authentication already ensures the authenticity of the Public Key. Thus, Public Key validation can be omitted.

For PKCS#3 the dhKeyAgreement OID is intended to be used in the algorithm field of a value of type AlgorithmIdentifier, while the parameters field of that type have the ASN.1 type DHParameter:

```
dhKeyAgreement OBJECT IDENTIFIER ::= {
    iso(1) member-body(2) us(840) rsadsi(113549)
    pkcs(1) pkcs-3(3) 1
}
DHParameter ::= SEQUENCE {
     prime INTEGER, -- p
     base INTEGER, -- g
     privateValueLength INTEGER OPTIONAL
}
```

For X9.42 the dhpublicnumber OID is intended to be used in the algorithm field of a value of type AlgorithmIdentifier, while the parameters field of that type have the ASN.1 type DomainParameters:

```
dhpublicnumber OBJECT IDENTIFIER ::= {
     iso(1) member-body(2) us(840) ansi-x942(10046)
     number-type(2) 1
}
DomainParameters ::= SEQUENCE {
     p INTEGER, -- odd prime, p=jq +1
     g INTEGER, -- generator, g
     q INTEGER, -- factor of p-1
     j INTEGER OPTIONAL, -- subgroup factor
     validationParms  ValidationParms OPTIONAL
}
```