Network Device Interpretation # 202101

Key Pair Generation for Authentication

Status: ☒ *Active* ☐ *Inactive*

Date: *7-Jun-2021*

End of proposed Transition Period (to be updated after TR2TD process): *7-Jul-2021*

Type of Change: ☐ Immediate application ☒ Minor change ☐ Major change

Type of Document: ☒ *Technical Decision* ☐ *Technical Recommendation*

Approved by: ☒ *Network iTC Interpretations Team* ☒ *Network iTC*

Affected Document(s): *NDcPPv2.2e, NDSDv2.2*

Affected Section(s): *FCS_CKM.1*

Superseded Interpretation(s): *None*

Issue:

Several portions of the PP suggest that the TOE must have the ability to generate asymmetric cryptographic keys for all instances in which the TOE presents that key to authenticate itself to a peer. When x.509 is involved, the PP is clear that the TOE must have the ability to generate key pairs for authentication (i.e. generate a Certificate Signing Request). A section of the PP suggests that with Distributed TOEs, these keys are ONLY allowed to be generated on the TOE. In all cases, it is not clear if generating these keys on a non-TOE entity and importing them into the TOE is allowed or not.

Our intent is to get clarification if the TOE must have the ability to generate authentication key pairs and if importing of these keys is allowed in the following instances:

1. SSHS (Server Host Key Pair)

2. SSHC (User's Identity Key Pair)

3. IPsec (Endpoint Cert Key Pair)

4. TLSS (Server Cert Key Pair)

a. Ability to generate Certificate Signing Request is required

5. TLSC (Client Cert for Mutual Auth)

a. Ability to generate Certificate Signing Request is required

6. Distributed TOE

a. In all instances listed above

BACKGROUND: The following two excerpts from Application Note 9 seem to apply globally to the PP, which suggest that the ability for the TOE to generate key pairs for authentication of the TOE to a peer is required in all instances (i.e. TLS, SSH, IPsec, etc.):

"The ST author selects all key generation schemes used for key establishment (including generation of ephemeral keys) and device authentication."

"If the TOE acts as a receiver in the key establishment schemes and is not configured to support mutual authentication, the TOE does not need to implement key generation." Stating that the TOE is not required to implement key generation in the scenario when the TOE only receives a public key AND is not performing mutual authentication, suggests that if the TOE was presenting a key for authentication then that key must have been able to be generated on the TOE.

Section B.4.1 makes it clear that the TOE must support the ability to generate key pairs for authentication when the TOE supports x.509 certificates (via a CSR) to authenticate itself to a peer; however, it is not clear if in addition to the support for generating CSRs, if generating key pairs on a non-TOE IT entity then importing them into the TOE is allowed or not:

"the TOE only needs to be able to generate a Certification Request if the TOE needs to present an X.509 certificate to another endpoint via the TSF for authentication".

The PP seems to be strict when it comes to Distributed TOEs, as in Footnote #4 of the PP which seems to suggest that the TOE must ONLY generate these key pairs, and thus importing of key pairs for this use is not allowed:

"Each component of a distributed TOE will be required either to perform on-board key generation and (if the TOE uses X.509 certificates as in Appendix B.4.1) RFC 2986 Certificate Request generation, or else to receive its keys and certificates, generated on some other component of the TOE…".

Resolution:

The TOE must be able to generate asymmetric keys (public/private key pairs) for each FCS_CKM.1 selection made by the ST author. In FCS_CKM.1 the ST author must select all the key generation schemes necessary to cover all protocols selected in FTP_ITC.1, FTP_TRP.1/Admin, FTP_TRP.1/Join, and FPT_ITT.1 that depend on the TOE containing an asymmetric key pair (e.g. not needed for a D/TLS client that is not supporting mutual authentication).

Wherever asymmetric keys are used for any trusted path/channel using any protocol, the TOE must be able to use keys (public and private) generated by the TOE. While the TOE may also optionally support using keys generated by a non-TOE entity, defining secure key injection functionality is outside the scope of this cPP.

In NDcPPv2.2e, Table 1 footnote 4 shall be replaced as follows:

<old>

Each component of a distributed TOE will be required either to perform on-board key generation and (if the TOE uses X.509 certificates as in Appendix B.4.1) RFC 2986 Certificate Request generation, or else to receive its keys and certificates, generated on some other component of the TOE, using a secure registration channel at the point where the component is joined to the TOE. (subsequent changes of keys and certificates may then use the post-registration inter-component secure channel). Certificate request generation will be required from either the component that generates the key or the component that receives the key.

</old>

<new>

The overall TOE is required to support on-board key generation and (if the TOE uses X.509 certificates as in Appendix B.4.1) RFC 2986 Certificate Request generation. If not all TOE components are supporting on-board key generation (and generation of certificate requests, where applicable), the TOE shall support distribution of keys to the TOE components that are not supporting key generation themselves. Depending on the life-cycle phase, either a secure registration channel shall be used for key distribution at the point where the component is joined to the TOE or an inter-component secure channel shall be used for key distribution post-registration.

</new>

Rationale:

The following references from the NDcPPv2.2e explicitly allow importing of keys (including asymmetric keys), and do not mandate that imported keys (public or private) must have been generated by the TOE:

- *FAU_GEN.1 ("Generating/import of… cryptographic keys);*
- *FMT_MTD.1/CryptoKeys, Application Note 119 ("… included if cryptographic keys can be managed (… imported) by the Security Administrator.");*
- *FMT_SMF.1 ("Ability to manage the cryptographic keys")*
- *FMT_SMF.1 ("Ability to import X.509v3 certificates…")*
- *SFR Dependencies Analysis, relevant to FCS_CKM.2, FCS_CKM.4, and FCS_COP.1/* ("FTP_ITC.1 as a secure channel that could be used for import");*

For distributed TOEs, the resolution is modifying the wording of Table 1 footnote 4 to be consistent with other uses of asymmetric keys within this PP.

Further Action:

None.

Action by Network iTC:

None.