KINK issue list update

http://www.taca.jp/kink/kink-issue-list.txt

IETF-62 KINK WG
KAMADA Ken'ichi <Ken-ichi.Kamada@jp.yokogawa.com>
Yokogawa Electric Corporation

Issue list

- Discussions are done at the mailing list.
- Almost all issues were discussed and have proposed solutions.
- You can see the current list and the progress at
  - http://www.taca.jp/kink/kink-issue-list.txt

Categories of issues

- Issues are categorized to 8 groups in the following slides.
  - Clarifications related to kcrypto
  - U2U (and cross-realm) issues
  - Other Kerberos matters
  - Handshake clarifications
  - Error handling
  - IANA considerations
  - Other clarifications
  - Editorial issues

Clarifications related to kcrypto

Align KINK crypto operations with kcrypto.

- Checksum (#8, #9, #10, #11)
  - Proposal:
    - Use required-to-implement checksum types corresponding to the keys' etypes.
    - Omit the checksum field (not zeroing out) in calculating the checksum.
- Encryption is not decomposable (#20)
  - Proposal:
    - use the whole output of the kcrypto encryption as an opaque octet string.
- Key usage numbers (#28)
  - Proposal:
    - Get Two key usages numbers (KINK_ENCRYPT and checksum) from 1510ter.
- prf (#25, #26)
  - Proposal:
    - Use kcrypto prf to generate IPsec keys.

Resolved issues: #8, #9, #10, #11, #20, #25, #26, and #28

U2U (and cross-realm) issues

- Modify GETTGT scenario (#3, #19, #44)
  - Proposal:
    - Send the responder's principal name when retrieving TGT. (KINK_TGT_REQ/KINK_TGT_REP format change.)
    - The responder returns its non-cross-realm TGT.
    - The KDC authenticates whether the TGT was issued to the expected responder.
- How to detect an U2U peer rebooted (#7)
  - Proposal:
    - When an U2U responder rebooted and got a new TGT, it can't decrypt tickets using the old TGT. In this case, let the responder return its new TGT in KINK_TGT_REP, then the problem is resolved and the usual DPD mechanism will work.
- Other comments on U2U have not been cleared (#2)
  - Comments:
    - more examination needed on a situations where it might *not* be two PKINIT clients.
    - over-specifying things on U2U.

Resolved issues: #3, #19, #44, and #7

Still remain: #2 (U2U)

Other Kerberos matters

- Checksum when returning KRB-ERROR (#17)
  - 1510ter has checksum on KRB-ERROR but not yet been standardized.
  - Proposal:
    - Use KINK checksum.
- Kerberos error type limitation (#18)
  - Proposal:
    - Removing the limitation of error codes which the responder can return.
- Subsession keys (#12)
  - Do we use only base key, or allow to use subkey?
  - Comments:
    - There are already ISAKMP nonces, so more entropy from subkeys buy us nothing (so don't use subkey).
    - Being the same as everyone else is preferred if we have no reason (so use subkey).
    - If we allow subkeys, we need to describe what key is used where.

Resolved issues: #17 and #18
Still remain: #12 (Subsession keys)

Handshake clarifications

- KE exchange and 3-way handshake interoperability (#45, #23)
  - Proposal:
    - Add texts about SA installation timing when KE payloads are used.
    - Add texts about the usage of ACKREQ flag when KE payloads are used.
- Describe how to reject KE payload (#23)
  - Proposal:
    - Return an ISAKMP error (NO-PROPOSAL-CHOSEN or INVALID-KEY-INFORMATION) when the responder doesn't want to do KE exchange.
- What keys are used for the resulting SA on the each side? (#37)
  - Comment:
    - Need a review after the change of section 8.

Resolved issues: #45 and #23

Remains (Waiting review for a revision of section 8): #37

Error handling

- Clarify the error handling of the version number mismatch and unknown payload types. (#1)
  - Proposal:
    - KINK minor version brings no worth things to KINK so remove it.
    - Return KINK_PROTOERR on unknown KINK payloads.
    - (unknown QM version is already described in the section 12)
    - (unknown ISAKMP payload is described in RFC 2408)
- Need more words for the each error type (ISAKMP and KINK_ERROR) like IKEv2. (#31)
  - Proposal:
    - Describe when these errors are generated.
    - Describe how the initiator should act on these errors.

Resolved issues: #1 and #31

IANA considerations

IANA suggestions

- We need to decide which values are IANA matters.
  - Proposal:
    - KINK port number
    - KINK message types
    - KINK payload types
    - KINK_ERROR error codes
- We need to decide which values are assigned from existing registries, and which values need new registries.
  - Proposal:
    - The port number is to be assigned.
    - Request new registries for other values.

Resolved issues: #33

Other clarifications (1/2)

- How to get peer's principal name: why not store a principal instead of a hostname? (#15)
  - Proposal:
    - From where/How to get peer's principal name is an implementation matter, not necessarily generated from a FQDN. E.g. principal names may be stored in the PAD. (clarifications on the text may be needed to avoid misunderstandings.)
- EPOCH format ambiguity (#16)
  - Proposal:
    - Describe the semantics of the "4-octet" value more concretely.
- Text on PFS support (#22)
  - Proposal:
    - Remove the reasoning that Kerberos doesn't provide PFS so KINK doesn't need it.
    - Not to mandate PFS is ok.

Other clarifications (2/2)

- SPD Considerations (#27)
  - Proposal:
    - Move this consideration to the outside of the Security Considerations section.
    - Clarify matters on SPD and PAD using 2401bis words.
- Rekey description (#29)
  - Proposal:
    - Refine it with 2401bis words.
- IKEv2 or not? (#30)
  - Proposal:
    - Go with 2401bis but not IKEv2.

Resolved issues: #15, #16, #22, #27, #29, and #30

Editorial issues

- Typos
- Terminology
- Wording
- Ambiguity
- References

Resolved issues: many

Remaining issues

- Remaining non-editorial issues are:
  - #2 some U2U comments
  - #12 Subsession keys
  - #37 What keys are used for the resulting SA on the each side?
- Comments are welcome on the mailing list.

#2 some U2U comments

- Raeburn> More examination of user-to-user case, especially situations where it might *not* be two PKINIT clients, which section 3 says is possible.
- Raeburn> In the user-to-user case with TGTs, I think the KINK draft may be over-specifying things that should be dealt with at the Kerberos level. If things are underspecified in Kerberos Clarifications, let’s deal with that.

#12 Subsession keys

- Thomas> more entropy from subsession keys buy us almost nothing (ISAKMP NONCE is enough). There are people who have been writing to *this* spec for several years now.
- Sommerfeld> The session key is long-lived. (but it’s not really all that different from a per-exchange nonce.)
- Hartman> Being the same as everyone else is preferred if we have no reason.

#37 What keys are used for the resulting SA on the each side?

- Hartman> I'm somewhat concerned that 4.3 is not specific enough to describe exactly what key gets set up. I.E. I'm concerned it may not be detailed enough for interoperable implementations.
- I understand section 8's purpose is to answer this issue, but I did not find it clear. I believe that section 8 is going to need to change to specify use of the kcrypto prf.
- I think that once this change is made I should either say that the result is clear or explain exactly what I think is missing. I suspect I may want some text copied in from the IKE RFC.