Lightweight AKE for OSCORE Requirements

draft-ietf-lake-reqs-00

LAKE Virtual Interim, January 16, 2020

Status

- Latest version:
  draft-ietf-lake-reqs-00
- Minor updates on the LAKE Github:
  https://github.com/lake-wg/reqs

Github Issues

#1 Omit signature based protocol?

Question:

- **Why do we need a signature based mode?**
  - I.e. why isn't static DH sufficient?
  - Would reduce 50% of message overhead

Potential answer:

- Static DH currently not widely deployed. Enrolment protocol specified (RFC 6955) but not implemented.

- Conclusion: No, the AKE needs to support both signature and static DH public keys. (Implementations that support static DH need only use that.)

#2 Terminology of data chunks

Terminology used in the draft:

- AKE protocol units: “messages”
- Radio layer units: “frames” (6TiSCH), ”packets” (LoRaWAN)

- Number of frames/packets has performance impact
- Minimize the number of radio layer units
  - the size is dependent on technology, regulations, configuration, etc.
- The AKE needs to be transported over CoAP which has its own fragmentation ("blocks")

- **Do we need other terminology for data chunks?**

#3 Resumption

- OSCORE Appendix B.2
  - generates a new security context from an existing security context
  - based on client- and server-provided nonces
  - does not provide PFS

- The AKE should support a procedure for generating a new security context with PFS from a previous authenticated key exchange between the same endpoints.

- **Special resumption procedure or not?**

- Proposal: Reuse PSK authenticated mode of the AKE (Section 2.2) using a dedicated PSK derived after a previous AKE run.

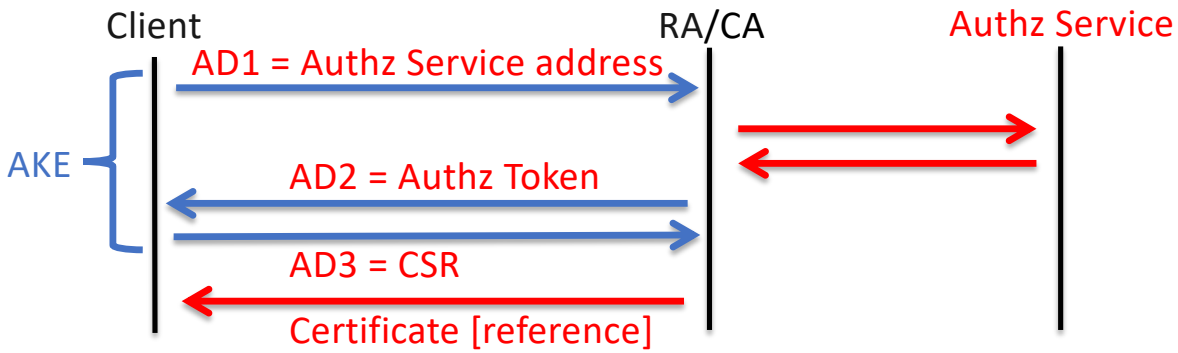
#4 Key separation

General: What key separation properties do we need?

— Different AKE messages and OSCORE messages should all use separate keys
— Keys used for resumption (issue #3) should be separate
— …

Particular: Does the ”Application Data” in the AKE need separate keys?

— This would add overhead to the AKE
— Used to transport authorization and certificate related information



#5 PQC formulation

Section 2.4 states

"PAKE and post-quantum key exchange is out of scope, but may be supported in a later version."

— Do we need another/different statement about PQC?

#6 Listing of specific attacks

Section 2.3 lists specific attacks, such as:

"The AKE shall provide Key Compromise Impersonation (KCI) resistance”

"The AKE shall protect against reflection attacks"

Is the high level description in section 2.3 sufficient?

#7 Extensibility vs. complexity

- The current version of the AKE does not target PAKE or PQC
- We want to allow future extensions
- Some extensibility is already built-in through COSE, for example
  - New algorithms
  - New certificate formats
  - New schemes for identifying and transporting credentials

Section 2.7 speaks of extensibility and adds a caveat:

"Since the main objective with this work is to create a simple yet secure AKE, care needs to be taken to avoid feature creep and extensions working against this."

Do we need a better formulation?

#8 Strength of the handshake integrity check

— What integrity do we require for the AKE?

— (See last item of https://mailarchive.ietf.org/arch/msg/lake/5iyqSkVEfp5rpxB2GFNAmFJUU1A)

#9 AKE vs OSCORE properties

Section 2.4: "The AKE shall support different AEAD/MAC algorithms for AKE and OSCORE".

— Only one example of relation of security properties between AKE and OSCORE. OSCORE needs AEAD, HKDF, Master Secret and Connection IDs. Any other related security properties to list as a requirement?

#10 Negotiation of AKE mode

- Assuming the AKE need to support signature keys or static DH
- Proposal: The AKE shall support negotiation of type of authentication credentials