Lightweight AKE for OSCORE Requirements

draft-ietf-lake-reqs-00

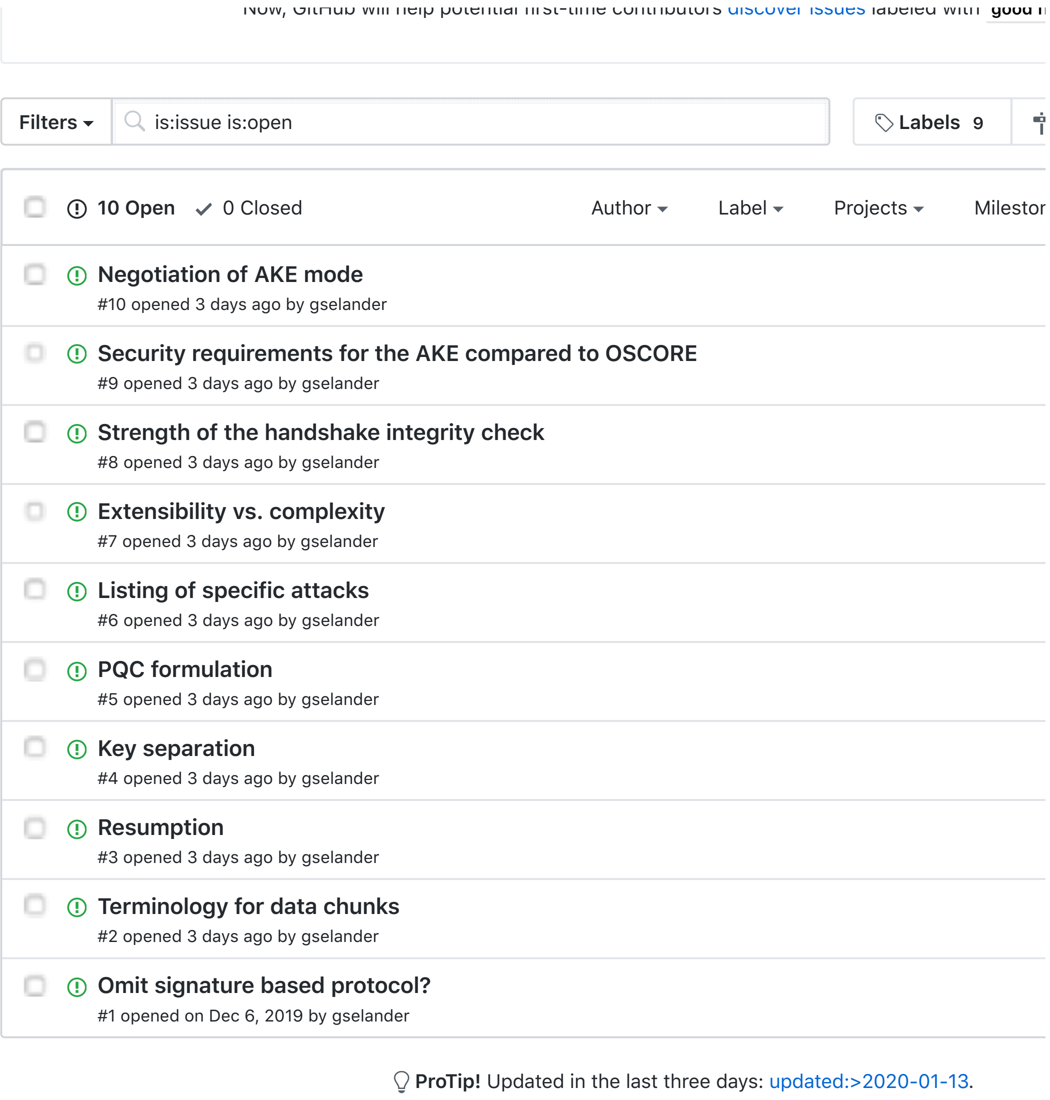
LAKE Virtual Interim, January 16, 2020

Background

- LAKE is about specifying a lightweight authenticated key exchange protocol for OSCORE (RFC 8613)

- Requirements draft
  - Latest submitted version: https://tools.ietf.org/html/draft-ietf-lake-reqs-00
  - Reviews by Ekr, Ivaylo, Karthik
  - Updates on the LAKE Github: https://github.com/lake-wg/reqs

Github Issues



Now, GitHub will help potential first-time contributors discover issues labeled with good fi

10 Open ✓ 0 Closed

□ Negotiation of AKE mode
#10 opened 3 days ago by gselander

□ Security requirements for the AKE compared to OSCORE
#9 opened 3 days ago by gselander

□ Strength of the handshake integrity check
#8 opened 3 days ago by gselander

□ Extensibility vs. complexity
#7 opened 3 days ago by gselander

□ Listing of specific attacks
#6 opened 3 days ago by gselander

□ PQC formulation
#5 opened 3 days ago by gselander

□ Key separation
#4 opened 3 days ago by gselander

□ Resumption
#3 opened 3 days ago by gselander

□ Terminology for data chunks
#2 opened 3 days ago by gselander

□ Omit signature based protocol?
#1 opened on Dec 6, 2019 by gselander

ProTip! Updated in the last three days: updated:>2020-01-13.

https://github.com/lake-wg/reqs/issues

#1 Omit signature based protocol?

Section 2.2
”Multiple public key authentication credential types may need to be supported for RPK and certificate-based authentication. In case of a Diffie-Hellman key exchange both the use of signature based public keys (for compatibility with existing ecosystem) and static DH public keys (for reduced message size) is expected.”

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.)

No need for new text

#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.
  - E.g. Initial public key run of AKE followed by symmetric key runs for resumption
  - Simpler than special resumption procedure

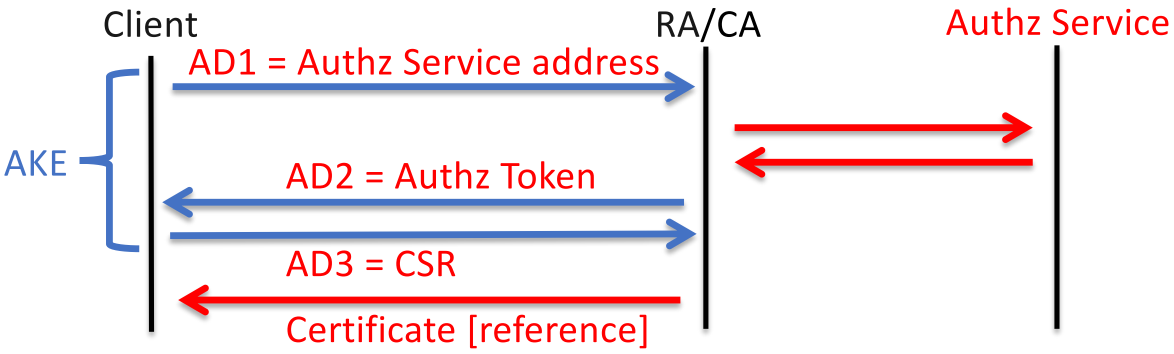
#4 Key separation

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
- Planned applications use application data to transport authorization and certificate related information
  - Why would this data need separate keys?



#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:

”The mutual authentication guarantees of the AKE shall guarantee the following properties:

— The AKE shall provide Key Compromise Impersonation (KCI) resistance.

— The AKE shall protect against identity misbinding attacks, when applicable. Note that the identity may be directly related to a public key such as for example the public key itself, a hash of the public key, or data unrelated to a key.

— The AKE shall protect against reflection attacks, but need not protect against attacks when more than two parties legitimately share keys (cf. the Selfie attack on TLS 1.3) as that setting is out of scope.”

Is this description 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."

Better formulation?

#8 Strength of the handshake integrity check

- **What integrity do we require for the AKE? Trade off with overhead.**

- Ekr: ”it's important that the AKE be covered by a full strength integrity check. There are a number of ways to do this depending on the overall architecture of the system.”
  - https://mailarchive.ietf.org/arch/msg/lake/5iyqSkVEfp5rpxB2GFNAmFJUU1A

#9 AKE vs OSCORE properties

What are the security requirements for the AKE as compared to the transport OSCORE?

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

- **Should we highlight other related security properties?**
  - OSCORE needs AEAD, HKDF, Master Secret and Connection IDs.

#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.