LAKE requirements

draft-ietf-lake-reqs-02
WLGC updates

LAKE Virtual Interim, March 31, 2020

Status

- 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-02
  - WGLC reviews by Christopher Wood, Richard Barnes, Eric Rescorla, Hannes Tschofenig
  - Responses and updates posted March 24
  - Github branch "WGLC-updates": https://github.com/lake-wg/reqs/tree/WGLC-updates
  - Updated comments from Ekr March 31

WGLC updates - main changes

A note from our chairs:

"As a reminder - our charter calls for us to park this draft after WGLC has successfully completed, and we'll not be sending this on for publication as an RFC at this time. So please try keep the focus of your comments on the meat of the content rather than on crossing all the i's and dotting all the t's :-)"

2.1 AKE for OSCORE

COSE: shall -> recommended

— "The AKE shall specify how it provides COSE algorithms to OSCORE. It is strongly recommended that COSE is reused by the AKE, for identification of credentials and algorithms, as extension point for new schemes, and to avoid duplicated implementation of crypto wrapper."

Other formulations with COSE

— "The AKE shall support negotiation of all COSE algorithms [IANA-COSE-Algorithms] to be used in OSCORE. The AKE shall support negotiation of algorithms used in the AKE. It is strongly recommended that the AKE algorithms are identified using [IANA-COSE-Algorithms] to reduce unnecessary complexity of a combined OSCORE/AKE implementation."

— "the AKE must support different schemes for transporting and identifying credentials, including x5t, x5u and x5chain (see Section 2 of [I-D.ietf-cose-x509])"

Certificates in order is relevant for constrained implementations. x5bag/cross-signatures needed?

2.2 Credentials

New formulation of negotiation of public key credential mix

— "The AKE shall support negotiation of public key credential mix and that both initiator and responder can verify the variant that was executed."

2.3 Mutual Authentication

Comment on session identifier:

— "At the end of the AKE protocol, each endpoint shall have authenticated the other's credential. In particular, both endpoints must agree on a fresh session identifier, and the roles and credentials of both endpoints, both endpoints must agree on a fresh session identifier, and the roles and credentials of both endpoints."

Ekr: What are the properties of this identifier? Is this a CK-style session identifier? I think this needs to be a lot clearer, given that you also use "connection identifier".

[Note: "connection identifier" is now removed from draft]

2.3 Mutual Authentication

Comment about identity misbinding:

— The AKE shall protect against identity misbinding attacks [Misbinding]. 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.

Ekr: It's very different to have the key pointed at by reference and have the endpoint identified by the key, and they have different binding properties.

2.3 Mutual Authentication

New formulation on replay protection

— "Replayed messages shall not affect the security of an AKE session.”

2.3 Mutual Authentication

Comment on verifying identity

— "Furthermore, the endpoints shall be able to verify that the identity of the other endpoint is an acceptable identity that it is intended to authenticate to."

Ekr: Can you give me an example of something that is an AKE that doesn't otherwise have this property?

GS: A protocol that accepts the other party as long as the provided public key verifies the signature, e.g., in the "opportunisitic" setting, but in that case it is intentional.

2.4 Confidentiality

The shared secret established by the AKE must be known only to the two authenticated endpoints.

A passive network attacker should never learn any session keys, even if it knows both endpoints' long-term keys. An active attacker who has compromised the initiator or responder credential shall still not be able to compute past session keys (Perfect Forward Secrecy, PFS). These properties can be achieved, e.g., with an ephemeral Diffie-Hellman key exchange.

PFS may also be achieved in other ways, for example, using hash-based ratcheting or with a nonce exchange followed by appropriately derived new session keys provided that state can be kept in the form of a session counter. Note that OSCORE specifies a method for session key update involving a nonce exchange (see Appendix B in [RFC8613]).

The AKE shall provide a mechanism to use the output of one handshake to optimize future handshakes, e.g., by generating keying material which can be used to authenticate a future handshake, thus avoiding the need for public key authentication in that handshake.

The AKE should give recommendations for frequency of re-keying potentially dependent on the amount of data.

Ekr: where do session tickets fit in?

2.5 Crypto Agility and Negotiation Integrity

”hybrid” AKE out of scope

— "hybrid" (simultaneously more than one) key exchange is out of scope, but may be supported in a later version.

Clarification of need for negotiation

— "The protocol shall allow negotiation of elliptic curves for Diffie-Hellman operations and signature-based authentication”

2.5 Crypto Agility and Negotiation Integrity

Comment on negotiation:

— "A successful negotiation shall result in the most preferred algorithms of one of the parties which are supported by the other."

Ekr: This is overly specific, and I don't see why it's a requirement. The way you want to phrase this is that it resists downgrade.

GS: A protocol that always results in the least preferred common algorithm would comply with "resist downgrade"

Ekr: TLS doesn't guarantee highest preferred algorithm. It seems like the definition people have come to is about integrity of negotiation not about "best preferred" https://eprint.iacr.org/2016/072.pdf.

GS: If best preferred is not required we can remove this, section already talks about negotiation integrity:

— "The AKE negotiation must provide strong integrity guarantees against active attackers. At the end of the AKE protocol, both endpoints must agree on both the crypto algorithms that were proposed and those that were chosen. In particular, the protocol must protect against downgrade attacks."

2.5 Crypto Agility and Negotiation Integrity

Comment on strong integrity:

— "The AKE negotiation must provide strong integrity guarantees against active attackers. At the end of the AKE protocol, both endpoints must agree on both the crypto algorithms that were proposed and those that were chosen. In particular, the protocol must protect against downgrade attacks."

Ekr: Would be good to quantify "strong". I suggest >= 128 bits.

GS: Discussed in connection with the BoF at IETF 105 and decided at the time not to do. Curve25519 may be characterized as < 128 bits. MACs shorter than 128 bits may be acceptable.

Ekr: 25519 thing is easy to handle by either just saying 127 bits or just saying that we consider 25519 to be about 128 bits. As for the MAC, the requirement I am talking about is the strength of the protocol, not the size of the MAC, and I think we should specify that.

2.6 Identity Protection

Clarification of identity protection properties of SIGMA-I and SIGMA-R

— ”In the case of public key identities, the AKE is required to protect the identity of one of the peers against active attackers and the identity of the other peer against passive attackers. SIGMA-I and SIGMA-R differ in this respect. SIGMA-I protects the identity of the initiator against active attackers and the identity of the responder against passive attackers. For SIGMA-R, the properties of the roles are reversed at the cost of an additional flight.”

New text on other identifying information

— "Other identifying information may also need to be transported in plain text, for example, identifiers to allow correlation between AKE messages, and cipher suites. Mechanisms to encrypt these kind of parameters, such as using pre-configured public keys typically adds to message overhead.”

“Connection identifier” removed from the draft.

2.7 Auxiliary Data

Removed mentioning of access token

Exemplified use of CSR

— For example, the auxiliary data in the first two messages of the AKE may transport authorization related information as in [I-D.selander-ace-ake-authz] followed by a Certificate Signing Request (CSR) in the auxiliary data of the third message.

New formulation of protection of auxiliary data

— "The auxiliary data must be protected to the same level as AKE data in the same flight."

2.8 Extensibility

— “While remaining extensible, the AKE should avoid optional mechanisms which introduce code paths that are less well tested.”

Ekr: I don't think this is a requirement that is actionable.

Mcr: I basically don't want us to include extension mechanisms that we don't know how we will use. For example, the minor version number if IKEv1 and IKEv2 is meaningless. Nobody knows when we would increment it, or what would it mean, so we don’t use it. Thus, the code path for minor_version != 0 is never executed.

2.8 Extensibility

— “The AKE should avoid mechanisms where an initiator takes a guess at the policy, and when it receives a negative response, must guess, based upon what it has tried, what to do next.

Ekr: There are good reasons to do this. In fact, it's the main way to get a three message protocol with confidentiality for the server’s first flight.
Mcr: If the initiator guesses wrong, then the responder should say what they support. Contrast IKEv1 "aggressive mode".

2.9 Denial of Service

Removed 1st paragraph (overlapped with DoS of underlying transport)

Title of section changed to "Availability"

2.10 Lightweight

Clarifications of benchmarks

Collecting information about benchmarks in section 2.10.4

- "Considering that an AKE protocol complying with these requirements is expected to have at least 3 messages, the optimal AKE has 3 messages and each message fits into as few frames as possible, ideally 1 frame per message. The target message sizes for minimal but realistic applications of PSK and RPK should be such that fragmentation can be avoided. For the case of certificate based authentication it may not be possible to transport certificates in the AKE with the minimal number of frames.

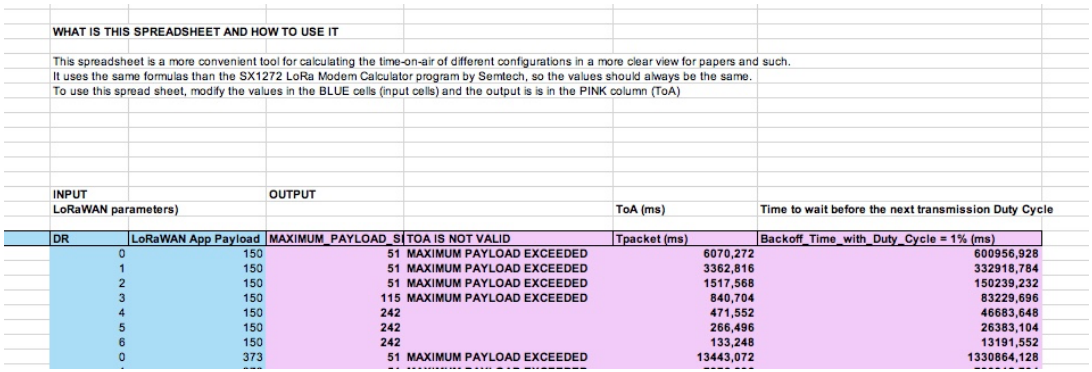
  For the LoRaWAN benchmark, the limit for fragmentation is 51 bytes at link layer. For the 6TiSCH benchmark, messages less than or equal to 45 bytes at CoAP payload layer need not be fragmented."

- **This text is intended to show what is expected to be achievable for the mentioned technologies.**
  - **RPK with static DH keys are feasible**

Benchmarks

- Secdispatch interim meeting March 2019 presented benchmarks e.g. LoRaWAN time-on-air, backoff time, etc.
- This provided calculations of cost at various sizes (packets, flights, etc.).
- Spreadsheets are available.
  - Enter message sizes and it calculates time-on-air.
  - The .xls contains a "how to"

https://github.com/EricssonResearch/EDHOC/blob/master/docs/LoRaWAN_ToA.xlsx



WHAT IS THIS SPREADSHEET AND HOW TO USE IT

This spreadsheet is a more convenient tool for calculating the time-on-air of different configurations in a more clear view for papers and such.
It uses the same formulas than the SX1272 LoRa Modem Calculator program by Semtech, so the values should always be the same.
To use this spread sheet, modify the values in the BLUE cells (input cells) and the output is is in the PINK column (ToA)

INPUT LoRaWAN parameters)		OUTPUT		ToA (ms)	Time to wait before the next transmission Duty Cycle
DR	LoRaWAN App Payload	MAXIMUM_PAYLOAD_SI	TOA IS NOT VALID	Tpacket (ms)	Backoff_Time_with_Duty_Cycle = 1% (ms)
0	150	51	MAXIMUM PAYLOAD EXCEEDED	6070,272	600956,928
1	150	51	MAXIMUM PAYLOAD EXCEEDED	3362,816	332918,784
2	150	51	MAXIMUM PAYLOAD EXCEEDED	1517,568	150239,232
3	150	115	MAXIMUM PAYLOAD EXCEEDED	840,704	83229,696
4	150	242		471,552	46683,648
5	150	242		266,496	26383,104
6	150	242		133,248	13191,552
0	373	51	MAXIMUM PAYLOAD EXCEEDED	13443,072	1330864,128

LoRaWAN Time-on-Air and Backoff Time Estimates

Assumption: SF12 (DR0)
Fragmentation into 51 byte packets, neglecting additional headers

— **PSK ECDHE**:

Slide from presentation at Secdispatch interim March 05, 2019

— **RPK ECHDE**:

Numbers are not up-to-date

Time-on-Air (s)

PSK ECHDE	EDHOC-12	DTLS 1.3
Flight1	2.6	10.7
Flight2	2.6	10.7
Flight3	1.5	4.1
Total	**6.7 s**	**25.5 s**

RPK ECDHE	EDHOC-12	DTLS 1.3
Flight1	2.5	8.4
Flight2	7.1	21.2
Flight3	4.9	12.7
Total	**14.5 s**	**42.2 s**

Duty Cycle backoff time estimates (min)

PSK ECHDE	EDHOC-12	DTLS 1.3
Flight1	4.3*)	13.8
Flight2	0*)	13.8
Flight3	0*)	4.6
Total	**4.3 min**	**32.3 min**

RPK ECDHE	EDHOC-12	DTLS 1.3
Flight1	0*)	9.2
Flight2	8.7	32.3
Flight3	4.3	18.4
Total	**13.0 min**	**59.9 min**

*) Since no fragmentation, the duty cycle overlaps with waiting for the next message

Next steps

— Resolve remaining issues on Github
— Submit a new version