Requirements for a Lightweight AKE for OSCORE

Mališa Vučinić, Göran Selander, John Mattsson, Dan Garcia-Carrillo

Network Working Group   M. Vucinic
Internet-Draft   Inria
Intended status: Informational   G. Selander
Expires: 10 December 2020   J. Mattsson
                                                           Ericsson AB
                                                             D. Garcia
                                                     Odin Solutions S.L.
                                                           8 June 2020

                  Requirements for a Lightweight AKE for OSCORE
                          draft-ietf-lake-reqs-04

Abstract

   This document compiles the requirements for a lightweight
   authenticated key exchange protocol for OSCORE. This draft has
   completed a working group last call (WGLC) in the LAKE working group.
   Post-WGLC, the requirements are considered sufficiently stable for
   the working group to proceed with its work. It is not currently
   planned to publish this draft as an RFC.

Status of This Memo

   This Internet-Draft is submitted in full conformance with the
   provisions of BCP 78 and BCP 79.

   Internet-Drafts are working documents of the Internet Engineering
   Task Force (IETF). Note that other groups may also distribute
   working documents as Internet-Drafts. The list of current Internet-
   Drafts is at https://datatracker.ietf.org/drafts/current/.

   Internet-Drafts are draft documents valid for a maximum of six months
   and may be updated, replaced, or obsoleted by other documents at any
   time. It is inappropriate to use Internet-Drafts as reference
   material or to cite them other than as "work in progress."

   This Internet-Draft will expire on 10 December 2020.

Copyright Notice

and restrictions with respect to this document. Code Components
extracted from this document must include Simplified BSD License text
as described in Section 4.e of the Trust Legal Provisions and are
provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction

   OSCORE [RFC8613] is a lightweight communication security protocol
   providing end-to-end security on application layer for constrained
   IoT settings (cf. [RFC7228]). OSCORE lacks a matching authenticated
   key exchange protocol (AKE). The intention with the LAKE WG
   [LAKE-WG] is to create a simple yet secure AKE for implementation in
   embedded devices supporting OSCORE.

   To ensure that the AKE is efficient for the expected applications of
   OSCORE, we list the relevant public specifications of technologies
   where OSCORE is included:

*   The IETF 6TiSCH WG charter identifies the need to "secur[e] the
    join process and mak[e] that fit within the constraints of high
    latency, low throughput and small frame sizes that characterize
    IEEE802.15.4 TSCH". OSCORE protects the join protocol as
    described in 6TiSCH Minimal Security
    [I-D.ietf-6tisch-minimal-security].

*   The IETF LPWAN WG charter identifies the need to improve the
    transport capabilities of LPWA networks such as NB-IoT and LoRa
    whose "common traits include ... frame sizes ... [on] the order of
    tens of bytes transmitted a few times per day at ultra-low
    speeds". The application of OSCORE is described in
    [I-D.ietf-lpwan-coap-static-context-hc].

*   OMA Specworks LwM2M version 1.1 [LwM2M] defines bindings to two
    challenging radio technologies where OSCORE is planned to be
    deployed: LoRaWAN and NB-IoT.

*   Open Connectivity Foundation (OCF) plans to use OSCORE for end-to-
    end security of unicast messages [OCF].

This document compiles the requirements for the AKE for OSCORE. It
summarizes the security requirements that are expected from such an
AKE, as well as the main characteristics of the environments where
the solution is envisioned to be deployed. The solution will
presumably be useful in other scenarios as well since a low security
overhead improves the overall performance.

2. Problem description

2.1. AKE for OSCORE

The rationale for designing this protocol is that OSCORE is lacking a
matching AKE. OSCORE was designed for lightweight RESTful operations
for example by minimizing the overhead, and applying the protection
to the application layer, thereby limiting the data being encrypted
and integrity protected for the other endpoint. Moreover, OSCORE was
tailored for use with lightweight primitives that are likely to be
implemented in the device, specifically CoAP [RFC7252], CBOR
[RFC7049] and COSE [RFC8152]. The same properties should apply to
the AKE.

In order to be suitable for OSCORE, at the end of the AKE protocol
run the two parties must agree on (see Section 3.2 of [RFC8613]):

* A shared secret (OSCORE Master Secret) with Perfect Forward
  Secrecy (PFS, see Section 2.4) and a good amount of randomness.
  (The term "good amount of randomness" is borrowed from [HKDF] to
  signify not necessarily uniformly distributed randomness.)

* OSCORE Sender IDs of peer endpoints, arbitrarily short.

  - Sender IDs are expected to be unique for a given Master Secret,
    more precisely the quartet (Master Secret, Master Salt, ID
    Context, Sender ID) must be unique, see Section 3.3. of
    [RFC8613].

* COSE algorithms to use with OSCORE

COSE provides the crypto primitives for OSCORE. 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.

The AKE cannot rely on messages being exchanged in both directions
after the AKE has completed, because CoAP/OSCORE requests may not
have a response [RFC7967]. Furthermore, there is no assumption of
dependence between CoAP client/server and AKE initiator/responder
roles, and an OSCORE context may be used with CoAP client and server
roles interchanged as is done, for example, in [LwM2M].

Moreover, the AKE must support transport over CoAP. When transported
over CoAP, the AKE must support the traversal of CoAP intermediaries,
as required by the 6TiSCH network formation setting
[I-D.ietf-6tisch-minimal-security].

Since the AKE messages most commonly will be encapsulated in CoAP,
the AKE must not duplicate functionality provided by CoAP, or at
least not duplicate functionality in such a way that it adds non-
negligible extra costs in terms of code size, code maintenance, etc.
It is therefore assumed that the AKE is being transported in a
protocol that provides reliable transport, that can preserve packet
ordering and handle message duplication [RFC7252], that can perform
fragmentation [RFC7959] and protect against denial of service attacks
as provided by the CoAP Echo option [I-D.ietf-core-echo-request-tag].

The AKE may use other transport than CoAP. In this case the
underlying layers must correspondingly handle message loss,
reordering, message duplication, fragmentation, and denial of service
protection.

2.2. Credentials

   IoT deployments differ from one another in terms of what credentials
   can be supported. Currently many systems use pre-shared keys (PSKs)
   provisioned out of band, for various reasons. PSKs are sometimes
   used in a first deployment because of their perceived simplicity.
   The use of PSKs allows for protection of communication without major
   additional security processing, and also enables the use of symmetric
   crypto algorithms only, reducing the implementation and computational
   effort in the endpoints.

   However, PSK-based provisioning has inherent weaknesses. There has
   been reports of massive breaches of PSK provisioning systems
   [massive-breach], and as many systems use PSKs without Perfect
   Forward Secrecy (PFS, see Section 2.4) they are vulnerable to passive
   pervasive monitoring. The security of these systems can be improved
   by adding PFS through an AKE authenticated by the provisioned PSK.

   Shared keys can alternatively be established in the endpoints using
   an AKE protocol authenticated with asymmetric public keys instead of
   symmetric secret keys. Raw public keys (RPK) can be provisioned with
   the same scheme as PSKs, which allows for a more relaxed trust model
   since RPKs need not be secret. The corresponding private keys are
   assumed to be provisioned to the party being authenticated beforehand
   (e.g. in factory or generated on-board).

   As a third option, by using a public key infrastructure and running
   an asymmetric key AKE with public key certificates instead of RPKs,
   key provisioning can be omitted, leading to a more automated ("zero-
   touch") bootstrapping procedure. The root CA keys are assumed to be
   provisioned beforehand. Public key certificates are important for
   several IoT settings, e.g., facility management with a large number
   of devices from many different manufacturers.

   These steps provide an example of a migration path in limited scoped
   steps from simple to more robust security bootstrapping and
   provisioning schemes where each step improves the overall security
   and/or simplicity of deployment of the IoT system, although not all
   steps are necessarily feasible for the most constrained settings.

   In order to allow for these different schemes, the AKE must support
   PSK- (shared between two nodes), RPK- and certificate-based
   authentication. These are also the schemes for which CoAP is
   designed (see Section 9 of [RFC7252]).

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.

To further minimize the bandwidth consumption it is required to
support transporting certificates and raw public keys by reference
rather than by value. Considering the wide variety of deployments,
the AKE must support different schemes for transporting and
identifying credentials. While there are many existing mechanisms
for doing so, ranging from PSK to raw public key by reference to
x5chain of in-band certificates [I-D.ietf-cose-x509], what is
appropriate for a given deployment will depend on the nature of that
deployment. In order to provide a clear initial effort,
Section 2.2.1 lists a set of credential types of immediate relevance;
the mechanism for selecting credential scheme is presumed to enable
future extensibility if needed.

The use of RPKs may be appropriate for the authentication of the AKE
initiator but not for the AKE responder. The AKE must support
different credentials for authentication in different directions of
the AKE run, e.g. certificate-based authentication for the initiating
endpoint and RPK-based authentication for the responding endpoint.

Assuming that both signature public keys and static DH public keys
are in use, then also the case of mixed credentials need to be
supported with one endpoint using a static DH public key and the
other using a signature public key. The AKE shall support
negotiation of public key credential mix and that both initiator and
responder can verify the variant that was executed.

2.2.1. Initial Focus

As illustrated above, the setting is much more diverse in terms of
credentials and trust anchors than that of the unconstrained web. In
order to deliver a timely result, there is a need to initially focus
on what is considered most important at the time of writing: RPK (by
reference and value) and certificate by reference. Information about
validity of a certificate may be omitted from the AKE if available
over unconstrained links. The case of transporting certificate
validation information over the AKE may be specified in the initial
phase if there is a lightweight solution that matches existing
standards and tools.

A subsequent extension beyond the initial focus may be inevitable to
maintain a homogenous deployment without having to implement a mix of
AKE protocols, for example, to support the migration path described

above. The AKE needs to make clear the scope of cases analysed in
the initial phase, and that a new analysis is required for additional
cases.

The initial scope as described in this subsection does not cover all
credentials as detailed previously in Section 2.2: an AKE which is
extensible but does not include PSK ECDHE would be conformant with
the requirements for the initial scope. A solution to the
requirements for the initial scope is intended to be a deliverable of
the LAKE WG.

2.3. Mutual Authentication

The AKE must provide mutual authentication during the protocol run.
At the end of the AKE protocol, each endpoint shall have freshly
authenticated the other's credential. In particular, both endpoints
must agree on a fresh session identifier, and the roles and
credentials of both endpoints.

Since the protocol may be initiated by different endpoints, it shall
not be necessary to determine beforehand which endpoint takes the
role of initiator of the AKE.

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

* The AKE shall provide Key Compromise Impersonation (KCI)
   resistance [KCI].

* 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.

* 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 [Selfie]) as that
   setting is out of scope.

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

As often is the case, it is expected that an AKE fulfilling these
goals would have at least three flights of messages (with each flight
potentially consisting of one or more messages, depending on the AKE
design and the mapping to OSCORE).

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.

   To mitigate against bad random number generators the AKE shall
   provide recommendations for randomness, for example to use
   [I-D.irtf-cfrg-randomness-improvements].

2.5. Cryptographic Agility and Negotiation Integrity

   Motivated by long deployment lifetimes, the AKE is required to
   support cryptographic agility, including the modularity of COSE
   crypto algorithms and negotiation of preferred crypto algorithms for
   OSCORE and the AKE.

   * The protocol shall support both pre-shared key and asymmetric key
      authentication. PAKE, post-quantum and "hybrid" (simultaneously
      more than one) key exchange is out of scope, but may be supported
      in a later version.

   * The protocol shall allow negotiation of elliptic curves for
      Diffie-Hellman operations and signature-based authentication.

* 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.

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

* The AKE may choose different sets of symmetric crypto algorithms
   (AEAD, MAC, etc.) for AKE and for OSCORE. In particular, the
   length of the MAC for the AKE may be required to be larger than
   for OSCORE.

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.

2.6. Cryptographic Strength

The AKE shall establish a key with a target security level
[keylength] of >= 127 bits. This level was chosen to include X25519
and applies to the strength of authentication, the established keys,
and the protection for the negotiation of all cryptographic
parameters.

2.7. Identity Protection

In general, it is necessary to transport identities as part of the
AKE run in order to provide authentication of an entity not
identified beforehand. In the case of constrained devices, the
identity may contain sensitive information on the manufacturer of the
device, the batch, default firmware version, etc. Protecting
identifying information from passive and active attacks is important
from a privacy point of view, but needs to be balanced with the other
requirements, including security and lightweightness.

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.

It is not required to protect the PSK identifier, and it may thus be sent in the first flight. Protection of PSK identifier in many cases require extra flights of the AKE.

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.

2.8. Auxiliary Data

In order to reduce round trips and the number of flights, and in some cases also streamline processing, certain security features may be integrated into the AKE by transporting "auxiliary data" together with the AKE messages.

One example is the transport of third-party authorization information from initiator to responder or vice versa. Such a scheme could enable the party receiving the authorization information to make a decision about whether the party being authenticated is also authorized before the protocol is completed, and if not then discontinue the protocol before it is complete, thereby saving time, message processing and data transmission.

Another, orthogonal, example is the embedding of a certificate enrolment request or a newly issued certificate in the AKE.

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.

The AKE must support the transport of such auxiliary data together with the protocol messages. The auxiliary data field must not contain data that violates the AKE security properties. The auxiliary data field must only be used with security analysed protocols.

The auxiliary data may contain privacy sensitive information. The auxiliary data must be protected to the same level as AKE data in the same flight. For example, for a SIGMA-I AKE it is expected that the 3 flights will provide the following protection of the auxiliary data:

* Auxiliary data in the first flight is unprotected

   * Auxiliary data in the second flight is confidentiality protected
      against passive attackers and integrity protected against active
      attackers

   * Auxiliary data in the third flight is confidentiality and
      integrity protected against active attackers

2.9. Extensibility

   It is desirable that the AKE supports some kind of extensibility, in
   particular, the ability to later include new AKE modes such as PAKE
   support. COSE provides an extension mechanism for new algorithms,
   new certificate formats, ways to identify credentials, etc.

   The main objective with this work is to create a simple yet secure
   AKE. The AKE should avoid having multiple ways to express the same
   thing. If the underlying encodings offered by CBOR offer multiple
   possibility the AKE should be strongly opinionated, and clearly
   specify which one will be used.

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

   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.

2.10. Availability

   Jamming attacks, cutting cables etc. leading to long term loss of
   availability may not be possible to mitigate, but an attacker
   temporarily injecting messages or disturbing the communication shall
   not have a similar impact.

2.11. Lightweight

   We target an AKE which is efficiently deployable in 6TiSCH multi-hop
   networks, LoRaWAN networks and NB-IoT networks. (For an overview of
   low-power wide area networks, see e.g. [RFC8376].) The desire is to
   optimize the AKE to be 'as lightweight as reasonably achievable' in
   these environments, where 'lightweight' refers to:

   * resource consumption, measured by bytes on the wire, wall-clock
      time and number of round trips to complete, or power consumption

   * the amount of new code required on end systems which already have
      an OSCORE stack

These properties need to be considered in the context of the use of
an existing CoAP/OSCORE stack in the targeted networks and
technologies. Some properties are difficult to evaluate for a given
protocol, for example, because they depend on the radio conditions or
other simultaneous network traffic. Additionally, these properties
are not independent. Therefore the properties listed here should be
taken as input for identifying plausible protocol metrics that can be
more easily measured and compared between protocols.

Per 'bytes on the wire', it is desirable for the AKE messages to fit
into the MTU size of these protocols; and if not possible, within as
few frames as possible, since using multiple MTUs can have
significant costs in terms of time and power. Note that the MTU size
depends on radio technology and its characteristics, including data
rates, number of hops, etc. Example benchmarks are given further
down in this section.

Per 'time', it is desirable for the AKE message exchange(s) to
complete in a reasonable amount of time, both for a single
uncongested exchange and when multiple exchanges are running in an
interleaved fashion, like e.g. in a "network formation" setting when
multiple devices connect for the first time. This latency may not be
a linear function depending on congestion and the specific radio
technology used. As these are relatively low data rate networks, the
latency contribution due to computation is in general not expected to
be dominant.

Per 'round-trips', it is desirable that the number of completed
request/response message exchanges required before the initiating
endpoint can start sending protected traffic data is as small as
possible, since this reduces completion time. See Section 2.11.4 for
a discussion about the trade-off between message size and number of
flights.

Per 'power', it is desirable for the transmission of AKE messages and crypto to draw as little power as possible. The best mechanism for doing so differs across radio technologies. For example, NB-IoT uses licensed spectrum and thus can transmit at higher power to improve coverage, making the transmitted byte count relatively more important than for other radio technologies. In other cases, the radio transmitter will be active for a full MTU frame regardless of how much of the frame is occupied by message content, which makes the byte count less sensitive for the power consumption as long as it fits into the MTU frame. The power consumption thus increases with AKE message size and the largest impact is on average under poor network conditions. Note that listening for messages to receive can in many cases be a large contribution to the power consumption, for which there are separate techniques to handle, e.g., time slots, discontinuous reception, etc. but this is not considered in scope of the AKE design.

Per 'new code', it is desirable to introduce as little new code as possible onto OSCORE-enabled devices to support this new AKE. These devices have on the order of 10s of kB of memory and 100 kB of storage on which an embedded OS; a COAP stack; CORE and AKE libraries; and target applications would run. It is expected that the majority of this space is available for actual application logic, as opposed to the support libraries. In a typical OSCORE implementation COSE encrypt and signature structures will be available, as will support for COSE algorithms relevant for IoT enabling the same algorithms as is used for OSCORE (e.g. COSE algorithm no. 10 = CCM* used by 6TiSCH). The use of those, or CBOR or CoAP, would not add to the footprint.

While the large variety of settings and capabilities of the devices and networks makes it challenging to produce exact values of some these dimensions, there are some key benchmarks that are tractable for security protocol engineering and which have a significant impact.

2.11.1. LoRaWAN

Reflecting deployment reality as of now, we focus on the European regulation as described in ETSI EN 300 220. LoRaWAN employs unlicensed radio frequency bands in the 868 MHz ISM band. For LoRaWAN the most relevant metric is the Time-on-Air, which determines the period before the next communication can occur and also which can be used as an indicator to calculate energy consumption. LoRaWAN is legally required to use a duty cycle with values such as 0.1%, 1% and 10% depending on the sub-band that is being used, leading to a payload split into fragments interleaved with unavailable times. For Europe, the duty cycle is 1% (or smaller). Although there are

exceptions from the use of duty cycle, the use of an AKE for
providing end-to-end security on application layer needs to comply
with the duty cycle.

2.11.1.1. Bytes on the wire

LoRaWAN has a variable MTU depending on the Spreading Factor (SF).
The higher the spreading factor, the higher distances can be achieved
and/or better reception. If the coverage and distance allows it,
with SF7 - corresponding to higher data rates - the maximum payload
is 222 bytes. For a SF12 - and low data rates - the maximum payload
is 51 bytes on data link layer.

The size and number of packets impact the Time-on-Air (ToA). The
benchmark used here is based on SF12 and a packet size of 51 bytes
[LoRaWAN]. The use of larger packets depend on good radio conditions
which are not always present. Some libraries/providers only support
51-bytes packet size.

2.11.1.2. Time

The time it takes to send a message over the air in LoRaWAN can be
calculated as a function of the different parameters of the
communication. These are the Spreading Factor (SF), the message
size, the channel, bandwidth, coding rate, etc. An important feature
of LoRaWAN is the duty cycle limitation due to the use of the ISM
band. The duty cycle is evaluated in a 1-hour sliding window. It is
legal for a device to transmit a burst for a total of up to 36
seconds ToA on a 1%-duty-cyle sub-band, but the device must then
pause the transmission for the rest of the hour [lorawan-duty-cycle].
In order to avoid extreme waiting times, the AKE needs to complete
before the duty cycle limit is exhausted, also taking into account
potential retransmissions and allowing additional air time for lower
level MAC frames and application data. As a challenging but
realistic example we assume each message is retransmitted 2 times and
allow a factor 2-3 for additional air time. With these assumptions
it is required with a ToA of 4-6 seconds for the uplink protocol
messages to ensure that the entire burst stays within the 36 seconds
duty cycle.

It should be noted that some libraries/providers enforce the duty
cycle limitation through a stop-and-wait operation, which restricts
the number of bytes to the size of the packets after which duty cycle
waiting times are incurred.

2.11.1.3. Round trips and number of flights

   Considering the duty cycle of LoRaWAN and associated unavailable
   times, the round trips and number of LoRaWAN packets needs to be
   reduced as much as possible.

2.11.1.4. Power

   The calculation of the power consumption in LoRaWAN is dependent on
   several factors, such as the spreading factor used and the length of
   the messages sent, both having a clear dependency with the time it
   takes to transmit the messages. The communication model (inherent to
   the different LoRaWAN classes of devices) also has an impact on the
   energy consumption, but overall the Time-on-Air is an important
   indication of the performance.

2.11.2. 6TiSCH

   6TiSCH operates in the 2.4 GHz unlicensed frequency band and uses
   hybrid Time Division/Frequency Division multiple access (TDMA/FDMA).
   Nodes in a 6TiSCH network form a mesh. The basic unit of
   communication, a cell, is uniquely defined by its time and frequency
   offset in the communication schedule matrix. Cells can be assigned
   for communication to a pair of nodes in the mesh and so be collision-
   free, or shared by multiple nodes, for example during network
   formation. In case of shared cells, some collision-resolution scheme
   such as slotted-Aloha is employed. Nodes exchange frames which are
   at most 127-bytes long, including the link-layer headers. To
   preserve energy, the schedule is typically computed in such a way
   that nodes switch on their radio below 1% of the time ("radio duty
   cycle"). A 6TiSCH mesh can be several hops deep. In typical use
   cases considered by the 6TiSCH working group, a network that is 2-4
   hops deep is commonplace; a network which is more than 8 hops deep is
   not common.

2.11.2.1. Bytes on the wire

   Increasing the number of bytes on the wire in a protocol message has
   an important effect on the 6TiSCH network in case the fragmentation
   is triggered. More fragments contribute to congestion of shared
   cells (and concomitant error rates) in a non-linear way.

   The available size for key exchange messages depends on the topology
   of the network, whether the message is traveling uplink or downlink,
   and other stack parameters. A key performance indicator for a 6TiSCH
   network is "network formation", i.e. the time it takes from switching
   on all devices, until the last device has executed the AKE and
   securely joined. As a benchmark, given the size limit on the frames

and taking into account the different headers (including link-layer security), for a 6TiSCH network 5 hops deep, the maximum CoAP payload size to avoid fragmentation is 47/45 bytes (uplink/downlink) [AKE-for-6TiSCH].

2.11.2.2. Time

Given the slotted nature of 6TiSCH, the number of bytes in a frame has insignificant impact on latency, but the number of frames has. The relevant metric for studying AKE is the network formation time, which implies parallel AKE runs among nodes that are attempting to join the network. Network formation time directly affects the time installers need to spend on site at deployment time.

2.11.2.3. Round trips and number of flights

Given the mesh nature of the 6TiSCH network, and given that each message may travel several hops before reaching its destination, it is highly desirable to minimize the number of round trips to reduce latency.

2.11.2.4. Power

From the power consumption point of view, it is more favorable to send a small number of large frames than a larger number of short frames.

2.11.3. NB-IoT

3GPP has specified Narrow-Band IoT (NB-IoT) for support of infrequent data transmission via user plane and via control plane. NB-IoT is built on cellular licensed spectrum at low data rates for the purpose of supporting:

* operations in extreme coverage conditions,

* device battery life of 10 years or more,

* low device complexity and cost, and

* a high system capacity of millions of connected devices per square kilometer.

NB-IoT achieves these design objectives by:

* Reduced baseband processing, memory and RF enabling low complexity device implementation.

   * A lightweight setup minimizing control signaling overhead to
      optimize power consumption.

   * In-band, guard-band, and stand-alone deployment enabling efficient
      use of spectrum and network infrastructure.

2.11.3.1. Bytes on the wire

   The number of bytes on the wire in a protocol message has a direct
   effect on the performance for NB-IoT. In contrast to LoRaWAN and
   6TiSCH, the NB-IoT radio bearers are not characterized by a fixed
   sized PDU. Concatenation, segmentation and reassembly are part of
   the service provided by the NB-IoT radio layer. As a consequence,
   the byte count has a measurable impact on time and energy consumption
   for running the AKE.

2.11.3.2. Time

   Coverage significantly impacts the available bit rate and thereby the
   time for transmitting a message, and there is also a difference
   between downlink and uplink transmissions (see Section 2.11.3.4).
   The transmission time for a message is essentially proportional to
   the number of bytes.

   Since NB-IoT is operating in licensed spectrum, in contrast to e.g.
   LoRaWAN, the packets on the radio interface can be transmitted back-
   to-back, so the time before sending OSCORE protected data is limited
   by the number of round trips/flights of the AKE and not by a duty
   cycle.

2.11.3.3. Round trips and number of flights

   As indicated in Section 2.11.3.2, the number of frames and round-
   trips is one limiting factor for protocol completion time.

2.11.3.4. Power

   Since NB-IoT is operating in licensed spectrum, the device is allowed
   to transmit at a relatively high power, which has a large impact on
   the energy consumption.

   The benchmark for NB-IoT energy consumption is based on the same
   computational model as was used by 3GPP in the design of this radio
   layer [NB-IoT-battery-life-evaluation]. The device power consumption
   is assumed to be 500mW for transmission and 80mW for reception.
   Power consumption for "light sleep" (~ 3mW) and "deep sleep" (~
   0.015mW) are negligible in comparison. The bitrates (uplink/
   downlink) are assumed to be 28/170 kbps for good coverage and
   0,37/2,5 kbps for bad coverage.

   The results [AKE-for-NB-IoT] show a high per-byte energy consumption
   for uplink transmissions, in particular in bad coverage. Given that
   the application decides about the device being initiator or responder
   in the AKE, the protocol cannot be tailored for a particular message
   being uplink or downlink. To perform well in both kind of
   applications the overall number of bytes of the protocol needs to be
   as low as possible.

2.11.4. Discussion and Summary of Benchmarks

   The difference between uplink and downlink performance must not be
   engineered into the protocol since it cannot be assumed that a
   particular protocol message will be sent uplink or downlink.

   For NB-IoT the byte count on the wire has a measurable impact on time
   and energy consumption for running the AKE, so the number of bytes in
   the messages needs to be as low as possible.

   While "as small protocol messages as possible" does not lend itself
   to a sharp boundary threshold, "as few flights as possible" does and
   is relevant in all settings above.

   The penalty is high for not fitting into the frame sizes of 6TiSCH
   and LoRaWAN networks. Fragmentation is not defined within these
   technologies so requires fragmentation scheme on a higher layer in
   the stack. With fragmentation increases the number of frames per
   message, each with its associated overhead in terms of power
   consumption and latency. Additionally the probability for errors
   increases, which leads to retransmissions of frames or entire
   messages that in turn increases the power consumption and latency.

   There are trade-offs between "few messages" and "few frames"; if
   overhead is spread out over more messages such that each message fits
   into a particular frame this may reduce the overall power
   consumption. For example, with a frame size of 50 bytes, two 60-byte
   messages will fragment into 4 frames in total, whereas three 40-byte
   messages fragment into 3 frames in total. On the other hand, a
   smaller message has less probability to collide with other messages
   and incur retransmission.

While it may be possible to engineer such a solution for a particular
radio technology and AKE protocol, optimizing for a specific scenario
may not be optimal for other settings. It is expected that specific
scenarios are evaluated in the design phase to ensure that the AKE is
fit for purpose. But in order to start the design work some general
criteria for the AKE performance need to be formulated that takes
into account the differences in the expected deployments.

There are benefits in terms of fewer flights/round trips for NB-IoT
(Section 2.11.3.3) and 6TiSCH (Section 2.11.2.3). An AKE protocol
complying with the requirements of this memo is expected to have at
least 3 messages. With a 3-message AKE, the initiator is able to
derive the OSCORE security context after receiving message 2,
rendering the AKE essentially one round trip before traffic data can
be exchanged, which is ideal.

If the AKE has 3 messages then optimal performance for 6TiSCH is when
each message fits into as few frames as possible, ideally 1 frame per
message.

For LoRaWAN, optimal performance is determined by the duty cycle
which puts a limit to ToA or, for certain libraries/providers, the
number of packets (see Section 2.11.1.2). If the AKE has 3 messages
and each message fits into a 51 byte packet then this is optimal for
the latter case. The same assumption incurs a ToA for uplink
messages in the interval of 4-6 seconds at SF12 both for a device-
initiated and infrastructure-initiated AKE, which complies with the
challenging example stated in Section 2.11.1.2.

One avenue to good performance is therefore to target message sizes
which avoids fragmentation or with as few fragments as possible. 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.

For the initial focus cases (Section 2.2.1), i.e. RPK (by reference
and value) and certificate by reference, it is required that the AKE
shall perform optimally with respect to the available criteria for
the radio technologies.

To determine with certainty what are the minimal number of fragments
for an AKE under different assumptions requires to design and analyse
the AKE, which is clearly beyond the requirements phase. However, by
means of an example we have reason to believe that an AKE with 3
messages can be designed to support RPK by reference in 3 fragments.
Thus the ideal number of fragments is expected for RPK by reference.

While such performance may not be possible for the other initial
focus cases, it is expected that if one of the peers send RPK by
value or certificate by reference, then one additional fragment is
sufficient, thus in total a maximum of 5 fragments. Alternatively,
for the LoRaWAN challenge (Section 2.11.1.2), it is expected that the
duty cycle for a burst can be complied with for RPK by value and
certificate by reference, assuming that each message only needs to be
retransmitted at most once (i.e. good AKE performance for RPK by
value and certificate by reference in not too poor radio
environments).

2.11.5. AKE frequency

One question that has been asked in the context of lightweightness
is: - How often is the AKE executed? While it may be impossible to
give a precise answer there are other perspectives to this question.

1. For some use cases, already one execution of the AKE is heavy,
    for example, because

    * there are a number of parallel executions of the AKE which
       loads down the network, such as in a network formation
       setting, or

    * the duty cycle makes the completion time long for even one run
       of the protocol.

2. If a device reboots it may not be able to recover the security
    context, e.g. due to lack of persistent storage, and is required
    to establish a new security context for which an AKE is
    preferred. Reboot frequency may be difficult to predict in
    general.

3. To limit the impact of a key compromise, BSI, NIST and ANSSI and
    other organizations recommend in other contexts frequent renewal
    of keys by means of Diffie-Hellman key exchange. This may be a
    symmetric key authenticated key exchange, where the symmetric key
    is obtained from a previous asymmetric key based run of the AKE.

To summarize, even if it we are unable to give precise numbers for
AKE frequency, a lightweight AKE:

* reduces the time for network formation and AKE runs in challenging
   radio technologies,

* allows devices to quickly re-establish security in case of
   reboots, and

   * enables support for recommendations of frequent key renewal.

3. Security Considerations

   This document compiles the requirements for an AKE and provides some
   related security considerations.

   The AKE must provide the security properties expected of IETF
   protocols, e.g., providing mutual authentication, confidentiality,
   and negotiation integrity as is further detailed in the requirements.

4. Privacy Considerations

   In the privacy properties for the AKE, the transport over CoAP needs
   to be considered.

5. IANA Considerations

   None.

Acknowledgments

   The authors want to thank Richard Barnes, Dominique Barthel, Karthik
   Bhargavan, Stephen Farrell, Ivaylo Petrov, Eric Rescorla, Michael
   Richardson, Jesus Sanchez-Gomez, Claes Tidestav, Hannes Tschofenig
   and Christopher Wood for providing valuable input.

Informative References

   [RFC7228] Bormann, C., Ersue, M., and A. Keranen, "Terminology for
              Constrained-Node Networks", RFC 7228,
              DOI 10.17487/RFC7228, May 2014,
              <https://www.rfc-editor.org/info/rfc7228>.

   [RFC7049] Bormann, C. and P. Hoffman, "Concise Binary Object
              Representation (CBOR)", RFC 7049, DOI 10.17487/RFC7049,
              October 2013, <https://www.rfc-editor.org/info/rfc7049>.

   [RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained
              Application Protocol (CoAP)", RFC 7252,
              DOI 10.17487/RFC7252, June 2014,
              <https://www.rfc-editor.org/info/rfc7252>.

   [RFC7959] Bormann, C. and Z. Shelby, Ed., "Block-Wise Transfers in
              the Constrained Application Protocol (CoAP)", RFC 7959,
              DOI 10.17487/RFC7959, August 2016,
              <https://www.rfc-editor.org/info/rfc7959>.

   [RFC7967] Bhattacharyya, A., Bandyopadhyay, S., Pal, A., and T.
              Bose, "Constrained Application Protocol (CoAP) Option for
              No Server Response", RFC 7967, DOI 10.17487/RFC7967,
              August 2016, <https://www.rfc-editor.org/info/rfc7967>.

   [RFC8152] Schaad, J., "CBOR Object Signing and Encryption (COSE)",
              RFC 8152, DOI 10.17487/RFC8152, July 2017,
              <https://www.rfc-editor.org/info/rfc8152>.

   [RFC8613] Selander, G., Mattsson, J., Palombini, F., and L. Seitz,
              "Object Security for Constrained RESTful Environments
              (OSCORE)", RFC 8613, DOI 10.17487/RFC8613, July 2019,
              <https://www.rfc-editor.org/info/rfc8613>.

   [RFC8376] Farrell, S., Ed., "Low-Power Wide Area Network (LPWAN)
              Overview", RFC 8376, DOI 10.17487/RFC8376, May 2018,
              <https://www.rfc-editor.org/info/rfc8376>.

   [I-D.ietf-6tisch-minimal-security]
              Vucinic, M., Simon, J., Pister, K., and M. Richardson,
              "Constrained Join Protocol (CoJP) for 6TiSCH", Work in
              Progress, Internet-Draft, draft-ietf-6tisch-minimal-
              security-15, 10 December 2019, <http://www.ietf.org/
              internet-drafts/draft-ietf-6tisch-minimal-security-
              15.txt>.

   [I-D.ietf-lpwan-coap-static-context-hc]
              Minaburo, A., Toutain, L., and R. Andreasen, "LPWAN Static
              Context Header Compression (SCHC) for CoAP", Work in
              Progress, Internet-Draft, draft-ietf-lpwan-coap-static-
              context-hc-14, 26 May 2020, <http://www.ietf.org/internet-
              drafts/draft-ietf-lpwan-coap-static-context-hc-14.txt>.

   [I-D.ietf-cose-x509]
              Schaad, J., "CBOR Object Signing and Encryption (COSE):
              Header parameters for carrying and referencing X.509
              certificates", Work in Progress, Internet-Draft, draft-
              ietf-cose-x509-06, 9 March 2020, <http://www.ietf.org/
              internet-drafts/draft-ietf-cose-x509-06.txt>.

   [I-D.ietf-core-echo-request-tag]
              Amsuess, C., Mattsson, J., and G. Selander, "CoAP: Echo,
              Request-Tag, and Token Processing", Work in Progress,
              Internet-Draft, draft-ietf-core-echo-request-tag-09, 9
              March 2020, <http://www.ietf.org/internet-drafts/draft-
              ietf-core-echo-request-tag-09.txt>.

   [I-D.irtf-cfrg-randomness-improvements]
              Cremers, C., Garratt, L., Smyshlyaev, S., Sullivan, N.,
              and C. Wood, "Randomness Improvements for Security
              Protocols", Work in Progress, Internet-Draft, draft-irtf-
              cfrg-randomness-improvements-12, 5 May 2020,
              <http://www.ietf.org/internet-drafts/draft-irtf-cfrg-
              randomness-improvements-12.txt>.

   [I-D.selander-ace-ake-authz]
              Selander, G., Mattsson, J., Vucinic, M., Richardson, M.,
              and A. Schellenbaum, "Lightweight Authorization for
              Authenticated Key Exchange.", Work in Progress, Internet-
              Draft, draft-selander-ace-ake-authz-01, 9 March 2020,
              <http://www.ietf.org/internet-drafts/draft-selander-ace-
              ake-authz-01.txt>.

   [AKE-for-6TiSCH]
              "AKE for 6TiSCH", March 2019,
              <https://docs.google.com/document/
              d/1wLoIexMLG3U9iYO5hzGzKjkvi-VDndQBbYRNsMUlh-k>.

   [AKE-for-NB-IoT]
              "AKE for NB-IoT", March 2019,
              <https://github.com/EricssonResearch/EDHOC/blob/master/
              docs/NB%20IoT%20power%20consumption.xlsx>.

   [NB-IoT-battery-life-evaluation]
              "On mMTC, NB-IoT and eMTC battery life evaluation",
              January 2017,
              <http://www.3gpp.org/ftp/tsg_ran/WG1_RL1/TSGR1_AH/
              NR_AH_1701/Docs//R1-1701044.zip>.

   [HKDF]   Krawczyk, H., "Cryptographic Extraction and Key
              Derivation: The HKDF Scheme", May 2010,
              <https://eprint.iacr.org/2010/264.pdf>.

   [IANA-COSE-Algorithms]
              "COSE Algorithms", March 2020,
              <https://www.iana.org/assignments/cose/
              cose.xhtml#algorithms>.

   [LwM2M]   "OMA SpecWorks LwM2M", August 2018,
              <https://www.openmobilealliance.org/release/
              LightweightM2M/V1_1-20180710-A/OMA-TS-
              LightweightM2M_Transport-V1_1-20180710-A.pdf>.

   [OCF]   "OSCORE:OCF Status and Comments", March 2020,
               <https://github.com/t2trg/2020-03-ocf-
               oscore/blob/master/slides/Joint-OCF-IRTF-T2TRG-call-on-
               OSCORE-20200318.pdf>.

   [LoRaWAN]   "LoRaWAN Regional Parameters v1.0.2rB", February 2017,
               <https://lora-alliance.org/resource-hub/lorawantm-
               regional-parameters-v102rb>.

   [LAKE-WG]   "LAKE WG", March 2020,
               <https://datatracker.ietf.org/wg/lake/about/>.

   [KCI]   Hlauschek, C., Gruber, M., Fankhauser, F., and C. Schanes,
               "Prying open Pandoras box:KCI attacks against TLS", August
               2015,
               <https://www.usenix.org/system/files/conference/woot15/
               woot15-paper-hlauschek.pdf>.

   [Misbinding]
               Sethi, M., Peltonen, A., and T. Aura, "Misbinding Attacks
               on Secure Device Pairing and Bootstrapping", Proceedings
               of the 2019 ACM Asia Conference on Computer and
               Communications Security , May 2019,
               <https://arxiv.org/pdf/1902.07550.pdf>.

   [Selfie]   Drucker, N. and S. Gueron, "Selfie:Reflections on TLS 1.3
               with PSK", March 2019, <https://eprint.iacr.org/2019/347>.

   [massive-breach]
               "Sim card database hack gave US and UK spies access to
               billions of cellphones", February 2015,
               <https://www.theguardian.com/us-news/2015/feb/19/nsa-gchq-
               sim-card-billions-cellphones-hacking>.

   [lorawan-duty-cycle]
               Saelens, M., Hoebeke, J., Shahid, A., and E. De Poorter,
               "Impact of EU duty cycle and transmission power
               limitations for sub-GHz LPWAN SRDs an overview and future
               challenges. EURASIP Journal on Wireless Communications and
               Networking. 2019. 10.1186/s13638-019-1502-5.", 2019,
               <https://jwcn-
               eurasipjournals.springeropen.com/articles/10.1186/
               s13638-019-1502-5>.

   [keylength]
              Lenstra, A., "Key Lengths:Contribution to The Handbook of
              Information Security", 2018,
              <https://infoscience.epfl.ch/record/164539/files/NPDF-
              32.pdf>.

Authors' Addresses

   Malisa Vucinic
   Inria

   Email: malisa.vucinic@inria.fr

   Goeran Selander
   Ericsson AB

   Email: goran.selander@ericsson.com

   John Preuss Mattsson
   Ericsson AB

   Email: john.mattsson@ericsson.com

   Dan Garcia-Carrillo
   Odin Solutions S.L.

   Email: dgarcia@odins.es