Network Working Group   M. Vucinic
Internet-Draft   Inria
Intended status: Informational   G. Selander
Expires: June 8, 2020   J. Mattsson
                                                            Ericsson AB
                                                               D. Garcia
                                                       Odin Solutions S.L.
                                                       December 06, 2019

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

Abstract

   This document compiles the requirements for a lightweight
   authenticated key exchange protocol for OSCORE.

Status of This Memo

Copyright Notice

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 LAKE 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:

   o The IETF 6TiSCH WG charter (-02) 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].

   o The IETF LPWAN WG charter (-01) 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].

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

   Other industry fora which plan to use OSCORE:

   o Fairhair Alliance has defined an architecture [Fairhair] which
      adopts OSCORE for multicast, but it is not clear whether the
      architecture will support unicast OSCORE.

   o Open Connectivity Foundation (OCF) has been actively involved in
      the OSCORE development for the purpose of deploying OSCORE, but no
      public reference is available since OCF only references RFCs. We
      believe that these OSCORE consumers reflect similar levels of
      constraints on the devices and networks in question.

   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, CBOR and COSE. The
   same properties must 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]):

   o a shared secret (OSCORE Master Secret) with PFS (see Section 2.3)
      and a good amount of randomness. (The term "good amount of
      randomness" is borrowed from [HKDF] to signify not necessarily
      uniformly distributed randomness.)

   o OSCORE Sender IDs of peer endpoints, arbitrarily short

   o COSE algorithms to use with OSCORE

   COSE provides the crypto primitives for OSCORE, and shall therefore
   be used also by the AKE, for several reasons including maintenance of
   crypto library. COSE provides identification of credentials and
   algorithms for OSCORE and the AKE, and an extension point for new
   schemes.

   Moreover, the AKE must support transport over CoAP. 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 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,
   that can perform fragmentation and protect against denial of service
   attacks, such 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 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 often 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, and as
   many systems use PSKs without perfect forward secrecy (PFS) 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.

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 of the Diffie-Hellman (DH) key exchange.

Bandwidth is a scarce resource in constrained-node networks. The use of static DH public keys instead of signature public keys is a significant optimization and shall be supported.

To further minimize the bandwidth consumption it is required to support transporting the certificates by reference rather than by value. Considering the wide variety of deployments the AKE must support different schemes for transporting and identifying credentials, including those identified in Section 2 of [I-D.ietf-cose-x509].

The common lack of a user interface in constrained devices leads to various credential provisioning schemes. 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.

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 authenticated the other.

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 e.g. in [LwM2M]. 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.

Compromise of initiator or responder long-term keys shall not enable
an attacker to compromise past session keys (Perfect Forward Secrecy)
and shall not enable a passive attacker to compromise future session
keys. These two properties can be achieved with an ephemeral Diffie-
Hellman key exchange.

To mitigate against bad random number generators the AKE shall
mandate randomness improvements such as
[I-D.irtf-cfrg-randomness-improvements] and analogously for symmetric
keys.

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

The AKE shall protect against replay attacks (injective).

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. 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.

2.4. Crypto Agility and Security Properties

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

o The protocol shall support both pre-shared key and asymmetric key
   authentication. PAKE and post-quantum key exchange is out of
   scope, but may be supported in a later version.

o The protocol shall allow multiple elliptic curves for asymmetric
  keys

o The AKE shall support negotiation of the all COSE algorithms used
  in the AKE and that OSCORE supports. A successful negotiation
  shall result in the most preferred algorithms of one of the
  parties which are supported by the other.

o The AKE shall support different AEAD/MAC algorithms for AKE and
  OSCORE

The AKE negotiation must be protected against downgrade attacks.
[Further detailing is requested.]

2.5. 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. For certain
data we therefore need to make an exemption in order to obtain an
efficient protocol.

The AKE is required to protect the identity against active attackers
of one of the peers and protection against passive attackers of the
other peer in the case of public key identities.

In case of a PSK identifier, this may be protected against passive
attackers with a key derived from the Diffie-Hellman shared secret.
The responder has first access to the shared secret but does in
general not know from whom a message without PSK identifier is sent.
Therefore the protection of PSK identifier in general needs to be
performed by the initiator, i.e. at the earliest in message 3. As a
consequence, in order to authenticate the responder within the AKE,
at least four protocol messages are needed in case of symmetric key
authentication with identity protection. Considering the need to
keep the number of messages at a minimum (see Section 2.9.4), unless
there are other good reasons for having more than 3 messages, it is
not required to protect the PSK identifier, and it may thus be sent
in the first message.

Other identifying information that needs to be transported in plain
text is cipher suites and connection identifiers. Encrypting crypto
algorithms does not allow negotiation of cipher suite within 3

messages. Encryption of connection identifiers only works in
asymmetric case and does not enable arbitrarily short identifiers
(see Section 2.1).

2.6. Application Data

   In order to reduce round trips and number of messages, and in some
   cases also streamline processing, certain applications may want to
   transport application data within the AKE.

   One example is the transport of third-party signed authorization
   information such as an access token or a voucher 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 discontinue the protocol before it
   is complete, thereby saving time and message processing.

   Another example is the embedding of certificate enrolment request or
   a newly issued certificate.

   The AKE must support the transport of application data within the
   protocol messages.

   It is expected that an AKE with 3 messages will provide the following
   protection of the application data:

   o Application data in the first message is unprotected

   o Application data in the second message is confidentiality
      protected against passive attackers and integrity protected
      against active attackers

   o Application data in the third message is confidentiality and
      integrity protected against active attackers

   Application data may contain privacy sensitive information. The
   application data must not violate the AKE security properties. The
   assumptions on the application data need to be detailed in the
   specification of the AKE.

2.7. 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. Note that by supporting COSE, the AKE can already support
   new algorithms, new certificate formats, ways to identify
   credentials, etc.

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.

2.8. Denial of Service

The AKE shall protect against denial of service attacks on responder
and initiator to the extent that the protocol supports lightweight
deployments (see Section 2.9) and without duplicating the DoS
mitigation of the underlying transport (see Section 2.1).

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.

2.9. Lightweight

We target an AKE which is efficiently deployable in 6TiSCH multi-hop
networks, LoRaWAN networks and NB-IoT networks. The desire is to
optimize the AKE to be 'as lightweight as reasonably achievable' in
these environments, where 'lightweight' refers to:

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

o 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 2.9.4 for
a discussion about the tradeoff between message size and number of
messages.

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. Increased power
consumption is unavoidable in poor network conditions, such as most
wide-area settings including LoRaWAN.

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.

2.9.1. LoRaWAN

   LoRaWAN employs unlicensed radio frequency bands in the 868 MHz ISM
   band. As a case in point, we focus here on deployment in Europe,
   where this is regulated by ETSI EN 300 220. For LoRaWAN the most
   relevant metric is the Time-on-Air, which determines the back-off
   times and 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 back-off 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.

2.9.1.1. 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. LoRaWAN has a header size of 13 bytes, to
   which we have to add the maximum recommended payload depending on the
   SF used. 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.

   The benchmark used here is Data Rates 0-2 corresponding to a packet
   size of 51 bytes [LoRaWAN]. The use of larger frame size depend on
   good radio conditions which are not always present. Some libraries/
   providers only support 51-bytes packet size.

2.9.1.2. 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. A duty cycle of 1% implies that the time to complete a
   fragmentation of the payload increases by at least 10,000%. This
   limitation determines how long time the device will have to wait for
   next use, which encourages the reduction of the message size as much
   as possible.

2.9.1.3. Round trips and number of messages

   Considering the duty cycle of LoRaWAN and associated back-off times,
   the round trips and number of messages needs to be reduced as much as
   possible.

2.9.1.4. 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 message sent, both having a clear dependency with the time it
   takes to transmit the message. 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.

2.9.2. 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.

2.9.2.1. 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 an example, given the size limit on the frames

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

2.9.2.2. 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.

2.9.2.3. Round trips and number of messages

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.

2.9.2.4. 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.

2.9.3. 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:

o operations in extreme coverage conditions,

o device battery life of 10 years or more,

o low device complexity and cost, and

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

NB-IoT achieves these design objectives by:

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

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

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

2.9.3.1. 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.

2.9.3.2. 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 2.9.3.4). The
transmission time for the 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/messages of the AKE and not by a duty
cycle.

2.9.3.3. Round trips and number of messages

As indicated in Section 2.9.3.2, the number of messages and round-
trips is one limiting factor for protocol completion time.

2.9.3.4. 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.

2.9.4. Discussion

While "as small protocol messages as possible" does not lend itself
to a sharp boundary threshold, "as few protocol messages 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. While it may be possible to engineer such a solution
for a particular radio technology and signature algorithm, the
benefits in terms of fewer messages/round trips in general and for
NB-IoT in particular (see Section 2.9.3) are considered more
important than optimizing for a specific scenario. Hence an optimal
AKE protocol has 3 messages and each message fits into as few frames
as possible, ideally 1 frame per message.

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

2.9.5. 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

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

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

o   enables support for recommendations of frequent key renewal

3. Requirements Summary

o   The AKE must support PSK, RPK and certificate based authentication
    with PFS and crypto agility for AKE as well as OSCORE, be 3-pass
    and support transport over CoAP. It is required to support
    different schemes for transporting and identifying credentials.

o   After the AKE run, the peers must be mutually authenticated, agree
    on a shared secret with PFS and good amount of randomness, peer
    identifiers (potentially short), and COSE algorithms to use.

o   The AKE must reuse CBOR, CoAP and COSE primitives and algorithms
    for low code complexity and to avoid duplicate maintenance of a
    combined OSCORE and AKE implementation.

   o The messages should be as small as reasonably achievable. The
      messages shall fit into as few LoRaWAN packets and 6TiSCH frames
      as possible.

4. 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 confidentiality protection, integrity
   protection, and authentication as is further detailed in the
   requirements.

5. IANA Considerations

   None.

Acknowledgments

   The authors want to thank Richard Barnes, Karthik Bhargavan, Eric
   Rescorla, Michael Richardson, and Claes Tidestav for providing
   valuable input.

7. Informative References

   [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>.

   [Fairhair]
              "Security Architecture for the Internet of Things (IoT) in
              Commercial Buildings, Fairhair Alliance white paper",
              March 2018, <https://www.fairhair-
              alliance.org/data/downloadables/1/9/
              fairhair_security_wp_march-2018.pdf>.

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

   [I-D.ietf-6tisch-minimal-security]
             Vucinic, M., Simon, J., Pister, K., and M. Richardson,
             "Constrained Join Protocol (CoJP) for 6TiSCH", draft-ietf-
             6tisch-minimal-security-14 (work in progress), December
             2019.

   [I-D.ietf-core-echo-request-tag]
             Amsuess, C., Mattsson, J., and G. Selander, "CoAP: Echo,
             Request-Tag, and Token Processing", draft-ietf-core-echo-
             request-tag-08 (work in progress), November 2019.

   [I-D.ietf-cose-x509]
             Schaad, J., "CBOR Object Signing and Encryption (COSE):
             Headers for carrying and referencing X.509 certificates",
             draft-ietf-cose-x509-05 (work in progress), November 2019.

   [I-D.ietf-lpwan-coap-static-context-hc]
             Minaburo, A., Toutain, L., and R. Andreasen, "LPWAN Static
             Context Header Compression (SCHC) for CoAP", draft-ietf-
             lpwan-coap-static-context-hc-11 (work in progress),
             October 2019.

   [I-D.irtf-cfrg-randomness-improvements]
             Cremers, C., Garratt, L., Smyshlyaev, S., Sullivan, N.,
             and C. Wood, "Randomness Improvements for Security
             Protocols", draft-irtf-cfrg-randomness-improvements-08
             (work in progress), November 2019.

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

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

   [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>.

   [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>.

   [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>.

   [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>.

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