ACE – Design Considerations

Corinna Schmitt (schmitt@ifi.uzh.ch)

IETF ACE WG meeting
July 23, 2014

Relevant Drafts

- Main draft
  - draft-seitz-ace-design-considerations-00

- Drafts addressing topic:
  - Two-way Authentication for IoT
  - Delegated CoAP Authentication and Authorization Framework (DCAF)
  - Group Authentication
  - Authentication and Authorization for Constrained Environments (ACE): Overview of Existing Security Protocols
  - The OAuth 2.0 Bearer Token Usage over the Constrained Application Protocol (CoAP)
  - The OAuth 2.0 Internet of Things (IoT) Client Credentials Grant

Guiding Questions

(1) What design components could be re-used?

(2) What areas need to be explored in more detail?

(3) Should the design be based on symmetric or asymmetric crypto? (or both?)

Guiding Questions

(1) What design components could be re-used?
- Authorization possibilities
- Key management

(2) What areas need to be explored in more detail?

(3) Should the design be based on symmetric or asymmetric crypto? (or both?)

Authorization

- The authorization decision may be based on credentials presented by C, resource, RESTful method and local context in the RS at the time of the request.
- Apart from access, authorization may also apply to gaining knowledge about a resource.

- RS needs to know that C is allowed to access the resource as requested.
- RS needs to make sure that it provides the resource only to C.

- The authorization decision may be taken either by AS or RS.
- The authorization decision is enforced by RS.
- There may be mechanisms for a client to lookup the corresponding AS for an RS.

Authorization information

- The authorization information contains information to allow the RS to verify that a requesting client is authorized.

  - What should it contain?
    - e.g. Attributes, AuthZ decision, capability list
  - How should it be encoded?
    - e.g. JOSE, CBOR, XML
  - How does RS verify it?
    - MAC vs Signature
  - When does RS receive it?
    - Pre-provisioned vs On-demand

Securing authorization information

- The RS shall authenticate the authorization information coming from the AS.
  - Authorization information may be communicated via the client.
- Authorization information may also be encrypted end-to-end between the AS and the RS.
- The RS may not be able to communicate with the AS at the time of the request from a client.
- The RS may store or cache authorization information.
- Authorization information stored or cached in RS shall be possible to change. The change of such information shall be authorized.

Keys and cipher suites

- Access request/response between client and RS may be protected with symmetric and/or asymmetric keys.
- Authorization information is protected with symmetric and/or asymmetric keys.
- There may be a mechanism for the client to look-up the supported cipher suites of RS.

- Key Management
  - Enrollment
  - Binding (e.g., to specific authorization)
  - Revocation
  - Expiration

Guiding Questions

(1) What design components could be re-used?

(2) What areas need to be explored in more detail?
- Message style and size
- Fewer messages
- Reduction of computation

(3) Should the design be based on symmetric or asymmetric crypto? (or both?)

Message size

- Reducing message size will reduce composing/parsing and sending/receiving costs which is favorably impacting energy consumption and latency.
  - Smaller than CoAP payload size (1024 bytes) avoids fragmentation at the application layer.
  - Smaller than the maximum MAC-layer frame size (e.g. 127 bytes for IEEE 802.15.4) avoids fragmentation at the link layer.
- The largest messages are potentially those containing certificates or authorization tokens, so reducing their size significantly will have a large impact.

Number of messages

- Removing message exchanges or round trips have potentially large impact on energy consumption and latency.
- Challenge-response based authentication protocols may potentially be replaced with other protocols with alternative measures to ensure freshness, such as time or sequence numbers.

Reduction of computation

- Reducing the number of public key operations used during normal operations, e.g. by keeping existing sessions alive, or generating session resumption state on a less constrained device.

  → Either more RAM or more sending and receiving of messages are needed.

- Replacing public key operations with symmetric key operations.

  → It is not always possible to make this replacement , because it requires a change in trust-model.

Guiding Questions

(1) What design components could be re-used?

(2) What areas need to be explored in more detail?

(3) Should the design be based on symmetric or asymmetric crypto? (or both?)
- Code size
- Network considerations

Code size

- The overall size of the code is influenced mainly by the size of the libraries needed for cryptography and parsing messages.

- In general asymmetric cryptography requires larger libraries (e.g. BigInteger, Elliptic curves) than symmetric cryptography.
- Minimal libraries for parsing ASN.1 and JSON are roughly comparable in size (around 6 kB) while even minimal XML parsers generally have a significantly larger size.

Network considerations (1)

- The solution shall prevent network overload due to avoidable authorization requests to AS.
- The solution shall prevent network overload by compact authorization information representation.

- The solution shall optimize the case where authorization information does not change often.
- The solution where possible supports multicasting authorization information for e.g. when multiple entities need to be configured (change state).

Network considerations (2)

- The solution shall interwork with existing infrastructure.
- The solution shall support authorization of access to legacy devices through proxies as enforcement points.

- The CoRE architecture provides for the use of intermediaries. The solution shall not unduly restrict the CoRE architecture.

Idea: Framework should support both.

Guiding Questions

(1) What design components could be re-used?
- Authorization possibilities
- Key management

(2) What areas need to be explored in more detail?
- Message style and size
- Fewer messages
- Reduction of computation

(3) Should the design be based on symmetric or asymmetric crypto? (or both?)
- Code size
- Network considerations

Thank you!

Questions/comments?

Delegation

- Example: Door-Lock System
- Do we want to support delegation in proposed solutions?
- The solution shall allow authorization of delegation of access rights.
- Outsourcing of key management
  - Providing symmetric keys to support authentication (cf. Kerberos).
  - Providing protected assertions containing statements about client and server, including public key certificates.

Constrained Devices 1(2)

- C and/or RS may be constrained in terms of power, memory, storage space
  - may not have user interfaces and displays
  - may not have network connectivity all the time
  - are not able to manage complex authorization policies
  - are not able to manage a large number of secure connections
  - may need to save on wireless communication due to high power consumption
  - may not be able to precisely measure time
- The C and/or RS may scale down to class 1 devices with certain effort
- AS is not a constrained device

Constrained Devices 2(2)

- All devices can process symmetric cryptography without incurring an excessive performance penalty
  - We expect symmetric key standard algorithm such as AES
  - Except for the most constrained devices we expect to use a standardized cryptographic hash function such as SHA-256
- Public key cryptography requires additional resources (e.g. RAM, ROM, power)
  - Certificate-based DTLS handshake requires additional resources (e.g. ROM) [1]
  - The RAM requirements of DTLS handshake with public-key cryptography may be prohibitive for constrained devices [1]

[1] "Delegation-based Authentication and Authorization for the IP-based Internet of Things" Hummen, R. et al.

Communication security paradigm

- The solution shall support session based security and/or object security
  - There should be modes of operation that don't require both