Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 2, 2012
July 1, 2011

S. Das
Telcordia
Y. Ohba
Toshiba

Bootstrapping CoAP Applications for Resource-Constrained Devices using EAP

draft-ohba-core-eap-based-bootstrapping-00

Abstract

This document describes a mechanism to use EAP (Extensible Authentication Protocol) for bootstrapping DTLS-PSK (Pre-Shared Key) ciphersuites and PSK mode of IKEv2 that are used for establishing a secure communication channel between a CoAP (Constrained Application) client and a CoAP server to protect CoAP messaging.

Status of this Memo

Copyright Notice

Table of Contents

1. Introduction 3
   1.1. Specification of Requirements 3
2. Use Cases 3
   2.1. Use Case 1: Non-integrated with Network Access Authentication 3
   2.2. Use Case 2: Integrated with Network Access Authentication 4
3. Architecture 5
4. Proposed Solution 6
   4.1. Requirements 6
   4.2. Assumptions 6
   4.3. Call Flow 6
5. Security Considerations 7
6. IANA Considerations 7
7. References 8
   7.1. Normative References 8
   7.2. Informative References 8
Authors' Addresses 8

1. Introduction

Constrained Application Protocol (CoAP) [I-D.ietf-core-coap] is a web protocol defined over UDP to realize the Representational State Transfer (REST) architecture of the web in a suitable form for constrained environments and M2M (Machine-to-Machine) applications. CoAP supports a limited subset of HTTP functionality, which allows straightforward mapping to HTTP. Unicast CoAP messages are secured using Datagram TLS (DTLS) [RFC4347] and IPsec Encapsulating Security Payload (ESP) [RFC4303].

This document describes how EAP (Extensible Authentication Protocol) can be used to bootstrap DTLS-PSK (Pre-Shared Key) ciphersuites and PSK mode of IKEv2 [RFC5996] that are used for dynamically establishing unicast security associations.

Although CoAP supports multicast messaging in addition to unicast, current CoAP specification does not clearly specify which security protocol is used for securing multicast CoAP messages and how multicast keys are established. This version of document focuses on bootstrapping unicast security associations.

1.1. Specification of Requirements

In this document, several words are used to signify the requirements of the specification. These words are often capitalized. The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

2. Use Cases

The following uses cases are considered.

2.1. Use Case 1: Non-integrated with Network Access Authentication

This use case scenario is applicable where CoAP applications bootstrap is not facilitated by the access network authentication mechanisms. Typically this type of scenario exists when there are no business relationships exist between access network provider and service provider. Service provider here is a provider that provides CoAP-based application services. For example, access network provider could be a DSL or cable provider whereas the service provider could be an electric utility provider. The applicability of such scenarios is depicted in more details in [ETSIM2M].

Following are the advantages of bootstrapping applications in this

scenario:

- There is no requirement of having knowledge on how the access network security is provided and managed. Hence there is no need to have interface between access network device/gateway and application device/gateway.

- The security credentials can be provisioned and managed directly by the service provider.

- There is no need for manual provisioning of keys to the client and server

- Provides a scalable architecture that does not require establishing secure connection to other devices/gateways in the network rather than CoAP application server.

2.2. Use Case 2: Integrated with Network Access Authentication

This use case scenario is applicable where CoAP applications bootstrap is facilitated by the access network authentication mechanisms. Typically this type of scenario exists when there are business relationships exist between access network provider and service provider or both access network and service providers are managed by the same entity or organization. For example, the access network provider and the service provider both could be an electric utility provider where access network is Wi-Fi mesh and the CoAP application is a smart metering data application. Another example could be that access network is a cellular network and there is business relationship with the cellular provider and utility provider. The applicability of such scenarios is depicted in more details in [ETSIM2M].

Following are the advantages of bootstrapping applications in this scenario:

- The same credential and other provisioning parameters for network access authentication can be used to generate the key for CoAP applications

- No need for separate provisioning and management interface to the end devices

- There is no need for manual provisioning of keys to the client and server

- Provides a tightly coupled architecture that does not require separate management and provisioning infrastructure.

3. Architecture

The bootstrapping architecture is shown below where several functional elements are used. The placement and consideration of these functional elements do not provide any mapping to specific network architecture or deployment scenarios.

```
CoAP client
+----+                    +----+                      +----+
| UE |--------------------| AA |----------------------| AS |
+----+                    +----+                      +----+
  |                                                     |
  |                     CoAP server                     |
  |                       +----+                        |
  +-----------------------| ApS|------------------------+
                          +----+
```

Figure 1: Functional Architecture

UE (User Equipment):

A user terminal that has a CoAP client

ApS (Application Server): A CoAP server that provides a specific application service for the UE

AS (Authentication Server): A server that authenticates the UE for application services

AA (Authentication Agent): An agent that acts as an authentication relay for the UE for application access authentication (e.g., NAS (Network Access Server).

This architecture can support not only infrastructur-based bootstrapping but also infrastructure-less bootstrapping. The latter can be supported by implementing the AA and AS on the same device.

Also, as part of infrastructure-based bootstrapping, this architecture can support automated recommissioning with the use of service provider-independent authentication credentials that may be pre-provisioned to the UE (e.g. manufacturer provisioned credential). Each time recomissioning happens, new authentication credentials that are specific to the new application service provider need to be generated and cryptographically bound to the service provider-independent authentication credentials. Note that the AS maintaining the service provider-independent authentication credentials is

typically different from the AS maintaining application service provider-specific authentication credentials.

4. Proposed Solution

The proposed solution is based on the requirements described in Section 4.1 and assumptions described in Section 4.2.

4.1. Requirements

1. Solution should have the capability of integration of network access authentication and application access authentication

2. The following parameters are configured through the bootstrapping process:

   * Identity of CoAP client used for DTLS-PSK or IKEv2

   * Identity of CoAP server used for DTLS-PSK or IKEv2

   * Pre-shared key used for DTLS-PSK or IKEv2

3. EAP [RFC3748] must be supported for an application access authentication protocol. A session key must be derived from the EMSK key hierarchy [RFC5295].

4.2. Assumptions

- UE and AS pre-configure authentication credentials required to authenticate to each other.

- Communications between AA and AS are always secured.

- Communications between ApS and AS are always secured.

- Communications between UE and AS or ApS may not be secured prior to bootstrapping CoAP.

- UE can discover AA and ApS using mechanisms that are not specified in this document.

4.3. Call Flow

A general call flow for the proposal solution is illustrated in Figure 2.

```
   UE                       AA                   AS
    |                        |                     |
    | /-------------------\ | /-----------------\ |
    |/    (1a) EAP         \|/   (1b) EAP        \|
    |\                     /|\                   /|
    | \-------------------/ | \-----------------/ |
    |                                             |
    |                      ApS                    |
    | /-------------------\ | /-----------------\ |
    |/   (2a) PSK Pull     \|/   (2b) PSK Pull   \|
    |\                     /|\   w/PSK[AS->ApS]  /|
    | \-------------------/ | \-----------------/ |
    |                        |                     |
```

Figure 2: General Call Flow

Step 1: CoAP service access authentication is performed between the UE and AS via the AA using EAP. For Use Case 2, the authentication agent is integrated with network access authentication where the AA is co-located with NAS (Network Access Server). In Figure 2, the UE is an EAP peer, the AA is an EAP authenticator and the AS is an EAP server. To transport EAP message between the UE and AA (Step 1a), PANA [RFC5191] is used as EAP lower layer for Use Case 1, and for use case 2, any lower layer transport may be used. When the AA and AS are not co-located, a AAA protocol is used for transporting EAP messages between the AA and AS (Step 1b).

Step 2: A pull key operation is performed between the UE and AS via the ApS to distribute PSK from the AS to the ApS. The pull key operation is initiated by the UE when the UE has CoAP application data to send to the ApS for which a PSK is not configured yet. After successful completion of Step 2, the PSK is ready to use for DTLS or IKEv2 between the UE and ApS.

5. Security Considerations

TBD.

6. IANA Considerations

This document includes no request to IANA.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC5191] Forsberg, D., Ohba, Y., Patil, B., Tschofenig, H., and A. Yegin, "Protocol for Carrying Authentication for Network Access (PANA)", RFC 5191, May 2008.

[RFC5996] Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010.

[I-D.ietf-core-coap]
Shelby, Z., Hartke, K., Bormann, C., and B. Frank, "Constrained Application Protocol (CoAP)", draft-ietf-core-coap-06 (work in progress), May 2011.

7.2. Informative References

[RFC5295] Salowey, J., Dondeti, L., Narayanan, V., and M. Nakhjiri, "Specification for the Derivation of Root Keys from an Extended Master Session Key (EMSK)", RFC 5295, August 2008.

[ETSIM2M] European Telecommunications Standards Institute, "Machine-to- Machine communications (M2M); Functional architecture", ETSI TS 102 690, 2011.

Authors' Addresses

Subir Das
Telcordia Technologies Inc.
1 Telcordia Drive
Piscataway, NJ 08854
U.S.A.

Email: subir@research.telcordia.com

Yoshihiro Ohba
Toshiba Corporate Research and Development Center
1 Komukai-Toshiba-cho
Saiwai-ku, Kawasaki, Kanagawa 212-8582
Japan

Phone: +81 44 549 2127
Email: yoshihiro.ohba@toshiba.co.jp