Tutorial on KMIP and FCEAP/GPSK

Bob Nixon (Emulex)
bob.nixon@emulex.com
Where is this bus going?

• What’s the motivation for involving KMIP?
• What is KMIP?
• What does FC-EAP/GPSK need from KMIP?
What’s the motivation for involving KMIP?
Who are the players?

• RADIUS
  – RADIUS Server (an Authentication Server)
  – RADIUS Client
• KMIP
  – KMIP Server (a Key Management Server)
  – KMIP Client
• FC-SP-2
  – Authentication Initiator
  – Authentication Responder
When using DHCHAP/RADIUS - 1
When using DHCHAP/RADIUS - 2

DHCHAP Initiator <-> DHCHAP Responder

DHCHAP:

Exchange identities

RADIUS Server
When using DHCHAP/RADIUS - 3

RADIUS Server

DHCHAP Initiator

DHCHAPResponder

Initiator authentication keys
Responder authentication keys

Delegate authentication
When using DHCHAP/RADIUS - 4

Diagram:
- RADIUS Server
- DHCHAP Initiator
- DHCHAP Responder
- DHCHAP:
  - Negotiate Session Key
When using DHCHAP/RADIUS - 5

- Key management is centralized 😊
- Cryptographic math is delegated 😊
- ...but the cryptography of RADIUS and CHAP are falling out of favor 😞

EAP/GPSK chosen for flexibility and security
But lacks an obvious management system
Would EAP/GPSK work with RADIUS?
Would EAP/GPSK work with RADIUS?

+---+---+---+
<p>| RADIUS | Server |     |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
+---+---+---+
|     |     |     |
|     |     |     |
|     |     |     |
+---+---+---+
|     |     |     |
|     |     |     |
|     |     |     |
+---+---+---+
| EAP/GPSK Initiator | --- | EAP/GPSK Responder |
| Exchange identities |
+---+---+---+---
Would EAP/GPSK work with RADIUS?

- EAP/GPSK only works for the responder side.
- RADIUS: Delegate authentication not supported.
- RADIUS Server
  - Initiator authentication keys
  - Responder authentication keys

EAP/GPSK Initiator

Only for responder

EAP/GPSK Responder
Would EAP/GPSK work with RADIUS?

**Diagram:**
- RADIUS Server
- RADIUS/Initiator session key
- EAP/GPSK Initiator
- EAP/GPSK Responder
- Negotiate Session Keys
- ...but I won’t tell you what it is!

**Text:**

Would EAP/GPSK work with RADIUS? 

EAP/GPSK: 

...but I won’t tell you what it is!
Would EAP/GPSK work with RADIUS?

• ...only the Authentication Responder can delegate the cryptographic math 😞
• ...and you can’t extend the trust to an FC-SEC session 😞 😞

Not so useful
When using EAP/GPSK/KMIP - 1
When using EAP/GPSK/KMIP - 2

EAP/GPSK Initiator → EAP/GPSK : Exchange identities → EAP/GPSK Responder

KMIP Server
When using EAP/GPSK/KMIP - 3

KMIP Server

Get Peer’s authentication key

EAP/GPSK Initiator

Initiator/Responder peer authentication key

EAP/GPSK Responder

Initiator/Responder peer authentication key
When using EAP/GPSK/KMIP - 4

KMIP Server

EAP/GPSK Initiator

Initiator/Responder peer authentication key

EAP/GPSK Responder

Initiator/Responder peer authentication key

EAP/GPSK:
Authenticate identities
When using EAP/GPSK/KMIP - 5

- KMIP Server
- EAP/GPSK Initiator
- EAP/GPSK Responder
- Negotiate Session Key
When using EAP/GPSK/KMIP - 6

• More secure than DHCHAP/RADIUS 😊
• Key management is centralized 😊
• Cryptographic math is not delegated 😞
  – “EAP-GPSK should be easy to implement” (RFC 5433) 😊
  – …Doesn’t help for EAP-NEXT, though 😞

Net gain, it appears
What is KMIP?
KMIP

≡ Key Management Interoperability Protocol

- It’s a protocol standard, not a server design. Intention is that it be “front-ended” to existing and future proprietary server designs.

- It covers management, not authorization. Intention is that, although a certain minimum is expected, a design is free to elaborate its authorization capability (or import it, e.g., from a corporate directory).
KMIP 1.0 is an OASIS Standard

• Actually, two OASIS Standards
  – KMIP Specification
  – KMIP Profiles

• Two supporting OASIS Committee Specifications
  – KMIP Use Cases (consider them as test specs)
  – KMIP Usage Guide (“Informative Annex”)
KMIP Specification

Oversimplifying:

- The Protocol is composed of a sequence of Request/Response pairs
- A Request or Response is a Message
- A Message is a header followed by one or more Batch Items
- A Batch Item is an Operation Code and a Payload
- A Payload is a Sequence of Objects and Attributes
- An Object is zero or more subordinate Objects and zero or more Attributes
- An Attribute is one or more primitive data types
- Everything is encoded as a TTLV structure
TTLV

≡ Tag, Type, Length, Value
Tag (3 bytes), Type (1 byte), Length (4 bytes), Value (see Length)

– Tag: What is it? (e.g., a Symmetric Key, a Lease Time)
– Type: How is it encoded? (e.g., a byte string, a substructure)
– Length: How long is the Value (in bytes)? (e.g., an Integer Length is 4)
– Value: What is the value? (OK, so that’s circular. This is a KMIP tutorial, it’s not Philosophy 301)
TTLV Example
(from KMIP Specification)

• A Text String with the value "Hello World":
  42 00 20 | 07 | 00 00 00 0B |
    48 65 6C 6C 6F 20 57 6F 72 6C 64 00 00 00 1733 00 00

   Simple , right?
Another TTLV Example
(from KMIP Use Cases)

Create (symmetric key)
In: objectType="00000002" (Symmetric Key), attributes={ CryptographicAlgorithm="00000003" (AES), CryptographicLength="128", CryptographicUsageMask="0000000C" }

Tag: Request Message (0x420078), Type: Structure (0x01), Data:
  Tag: Request Header (0x420077), Type: Structure (0x01), Data:
    Tag: Protocol Version (0x420069), Type: Structure (0x01), Data:
      Tag: Protocol Version Major (0x42006A), Type: Integer (0x02), Data: 0x00000001 (1)
      Tag: Protocol Version Minor (0x42006B), Type: Integer (0x02), Data: 0x00000000 (0)
    Tag: Batch Count (0x42000D), Type: Integer (0x02), Data: 0x00000001 (1)
  Tag: Batch Item (0x42000F), Type: Structure (0x01), Data:
    Tag: Operation (0x42005C), Type: Enumeration (0x05), Data: 0x00000001 (Create)
  Tag: Request Payload (0x420079), Type: Structure (0x01), Data:
    Tag: Object Type (0x420057), Type: Enumeration (0x05), Data: 0x00000002 (Symmetric Key)
    Tag: Template-Attribute (0x420091), Type: Structure (0x01), Data:
      Tag: Attribute (0x420008), Type: Structure (0x01), Data:
        Tag: Attribute Name (0x4200A0), Type: Text String (0x07), Data: Cryptographic Algorithm
        Tag: Attribute Value (0x4200B0), Type: Enumeration (0x05), Data: 0x00000003 (AES)
      Tag: Attribute (0x420008), Type: Structure (0x01), Data:
        Tag: Attribute Name (0x4200A0), Type: Text String (0x07), Data: Cryptographic Length
        Tag: Attribute Value (0x4200B0), Type: Integer (0x02), Data: 0x00000080 (128)
      Tag: Attribute (0x420008), Type: Structure (0x01), Data:
        Tag: Attribute Name (0x4200A0), Type: Text String (0x07), Data: Cryptographic Usage Mask
        Tag: Attribute Value (0x4200B0), Type: Integer (0x02), Data: 0x0000000C (Encrypt, Decrypt)
KMIP Profiles

Oversimplifying:

- A profile specifies a subset of the optional features in the KMIP Specification that someone believes would be useful and sufficient for some class of applications
  - Note that what is required by the KMIP Specification is generic to the point that it is practically useless except when extended by a profile.
  - Note that the KMIP Specification requires compliance to at least one profile.
- A profile is a pairing of an Authentication Suite and a Conformance Clause.
- A profile is specified for servers. Clients’ requirements may be inferred.
Authentication Suite

• An Authentication Suite
  – Requires a channel security method providing confidentiality and integrity
  – May require certain options for the channel security method
  – May require a means of client authentication

• If the channel is TCP, TLS 1.0 support is required by the KMIP Specification. An Authentication Suite may add to that.
Example Authentication Suite

• KMIP Basic Authentication Suite includes
  – Requirement for TLSv1.0 protocol, exclusions of SSL
  – Requirement for the TLS_RSA_WITH_AES_128_CBC_SHA cipher suite
  – Requirement for TLS mutual authentication
  – Requirement for consideration of per request credentials if provided by the client
  – Etc.
Conformance Clause

- A Conformance Clause
  - Requires support for the KMIP Server conformance clause
  - Requires support for specific KMIP options
  - May forbid certain KMIP options
  - Typically explicitly permit any KMIP options and extensions outside the standard that are not explicitly listed and not contradictory of any requirements
Example Conformance Clause

• The KMIP Secret Data Server Conformance Clause includes
  – Requirement to support for the KMIP Server conformance clause
  – Requirement to support the optional Secret Data object, of type Password (the Server conformance clause requires support for a generic key, but no specific kind of key)
  – Requirement to support the optional Register operation (the Server conformance clause does not specify how any object gets into a server)
  – etc.
  – Permission to support anything that doesn’t conflict.
What does FC-EAP/GPSK need from KMIP?

My guesses...

Expert advice enthusiastically solicited!
Authentication

• Presuming our channel to be TCP/IP, TLSv1.0 support is required. Any expert advice why we would not use it?
  – TLSv1.2 fixes a published security issue. Is 1.2 generally implemented?
  – Is IPSEC an alternative?

• Require the server to support object authorization rules
  – Modify only administratively
  – Read only by an administratively specified group of two or more entities
Necessary information

• Symmetric key objects
  – Does TLSv1.0 provide sufficient confidentiality that we don’t need key wrapping?
  – Do we need Start/Stop usage dates?

• Uninterpreted Text String Names for keys (the only alternative is URI)

• 128-bit and 256-bit key lengths

• AES and HMAC-SHA256 algorithms
Questions?
Suggestions...
please?