An EAP Method for Extending EAP (draft-ohba-hokey-emu-eap-ext-00.txt)

Yoshihiro Ohba

Subir Das

Goal: Backwards Compatibility

- Allow EAP to add more functionalities including HOKEY, without loss of backwards compatibility with existing EAP and EAP methods implementations

Gap Analysis: EMSK

- HOKEY is defining some usage on EMSK
  - EMSK is mandatory to export in RFC 3748
  - In reality, most existing implementations do not export EMSK
    - WPA and WPA2 certificates do not require EMSK
    - We can't blame them because we did not define EMSK usages
- Defining EMSK usages with expectation of support from all EAP methods will create a serious deployment gap
  - Industry may not use HOKEY if there is no smooth migration path
    - e.g., 802.11i
- In addition, a mechanism for enabling and bootstrapping each EMSK usage is needed. However…
  - Relying on pre-configuration is a bad idea
  - Defining such a mechanism for every EAP method is also bad
  - Defining such a mechanism in EAP lower layer could make the situation even more worse

Gap Analysis: Channel Binding

- EAP keying identifies two Channel Binding approaches:
  - Binding based on a KDF
  - Binding based on parameter exchange
- There is no EAP method that “actively” supports Channel Binding
  - The deployment bar is too high if Channel Binding is required for each EAP method

EAP Facts

- EAP is not extensible without providing backwards compatibility for itself
  - No version field
  - No extension header
  - Silent discarding a message with a new Code
- Is there any way to add more functionalities to EAP without coming up with EAPv2?
  - Yes, by defining a new EAP method used for extending EAP
  - Basic backwards compatibility is provided with NAK

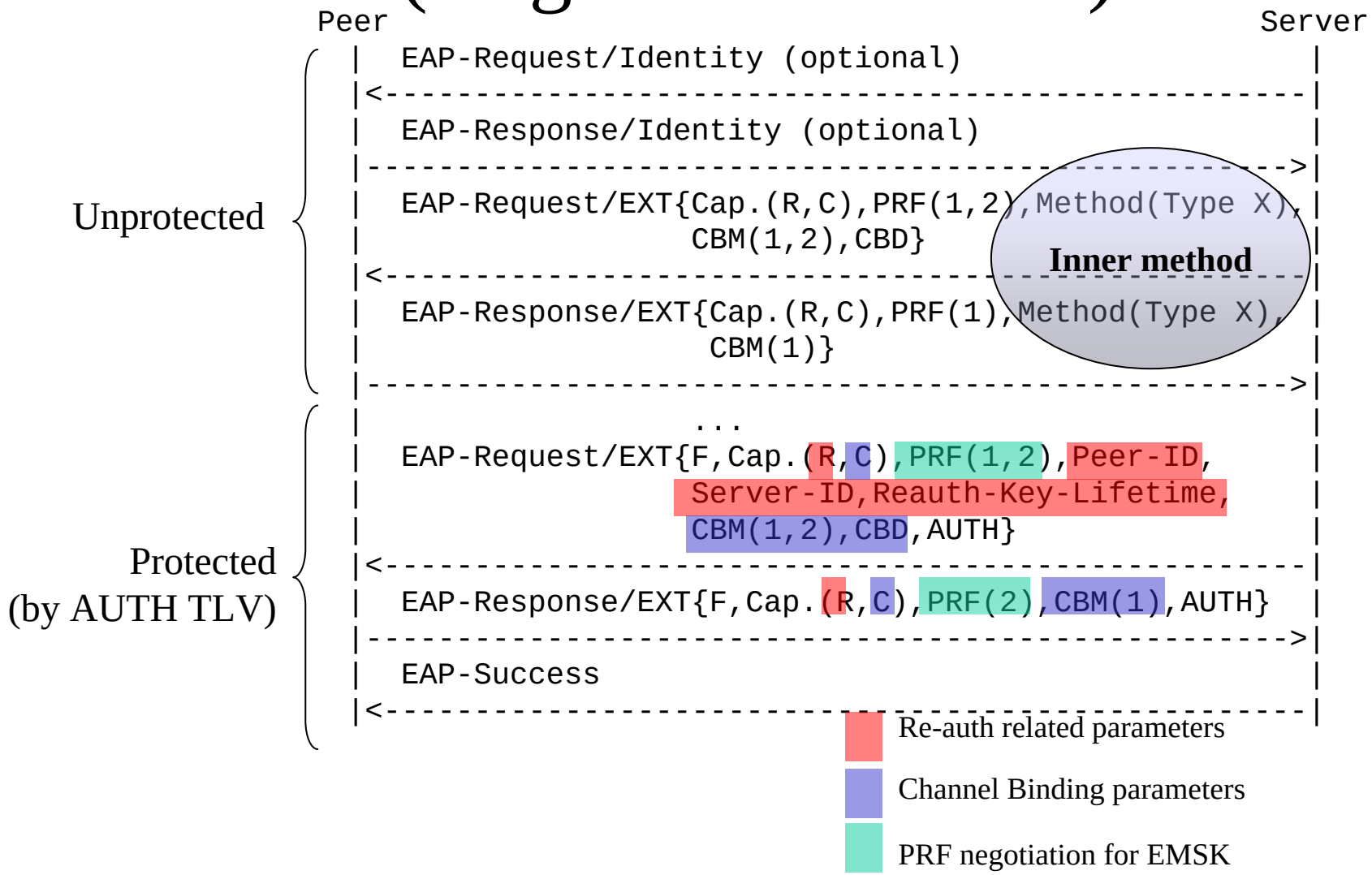
Design choices for a new EAP method to extend EAP

- Sequencing in a single EAP conversation
  - I.e., an authentication method followed by the new EAP method followed by EAP-Success/Failure
  - Sequencing multiple authentication methods (Types 4 and greater) is not allowed in RFC 3748 except inside a tunneling method
- Sequencing EAP conversations
  - I.e., run an authentication method in an EAP conversation and then start another EAP conversation with the new EAP
  - Many lower layers do not support sequencing EAP conversations to generate a single network access authorization
- Tunneling
  - I.e., run an authentication method within the new EAP method
  - Sounds like the most backwards-compatible way

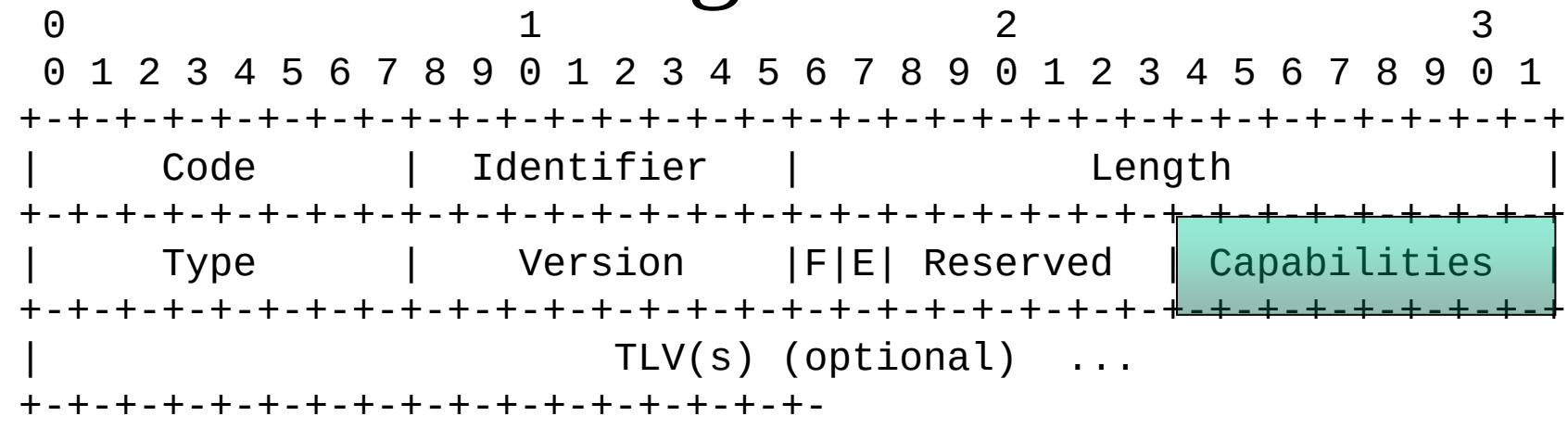
EAP-EXT in a Nutshell

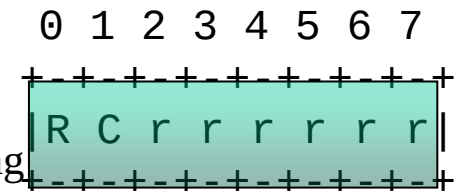
- EAP-EXT provides capabilities exchange.
  - Capabilities: re-authentication and channel binding. Other capabilities such as handover keying can also be added
- At least one EAP method (e.g., EAP-TLS) is run inside EAP-EXT for authenticating the peer
- After an inner method generates EAP keying material, exchanged capabilities are protected
- Even if capability negotiations fail, the peer is still authorized for network access using the basic EAP functionality which is available now
- It is allowed to run multiple authentication methods inside EAP-EXT with cryptographic binding
  - N-th auth method is protected with MSK from (N-1)-th auth method (Integrity chaining)
- EAP-EXT exports MSK and EMSK even if inner methods do not generate EMSK
  - (MSK,EMSK)=KDF(MSK_i, "EAP-EXT-EAP-Keying-Material", 128)
    - MSK_i : MSK from the last successful inner method

EAP-EXT Example (single auth method)

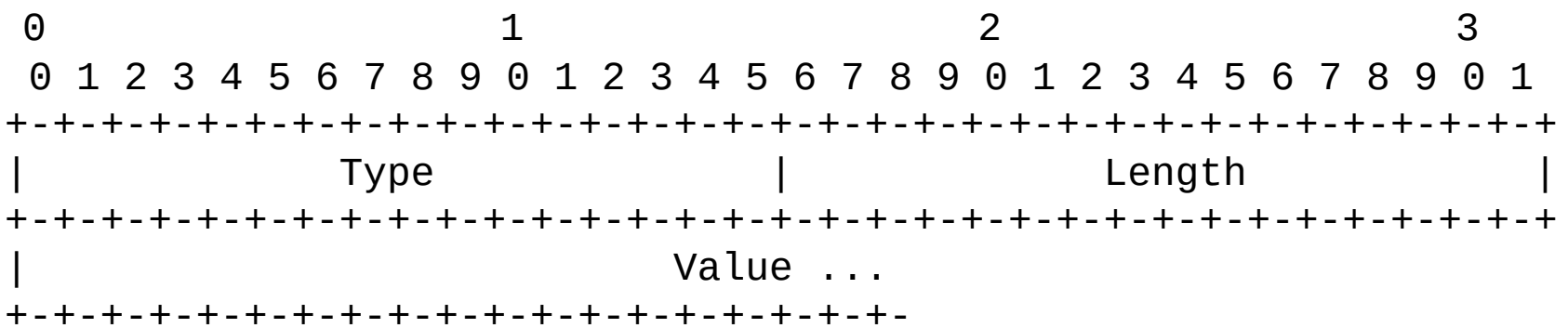


Message Format





- F-bit indicates whether this is the final message from the sender
- E-bit indicates an error
- Capabilties: R-bit for re-authentication and C bit for Channel Binding
- TLV(s): See below



TLVs

- PRF TLV: contains a list of PRF algorithms for USRK derivation
- Re-auth related TLVs
  - Peer-ID TLV, Server-ID TLV, Reauth-Key-Lifetime
  - Actual re-auth mechanism is not specified in this draft
- Channel Binding related TLVs
  - Channel Binding Mechanism TLV: contains a list of CB mechanisms
  - Channel Binding Data TLV: contains parameters specific to a CB mechanism (some CB mechanism does not require this)

Additional work to be done

- Add a TLV for encrypting other TLVs

Summary

- Without addressing backwards compatibility issues, industry may not use new functionalities relating to EAP, including HOKEY
- This proposal addresses the backwards compatibility issues and allows a smooth migration path to HOKEY