P3 Introduction

Lownet – Secure (?)

What is to be done?

- Implementation for provided encryption/decryption hooks, using AES encryption.
- Implement a mechanism to apply / change AES encryption keys in use.
- Implement a new protocol handler for `LOWNET_PROTOCOL_COMMAND` packets.
- Verify authenticity of signed command packets
  - Reject inauthentic commands
  - Correctly process authentic commands.

What has changed in Lownet skeleton?

- New structure `lownet_secure_frame_t` – wraps a plaintext frame.
- Updated Lownet inbound packet handler design
  - Where an AES key has been set, plaintext received frames are dropped and encrypted frames are handled.
  - Where NO key is set, plaintext frames are handled and encrypted frames are dropped.
  - [Decryption is performed in network driver task, esp-noew recv callback – don't block! (This is a poor design for lownet)]
- New Lownet API methods (next slide)
- New Lownet utility module, `lownet_crypt.c/.h` containing (optional) AES keystore.

New Lownet API methods

- lownet.h
  - `lownet_set_time( )`
  - `lownet_get_key( )`
  - `lownet_set_key( )`
  - `lownet_set_stored_key( )`
  - `lownet_get_signing_key( )`
- lownet_crypt.h [Included by <lownet.h>]
  - `lownet_keystore_init( )` [Called by lownet_init(..)]
  - `lownet_keystore_free( )`
  - `lownet_keystore_write( )`
  - `lownet_keystore_read( )`

Where should I start?

- In `app_main.c` the following methods can be given an implementation:
  - `app_frame_dispatch( )`
  - `lownet_decrypt( )`
  - `Lownet_encrypt( )`

- Correctly implementing the methods above suffices to fulfill milestone 1.
- Milestone 2 is a freestyling problem!
  - All of the necessary functionality should be implemented at “application level” – no need to modify lownet source files.

FAQ

- What if I like my P2 implementation and want to build upon it?
  - Add the lownet source files (lownet, lownet_util, lownet_crypt) from the skeleton to your project, overwriting where necessary.
  - In the inner CMakeLists file (within the main/ directory), to the `REQUIRES` line append "`mbedtls`" (with quotation marks).
  - Adjust your preexisting call to `lownet_init(..)` providing function pointers for encrypt / decrypt functions.
- What should I do if I've finished Milestone 1 but I don't know where to start Milestone 2?
  - Take it to Ed! This is a design problem as well as an implementation problem, and there are myriad viable ways to approach this.