SEC4 - Advanced wolfSSL for Embedded Security

Objectives

- Establish fundamental knowledge about cryptographic, algorithms, and protocols.
- Discover how encryption works and how to manage secret keys
- Understanding and Implementing Cryptography
- Learn how to implement secure authentication with wolfSSL
- Develop using wolfSSL's cryptography library (wolfCrypt)
- Understand how to build wolfMQTT on standard platforms and use it in an IoT application
- Building wolfSSH on standard Platforms
- Secure boot using wolfBoot (with wolfCrypt and WolfSSL)

Prerequisites

- C programming
- Experience with embedded systems development.
- oSEC3 Embedded Security with wolfSSL

Course environment

- Theoretical course
  - PDF course material (in English)
  - Course dispensed using the Teams video-conferencing system
  - The trainer to answer trainees' questions during the training and provide technical and pedagogical assistance through the Teams video-conferencing system
- Practical activities
  - Practical activities represent from 40% to 50% of course duration
  - One Online Linux PC per trainee for the practical activities
  - The trainer has access to trainees' Online PCs for technical and pedagogical assistance

Duration

- Total : 12 hours
- 2 sessions of 6 hours

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

First Session

Advanced WolfCrypt usage

- Public Key Cryptography
  - RSA
  - DH (Diffie-Hellman)
  - EDH (Ephemeral Diffie-Hellman)
  - DSA (Digital Signature Algorithm)

- PKCS Public Key Cryptography Standards
  - PKCS#1 RSA Cryptography Standard
  - PKCS#3 Diffie-Hellman Key agreement Standard
  - PKCS#5 Password Based Cryptography Standard
  - PKCS#6 Extended-Certificate Syntax Standard Historic
  - PKCS#7 and RFC 3369 : Cryptographic Message Syntax (CMS)
  - PKCS#8 Private Key Information Syntax Standard
  - PKCS#9 Selected Object Classes and Attribute Types
  - PKCS#10 Certification Request Syntax Standard
  - PKCS#11 Cryptographic Token Interface Standard
  - PKCS#12 Personal Information Exchange Syntax Standard
  - PKCS#15 Cryptographic Token Information Syntax Standard
- Cryptographic Certification
  - FIPS 140-2 Certification
  - NSA Certification
- Progressive Cryptography
- Hardware Accelerated Cryptography
- X.509 Certificates
- Key and Certificate genration
- WolfCrypt CertManager API

Exercise: Sign and Verify data with RSA
Exercise: Sign and Verify data with ECC
Exercise: Sign and Verify data with Ed25519
Exercise: Key Agreement with ECDH and Curve25519
Exercise: PKCS#7 and CMS bundle Generation and Verification
Exercise: wolfCrypt “FIPS Readyâ€Â•
Exercise: Progressive Cryptography
Exercise: Creating Keys and Certificates
Exercise: Generate RSA/ECC Key and Certificates via API
Exercise: WolfCrypt Certificate Manager

Second Session

wolfBoot

- • Introduction
- Features
- Components
- Wolfboot bootloader
- Integrating wolfBoot
- Upgrading the firmware

Exercise: Upgrading the firmware using wolfBoot

wolfSSH

- Overview
- Building wolfSSH
- wolfSSH User authentication Callback
- wolfSSH SFTP
- wolfSSH SCP

Exercise: wolfSSH examples with passwords and Keys
Exercise: SFTP with wolfSSH
Exercise: SCP with wolfSSH

wolfMQTT

- MQTT and MQTT-SN Overview

- MQTT client
- MQTT packet
- MQTT socket
- API reference

Exercise: MQTT Broker