SILICON LABS

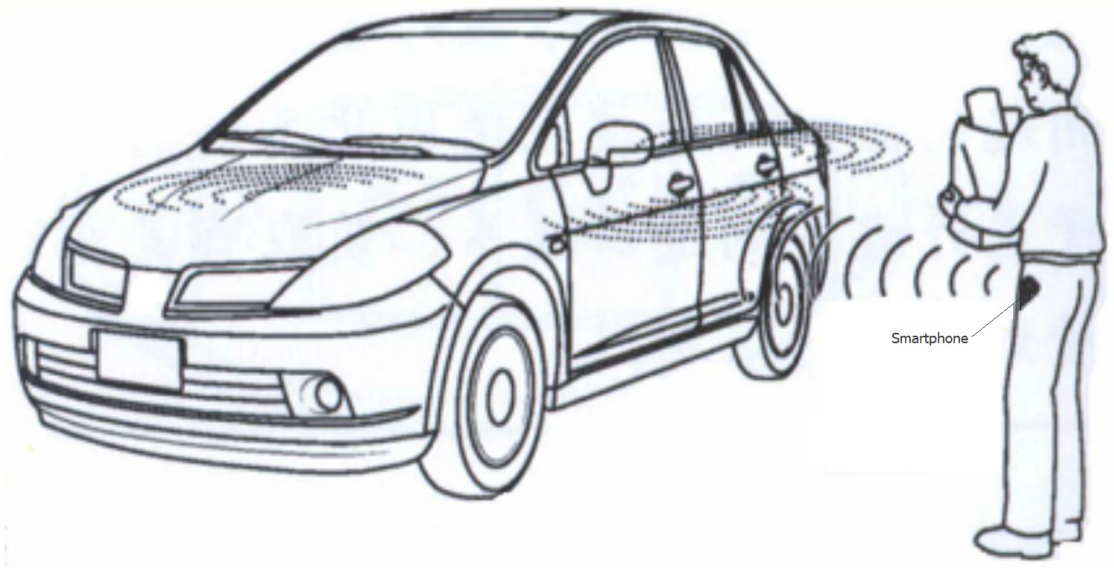
Bluetooth Smart Key and Demo

APAC RA

Agenda

- Overview
- Technical Introduction
  - HID
  - RSSI to Distance
  - Workflow
- Demo Setup
  - Prerequisites
  - Project Setup
  - Demo

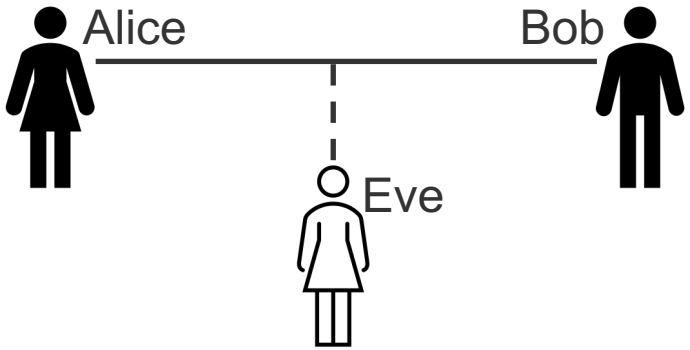
Overview



- PKE(Passive Keyless Entry)
  - Smart key that operate while stored in the user's pocket or bag
  - Based on wireless technology, can be BLE, NFC, UWB, UHF etc.
- Smartphone PKE
  - Almost everyone has a smartphone
  - Almost all smartphone support BLE
  - Add BLE HID device into the car, smartphone based smart key should be possible
  - Simplify people travel and car unlock way

HID

- Security #1
  - Common threats in wireless communications
    - Passive eavesdropping
    - Active eavesdropping = Man in the Middle (MITM)
    - Privacy (tracking)



  - Bluetooth security features
    - Pairing - creating trusted relationships between devices (Key Generation, Key Exchange, Identity Information Exchange)
    - Bonding - storing the keys created during pairing for later connections
    - Device authentication - verification that devices have the same keys (Protected from MITM)
    - Encryption - data confidentiality (FIPS or NIST approved AES128-CCM algorithm used)
    - Data signing (Message integrity) - protection against data alteration

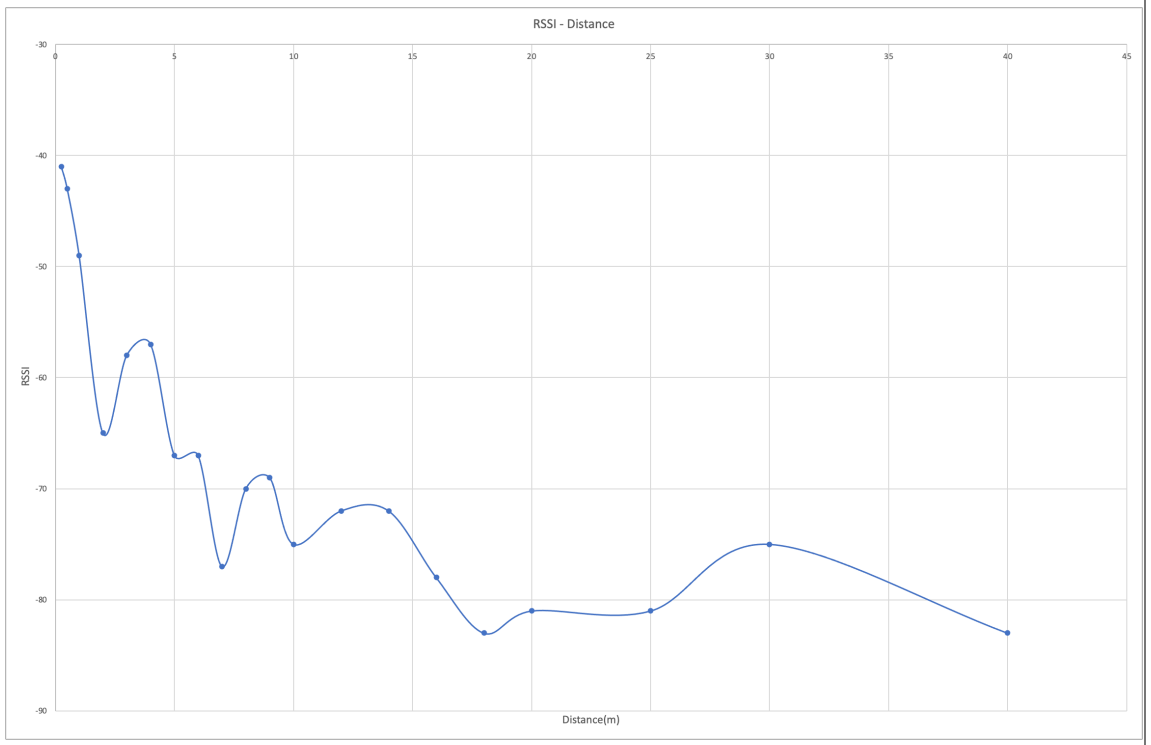
HID

- Security #2
  - BLE Connection Security Modes/Levels
    - Security Mode 1, enforces security by means of encryption, contains four levels:
      - Level 1: No Security (No authentication and no encryption)
      - Level 2: Unauthenticated pairing with encryption
      - Level 3: Authenticated pairing with encryption
      - Level 4: Authenticated LE Secure Connections pairing with encryption
    - Security Mode 2, enforces security by means of data signing, contains two levels:
      - Level 1: Unauthenticated pairing with data signing
      - Level 2: Authenticated pairing with data signing
  - Silicon Labs' BLE stack only support Security Mode 1
  - HID devices request bond and use LE Security Mode 1, Security Level 2 or 3

HID

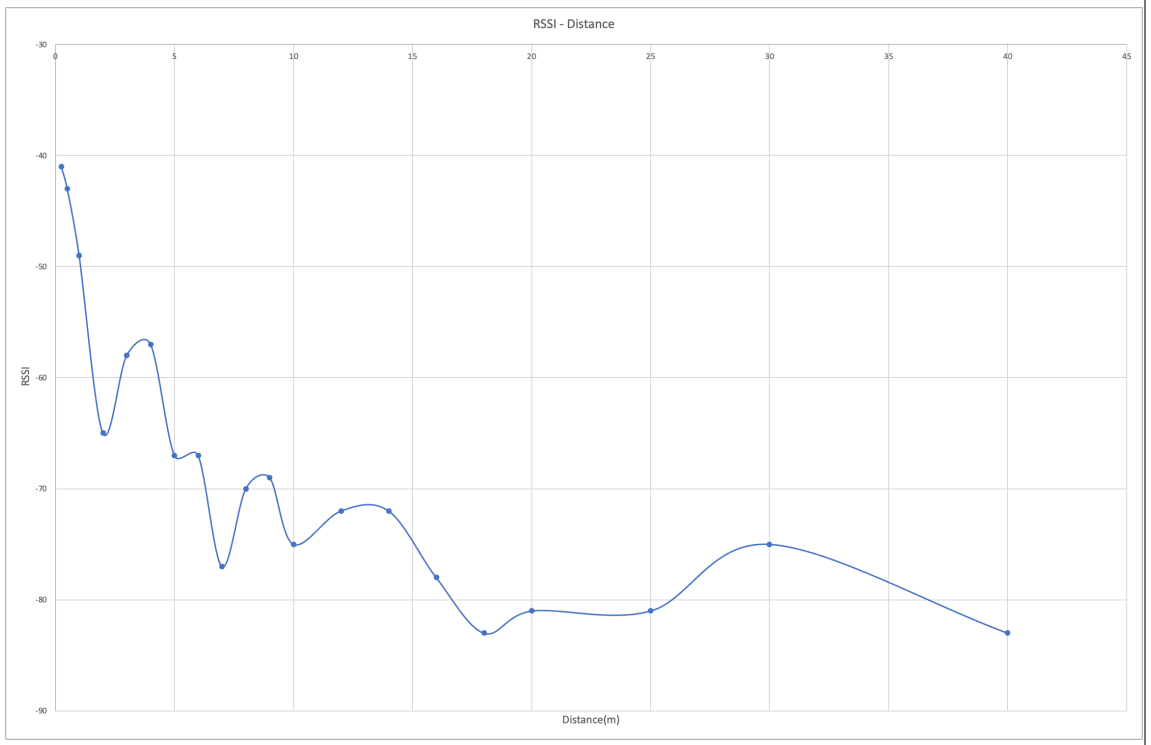
- **Auto Reconnection**
  - **HID Host reconnect to the HID device while link loss disconnect, refer to HID over GATT Profile**
  - **For compatibility with iOS devices, refer to Apple Accessory Design Guidelines**

RSSI



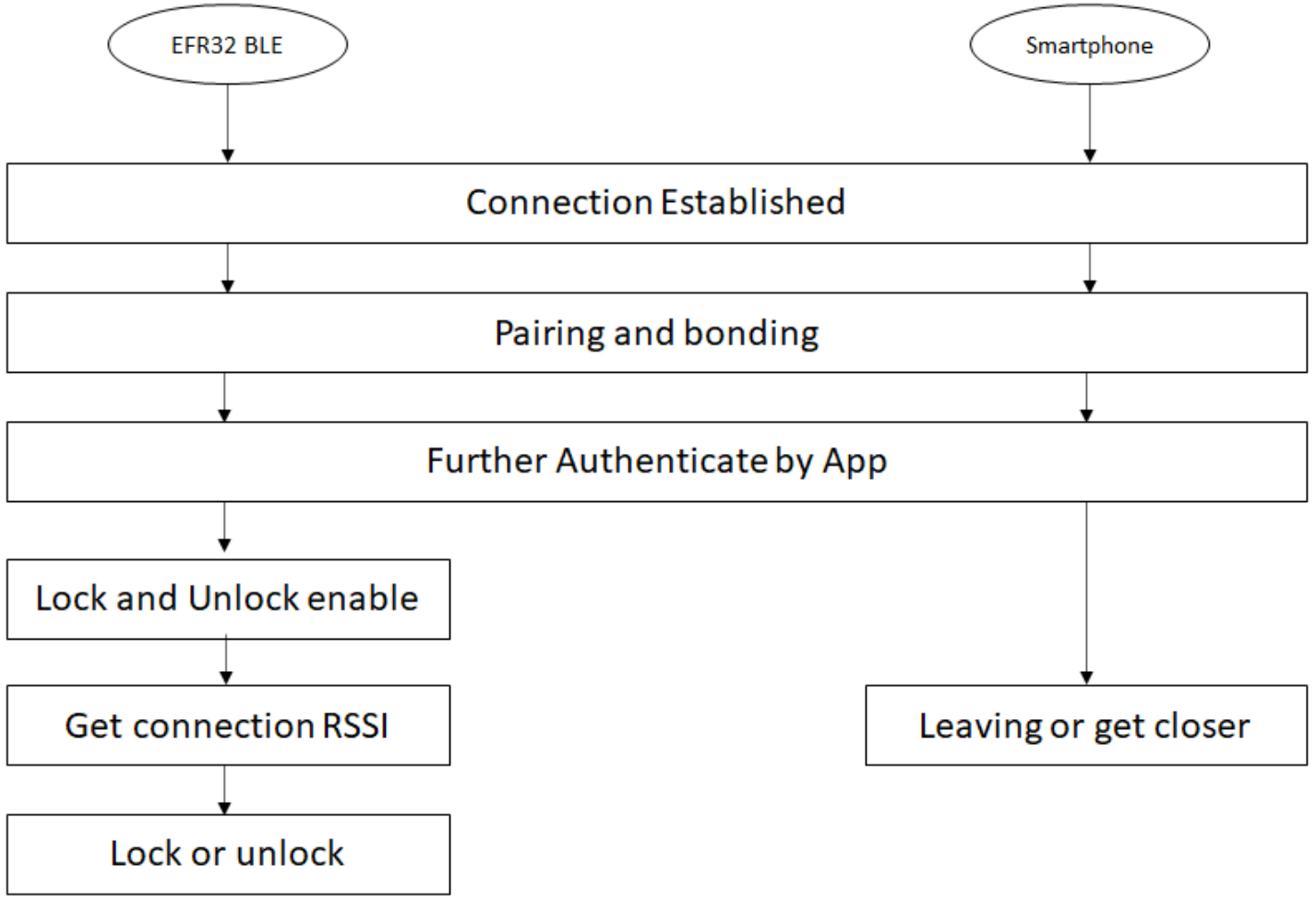
- Indication of the power level being received
- More negative number means more further away
- Affected by many factors, output power, obstacles, distance etc.
- Basically reflects the far and near, calculate average RSSI can improve accuracy

RSSI to Distance



- Formula
  - d = (r/t)^10.0
  - d = A*(r/t)^B+C
- Parameter
  - d – estimated distance in meters
  - t – TX power in dBm at 1 m
  - r – measured signal value (RSSI) in dBm
  - A, B, and C are constants. For how to calculate, refer to Calculating Formula Constants
- Example(A=0.7, B=5.1, C=0.111, t=-40, r=-55)
  - d = 0.7 * powf(ratio, 5.1) + 0.111
    = 0.7 * ((-55 / -40)^5.1) + 0.111
    = 0.7 * 5.074 + 0.111
    = 3.662745

Workflow



- Lock operation enable after authentication via App
- The App
  - Self-developed App
  - Mini program (WeChat/Alipay)
  - EFR Connect (temporary use)
- Come near or go far

Demo Setup

- Project Setup
  - Create a new 'Bluetooth - SoC Empty' project for the EFR32xG24 device. Rename it as "xG24_training_HID", check the "Copy contents" on
  - Install the following software components:
    - Services -> IO Stream -> IO Stream: USART
    - Application -> Utility -> Log
    - Services -> Simple timer service
    - Platform -> Driver -> LED -> Simple LED
    - Platform -> Driver -> Button -> Simple Button
    - Bluetooth -> NVM -> NVM Support
  - Enable floating point printf(), refer to KBA.
  - Import the attached gatt_configuration.btconf file in the GATT Configurator
  - Copy the attached app.c file into the project (overwriting the existing one)
  - Build and flash the project to the EFR32 device

Demo Code

Demo

Demo

SILICON LABS

Thank You!

www.silabs.com