PhD Thesis Defense 2019 @ SUTD

Design, Implementation, and Evaluation of Secure Cyber-Physical and Wireless Systems

Daniele Antonioli

Singapore University of Technology and Design (SUTD)

Design, Implementation, and Evaluation of Secure Cyber-Physical and Wireless Systems

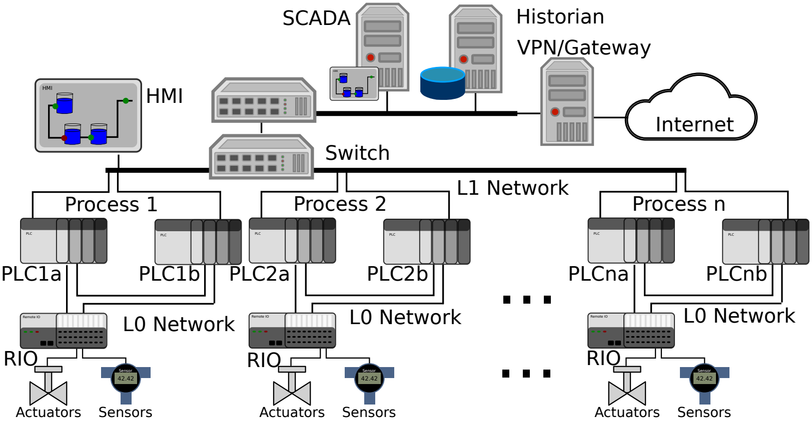
- Thesis's structure
  - Part I: Cyber-physical systems security (Chapter 1-5)
  - Part II: Wireless systems security (Chapter 6-10)
  - TL;DR: Read sections 1.3 and 6.3
- Main collaborations
  - SUTD (P. Szalachowski), University of Oxford (K. Rasmussen), and CISPA (N. O. Tippenhauer)

SINGAPORE UNIVERSITY OF TECHNOLOGY AND DESIGN

CISPA
HELMHOLTZ CENTER FOR INFORMATION SECURITY

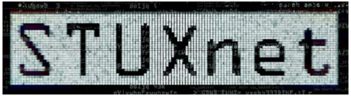
Cyber-Physical Systems (CPS)

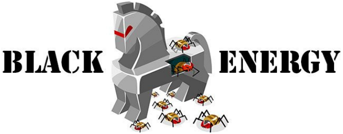
- Interconnected devices managing a physical process
  - Information technology (IT)
  - Operational technology (OT)



Cyber-Physical Systems (CPS) Security

- Securing CPS is paramount, yet challenging
  - Cyber, physical, and cyber-physical attacks
  - Wired and wireless connections (to the Internet)
- High impact attacks on CPS
  - E.g. Stuxnet (nuclear), BlackEnergy (smart grid), TRISIS/TRITON (safety)



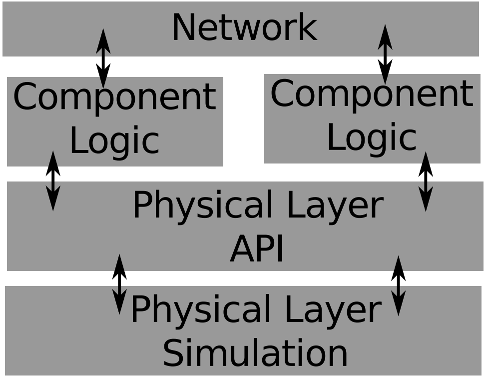


CPS Security Challenges and Research Questions

- **C1: Evaluation of CPS (IT and OT) technologies**
  - Q1: Can we build a low-cost real-time simulation environment for CPS? [CPS-SPC15]
- **C2: Cyber-physical attacks**
  - Q2: Can we detect and mitigate cyber-physical attacks? [CPS-SPC16]
- **C3: CPS security education**
  - Q3: Can we fill the gaps between IT and OT security professionals? [CPS-SPC17]

MiniCPS: A toolkit for security research on CPS networks [CPS-SPC15]

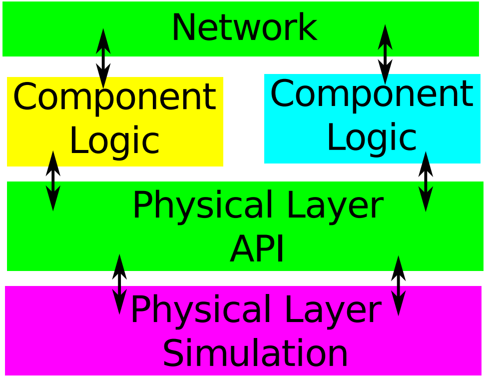
- Q1: Can we build a low-cost real-time simulation environment for CPS?



(C)yber $\longrightarrow$ Network Emulation
(P)hysical $\longrightarrow$ Physical Layer Simulation and API
(S)ystem $\longrightarrow$ Simulation of Control Devices

MiniCPS: A toolkit for security research on CPS networks [CPS-SPC15]

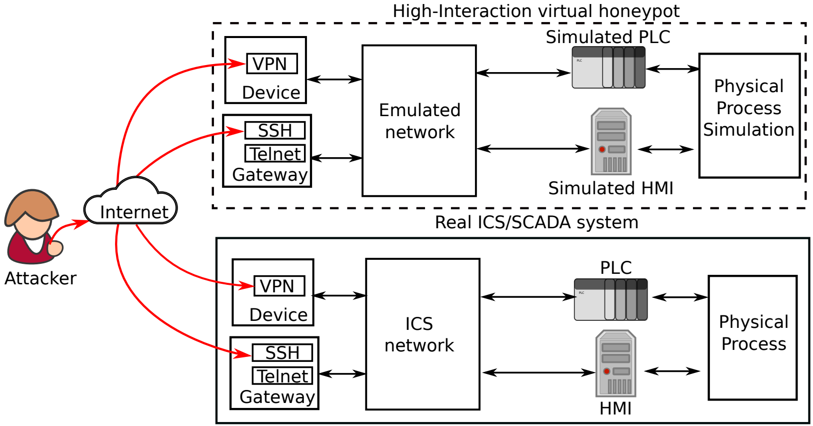
- Q1: Can we build a low-cost real-time simulation environment for CPS?



(C)yber ⟶ Network Emulation
(P)hysical ⟶ Physical Layer Simulation and API
(S)ystem ⟶ Simulation of Control Devices

Towards high-interaction virtual ICS honeypots-in-a-box [CPS-SPC16]

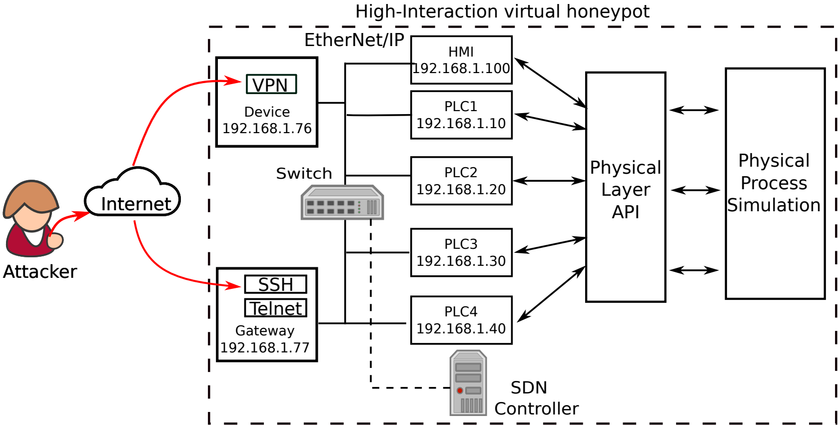
- Q2: Can we detect and mitigate cyber-physical attacks?



High Interaction ⟶ Simulate physical process and ICS devices
Virtual ⟶ Linux container virtualization
In–a-box ⟶ Runs on a single Linux kernel

Towards high-interaction virtual ICS honeypots-in-a-box [CPS-SPC16]

- Q2: Can we detect and mitigate cyber-physical attacks?



High Interaction ⟶ Simulate physical process and ICS devices
Virtual ⟶ Linux container virtualization
In–a-box ⟶ Runs on a single Linux kernel

Gamifying ICS Security Training and Research: Design, Implementation, and Results of S3 [CPS-SPC17]

- Q3: Can we fill the gaps between IT and OT security professionals?
- SWaT Security Showdown (S3) contest
  - ICS-centric, gamified security competition
  - We run it at SUTD in 2016 and 2017
  - IT and OT security professionals from academia and industry
- MiniCPS based security challenges
  - Evaluate MiniCPS as an educational tool
  - E.g. MitM attacks, sensor and actuator manipulations
- Main outcomes
  - Conducted (novel) attacks
  - Evaluated (novel) defenses

CPS includes Wireless Communication Systems

- Wireless systems (thesis's Part II)
  - Transmission and reception of electro-magnetic (EM) signals
  - Over a wireless physical layer (e.g. over the air)
- Pervasive use cases
  - Mobile communications: Wi-Fi, Bluetooth, and cellular
  - Localization: GPS and RFID

Wireless Systems Security

- Wireless systems security is important, yet hard
  - Wireless channel is *broadcast*
  - Threats: eavesdropping, jamming, etc.
- Recent high impact attacks
  - Wi-Fi: Key Reinstallation AttaCK (KRACK) on WPA2
  - Bluetooth: BlueBorne implementation flaws on Android and Linux

Our Wireless Security Challenges and Research Questions

- **C1: Wireless physical layer as a defense mechanism**
  - Q1: Can we leverage deployed physical layer features to secure communications? [CANS17]
- **C2: Complexity and accessibility of wireless technologies**
  - Q2: Can we analyze and evaluate (proprietary) wireless technologies? [NDSS19]
- **C3: Security evaluations and hardening of wireless technologies**
  - Q3: Can we harden already deployed technologies? [USEC19]

Our Wireless Security Challenges and Research Questions

- **C1: Wireless physical layer as a defense mechanism**
  - Q1: Can we leverage deployed physical layer features to secure communications? [CANS17]

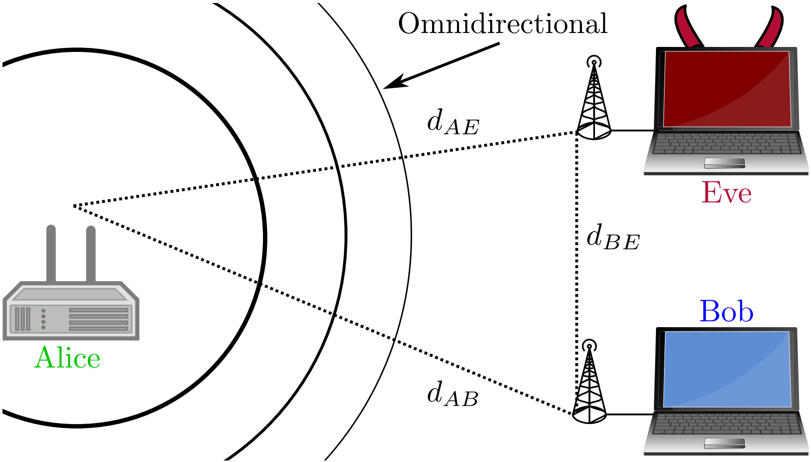
C1: Wireless physical layer as a defense mechanism

- Physical layer (PHY)
  - From bits to EM signals and vice versa
- Wireless PHY security
  - Security guarantees from some physical layer features
  - E.g. beamforming
- Q1: Can we leverage deployed physical layer features to secure communications?
  - *Practical Evaluation of Passive COTS Eavesdropping in 802.11b/n/ac WLAN* [CANS17]

Practical Evaluation of Passive COTS Eavesdropping in 802.11b/n/ac WLAN [CANS17]

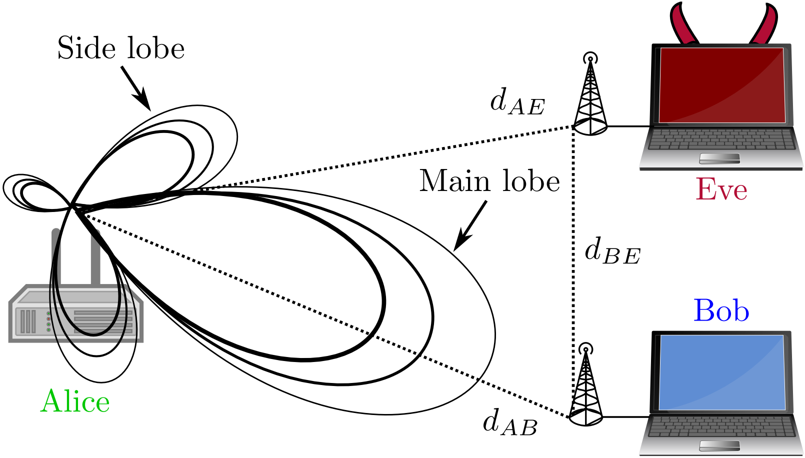
- IEEE 802.11 PHY features
  - 802.11b: single antenna, omnidirectional (SISO)
  - 802.11n/ac: multiple antenna, beamforming (MIMO)
- Threat model
  - Alice (access point) communicates with Bob (user)
  - Eve (attacker) wants to eavesdrop the downlink from Alice to Bob
- *Is Eve affected by 802.11n/ac PHY features compared to 802.11b?*
  - If yes, we should use it (together with crypto)

802.11b Downlink (SISO, omnidirectional)



- **802.11b**
  - Alice uses 1 antennas
  - Eve's eavesdropping success depends on: d_{AE}

802.11n/ac Downlink (MISO, beamforming)



- **802.11n/ac**
  - Alice uses L antennas to dynamically beamform towards Bob
  - Bob experiences a gain but Eve does not
  - Eve's eavesdropping success depends on: d_{AE}, d_{BE}, and L

Metrics

- Signal-to-Noise-Ratio (SNR)
  - Power of the useful signal divided by the noise power at the receiver
  - Usually expressed in dB ($10 \log_{10} \mathrm{SNR} = \mathrm{SNR}_{\mathrm{dB}}$)
- Bit-Error-Rate (BER)
  - Probability of erroneously decoding 1-bit at the receiver
  - Not an exact quantity (MCS, fading model)
  - 10^{-6} considered reasonable
- Packet-Error-Rate (PER)
  - $\mathrm{PER} = 1 - (1 - \mathrm{BER})^N$
  - N is the average packet size in bits

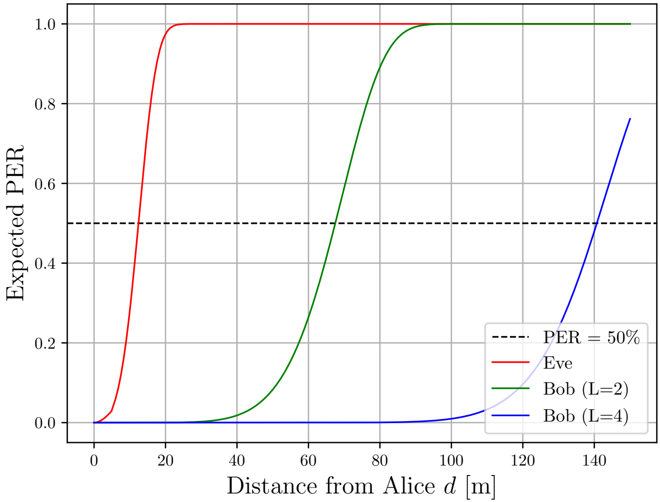
Predictions and Experiments

- 802.11n/ac (beamforming) vs. 802.11b (omnidirectional)
  - Eve targets the downlink from Alice to Bob
  - Is Eve affected by n/ac PHY features?
- Predictions (numerical analysis)
  - Eve's SNR disadvantage in b vs. n/ac
  - Eve's PER disadvantage compared to Bob in n/ac
- Experiments (COTS devices)
  - Measure PER and SNR of Eve and Bob
  - Compare the results with predictions

Wireless Path Loss Models

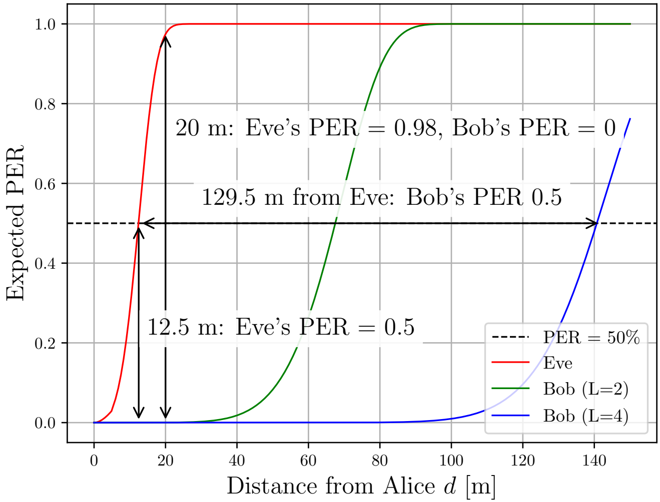
- Path loss model
  - Parametric simulation ot wireless links (indoor, outdoor)
  - d_{BP} is the breakpoint distance
  - σ_{SF} is the shadowing std dev (log-normal)
  - s_{PL} LOS and NLOS path loss slopes
- Model B: Residential (intra-room)
  - $d_{BP} = 5$ m
  - σ_{SF} = 3, 4 dB
  - s_{PL} = 2, 3.5
- Model D: Office (large conference room)
  - $d_{BP} = 10$ m
  - σ_{SF} = 3, 5 dB
  - s_{PL} = 2, 3.5

Model B (Residential) Expected PER



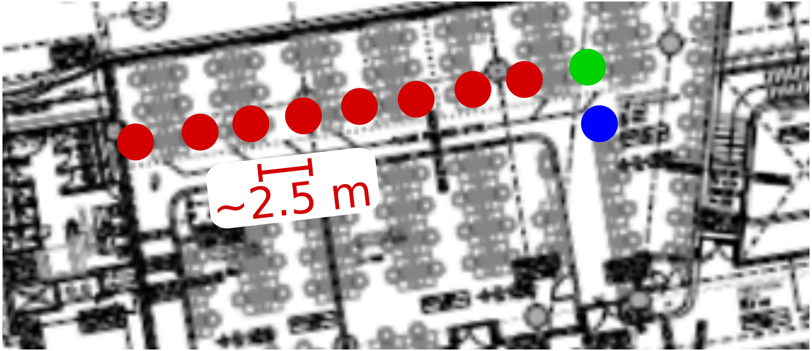
- PER of Eve, Bob(L=2) and Bob(L=4) in 802.11n (BPSK)

Model B (Residential) Expected PER



- PER of Eve, Bob(L=2) and Bob(L=4) in 802.11n (BPSK)

Experimental Office Layout (NLOS)

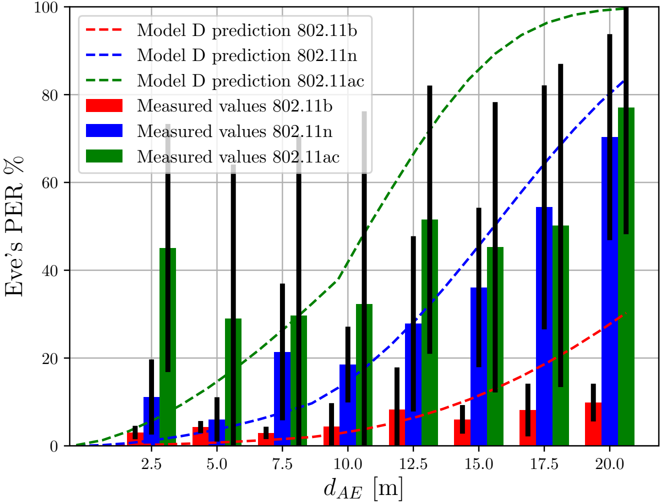


- Alice, Bob, and Eve locations
  - $d_{AB} = 2$ m
  - $\vec{d}_{AE} = [2.5, 5.0, \ldots, 20]$ m (8 distances)
  - $\Delta d_{AE} = 2.5$ m
  - Constant angle and elevation
  - NLOS (exploit multipath)

Experimental Setup: Traffic and Metrics

- UDP packets from Alice to Bob (targeted by Eve)
  - Wireshark running on Alice, Eve, and Bob
  - 30 repetitions per distance (2.5 m, 5.0 m, . . . , 20 m)
- SNR measurements
  - Received Signal Strength Indication (RSSI) and noise floor
  - From radiotap headers
- PER measurements
  - From incorrect UDP checksums
  - Over the total number of packet sent

Eve's Measured PER vs. Model D (Office)



- Eve's PER is *increasing among 802.11b/n/ac*

Conclusions about 802.11 Eavesdropping

- Q1: Can we leverage deployed physical layer features to secure communications?
  - *Yes, 802.11n/ac PHY features disadvantage an eavesdropper*
- Predicted 802.11n/ac disadvantages for Eve
  - SNR is bounded by 6-41 dB
  - PER increases to 98% when $d_{AE} > 20$ m
  - Eve has to be 129.5 m closer to get same performance as Bob
- Experimental results about Eve
  - PER increases significantly when $d_{AE} > 15$ m
  - PER is 20% higher in 802.11n than in 802.11b
  - PER is 30% higher in 802.11ac than in 802.11b

Our Wireless Security Challenges and Research Questions

- **C1: Wireless physical layer as a defense mechanism**
  - Q1: Can we use physical layer features to build security mechanisms? [CANS17]
- **C2: Complexity and accessibility of wireless technologies**
  - Q2: Can we analyze and evaluate (proprietary) wireless technologies? [NDSS19]
- **C3: Security evaluations and hardening of wireless technologies**
  - Q3: Can we harden already deployed technologies? [USEC19]

Our Wireless Security Challenges and Research Questions

- **C2: Complexity and accessibility of wireless technologies**
  - Q2: Can we analyze and evaluate (proprietary) wireless technologies? [NDSS19]

C2: Complexity and accessibility of wireless technologies

- Wireless technologies are complex
  - Specifications have amendments (revisions)
  - Different implementations of a specification
- Wireless technologies are difficult to access
  - Proprietary specifications
  - Closed-source implementations
- Q2: Can we analyze and evaluate (proprietary) wireless technologies?
  - *Nearby Threats: Reversing, Analyzing, and Attacking Google's 'Nearby Connections' on Android* [NDSS19]

Nearby Threats: Reversing, Analyzing, and Attacking Google’s ‘Nearby Connections’ on Android [NDSS19]

- Nearby Connections
  - API for Android and Android Things
  - In-app proximity-based services
- Implemented in the Google Play Services
  - Available across different Android versions
  - Applications use it as a shared library

Google Nearby

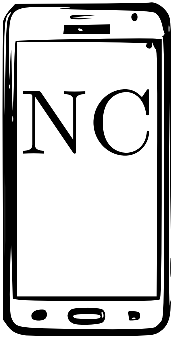
Why Analyzing Nearby Connections?

- Wide attack surface
  - Any Android (version $\geq$ 4.0) and Android Things device
  - Uses Bluetooth and Wi-Fi (even at the same time)
- Proprietary technology
  - No public specifications
  - Implementation is closed-source and obfuscated

Our Core Contributions

- First (security) analysis of Nearby Connections
  - Uncovers its proprietary mechanisms and protocols
  - Based on reversing its Android implementation
- Re-implementation of Nearby Connections (REarby)
  - Exposes parameters not accessible with the official API
  - Impersonates nearby devices from any application
- Attacking Nearby Connections on Android
  - Connection manipulation and range extension attacks
  - Responsible disclosure with Google

Nearby Connections Public Information

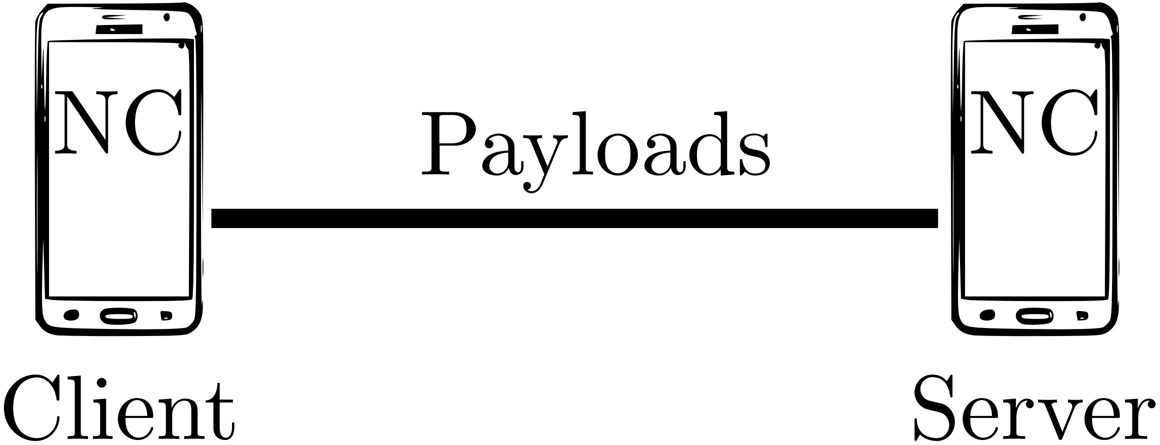


Client

Server

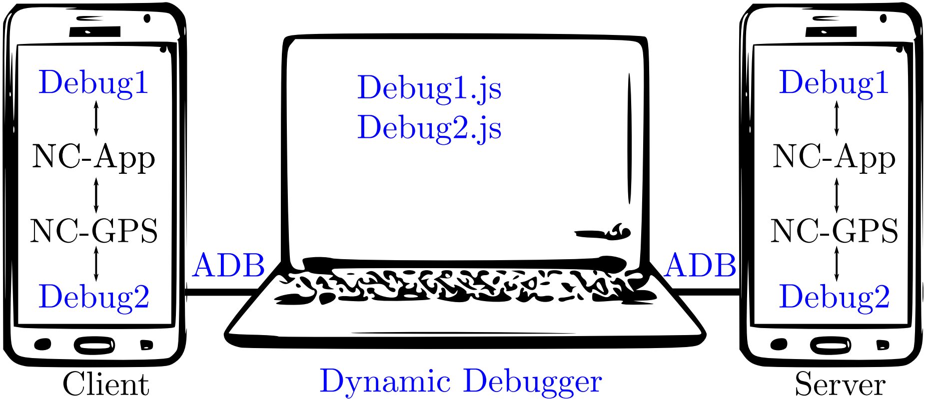
- The server advertises a service (`sid`) and the client discovers it
- Two connection strategies: `P2P_STAR` and `P2P_CLUSTER`

Nearby Connections Public Information 2



- Automatic connection using Bluetooth and/or Wi-Fi
- Node exchanges encrypted payloads (peer-to-peer)

Our Dynamic Binary Instrumentation



- Workhorse: Frida, `https://www.frida.re`
  - Profiling of processes, e.g. NC-App, NC-GPS
  - Hook function and methods calls
  - Override parameters and return values
  - Read and write processes' memory

Reversed Phases of a Nearby Connection

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret
4. **Optional Authentication**: Based on the shared secret

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret
4. **Optional Authentication**: Based on the shared secret
5. **Application Layer Connection Establishment**: Interactive

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret
4. **Optional Authentication**: Based on the shared secret
5. **Application Layer Connection Establishment**: Interactive
6. **Key Derivation Functions**: Session, AES and HMAC keys

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret
4. **Optional Authentication**: Based on the shared secret
5. **Application Layer Connection Establishment**: Interactive
6. **Key Derivation Functions**: Session, AES and HMAC keys
7. **Optional Physical Layer Switch**: Bluetooth to Wi-Fi

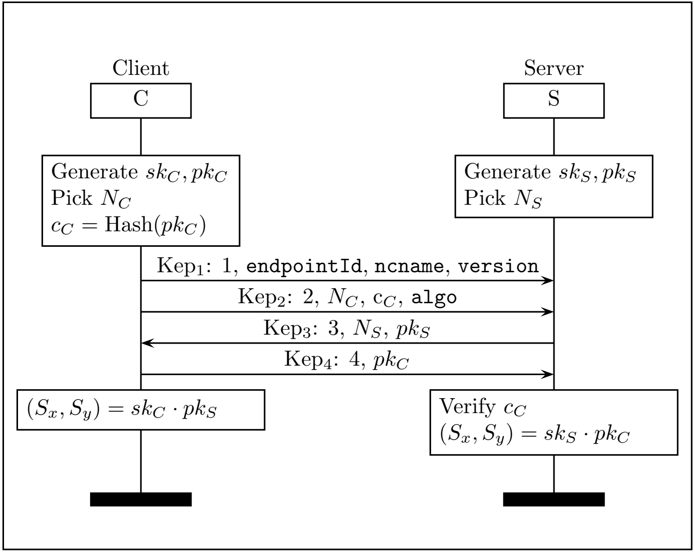
Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret
4. **Optional Authentication**: Based on the shared secret
5. **Application Layer Connection Establishment**: Interactive
6. **Key Derivation Functions**: Session, AES and HMAC keys
7. **Optional Physical Layer Switch**: Bluetooth to Wi-Fi
8. **Exchange Encrypted Payloads**: Proximity-based service

Reversed Phases of a Nearby Connection

1. **Discovery**: Bluetooth name (BR/EDR) and BLE reports
2. **Connection Request**: automatic over Bluetooth, not authenticated
3. **Key Exchange Protocol**: Establishment of a shared secret
4. **Optional Authentication**: Based on the shared secret
5. **Application Layer Connection Establishment**: Interactive
6. **Key Derivation Functions**: Session, AES and HMAC keys
7. **Optional Physical Layer Switch**: Bluetooth to Wi-Fi
8. **Exchange Encrypted Payloads**: Proximity-based service
9. **Disconnection**: automatic after a 30 seconds timeout

Key Exchange Protocol (KEP)



- Based on ECDH, NIST P256 curve, shared secret is S_x

Optional Physical Layer Switch

- **Bluetooth to soft access point (Wi-Fi Direct, hostapd)**
  - Server instructs the client over Bluetooth (e.g. ESSID, password)
  - Client contacts the server over Wi-Fi

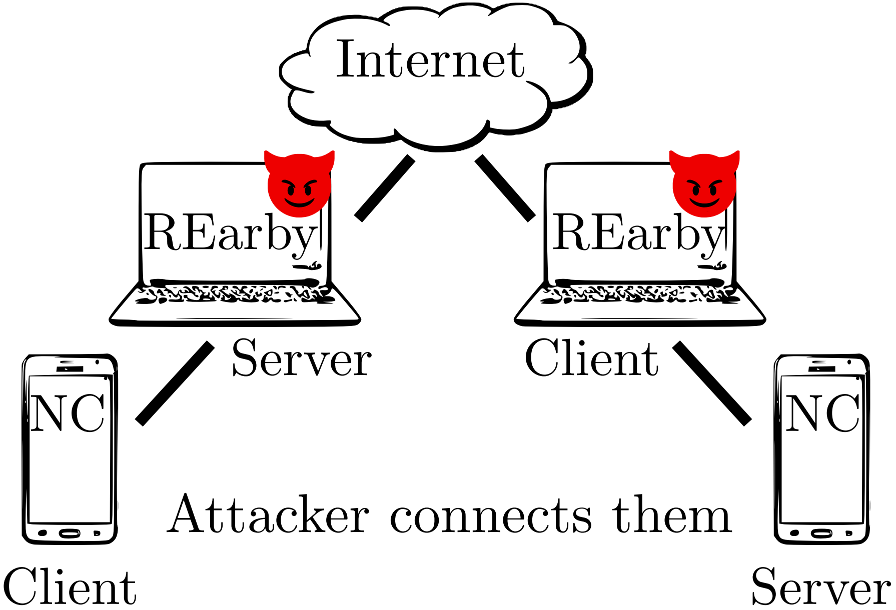
Range Extension MitM Attack

Client

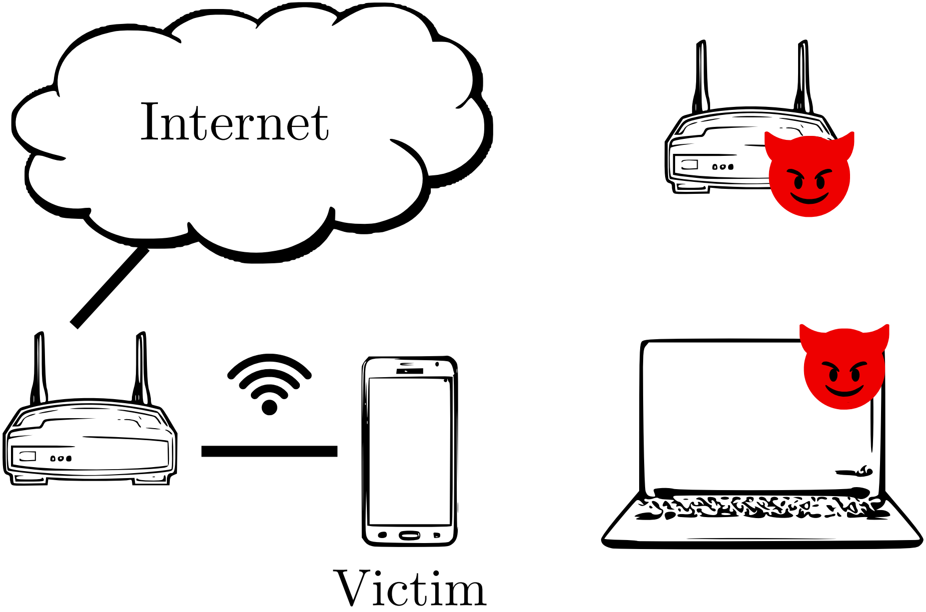
Victims are not nearby

Server

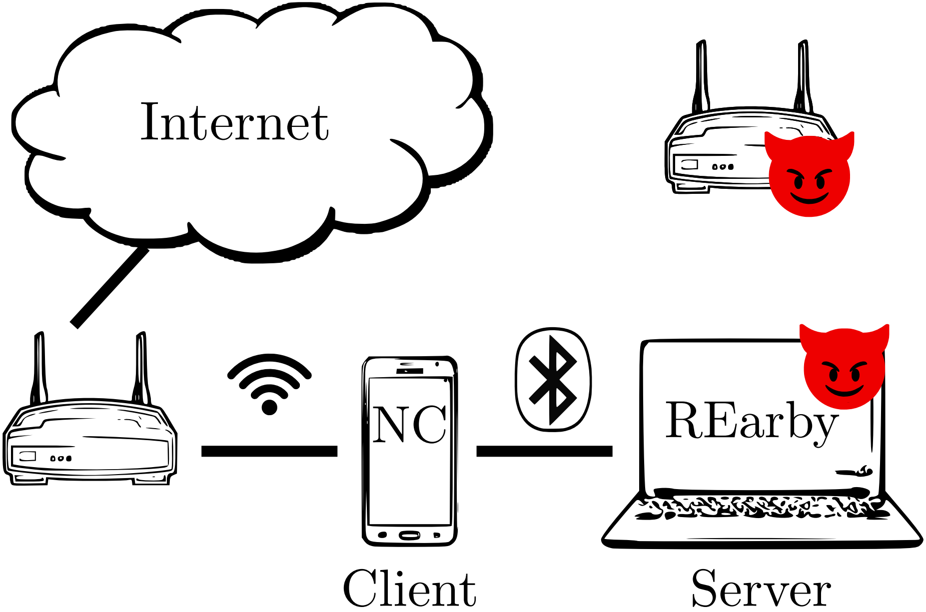
Range Extension MitM Attack



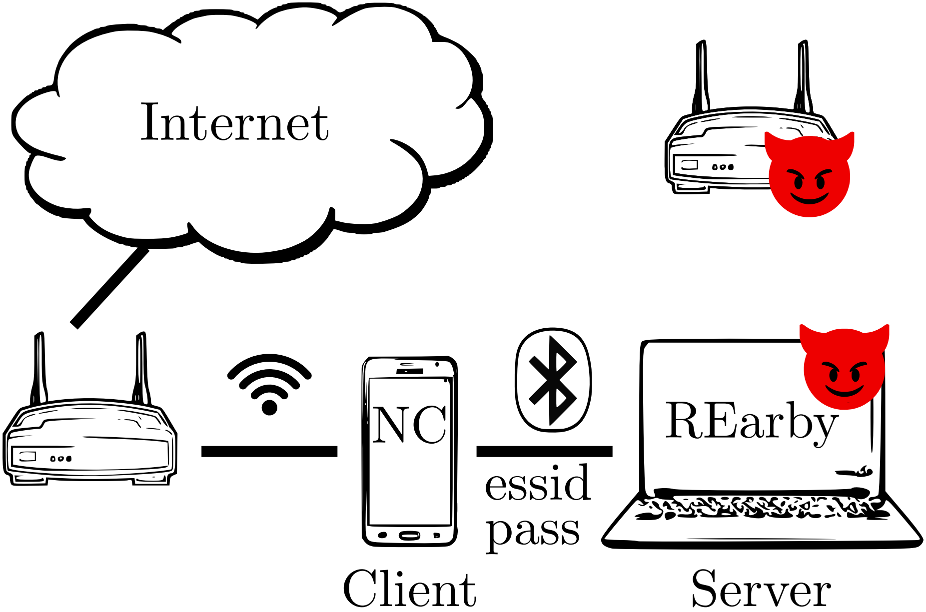
Soft Access Point Manipulation Attack



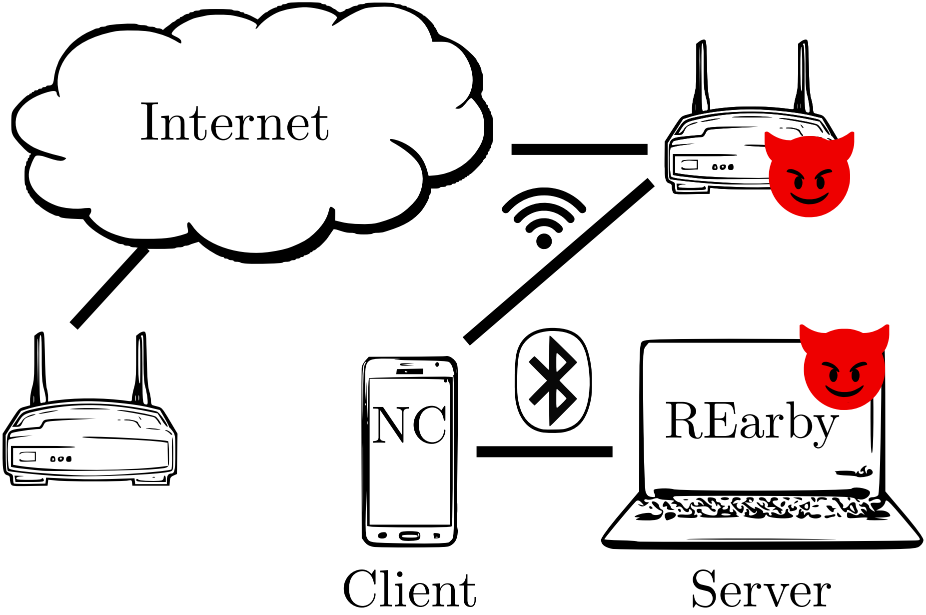
Victim Connects to Attacker's REarby Server



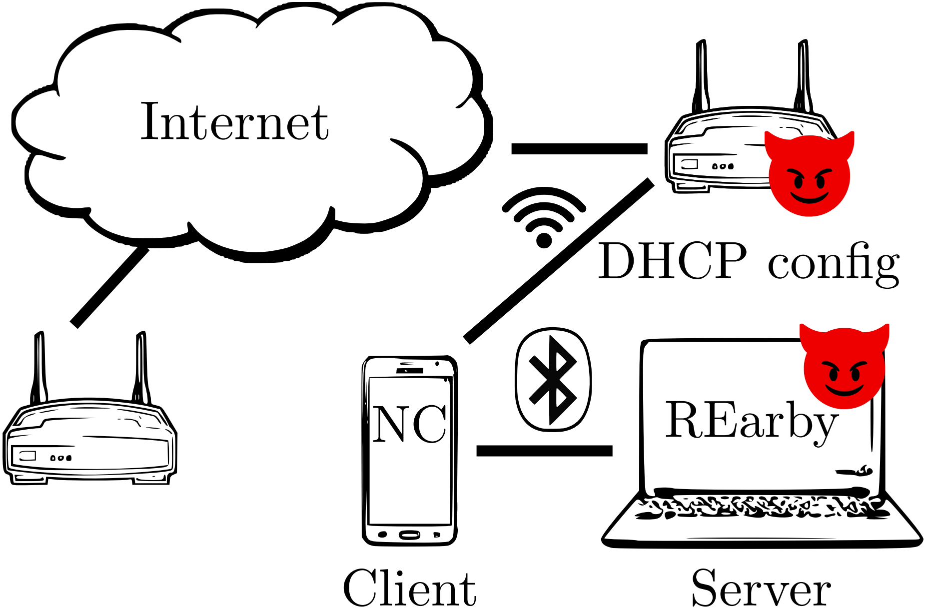
Attacker Manipulates Bluetooth to Wi-Fi Switch



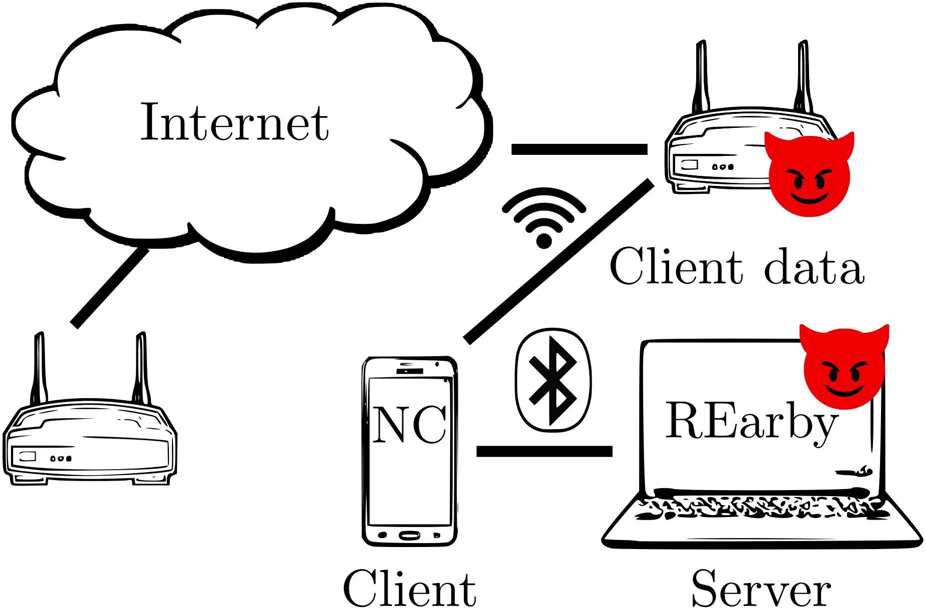
Victim Connects to Attacker's Wi-Fi AP



Attacker Configures Victim's Network Interface



Attacker Eavesdrops All Wi-Fi Traffic



Conclusions about Nearby Connections

- Q2: Can we analyze and evaluate (proprietary) wireless technologies?
  - *Yes, and they should not use security through obscurity.*
- First security analysis of Nearby Connections
  - Android and Android Things API for proximity-based services
- Reversed its Android implementation and re-implemented it
  - REarby `https://francozappa.github.io/project/rearby/`
- Demonstrate attacks and proposed countermeasures
  - Range extension MitM: authenticate nodes and check proximity
  - Soft access point manipulation: authenticate nodes

Conclusion and Q&A

- **CPS security contributions (Thesis Part I, Chapter 1-5)**
  - C1: Evaluation of CPS (IT and OT) technologies
    - *MiniCPS: A toolkit for security research on CPS networks* [CPS-SPC15]
    - *Legacy-Compliant Data Authentication for Industrial Control System Traffic* [ACNS17]
  - C2: Cyber-physical attacks
    - *Towards high-interaction virtual ICS honeypots-in-a-box* [CPS-SPC16]
    - *State-Aware Anomaly Detection for Industrial Control Systems* [SAC18]
  - C3: CPS security education
    - *Gamifying ICS Security Training and Research: Design, Implementation, and Results of S3* [CPS-SPC17]
- **Wireless systems security contributions (Thesis Part II, Chapter 6-10)**
  - C1: Wireless physical layer as a defense mechanism
    - *Practical Evaluation of Passive COTS Eavesdropping in 802.11b/n/ac WLAN* [CANS17]
  - C2: Complexity and accessibility of wireless technologies
    - *Nearby Threats: Reversing, Analyzing, and Attacking Google's 'Nearby Connections' on Android* [NDSS19]
  - C3: Security evaluations and hardening of wireless technologies
    - *The KNOB is broken: Exploiting low entropy in the encryption key negotiation of Bluetooth BR/EDR* [USEC19]

Thanks for your time! Questions? More at: `https://francozappa.github.io`

Our Wireless Security Challenges and Research Questions

- **C1: Wireless physical layer as a defense mechanism**
  - Q1: Can we leverage deployed physical layer features to secure communications? [CANS17]
- **C2: Complexity and accessibility of wireless technologies**
  - Q2: Can we analyze and evaluate (proprietary) wireless technologies? [NDSS19]
- **C3: Security evaluations and hardening of wireless technologies**
  - Q3: Can we harden already deployed technologies? [USEC19]

Our Wireless Security Challenges and Research Questions

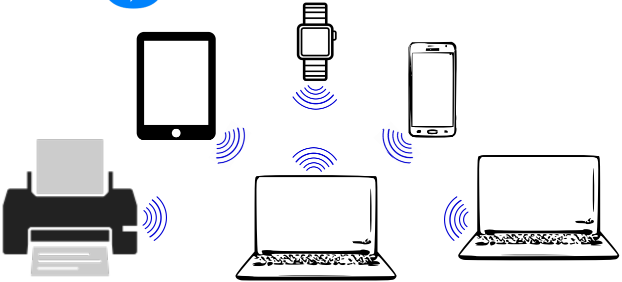
- **C3: Security evaluations and hardening of wireless technologies**
  - Q3: Can we harden already deployed technologies? [USEC19]

C3: Security evaluations and hardening of wireless technologies

- Bluetooth is a pervasive wireless technology
  - Wide attack surface: IT, mobile, automotive, medical, and industrial
- Bluetooth security posture
  - Open specification
  - Custom security mechanisms
  - No public reference implementation
- Q3: Can we evaluate and harden already deployed technologies?
  - *The KNOB is broken: Exploiting low entropy in the encryption key negotiation of Bluetooth BR/EDR* [USEC19]

The KNOB is broken: Exploiting low entropy in the encryption key negotiation of Bluetooth BR/EDR [USEC19]

- *Bluetooth BR/EDR (Basic Rate/Extended Data Rate)*
  - P2P, master-slave
  - Better performance, yet less battery life than Bluetooth Low Energy (BLE)

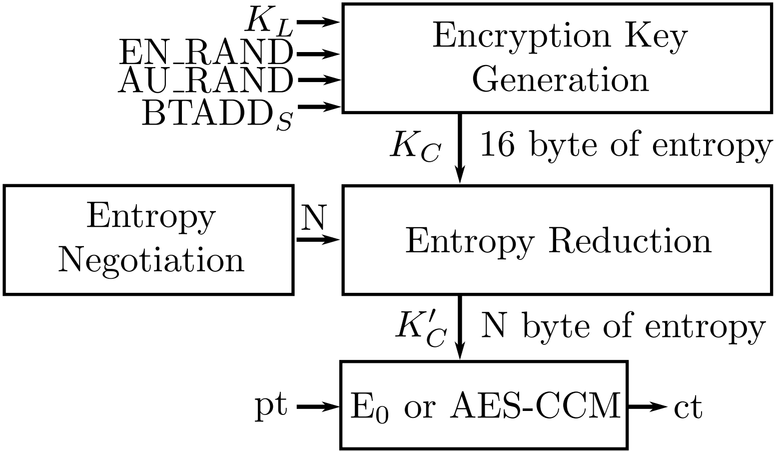


Bluetooth BR/EDR's Security

- Bluetooth BR/EDR link layer security guarantees
  - Confidentiality, integrity, and authentication
- *Secure Simple Paring (SSP)*, since Bluetooth v2.1
  - Pairing to generate a *link key* (long term secret)
  - ECDH and nonce-based key authentication
  - Session keys derived from the link key (AES, HMAC)
- *Secure Connections (SC)*, since Bluetooth v4.1
  - AES-CCM rather than E0
  - P-256 curve rather than P-192 curve

Key Negotiation of Bluetooth (KNOB)

- Paired devices share K_L and negotiate a new K'_C per connection



- Q: **What is the smallest yet standard-compliant N?**

KNOB from the Bluetooth core spec v5.0 (page 1650)

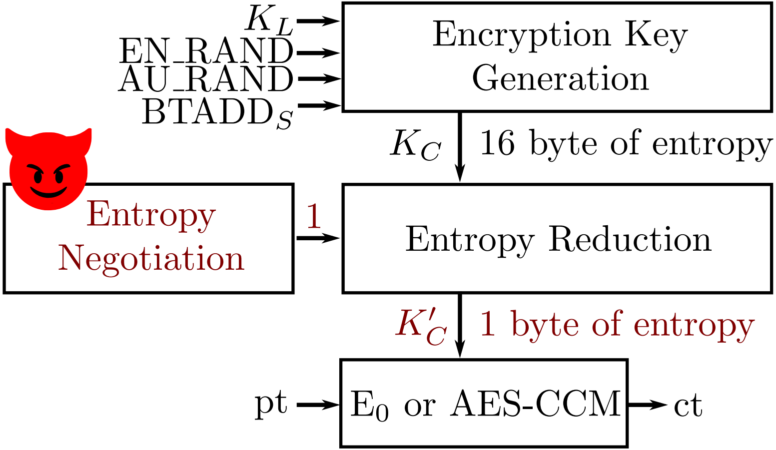
> *"For the encryption algorithm, **the key size may vary between 1 and 16 octets (8-128 bits)**. The size of the encryption key is configurable for two reasons. The first has to do with the many **different requirements imposed on cryptographic algorithms in different countries** - both with respect to export regulations and official attitudes towards privacy in general. The second reason is to **facilitate a future upgrade** path for the security without the need of a costly redesign of the algorithms and encryption hardware; **increasing the effective key size is the simplest way to combat increased computing power at the opponent side**."*

https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043

- Q: **How hard is to decrease the key size (entropy) to 1 Byte?**

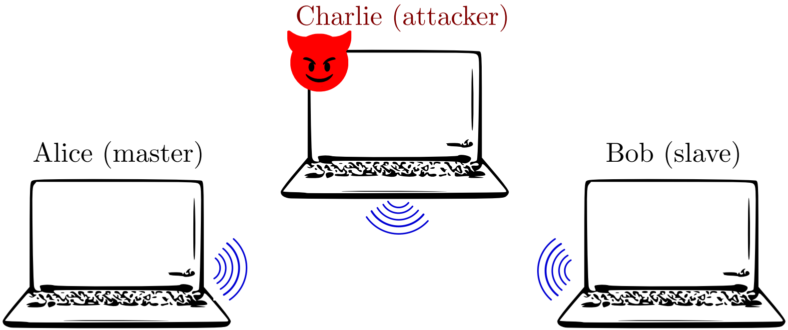
Our Contribution: the KNOB Attack

- How hard is to adversarially set N=1 (break the KNOB)?



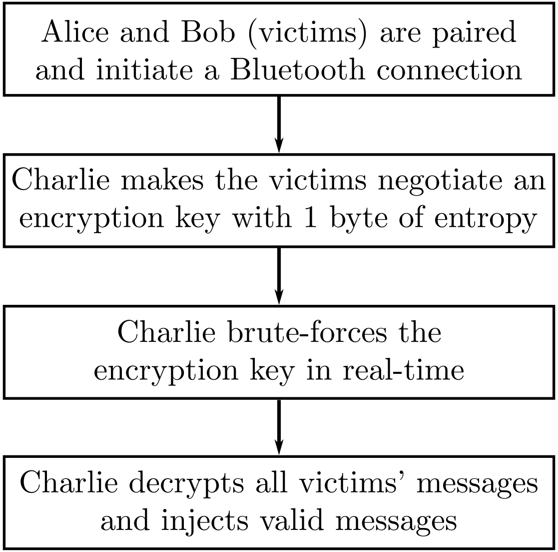
- Well, **we demonstrated that the KNOB is broken**

Threat Model



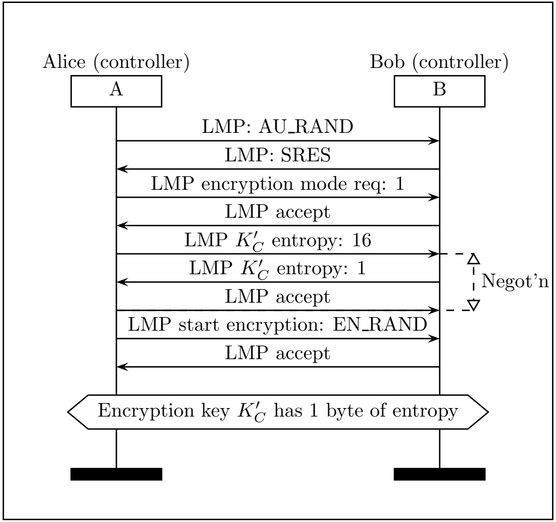
- Alice (master) establishes a secure Bluetooth connection with Bob (slave)
  - Victims already performed pairing (they share K_L)
  - Link layer is encrypted (using K'_C)
- Charlie (attacker)
  - In range with the Alice and Bob
  - Wants to eavesdrop and manipulate the victims's information

KNOB Attack Stages



Entropy Negotiation is Not Integrity Protected

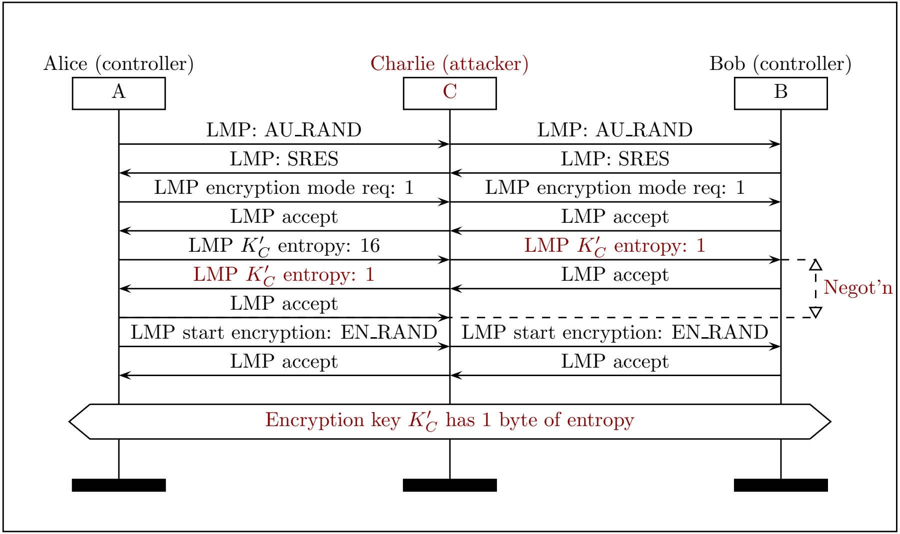
- Devices negotiate N, between 1 and 16, according to their L_{min} and L_{max}



- **Over the air LMP packets are not integrity protected**

Adversarial Entropy Negotiation

- Charlie (attacker) forces Alice and Bob to negotiate N=1



Brute Forcing the Encryption Key (K'_C)

- Alice and Bob
  - Use an encryption key (K'_C) with 1 Byte of entropy
  - K'_C is one within 256 candidates
- Charlie
  - Eavesdrops the ciphertext
  - Tests the 256 K'_C candidates against the ciphertext (in parallel)
  - Use K'_C to decrypt all packets and inject new packets

Example of a KNOB Attack Scenario

- Victims: Nexus 5 and Motorola G3 (SSP, no SC)
- Attacker: ThinkPad X1 and Ubertooth (Bluetooth sniffer)
- Attacker decrypts a file exchanged over a secure Bluetooth link (OBEX)

KNOB Attack Evaluation

- The KNOB attack is at the *architectural* level
  - All standard compliant Bluetooth devices are (potentially) vulnerable
  - Regardless their implementations, SSP, and SC
- KNOB Attack Evaluation
  - We tested all the Bluetooth devices that we had access to

Vulnerable chips and devices (Bluetooth 5.0, 4.2)

Bluetooth chip	Device(s)	Vulnerable?
Bluetooth Version 5.0		
Snapdragon 845	Galaxy S9	✓
Snapdragon 835	Pixel 2, OnePlus 5	✓
Apple/USI 339S00428	MacBookPro 2018	✓
Apple A1865	iPhone X	✓
Bluetooth Version 4.2		
Intel 8265	ThinkPad X1 6th	✓
Intel 7265	ThinkPad X1 3rd	✓
Unknown	Sennheiser PXC 550	✓
Apple/USI 339S00045	iPad Pro 2	✓
BCM43438	RPi 3B, RPi 3B+	✓
BCM43602	iMac MMQA2LL/A	✓

✓ = Entropy of the encryption key (K'_C) reduced to 1 Byte

Vulnerable chips and devices (Bluetooth 4.1 and below)

Bluetooth chip	Device(s)	Vulnerable?
Bluetooth Version 4.1		
BCM4339 (CYW4339)	Nexus5, iPhone 6	✓
Snapdragon 410	Motorola G3	✓
Bluetooth Version $\leq$ *4.0*		
Snapdragon 800	LG G2	✓
Intel Centrino 6205	ThinkPad X230	✓
Chicony Unknown	ThinkPad KT-1255	✓
Broadcom Unknown	ThinkPad 41U5008	✓
Broadcom Unknown	Anker A7721	✓
Apple W1	AirPods	*

✓ = Entropy of the encryption key (K'_C) reduced to 1 Byte
* = Entropy of the encryption key (K'_C) reduced to 7 Byte

Countermeasures for the KNOB Attack

- *Legacy compliant* (do not require to change the specification)
  - Set N to 16 (set $L_{min} = L_{max} = 16$)
  - Check N from the host (OS) upon connection
  - Security mechanisms on top of the link layer
- *Non legacy compliant*
  - Secure entropy negotiation with K_L (ECDH shared secret)
  - Get rid of the entropy negotiation protocol

Conclusion

- Discovered an architectural vulnerability of Bluetooth BR/EDR
  - The entropy of any encryption key can be reduced to 1 Byte
  - All standard compliant devices are (potentially) vulnerable
- Demonstrated the exploitability of this vulnerability
  - Key Negotiation Of Bluetooth (KNOB) attack
  - Evaluated on more than 14 chips (e.g. Intel, Broadcom, Apple, Qualcomm)
- Provided effective countermeasures (while doing disclosure)
  - Legacy and non legacy compliant
  - Today the embargo is over and the KNOB should be fixed

`https://github.com/francozappa/knob`

- Thanks for your time! Questions?