E-Spoofer: Attacking and Defending Xiaomi Electric Scooter Ecosystem

WiSec 2023, Guildford, Surrey (UK), May 29 - June 01

Marco Casagrande*, Riccardo Cestaro, Eleonora Losiouk, Mauro Conti, and Daniele Antonioli*

EURECOM *
Sophia Antipolis

Università degli Studi di Padova

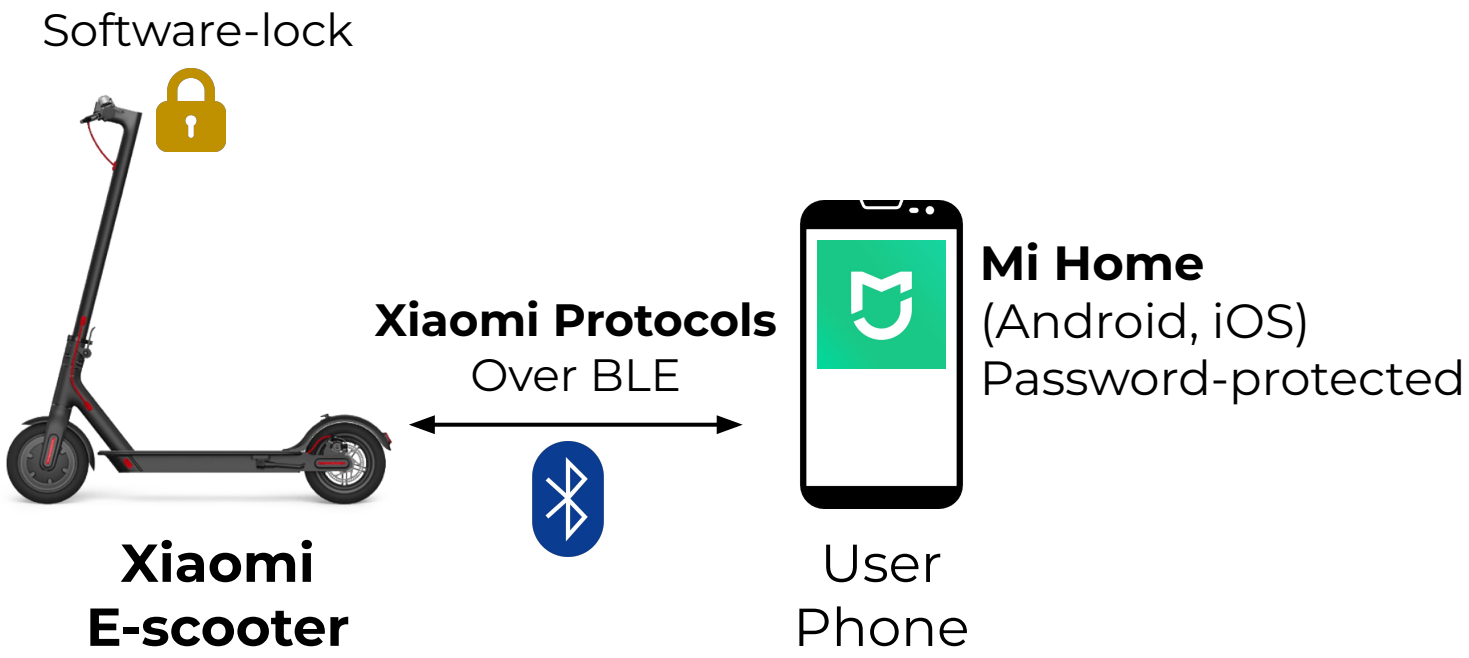
Motivation

- **E-scooters** are a critical **wireless** attack surface
  - Security (theft), privacy (data leak), safety (break)
- We know little about their security mechanisms
  - Proprietary, undocumented, untestable
- Millions of e-scooters and users
  - Controlled by a couple of companies (e.g., **Xiaomi**)
- One attack on Xiaomi has a huge impact
  - E.g., 2019 Zimperium remote braking system exploit (ref)

Contributions

- RE all **Xiaomi e-scooter protocols** since 2016
  - Pairing and Session phases
- Uncover critical **protocol-level vulnerabilities**
  - E.g., unauthorized pairing, no password enforcement
- **Proximity** and **remote** wireless attacks
  - Malicious Pairing (MP), Session Downgrade (SD)
- **E-Spoofer** open-source toolkit
  - Reproduce the attacks, tamper with protocols
- **Countermeasures** and **disclosure** to Xiaomi

System Model



Attacker Models

Proximity Attacker

Xiaomi Protocols
Over BLE

Xiaomi
E-scooter

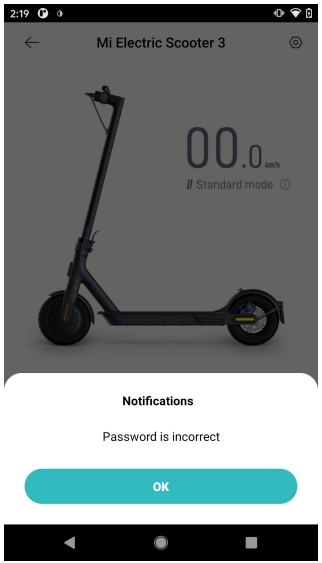
Xiaomi Protocols
Over BLE

Remote Attacker (Android app)

User
Phone

Attacker Goals

Spoof Mi Home to the e-scooter. Send arbitrary and unauthorized **read** and **write** commands without user consent and notice.



Xiaomi E-Scooter Protocols Introduction

- **P1, P2, P3, P4 (since 2016)**
  - *Application-layer* Pairing and Session phases
  - *No BLE link-layer* security
- **Pairing** phase
  - Devices agree on a **Pairing Key** (**PK**)
- **Session** phase
  - Devices compute a **Session Key** (**SK**) from PK
  - Devices use SK to establish a secure channel

P1: No Security Mechanisms

NO PAIRING

NO SESSION

BLE Connection Req

BLE Connection Resp

Read/Write Req

Read/Write Resp

Proprietary BLE GATT characteristics and binary data over GATT

P1: Proximity/Remote Attacks

NO PAIRING

NO SESSION

Spoofed BLE Connection Req

BLE Connection Resp

Arbitrary Read/Write Req

Read/Write Resp

Read all data: **password, mileage.**

Write all data: **lock/unlock**, **password, max speed.**

P2: Public PK and XOR Obfuscation

PAIRING

BLE Connection Req

BLE Connection Resp

Read `pk_characteristic`

PK [24 B]

SESSION

Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

Characteristic exposes new **PK** every boot.

Obfuscation by XORing **PK**.

P2: Proximity/Remote Attacks

PAIRING

Spoofed BLE Connection Req

BLE Connection Resp

Read `pk_characteristic`

PK [24 B]

SESSION

Arbitrary Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

PK exposed **publicly**.

No encryption but XOR obfuscation.

P2: Proximity/Remote Attacks

MALICIOUS PAIRING

Spoofed BLE Connection Req

BLE Connection Resp

Read `pk_characteristic`

PK [24 B]

SESSION

Arbitrary Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

PK exposed **publicly**.

No encryption but XOR obfuscation

P2: Proximity/Remote Attacks

SESSION

Spoofed BLE Connection Req

BLE Connection Resp

p2_session_downgrade_cmd

Arbitrary Read/Write Req

Read/Write Resp

Session **downgraded to P1** (more insecure).

P2: Proximity/Remote Attacks

SESSION DOWNGRADE

Spoofed BLE Connection Req

BLE Connection Resp

`p2_session_downgrade_cmd`

Arbitrary Read/Write Req

Read/Write Resp

Session **downgraded to P1** (more insecure).

P3: Const PK and XOR Obfuscation

PAIRING

esc_name

BLE Connection Req

BLE Connection Resp

PK = AES-ECB(key=**const**,inp=**esc_name**)

PK derived from **esc_name**, and **const** from Mi Home.

SESSION

Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

P3: Proximity/Remote Attacks

PAIRING

esc_name

Spoofed BLE Connection Req

BLE Connection Resp

PK = AES-ECB(key=**const**,inp=**esc_name**)

SESSION

Arbitrary Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

PK derived from **publicly** known data.

RE **const** from Mi Home.

Recap: P1, P2, P3 insecurity

- **P1**, **P2**, **P3** are **insecure by design**
  - Security through obscurity (XOR, public seeds, binary data)
  - **Proximity/remote impersonation is trivial**
- **P4** to the rescue?
  - **NOT** really

P4: Pairing (ECDH, AES-CCM)

PAIRING

chal, **PUB_e**

PUB_a

SS = ECDH(**PUB_e**,**PRV_a** or **PUB_a**,**PRV_e**)
PK||**OTK** = HKDF(key=**SS**,inp=**const**)

resp = AES-CCM(key=**OTK**,inp=**chal**)

ECDH to generate **PK** from **SS**.

App auth using **OTK**.

P4: Proximity/Remote Attacks

PAIRING

chal, **PUB_e**

PUB_a

Unauthorized and **non authenticated** pairing.

SS = ECDH(**PUB_e**,**PRV_a** or **PUB_a**,**PRV_e**)
PK||OTK = HKDF(key=**SS**,inp=**const**)

resp = AES-CCM(key=**OTK**,inp=**chal**)

P4: Proximity/Remote Attacks

MALICIOUS PAIRING

chal, **PUB_e**

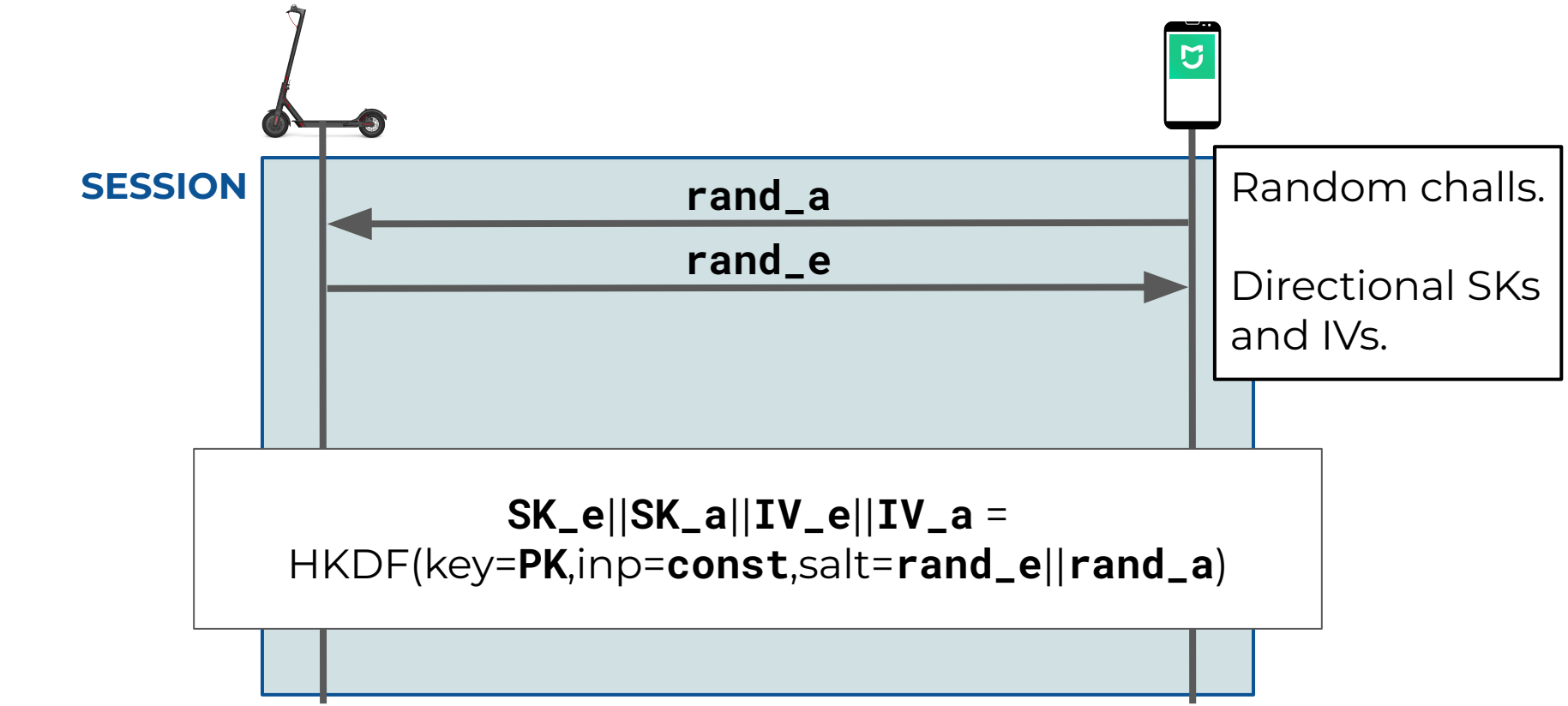
PUB_a

Unauthorized and **non authenticated** pairing.

SS = ECDH(**PUB_e**,**PRV_a** or **PUB_a**,**PRV_e**)
PK||OTK = HKDF(key=**SS**,inp=**const**)

resp = AES-CCM(key=**OTK**,inp=**chal**)

P4: Session (HKDF, AES-CCM) (1)



P4: Session (HKDF, AES-CCM) (2)

SESSION

resp_a = HKDF(key=**SK_a**,inp=**rand_a**||**rand_e**)

resp_e = HKDF(key=**SK_e**,inp=**rand_e**||**rand_a**)

AES-CCM(key=**SK_a**,IV=**IV_a**||**cnt**, inp=**Read/Write Req**)

AES-CCM(key=**SK_e**,IV=**IV_e**||**cnt**, inp=**Read/Write Resp**)

Mutual session auth.

Encrypted and integrity protected comms.

P4: Proximity/Remote Attacks

SESSION

p4_session_downgrade_cmd

esc_name

PK = AES-ECB(key=**const**,inp=**esc_name**)

Arbitrary Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

Session **downgraded to P3** (more insecure).

P4: Proximity/Remote Attacks

SESSION DOWNGRADE

p4_session_downgrade_cmd

esc_name

PK = AES-ECB(key=**const**,inp=**esc_name**)

Arbitrary Read/Write Req XOR **PK**

Read/Write Resp XOR **PK**

Session **downgraded to P3** (more insecure).

Implementing the attacks: E-Spoofer

- **E-Spoofer** is open-source
  - Automated Proximity MP (link)
  - Automated Remote SD (link)
- **Reversed BLE firmware** on Ghidra
- Xiaomi protocol **dissectors**
- **Frida hooks** for Mi Home crypto calls
- WiSec Artifact approval

Evaluation Setup

M365

Essential

Mi 3

- 5 BLE boards (M365, Pro 1, Pro 2, Essential, Mi 3)
- 8 BLE firmware (P1, P2, P3, P4)

Evaluation Results

E-scooter	BLE Board	BLE Fw	Protocol	Strategy	Prox/Rem Adv.	
					Spoof Mi Home	*Arb R/W*
M365	M365	072	P1	RE	✓	✓
M365	M365	081	P2	RE, MP, SD	✓	✓
M365	Pro 1	090	P3	RE	✓	✓
M365	M365	122	P4v1	RE, MP, SD	✓	✓
M365	Pro 2	129	P4v1	RE, MP, SD	✓	✓
Essential	Essential	152	P4v1	RE, MP, SD	✓	✓
Mi 3	Mi 3	153	P4v1	RE, MP, SD	✓	✓
Mi 3	Mi 3	157	P4v2	RE, MP	✓	✓

Countermeasures

- Update firmware via Mi Home
  - From P1, P2, P3 to P4v1 or P4v2
- **Password-protected and authorized Pairing**
  - Addresses MP on P4v1 and P4v2
  - More details in Section 8.1
- **Anti-downgrade patching script for BLE fw**
  - Addresses SD on P4v1
  - Evaluated on a real M365
  - More details in Section 8.2

Conclusion and Q&A

- RE all **Xiaomi e-scooter protocols** since 2016
  - Pairing and Session phases
- Uncover critical **protocol-level vulnerabilities**
  - Unwanted pairing, weak authentication
- **Proximity** and **remote** wireless attacks
  - Malicious pairing, session downgrade
- **E-Spoofer** open-source toolkit
  - Reproduce the attacks, tamper with protocols
- **Countermeasures** and **disclosure** to Xiaomi