~~BLE authentication design challenges on smartphone controlled IoT devices: analyzing Gogoro Smart Scooter~~

Is Smart Phone a secure key for vehicle?

GD、CSC

Privacy and Risk Management Lab
IM, NTUST, Taiwan

Speakers

G D

- Graduate Student at NTUST IM
- CHROOT/HITCON Coordinator
- Team T5 CTO
  - Digital Forensics & Incident Response
  - Threat Intelligence Program & Plat.
- Research on Foods, plays CTFs
- Occasionally got vulnerabilities
  - Synology Bounty Program (2015)

TEAM T5
Cyber Security Research

CSC

- Associate Professor at NTUST IM
- Ph.D., Dept of IM, NTU
- Gomaji (TW.8472) Board member
- CISSP, CCFP, CSSLP, CISM, PMP
- Published many practical security papers on journals. Helped many private and public sectors to establish info security policy.

HITCON

Hacks In Taiwan Conference

- CTF Team
  - DEF CON 2nd Place
- CTF Event
  - DEF CON Qualifier

- Community Conf
- Enterprise Conf
- Girls Conf

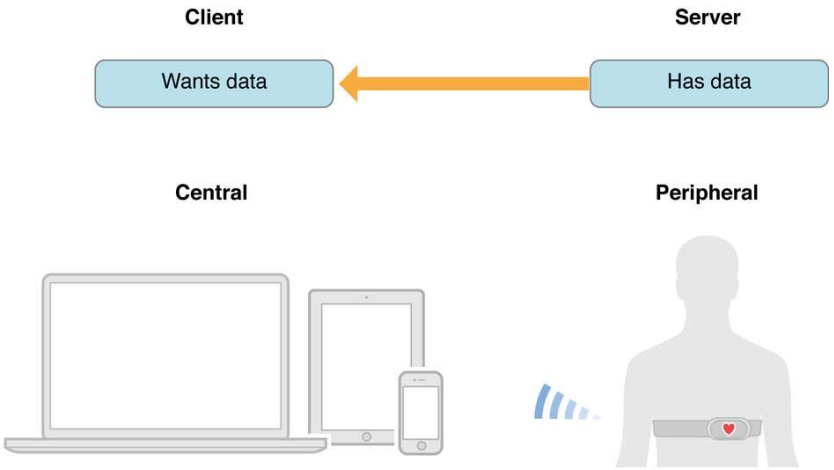
HITCON GIRLS

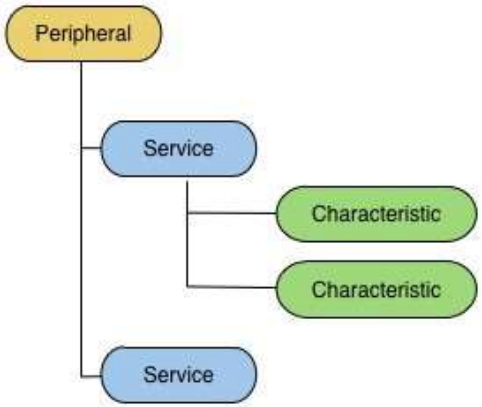
Table of content

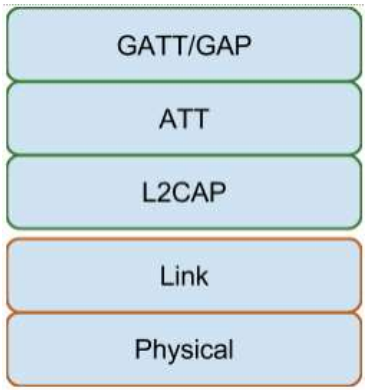
Bluetooth 4.0

High Speed	Classic	Low Energy
WiFi mixed BT	Most common BT	Originally "Wibree"
Persistent connections	Persistent connections	Non-persistent
High power consump.	Mid power consump.	Low power consump.
High bandwidth	Mid bandwidth	Low bandwidth
Short range	Mid range	Long range
(never tried)	Headphones, Keyboards, Mouse	Health wrists, Temp. sensors, IoT devices

Bluetooth 4.0 Low Energy







BLE is session-less, 7 methods, similar to HTTP

Method	方向	功能
Request	Central -> Peripheral	一般發送訊息
Response	Peripheral -> Central	回覆 Request 用
Commands	Central -> Peripheral	不用 Response
Notifications	Peripheral -> Central	不用 Confirm
Indications	Peripheral -> Central	需要 Confirm
Confirmations	Central -> Peripheral	回覆 Indication 用

Usually Server has smaller battery and operates only when requested. Fig. Ref: Stanfy Inc, 2015

BLE widely adopted in Health & IoT

Curiosity to understand how it works.

BLE built-in profiles

- Time, Temp, Energy
- Weight, User profile
- Blood pressure, glucose
- Body mass, heart rate
- Speed, direction, location

GATT-Based Specifications

Profile Specification		Version	Status	Date Adopted
ANP	Alert Notification Profile	1.0	Active	13 September 2011
ANS	Alert Notification Service	1.0	Active	13 September 2011
AIOP	Automation IO Profile	1.0	Active	14 July 2015
AIOS	Automation IO Service	1.0	Active	14 July 2015
BAS	Battery Service	1.0	Active	27 December 2011
BCS	Body Composition Service	1.0	Active	21 October 2014
BLP	Blood Pressure Profile	1.0	Active	25 October 2011
BLS	Blood Pressure Service	1.0	Active	25 October 2011
BMS	Bond Management Service	1.0	Active	21 October 2014
CGMP	Continuous Glucose Monitoring Profile	1.0.1	Active	15 December 2015
CGMS	Continuous Glucose Monitoring Service	1.0.1	Active	15 December 2015
CPP	Cycling Power Profile	1.1	Active	03 May 2016
CPS	Cycling Power Service	1.1	Active	03 May 2016
CSCP	Cycling Speed and Cadence Profile	1.0	Active	21 August 2012
CSCS	Cycling Speed and Cadence Service	1.0	Active	21 August 2012
CTS	Current Time Service	1.1	Active	07 October 2014
DIS	Device Information Service	1.1	Active	29 November 2011
ESP	Environmental Sensing Profile	1.0	Active	18 November 2014
ESS	Environmental Sensing Service	1.0	Active	18 November 2014
FMP	Find Me Profile	1.0	Active	21 June 2011
GLP	Glucose Profile	1.0	Active	10 April 2012
GLS	Glucose Service	1.0	Active	10 April 2012
HIDS	HID Service	1.0	Active	27 December 2011
HOGP	HID over GATT Profile	1.0	Active	27 December 2011
HPS	HTTP Proxy Service	1.0	Active	06 October 2015
HRP	Heart Rate Profile	1.0	Active	12 July 2011
HRS	Heart Rate Service	1.0	Active	12 July 2011
HTP	Health Thermometer Profile	1.0	Active	24 May 2011
HTS	Health Thermometer Service	1.0	Active	24 May 2011
IAS	Immediate Alert Service	1.0	Active	21 June 2011
IPS	Indoor Positioning Service	1.0	Active	19 May 2015
IPSP	Internet Protocol Support Profile	1.0	Active	16 December 2014
LLS	Link Loss Service	1.0.1	Active	14 July 2015
LNP	Location and Navigation Profile	1.0	Active	30 April 2013
LNS	Location and Navigation Service	1.0	Active	30 April 2013
NDCS	Next DST Change Service	1.0	Active	13 September 2011
OTP	Object Transfer Profile	1.0	Active	17 November 2015
OTS	Object Transfer Service	1.0	Active	17 November 2015
PASP	Phone Alert Status Profile	1.0	Active	13 September 2011
PASS	Phone Alert Status Service	1.0	Active	13 September 2011
PXP	Proximity Profile	1.0.1	Active	14 July 2015
PLXP	Pulse Oximeter Profile	1.0	Active	14 July 2015
PLXS	Pulse Oximeter Service	1.0	Active	14 July 2015
RSCP	Running Speed and Cadence Profile	1.0	Active	07 August 2012
RSCS	Running Speed and Cadence Service	1.0	Active	07 August 2012
RTUS	Reference Time Update Service	1.0	Active	13 September 2011
ScPP	Scan Parameters Profile	1.0	Active	27 December 2011
ScPS	Scan Parameters Service	1.0	Active	27 December 2011
TDS	Transport Discovery Service	1.0	Active	17 November 2015
TIP	Time Profile	1.0	Active	13 September 2011
TPS	Tx Power Service	1.0	Active	21 June 2011
UDS	User Data Service	1.0	Active	27 May 2014
WSP	Weight Scale Profile	1.0	Active	21 October 2014
WSS	Weight Scale Service	1.0	Active	21 October 2014

BLE playgrounds

- Nordic nRF App

Non-connectable -71 dBM

N/A
Address not available
Non-connectable -70 dBM

N/A (Eddystone™)
Address not available
Non-connectable -83 dBM

Complete Local Name N/A
ManufacturerData N/A
TxPowerLevel N/A
SolicitedServiceUUID N/A
ServiceUUID 0xFEAA
ServiceData Frame Type: URL
16-bits Service UUID: 0xFEAA
TxPower: -18
URL: http://www.nordicsemi.com/
OverflowServiceUUID N/A

Pebble
Address not available
Connectable -60 dBM

N/A
Address not available
Non-connectable -82 dBM

N/A (Eddystone™)
Address not available
Non-connectable -88 dBM

Device: Nordic_HRM
Status: Connected

DFU Packet
UUID 00001532-1212-EFDE-1523-785FEABCD123
Properties WriteWithoutResponse
Value N/A
Descriptor false

DFU Control Point
UUID 00001531-1212-EFDE-1523-785FEABCD123
Properties Write Notify
Value N/A
Descriptor true

DFU Version
UUID 00001534-1212-EFDE-1523-785FEABCD123
Properties Read
Value 0x0100
Descriptor false

- Node.js bleno

Primary Service

```
var PrimaryService = bleno.PrimaryService;

var primaryService = new PrimaryService({
    uuid: 'fffffffffffffffffffffffffffffff0', // or 'fff0' for 16-bit
    characteristics: [
        // see Characteristic for data type
    ]
});
```

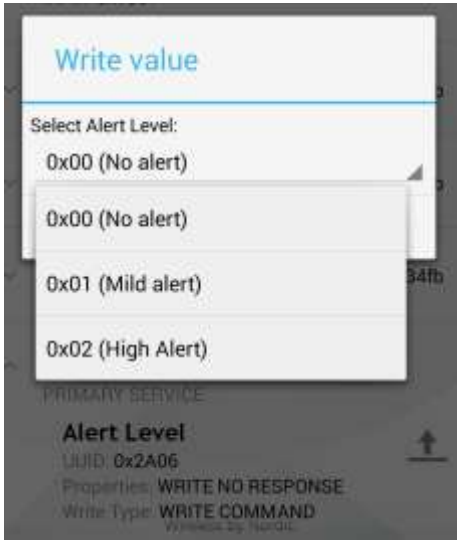
Characteristic

```
var Characteristic = bleno.Characteristic;

var characteristic = new Characteristic({
    uuid: 'fffffffffffffffffffffffffffffff1', // or 'fff1' for 16-bit
    properties: [ ... ], // can be a combination of 'read', 'write', 'writeWithoutResponse', 'notify', 'indic
    secure: [ ... ], // enable security for properties, can be a combination of 'read', 'write', 'writeWithou
    value: null, // optional static value, must be of type Buffer - for read only characteristics
    descriptors: [
        // see Descriptor for data type
    ],
    onReadRequest: null, // optional read request handler, function(offset, callback) { ... }
    onWriteRequest: null, // optional write request handler, function(data, offset, withoutResponse, callback
    onSubscribe: null, // optional notify/indicate subscribe handler, function(maxValueSize, updateValueCallb
    onUnsubscribe: null, // optional notify/indicate unsubscribe handler, function() { ...}
    onNotify: null // optional notify sent handler, function() { ...}
    onIndicate: null // optional indicate confirmation received handler, function() { ...}
});
```

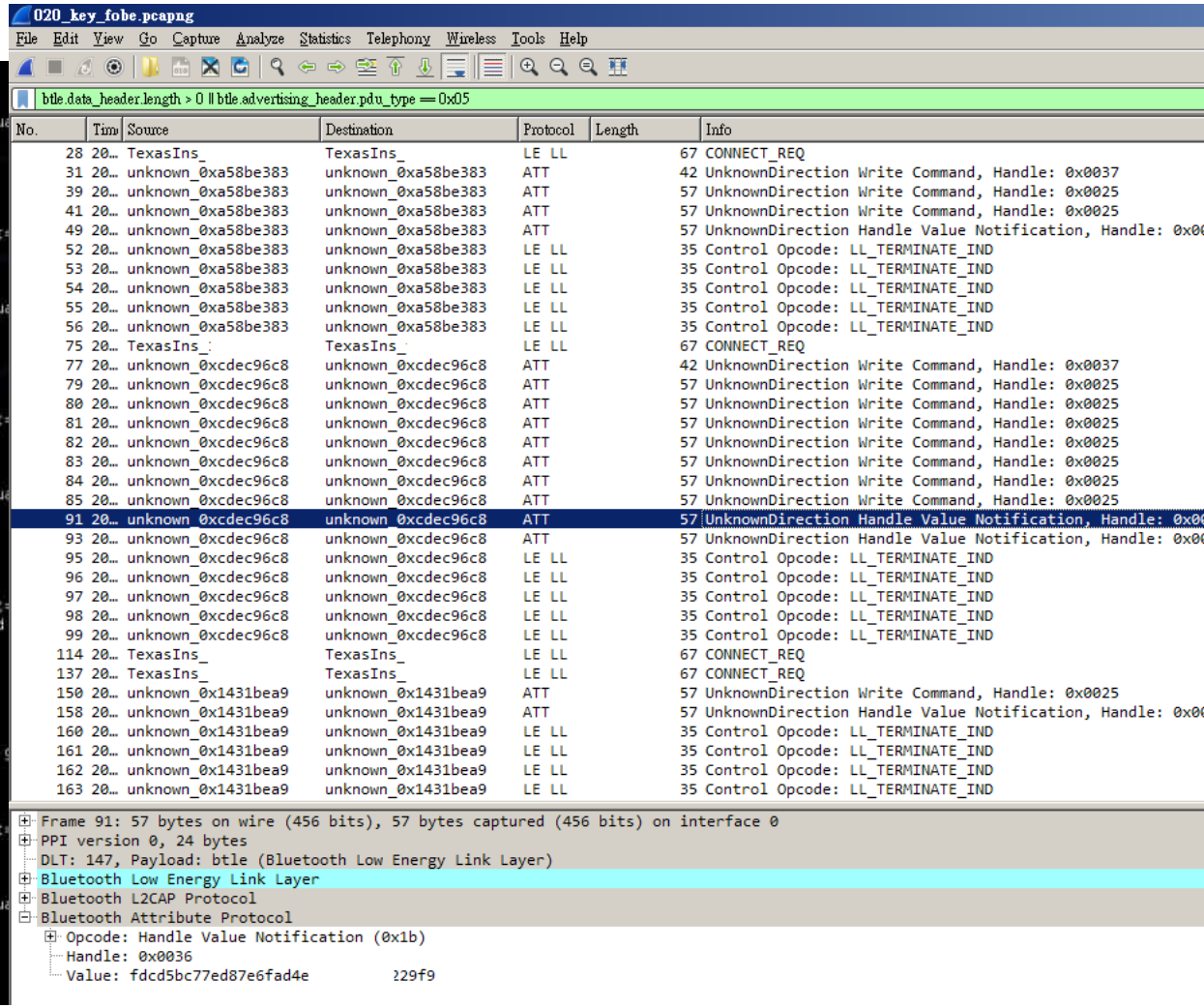
BLE is easy to hijack

- Sending vibrate message to nearby MI wristbands



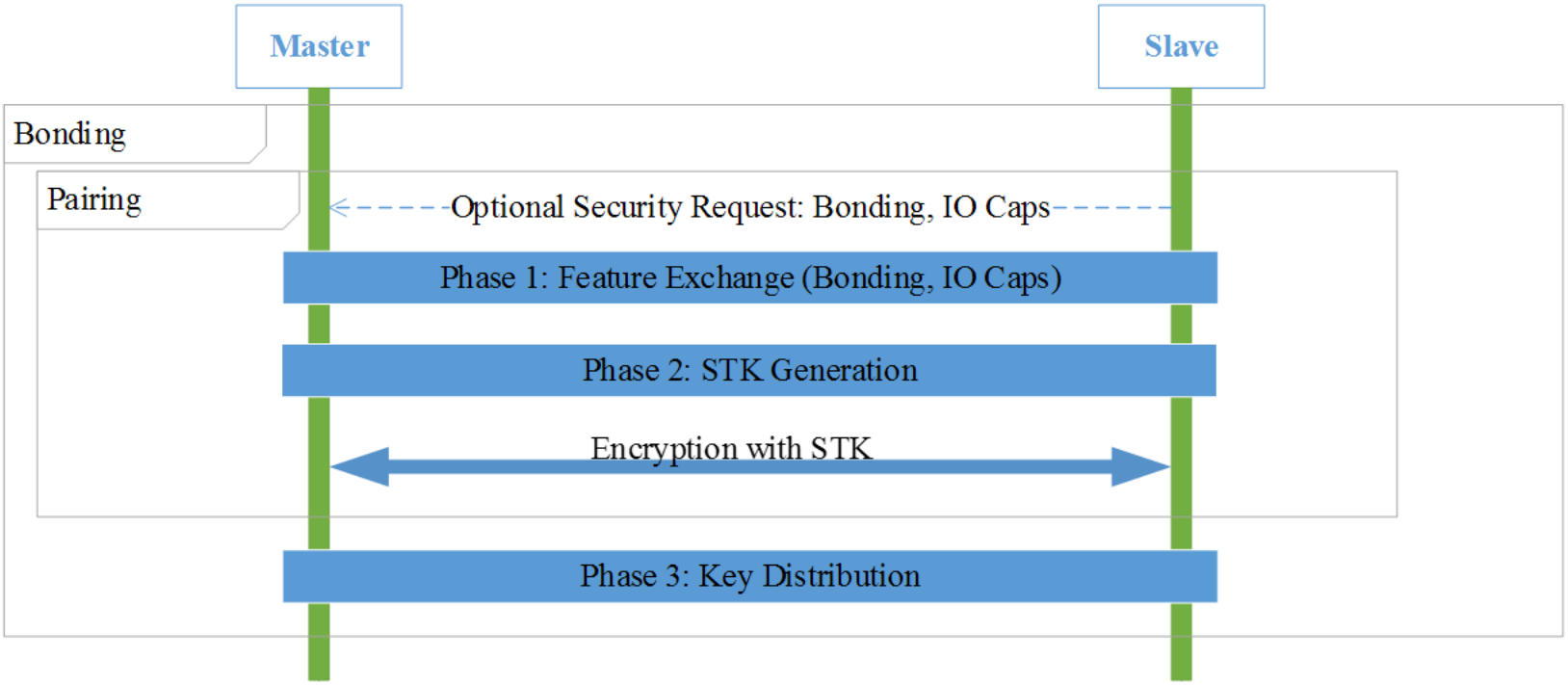
Ref: “Le IoT 想想物聯網 blog”

All BLE sniffer got is in plaintext ?!



Security Manager Protocol

Pairing	Bonding	Re-establishment
Short Term Key	Permanent Key	Permanent Key



BLE 4.0 SMP pairing

Pairing Mtd.	MitM attacks	Usability
Just Works	Vulnerable	Convenient, Un-authed
Passkey Entry	If you brute-PIN	Needs screen & Keyboard
Out-Of-Band	Secure via NFC	Needs NFC transceivers

Just Works is Un-authed

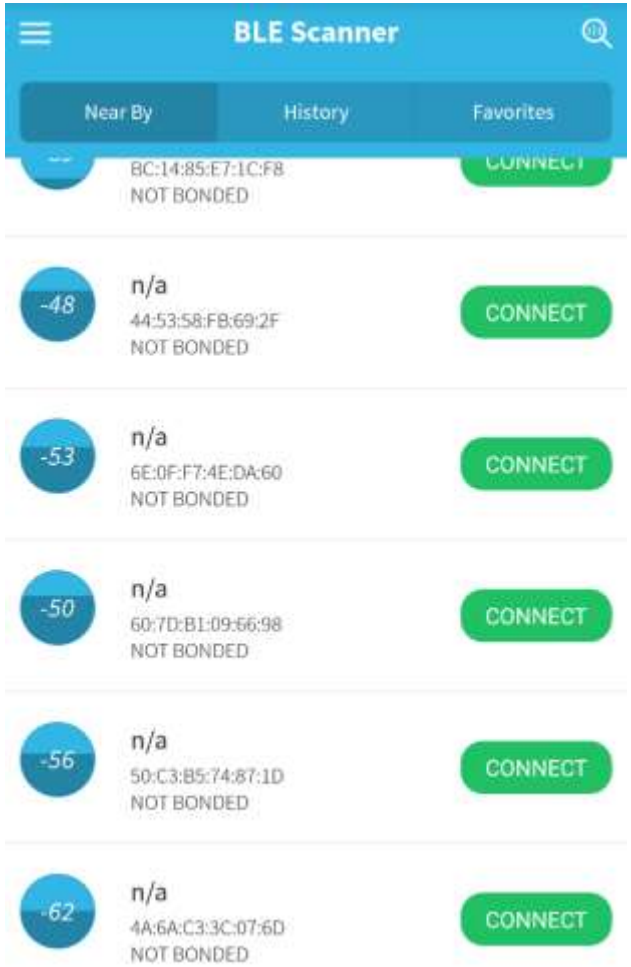
	Initiator				
Responder	**DisplayOnly**	**Display YesNo**	**Keyboard Only**	**NoInput NoOutput**	**Keyboard Display**
NoInput NoOutput	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated	Just Works Unauthenticated
Display YesNo	Just Works Unauthenticated	Just Works (For LE Legacy Pairing) Unauthenticated Numeric Comparison (For LE Secure Connections) Authenticated	Passkey Entry: responder displays, initiator inputs Authenticated	Just Works Unauthenticated	Passkey Entry (For LE Legacy Pairing): responder displays, initiator inputs Authenticated Numeric Comparison (For LE Secure Connections) Authenticated

Why vendors did not use SMP pairing:

1. Pairing takes time and effort
2. Paired does not always means secure
3. No screen for numeric comparison

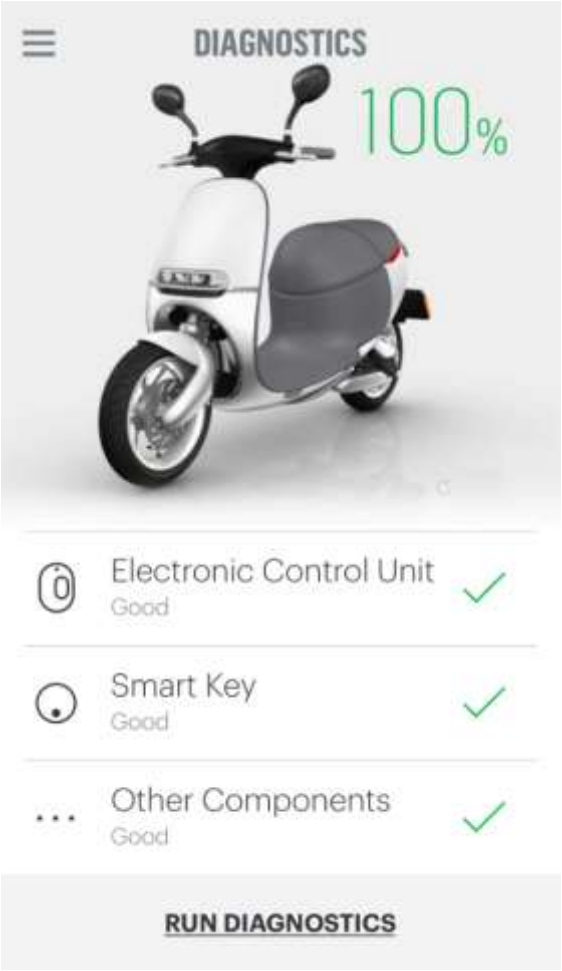
BLE 4.2 screen numb. comp.

BLE 4.0 privacy features

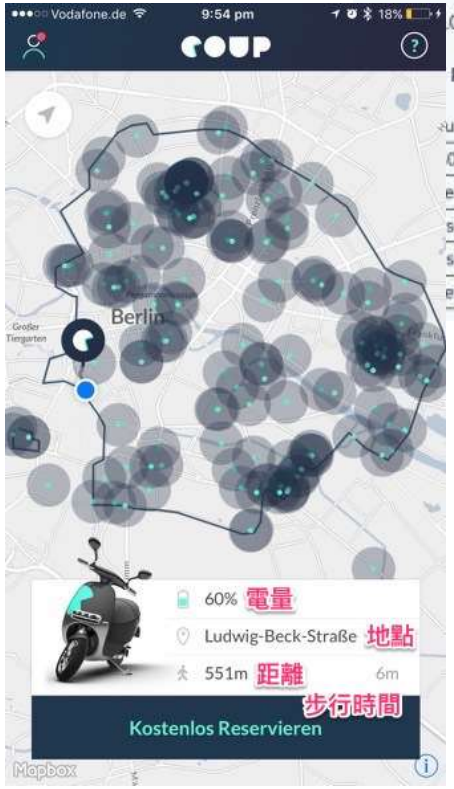


- HW Identifier Read Limitations
  - Prevents App/Ads tracking user
  - MAC Address always 02000000000000
- HW Identifier Randomization
  - Prevents AP tracking/nearby scanning
  - MAC Address different per power-cycle
  - SMP paired device gets fixed MAC via IRK
- How to authenticate device without HW identifier?

Gogoro Smart Scooter



Public Rental in Berlin



COUP Tarifübersicht

08.2016

Preise

		費用/歐元
ung (Entfällt für September 2016)		30 €
0 Minuten	30 分鐘	3 €
ere 10 Minuten	每10分鐘	1 €
schale (07.00 bis 19.00)	白天 AM 7:00 - PM 19:00	20 €
schale (19.00 bis 07.00)	晚上 PM 19:00 - AM 7:00	10 €
etrag pro Tag (24h Buchungszeitraum)	整天 24小時	30 €

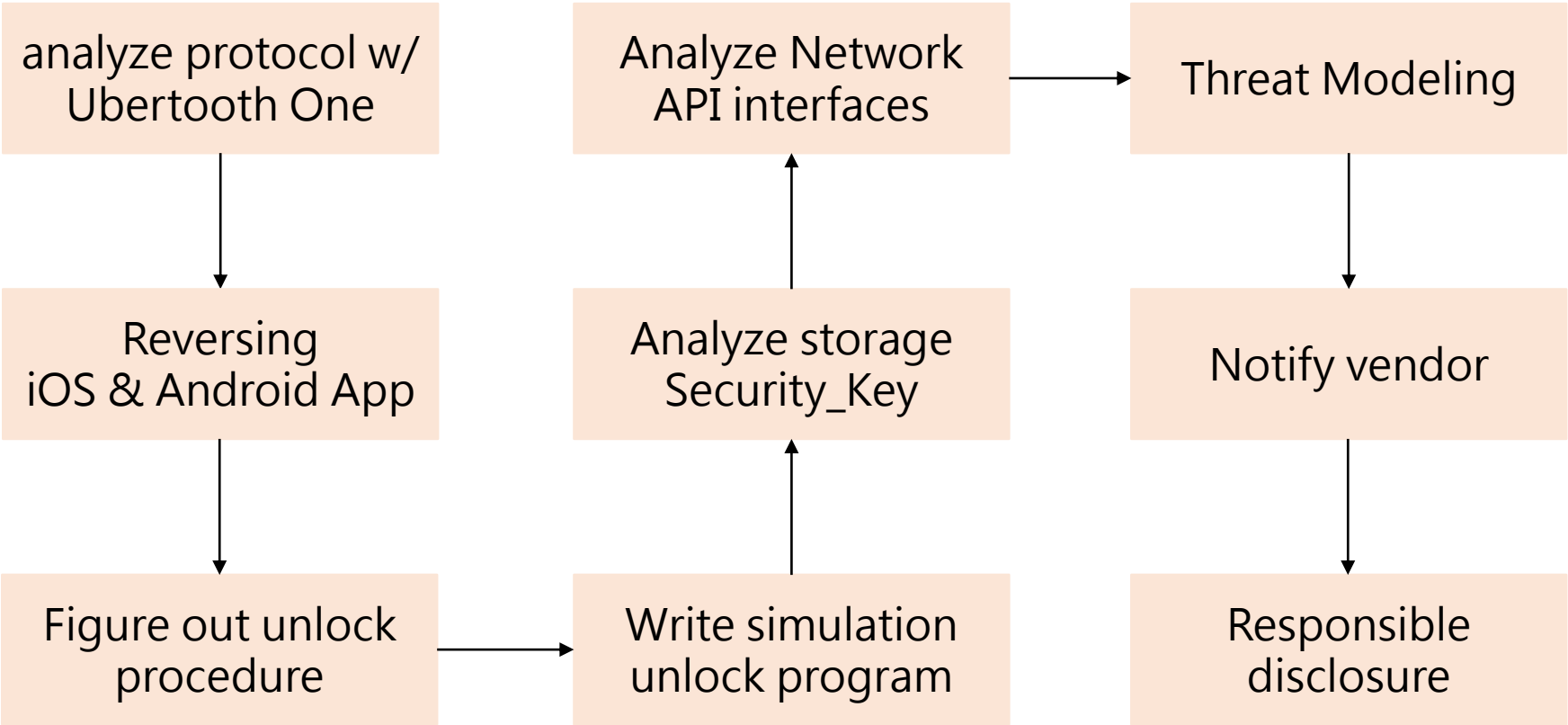
Gogoro's awesome electric scooter is coming to Europe next year, Amsterdam first

Bosch Teams Up with Gogoro to Bring Electric Scooter-Sharing to Berlin

by Kirsten Korosec AUGUST 4, 2016, 1:51 AM EDT

Our current research is based on Taiwan Gogoro.
Berlin Gogoro might work different from Taiwan's.

Analyzing method



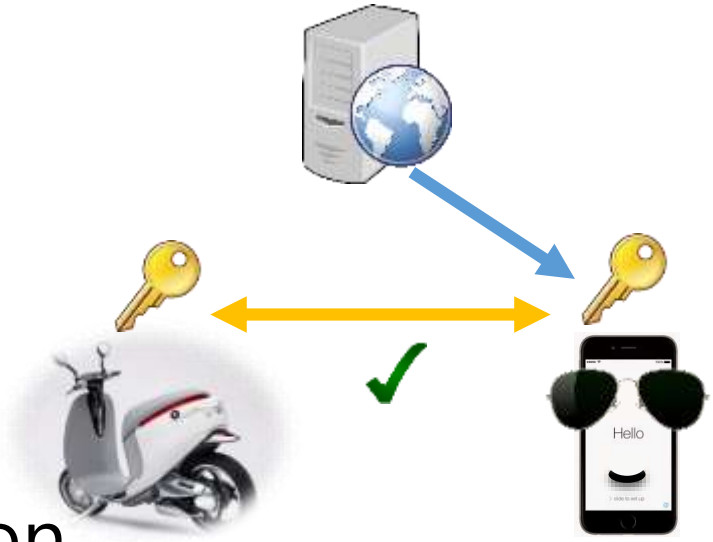
Key Fob Unlock (BLE)

	Source	Destination	Protocol	Length	Info
16:40:39.125904745	TexasIns[redacted]	TexasIns[redacted]	LE LL	67	CONNECT_REQ
16:40:39.142823445	unknown_0xa58be383	unknown_0xa58be383	ATT	42	UnknownDirection Write Command, Handle: 0x0037
16:40:39.230913045	unknown_0xa58be383	unknown_0xa58be383	ATT	57	UnknownDirection Write Command, Handle: 0x0025
16:40:39.231566145	unknown_0xa58be383	unknown_0xa58be383	ATT	57	UnknownDirection Write Command, Handle: 0x0025
16:40:39.306336345	unknown_0xa58be383	unknown_0xa58be383	ATT	57	UnknownDirection Handle Value Notification, Handle: 0x0036

Origin	**Handle**	**Value**	**Function**
Key Fob		CONNECT_REQ	Init connection
Scooter	0x37	01 00	Command ID
Scooter	0x25	c2 e7 20 bf d2 99 9d 43 68 c6 2d 65 39 3d 72 c9 f3	Rand. Challenge
Key Fob	0x36	d2 25 57 33 19 18 51 fd ae 7d 1b ed 85 e0 10 78 e2	Signed. Response
Scooter		LL_TERMINATE_IND	Ends connection

(this is much better than widely adopted Keeloq protocol)

Mobile App (Gateway)



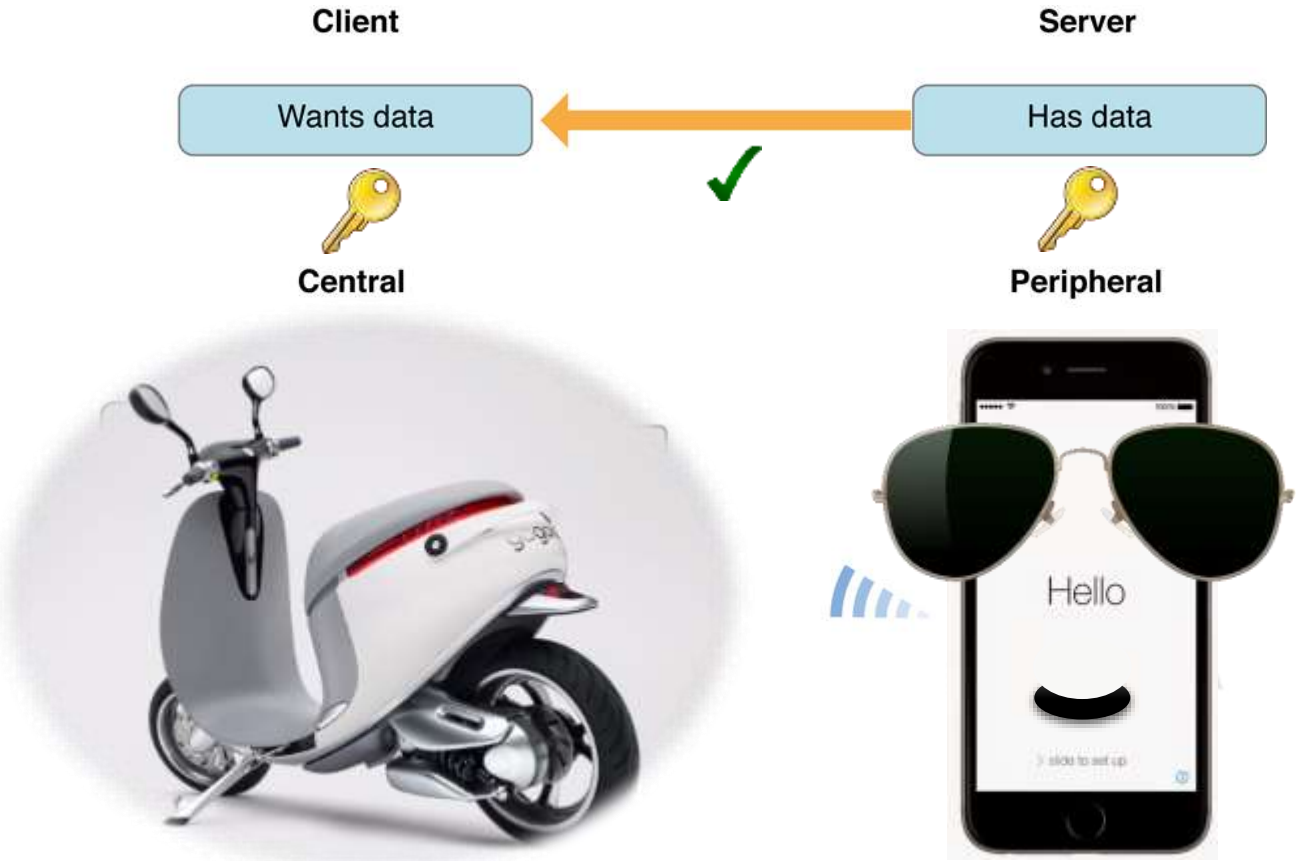
- My Gogoro single-sign-on
- App gets scooter information

Mobile App Pairing & Unlock

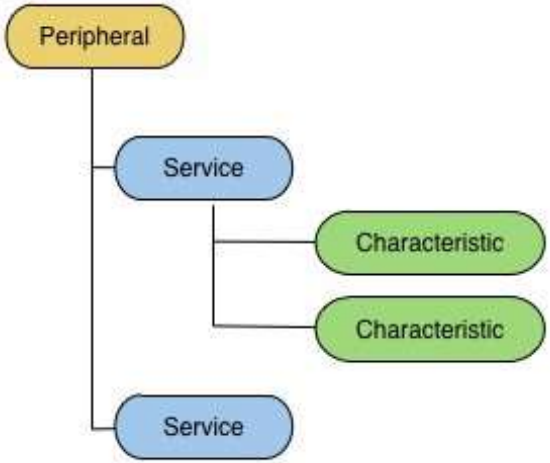
Only GATT protocol, no BLE SMP pairing observed.

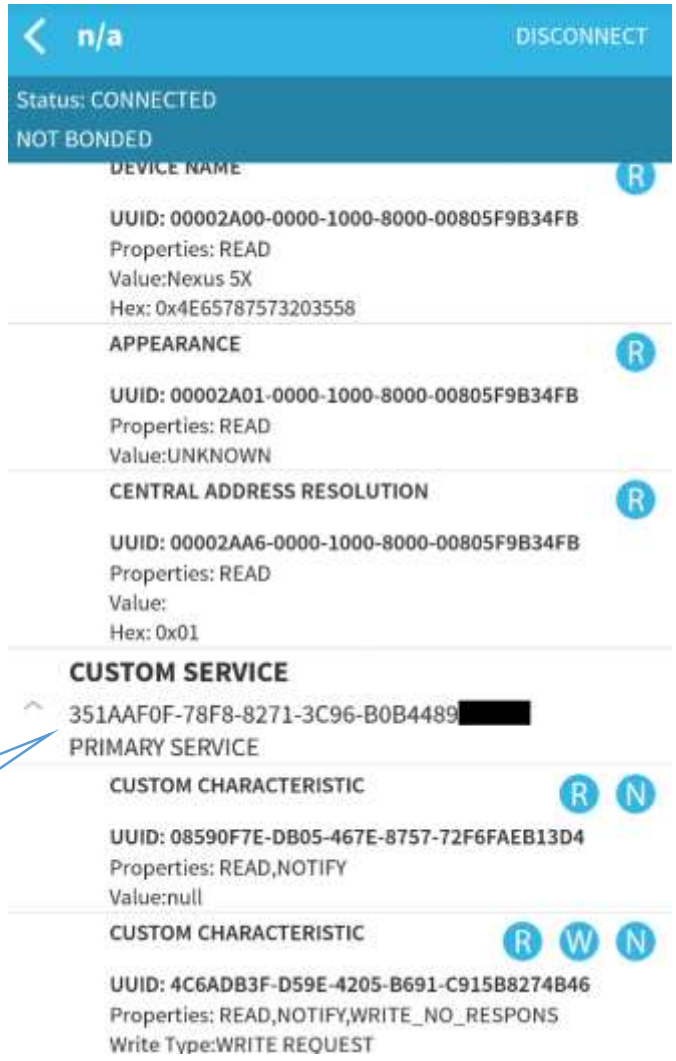
Vendor's challenge

- How to design authentication protocol when we did not BLE pairing and have no HW identifier?



BLE Gogoro Service





BLE Service UDID last 6-bytes must be Scooter MAC Address

Gogoro App Protocol

A-prefix: querying information

```
473 ATT        52 UnknownDirection Write Command, Handle: 0x0014
476 ATT        47 UnknownDirection Handle Value Notification, Hand
485 ATT        48 UnknownDirection Write Command, Handle: 0x0014
488 ATT        48 UnknownDirection Handle Value Notification, Hand
```

```
⊞ Frame 470: 48 bytes on wire (384 bits), 48 bytes captured (384 bits)
⊞ PPI version 0, 24 bytes
  DLT: 147, Payload: btle (Bluetooth Low Energy Link Layer)
⊞ Bluetooth Low Energy Link Layer
⊞ Bluetooth L2CAP Protocol
⊟ Bluetooth Attribute Protocol
  ⊞ Opcode: Handle Value Notification (0x1b)
    Handle: 0x0011
    Value: 90a20800000002c4
```

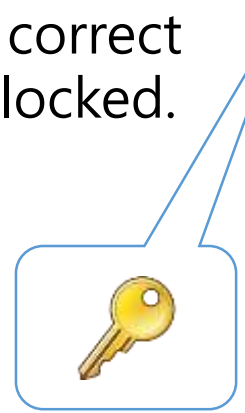
90 **A2** 08 00 00 00 **02** C4 (hex)
90: Header, **A2**: Command, 08: Length,
02: Parameter, C4: Checksum

Origin	Cmd	Function
App	A0	GetScooterSettingWithType
App	A1	GetScooterErrors
App	A2	GetScooterInfo
App	A3	SetScooterSetting
Scooter	A4	ScooterGetSettingStatus
Scooter	A5	ScooterErrorStatus
Scooter	A6	ScooterInfoState
Scooter	A7	ScooterSetSettingStatus
Scooter	A8	NotifyScooterError
Scooter	A9	NotifyInfo
Scooter	AE	PurchasedStatus
Scooter	AF	ScooterInfoState
Scooter	B0	ECU Challenge nonce
App	B1	ECU Response digest
Scooter	B2	ECU unknown
Scooter	B3	ECU Error
App	B4	ECU Cmd (Lock, Unlock, Open Trunk)

B-prefix: ECU Challenge Response

Gogoro Unlock flow

1. Scooter scans nearby peripheral for GATT Gogoro Service
   And if UUID {351AAF0F-}last 6-bytes matches its MAC Address
2. Mobile App reads GATT Scooter status, enable unlock button.
   Click to send ECU_Cmd(0xB4): 00 Lock, 01 Unlock, 02 Op Truck
3. Scooter writes ECU_Challenge (0xB0), a random 256 bit nonce
4. Mobile App notify ECU_Response (0xB1), also 256 bits
   ECU_Response =SHA256(ECU_Challenge, **Security_Key**)
5. Scooter compares ECU_Response if correct
   then ECU_Cmd will be executed, Unlocked.



the Security_Key

- ECU_Response =SHA256(ECU_Challenge, **Security_Key**)
- Early App put Security_Key in Document folder (slightly encrypted)
  - iOS MobileAppProp.plist has ScooterSKey
  - Android Settings.xml has AppSettings_DefScooter/encryptedkey2
  - Decrypting: AES-256, CBC/PKCS7Padding, IV=UserId, Key = ScooterUUID
- Document folder can be backed-up via iTunes / Android adb
  - Various methods: cable Juicy Attack, iTunes backup folder extraction etc.
  - AndroidManifest.xml has allowBackup flag set to true

- Security_Key can be retrieved from WebAPI
  - Attacker can brute My Gogoro membership
  - App Cookie can be stolden (MobileAppProp.plist has Web_Token)
  - https://mobile-pro.gogoroapp.com/WebService/Web/GetKey

Insecure App Data Storage

- Token, Certificate should be stored encrypted
  - Manages Timeout, Password Tamper etc.
  - Limits user, process access and key export

- Most OS platforms has secure storage zone
  - Apple iOS/macOS Keychain
    - iPhone 6~ Secure Enclave
  - Android Keystore
    - Samsung S6~ KNOX
  - Windows Protected Storage
    - HSM Such as UbiKey

Unlock code generator

- We wrote our Android App to generate ECU_Response and unlocked scooters successfully if Security_Key is known.
- Demo

Via this experiment we proved:

1. Security_Key is necessary to unlock scooter.
2. Security_Key can be cloned or transferred.
3. Gogoro Scooter cannot identify Mobile App hardware.

Gogoro Analysis Summary

- HW identifier privacy makes authentication difficult
  - IoT device trusts Security_Key rather than your Mobile Phone
  - Protect your Security_Key hard !!!
- Insecure App Data Storage vulnerable
  - Security_Key should not be stored in Document folder
  - Should be stored at Keychain / KeyStore
- Other possible weakness
  - WebAPI should do SSL Cert Pining to prevent MitM
  - Relay-Attack for Challenge-Response might be possible
  - Dumping Security_Key from Key Fob MCU or Scooter ECU ?

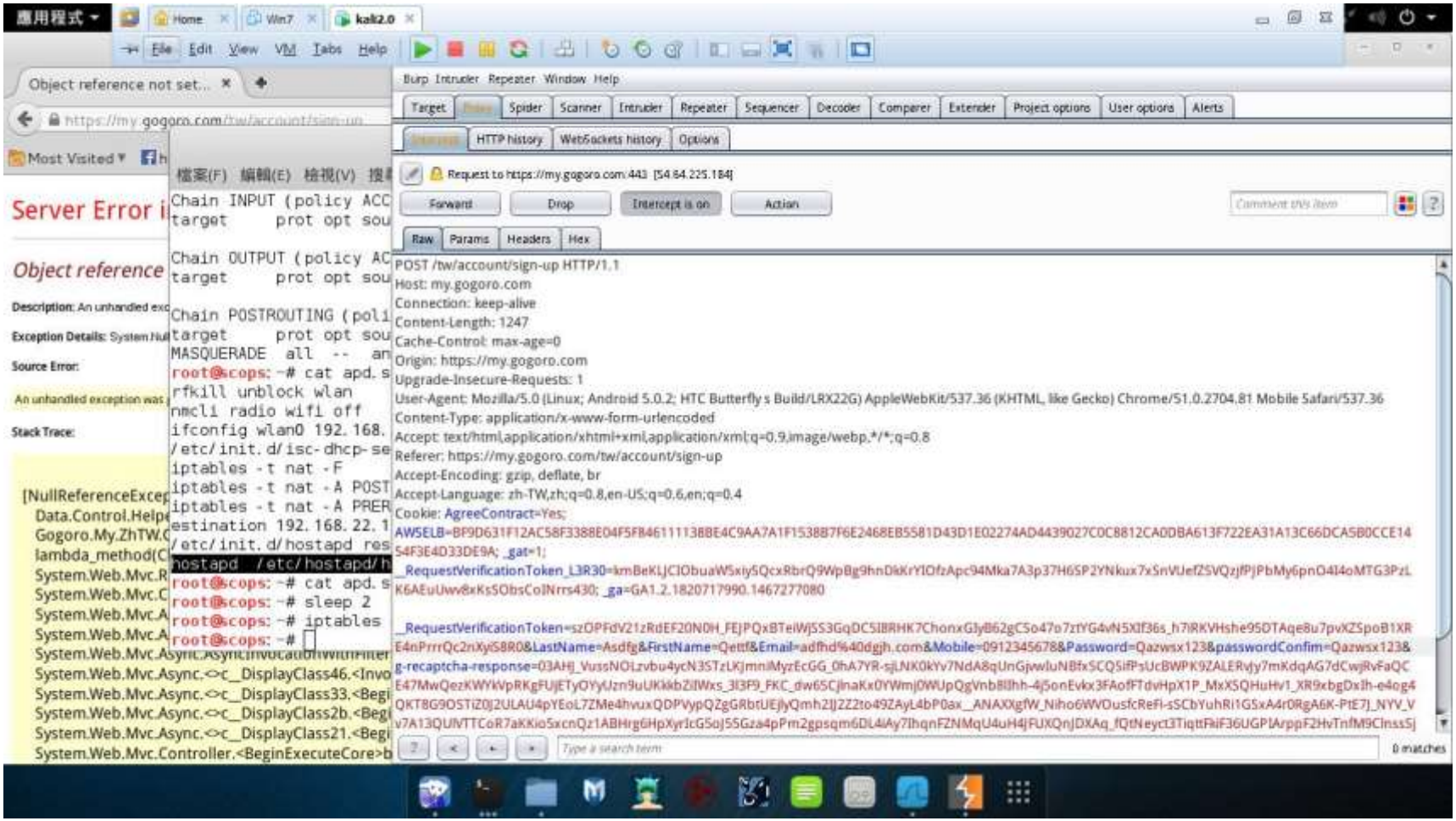
Gogoro system is generally safe…

- Although BLE SMP pairing is not adapted, Challenge/Response is better than Keeloq OTP
- Obtaining Security_Key from mobile phone is possible only when malware infected/jailbroken.
- Obtaining Security_Key from PC backup folder still needs to infect PC and decrypt slightly AES.
- Obtaining Security_Key from WebAPI might be the easiest way if username / password can be retrieved, brute-force or from other leaked database.

How to steal a Gogoro Scooter

- Infect the owner's phone or backup PC
  - Obtain and decrypt Security_Key from plist
- Owner open App to check fuel in Public Wifi
  - Do SSL MitM to get his cookie
  - Ask WebAPI for Security_Key
- Simulate the BLE Gogoro Service
  - With target scooter's MAC UUID
  - Approach target scooter and do ECU Challenge Response
  - Rode away as soon as possible.
- But you still cannot exchange battery :-(
  - Gogoro Battery has NFC authentication.

SSL MitM to retrieve Security_Key



Responsible disclosure

- 2016/02 App supports BLE unlock

- 2016/04 We notified Gogoro Vendor
- 2016/04 Fixed Security Key store
- 2016/07 Fixed SSL Cert verification
- 2016/07 Issued force logout update

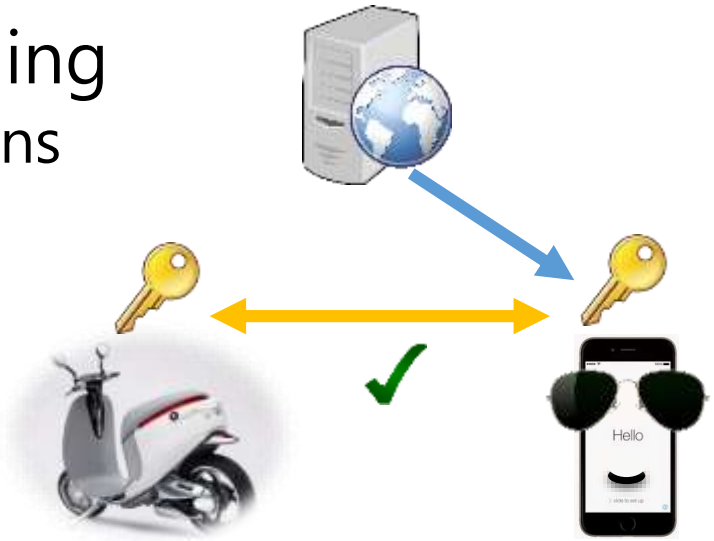
- 2016/12 Full Recall / Replace ECU
- Better Bluetooth Pairing Function

We will keep investing on security area and have more frequently release for security improvement in the future.

Designing good IoT-phone authentication

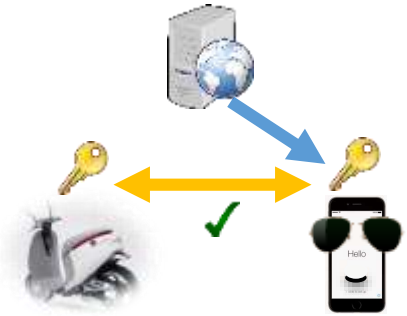
- Device does not know each other
  - IoT device does not know phone
  - IoT device knows secret key
  - IoT server provision secret key to phone

- Preventing Security_Key cloning
  - BLE 4.2 SMP Secure Connections
  - Phone has hardware identifier
  - store it in Secure Element
  - use OOB OTP such as SMS
  - add dual-counter to detect



Auth Methods Comparisons

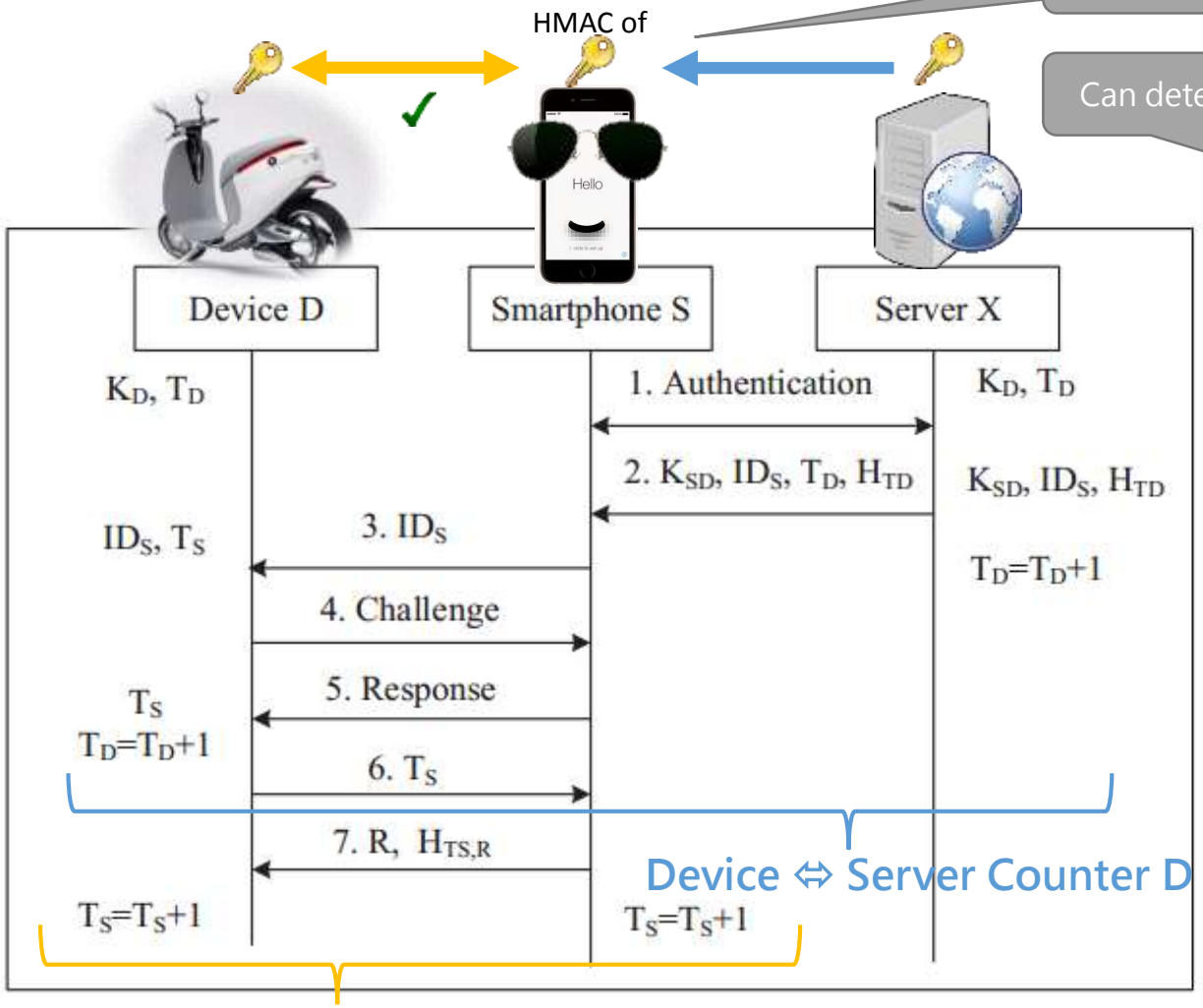
Method	Advantage	Disadvantage
Server Provision Secret Key	Phone device independent	Easy to steal, hard to detect
BLE 4.2 Secure Connections	Prevents MITM and clone.	Need a numeric display
Hardware Identifier	ID device. Prevents clone.	Privacy concern
Store in Secure Element	Encrypted, difficult to clone	Not every phone has SE
OOB OTP such as SMS	Trusting phone number.	OOB Channel cost (SMS)
Dual-counter detection 勝	Can easily detect abuse.	Cannot prevent abuse.



Dual-counter enhanced

Can revoke HMAC(Key) when phone lost

Can detect when HMAC(Key) is abused



Device ⇔ Server

K_D Permeant Shared Key
T_D Counter D
ID_S Identification
K_{SD} HMAC(K_D, ID_S) (temp)
H_{TD} HMAC(K_D, T_D) (temp)

Device ⇔ Smart Phone

Cha. RAND()
Res. HMAC(K_{SD}, H_{TD}, T_D)
T_S Counter S
R Command Request
$H_{TS,R}$ HMAC(K_{SD}, T_S, R)

When HMAC(Key) is used,
Counter will change.
If counter de-synced,
User can detect abuse.

Conclusion

1. introduction to Bt Low Energy, Security Manager Protocol, Smartphone authentications to controls IoT devices via BLE.
2. BLE 4.0 has many privacy features, restricting vendor powers, Hardware identifiers are either limited or randomized.
3. Challenges when designing auth methods, many vendors giving up SMP pairing, using just plaintext transmissions.
4. How to analyze BLE protocols, we examined many health and IoT devices, including Gogoro Smart Scooter vehicle.
5. Without SMP pairing, these vendor-designed authentications are sometimes flawed, so we are able to ignite other Gogoro.
6. We propose a better auth protocol: Dual-counter enhanced.

Future research

- Hardware hacking
  - Dump Security_Key from Key Fob MCU (TI CC2540)
  - Dump Security_Key from Scooter ECU (Atmel)
- Cryptography analysis
  - Challenge nonce randomization strength?
  - Challenge response acceptance timeframe?
- Relay-Attack on challenge responses
  - Attacker A approach Owner
  - Attacker B approach Scooter
  - A & B Relay challenge response over internet

Special thanks to

- Professor CSC's guidance and research
- Gogoro designed a BLE Smart Scooter
- Hiraku help dumping iOS app
- Support from lab and company colleagues

Q&A

- IoT is Security or Nothing
- Any questions?
  - GD@TeamT5.org

References

- Bluetooth SIG, Bluetooth Smart (Low Energy) Security. Bluetooth SIG, 2016 https://developer.bluetooth.org/TechnologyOverview/Pages/LE-Security.aspx
- Bluetooth SIG, Bluetooth Specification Version 4.0, Bluetooth SIG, 2010
- Andrew Garkavyi, Bluetooth Low Energy. Essentials for Creating Software with Device to Smartphone Connectivity, Stanfy Inc, 2015 https://medium.com/@stanfy/bluetooth-low-energy-essentials-for-creating-software-with-device-to-smartphone-connectivity-5164c71963e7
- Mike Ryan, Bluetooth: With Low Energy comes Low Security, iSEC Partners, USENIX WOOT, 2013.
- Mike Ryan, Hacking Bluetooth Low Energy: I Am Jack's Heart Monitor, ToorCon 14, 2012.
- Lindell, A. Y. Attacks on the pairing protocol of bluetooth v2.1, BlackHat US, 2008.
- Samy Kamkar, Drive It Like You Hacked It, Defcon 23, 2015 http://samy.pl/defcon2015/2015-defcon.pdf
- Gogoro, Gogoro Smart Scooter 規格書, 睿能創意股份有限公司, 2015. http://images.gogoroapp.com/download/PDF/tw/Gogoro-Smartscooter-Spec-Sheet-2015-06-17-02-Chinese.pdf
- Google, Android Physical Identifier Privacy, Google, 2016.
- https://developer.android.com/about/versions/marshmallow/android-6.0-changes.html#behavior-hardware-id
- Apple, iOS Physical Identifier Privacy, Apple, 2016. https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIDevice_Class
- N. Gupta, Inside Bluetooth Low Energy. Artech House, 2013.
- Le IoT 想想物聯網 Blog, 2016 https://thinkingiot.blogspot.tw/