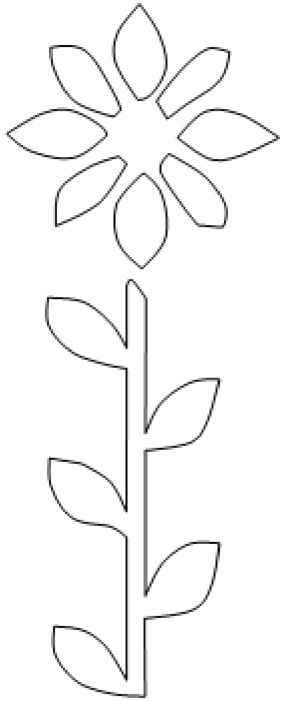
Fitbit Firmware Hacking

ReCon 0xE

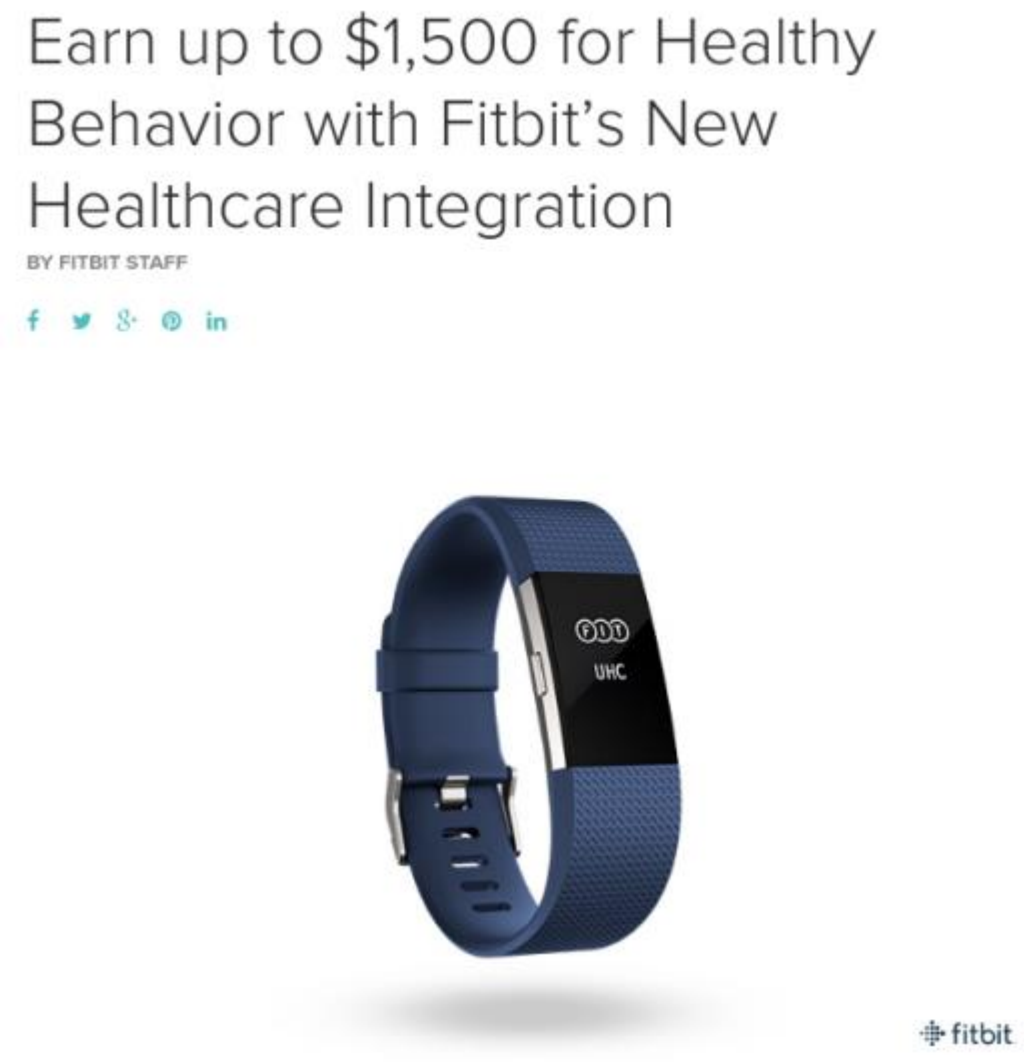


Jiska Classen & Daniel Wegemer

jclassen@seemoo.de
dwegemer@seemoo.de

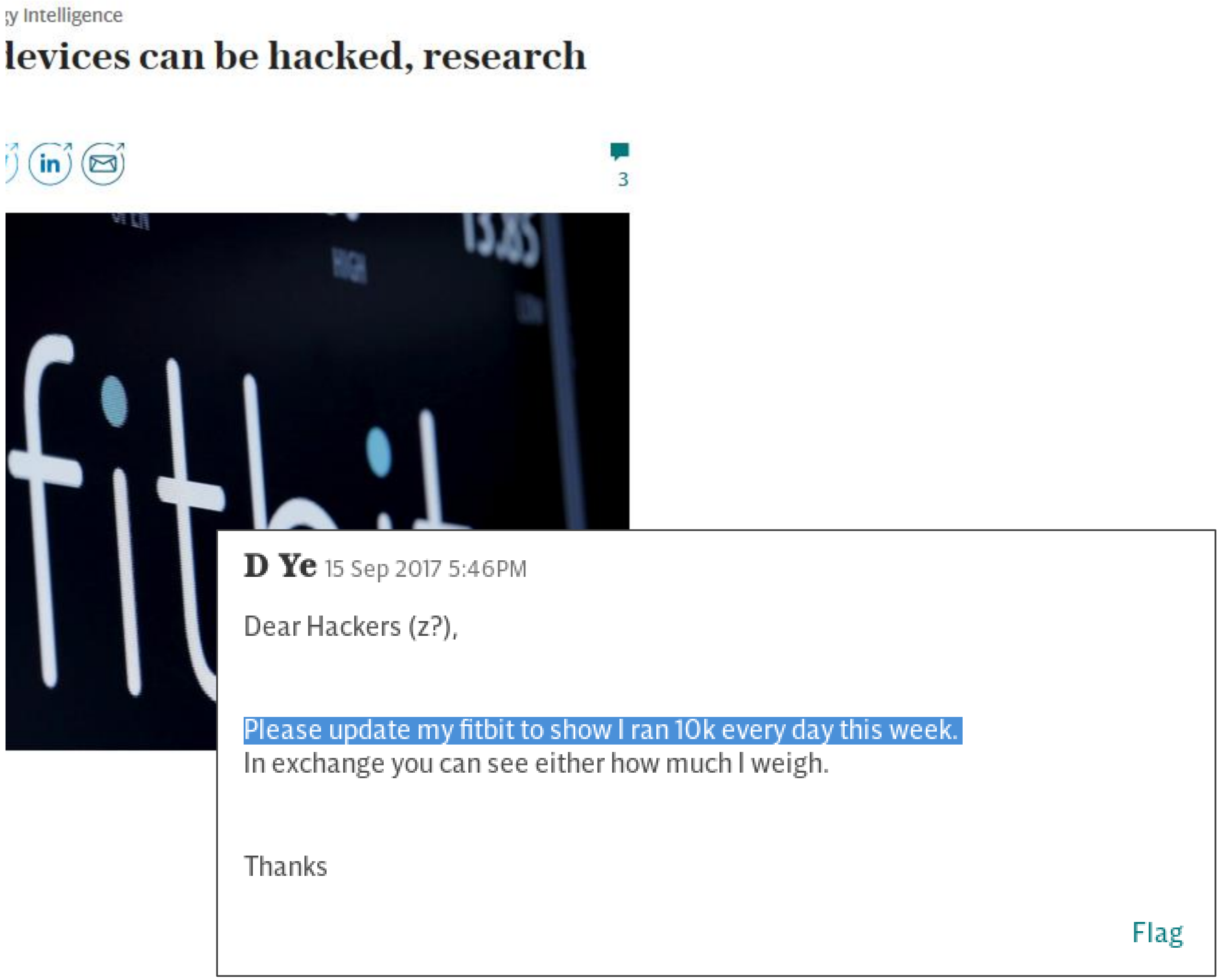
Technische Universität Darmstadt
Secure Mobile Networking Lab - SEEMOO
Department of Computer Science

Motivation



Earn up to $1,500 for Healthy Behavior with Fitbit's New Healthcare Integration

BY FITBIT STAFF



y Intelligence

levices can be hacked, research

D Ye 15 Sep 2017 5:46PM

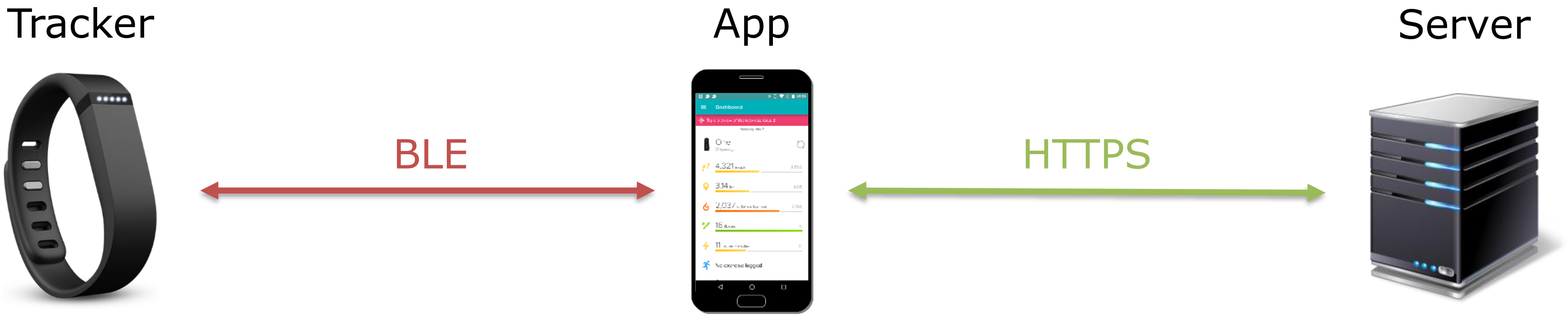
Dear Hackers (z?),

Please update my fitbit to show I ran 10k every day this week.
In exchange you can see either how much I weigh.

Thanks

Communication

Communication Paradigm



Tracker

BLE

App

HTTPS

Server

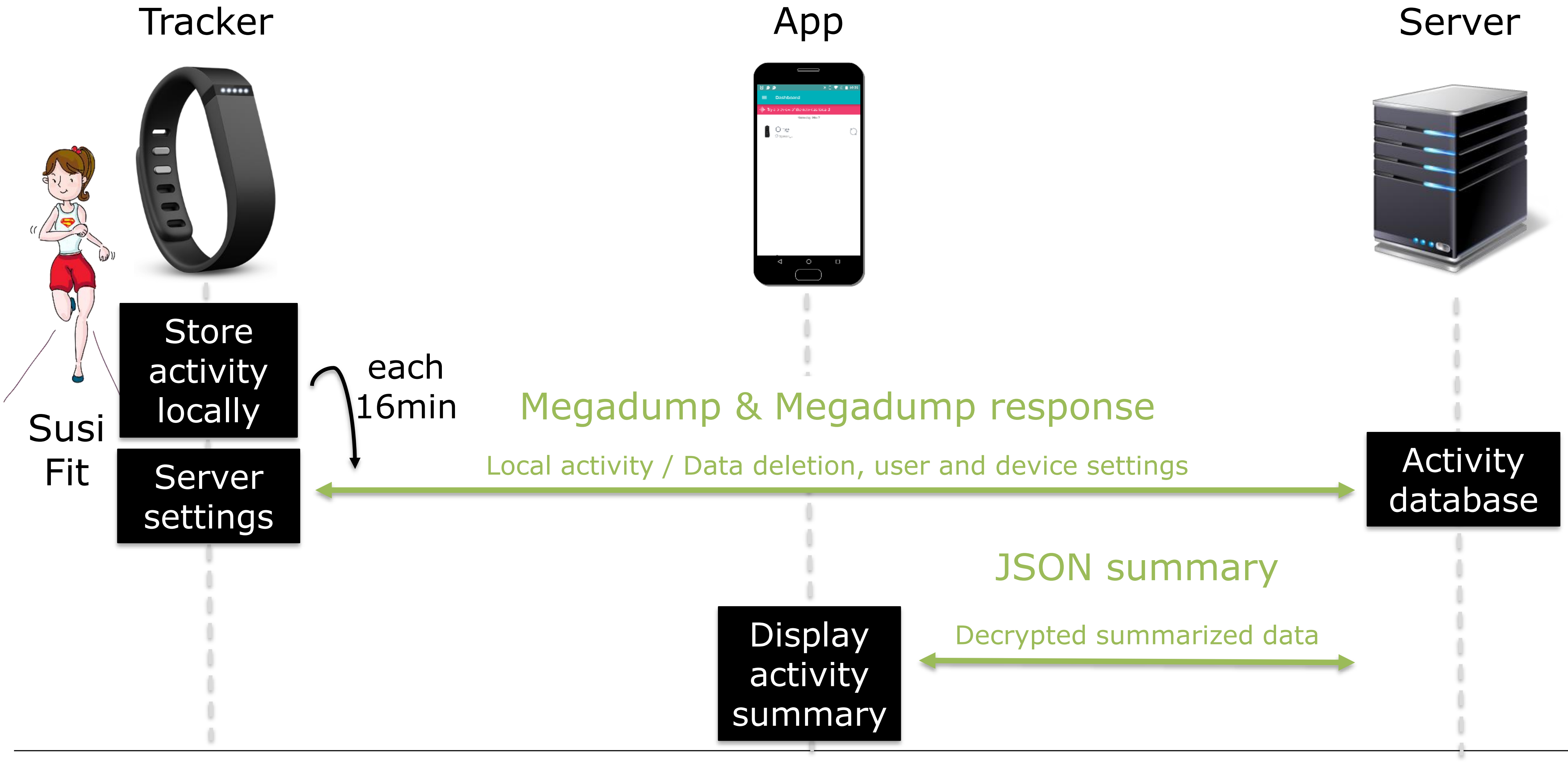
End-to-end encryption

Device-specific symmetric key

Recent trackers only …

Memory readout attack in firmware before October 2017

Activity Record Synchronization



Tracker

App

Server

Susi Fit

Store activity locally

Server settings

each 16min

Megadump & Megadump response

Local activity / Data deletion, user and device settings

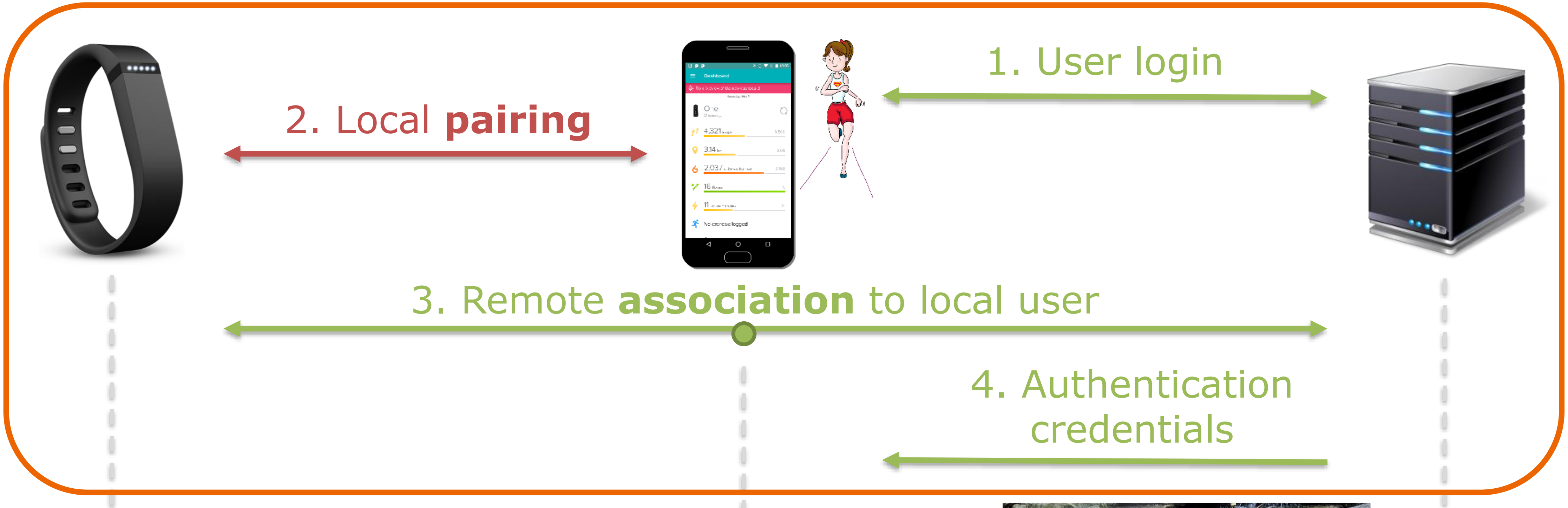
Activity database

JSON summary

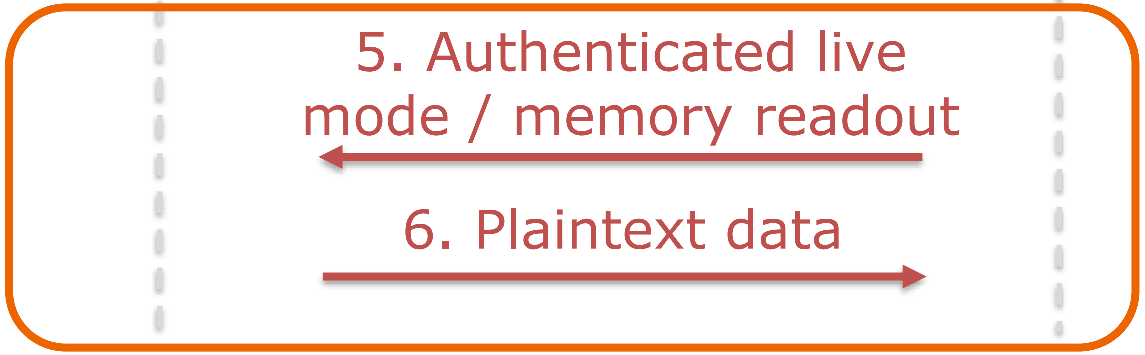
Decrypted summarized data

Display activity summary

Authenticated Live Mode


1. User login
2. Local pairing
3. Remote association to local user
4. Authentication credentials

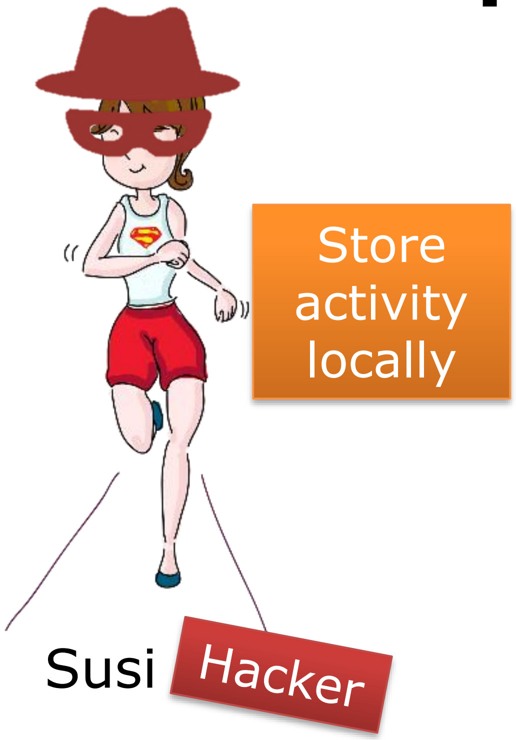
obtain once

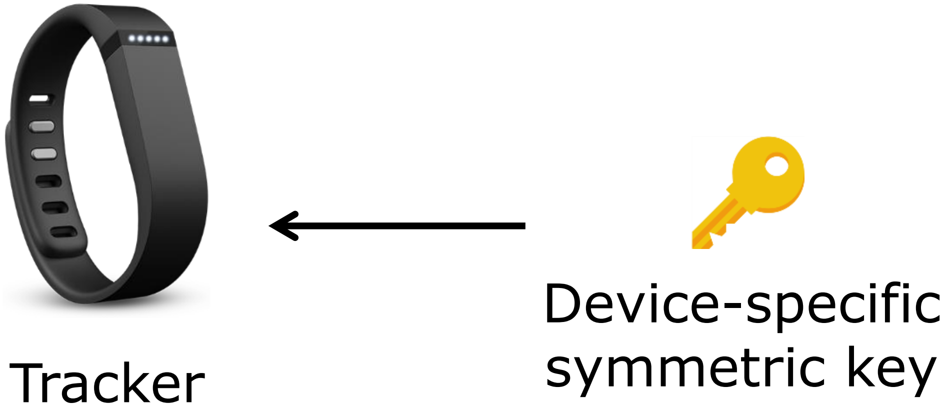


reuse forever

Accessing the Fitbit Hardware

Hardware Access Threats

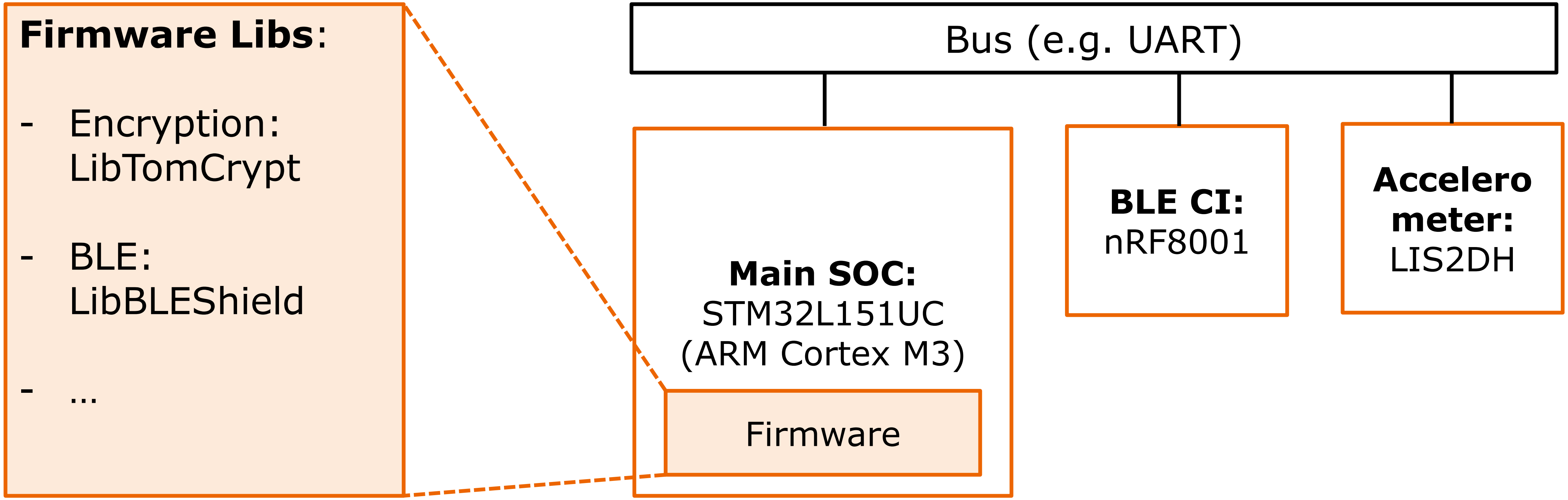




Goals:

1. Access/Modify local storage
2. Get encryption keys

Fitbit Flex Hardware & Software



Firmware Libs:

- Encryption: LibTomCrypt
- BLE: LibBLEShield
- …

Bus (e.g. UART)
Main SOC: STM32L151UC (ARM Cortex M3)
Firmware
BLE CI: nRF8001
Accelerometer: LIS2DH

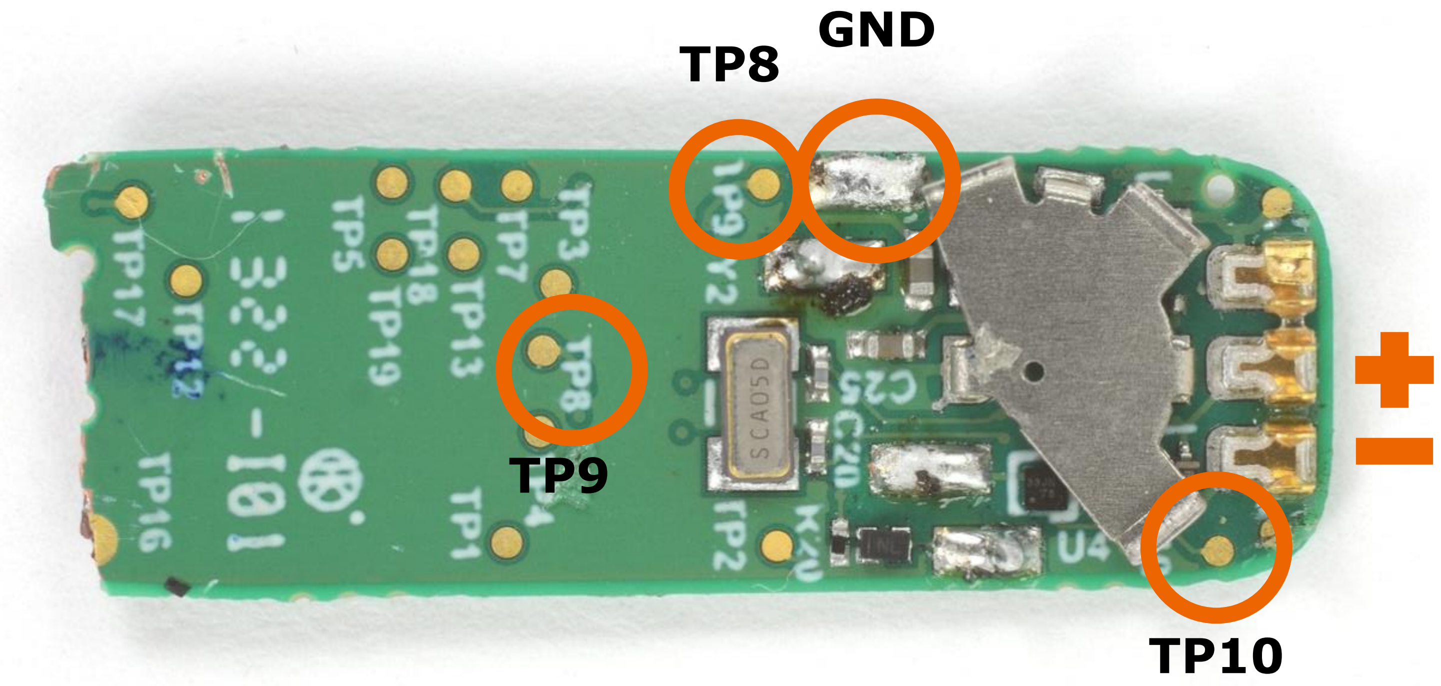
Hardware Access

Testing points to connect to debugger:

- TP8 SWDIO
- TP9 SWCLK
- TP10 NRST
- GND (from battery)

Goals:

- Dump firmware
- Modify stored data



CC BY-NC-SA 3.0 ifixit.com
(Sam Lionheart)

Memory Layout

Flash

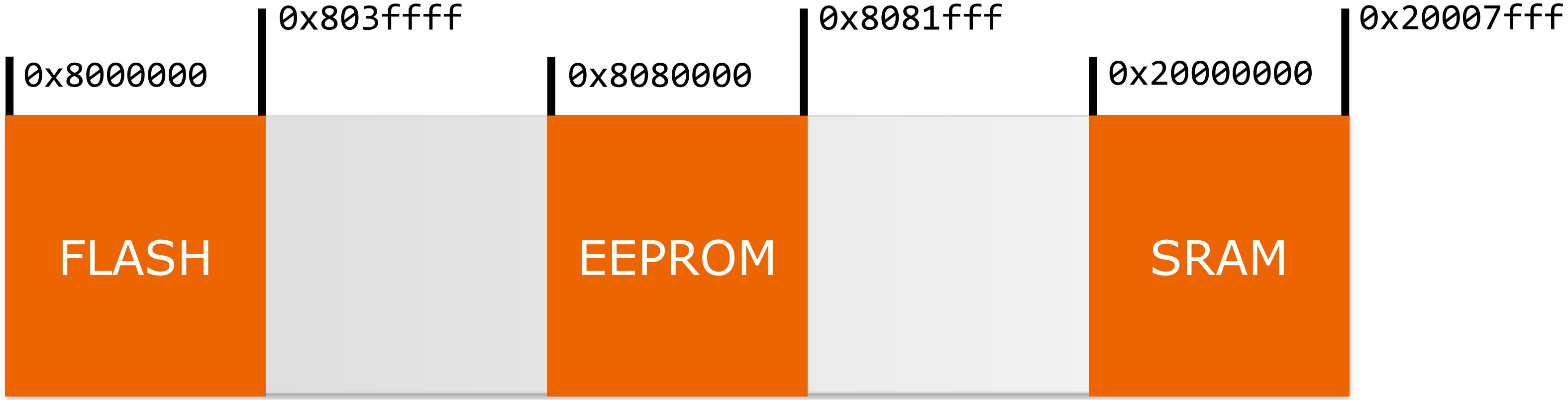
- Firmware code

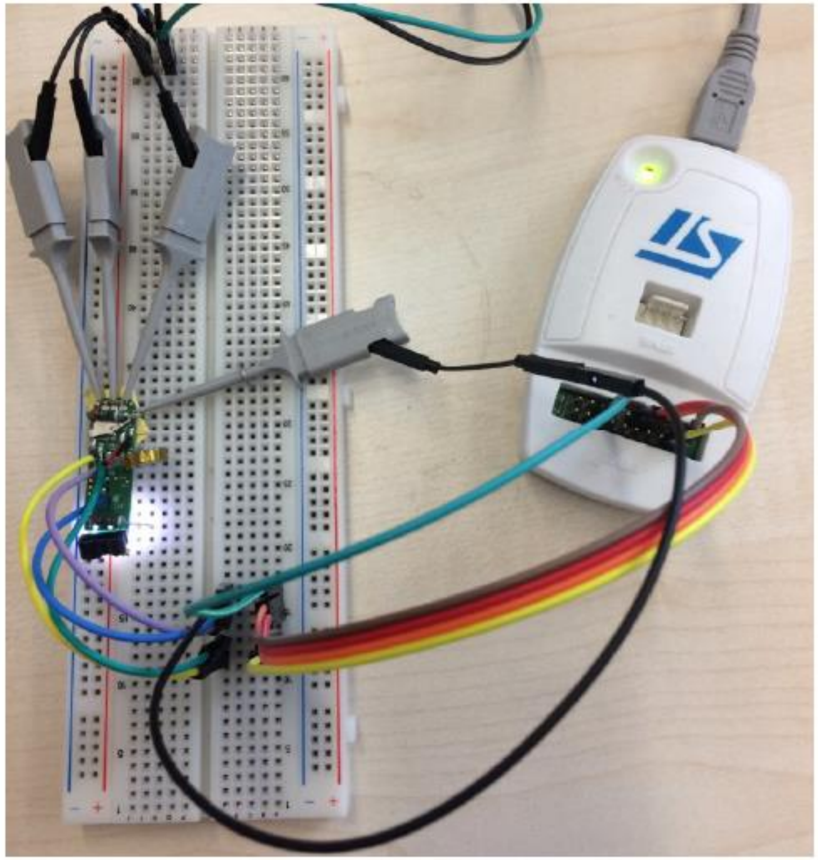
EEPROM

- Information that should survive empty battery

SRAM

- Firmware variables



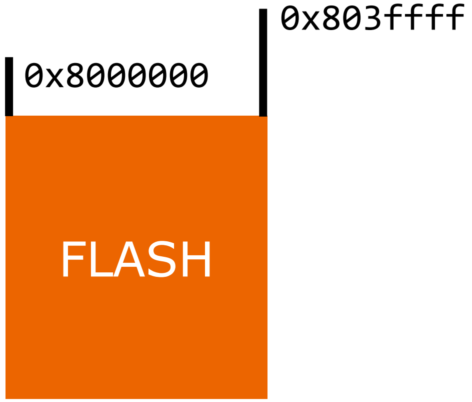


Fitbit Flex Hardware is Cheap!

Some Fitbits were harmed during our experiments…

Flash Contents

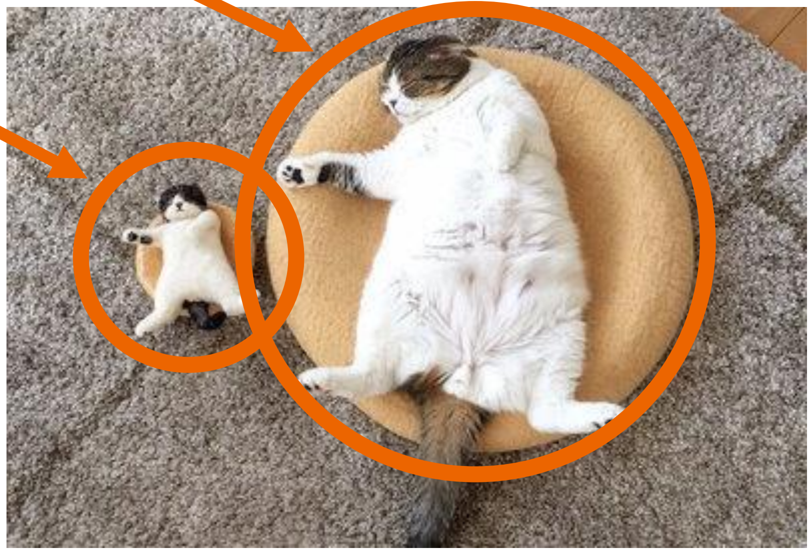
- **APP** ~ 1000 functions (including BSL duplicates)
- **BSL** ~ 500 functions
- Both BSL and APP run **independently**



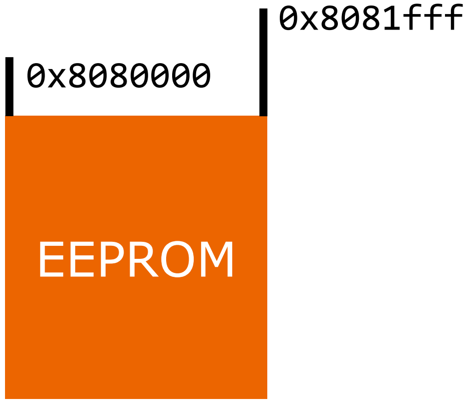
0x8000000

0x803ffff

FLASH



- **Serial number**
- Encryption **key**
- Encryption switch
- Fitness data



0x8080000

0x8081fff

EEPROM

Re-Enabling GDB Access (1)

Debugger Access

- Debugging is only enabled **during reset**
- Firmware initialization **disables GPIO ports** necessary for debugging
- Lets reset them!

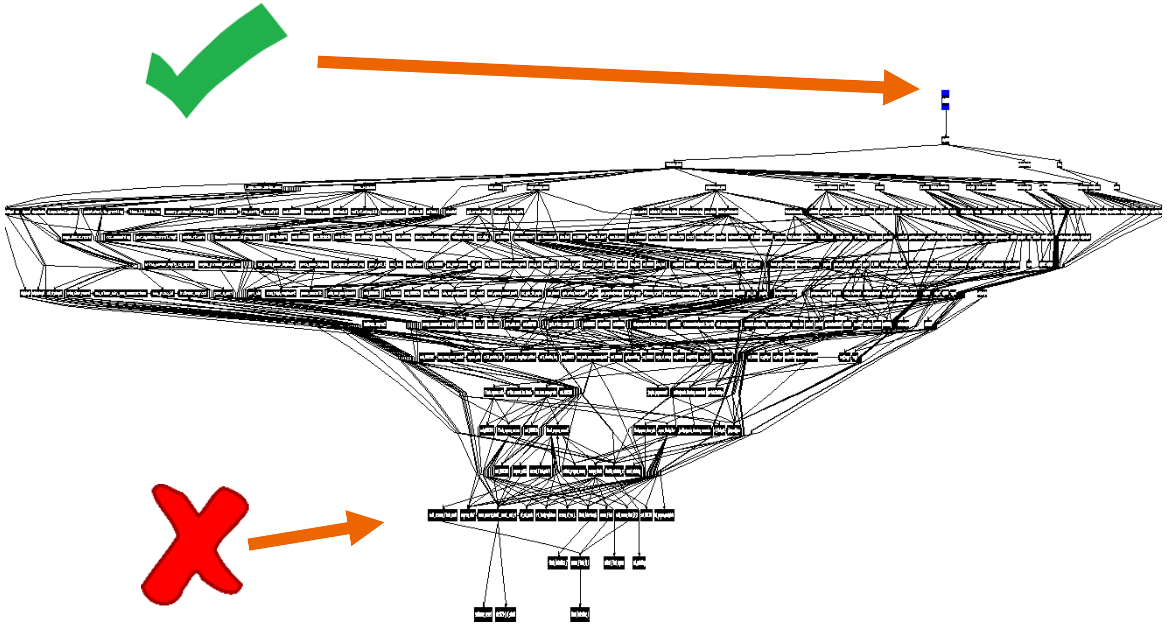
How? NexMon!

- NexMon is a **binary patching framework**
- We adapt NexMon for the Fitbit firmware
- Goal:
  - **Modify** firmware
  - Enable **dynamic debugging** (GDB)

nexmon.org

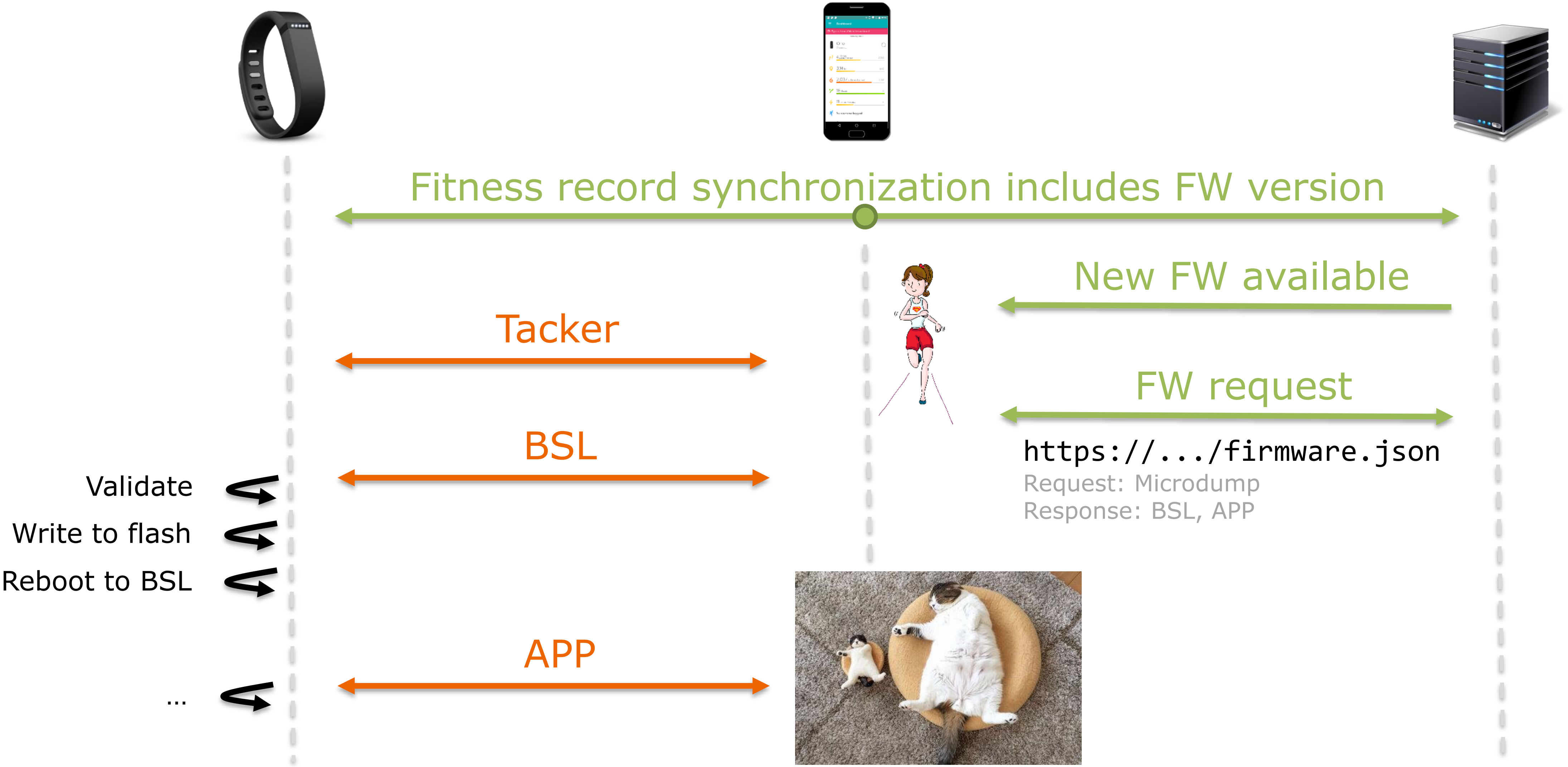
Re-Enabling GDB Access (2)

- **Connect after reset** works fine
- GPIOs needed for Debugging get **reassigned** during initialization
  → Where? No idea!
- Let **patch the firmware** to reenable the GPIOs necessary for debugging **after initialization has finished**
  → This might comes with **side effects**
  → Use **Bluetooth commands** to trigger the reprogramming



Wireless Fitbit Firmware Flashing

Update Process



Fitness record synchronization includes FW version

New FW available

FW request

`https://.../firmware.json`

Request: Microdump

Response: BSL, APP

Tacker

BSL

APP

Validate

Write to flash

Reboot to BSL

...

Firmware & Dump Encryption

Newer trackers come with encryption **enabled by default**
→ We need to know how **encrypted firmware updates** work

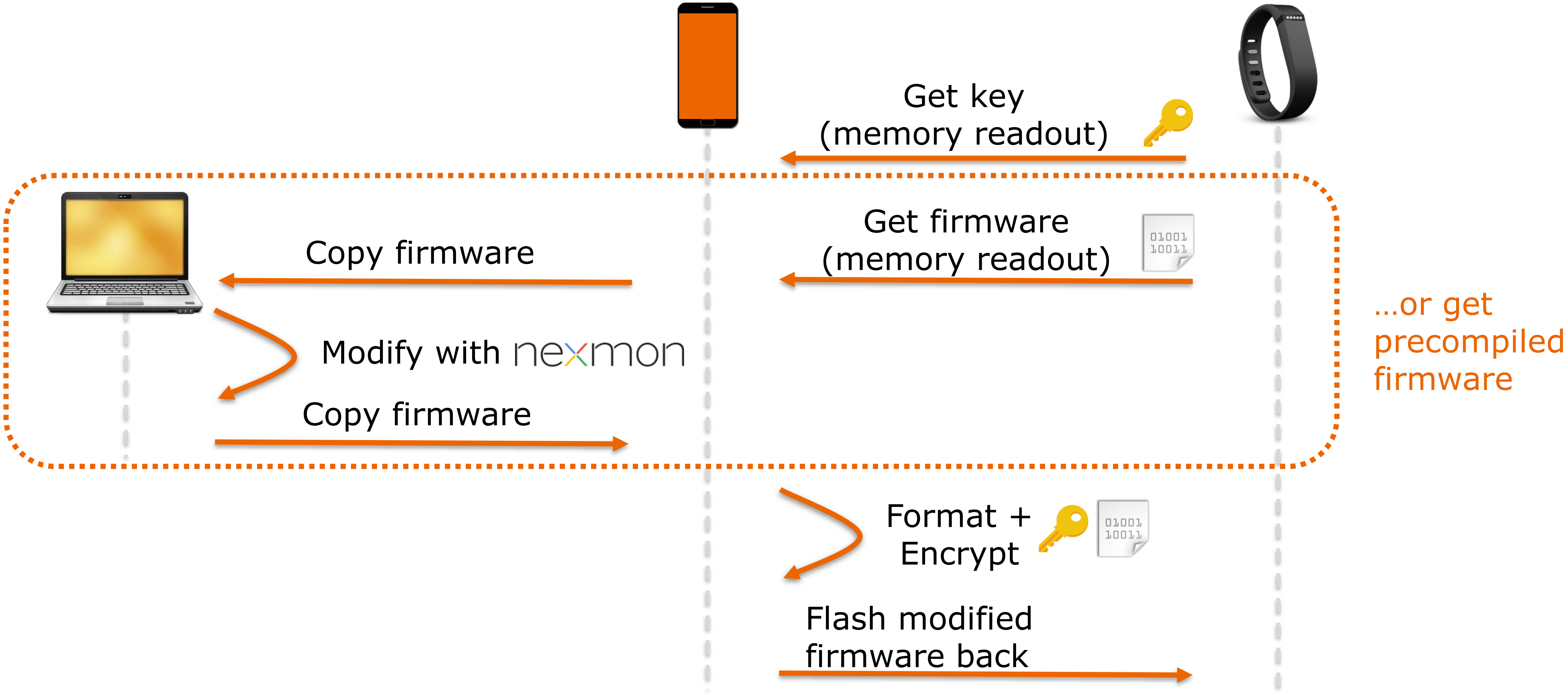
Trackers use **XTEA/AES in EAX mode**:

- **2 byte nonce** in beginning of each dump
- **128 bit encryption key**, extractable from EEPROM via memory readout attack
- **8 byte authentication MAC** in the end of each dump before length field

→ Firmware is based on **LibTomCrypt** (C)

Based on own reverse-engineering and insights of M. Schellevis

Steps to Flash Modified Firmware



Get key (memory readout)

Copy firmware

Get firmware (memory readout)

Modify with nexmon

Copy firmware

…or get precompiled firmware

Format + Encrypt

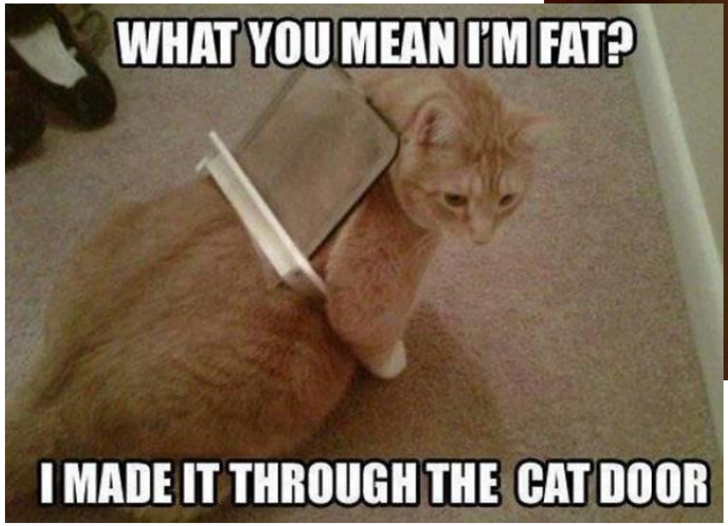
Flash modified firmware back

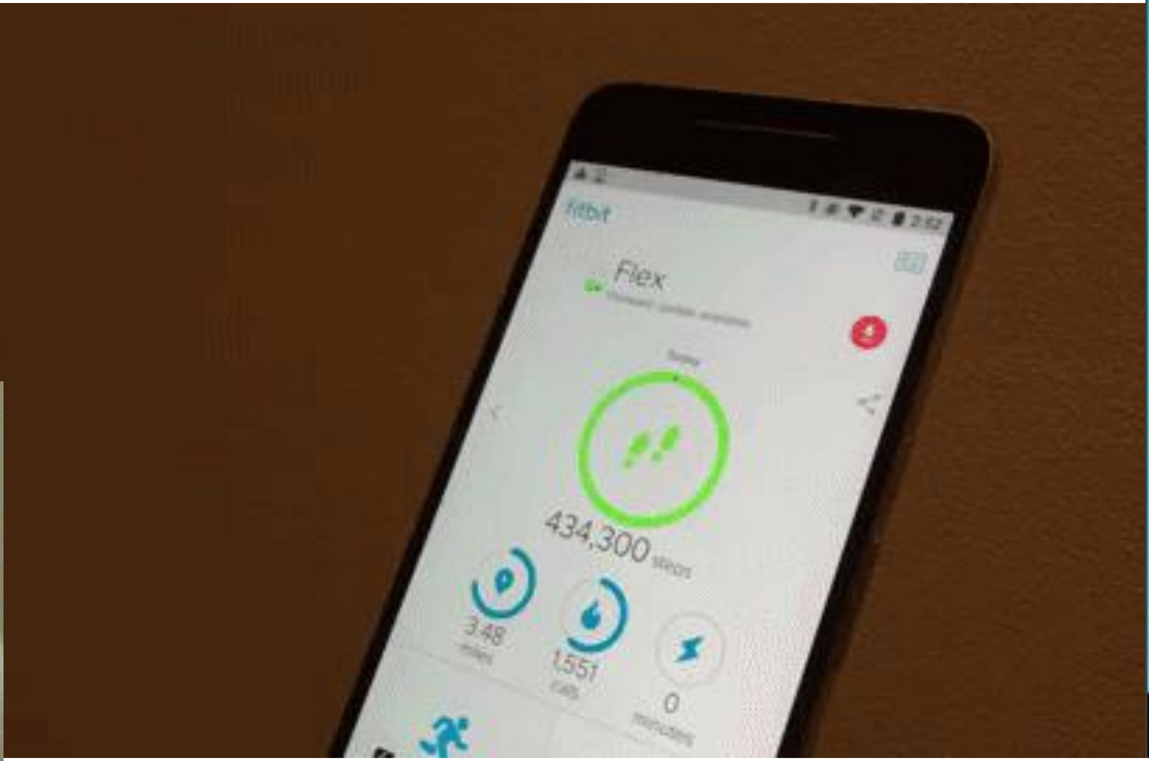
https://github.com/seemoo-lab/ fitness-firmware

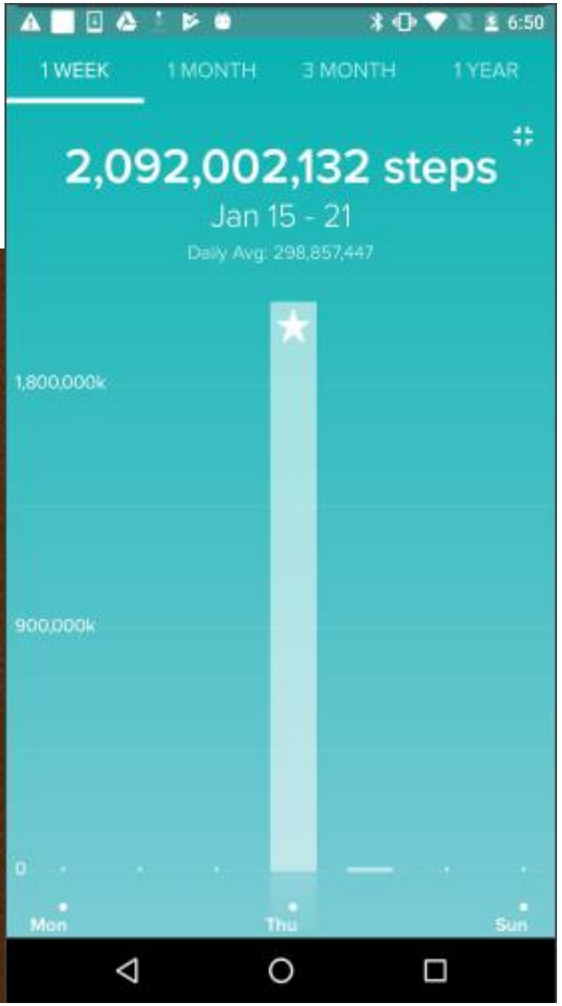
https://github.com/seemoo-lab/ fitness-app

Demo

```
return steps * 100;
```







2,092,002,132 steps

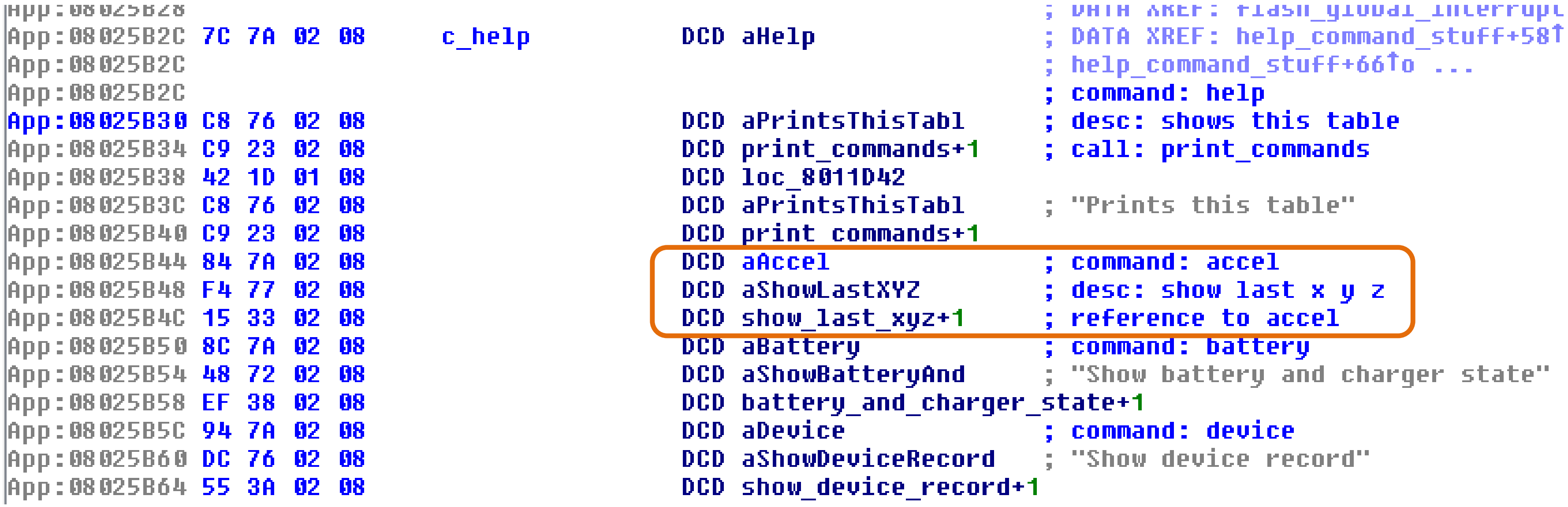
Jan 15 - 21

Daily Avg: 298,857,447

Accessing the Accelerometer

Locating Accelerometer Values

- **Factory test** functions include accelerometer printout
- Points to the correct register wich is updated by the accelerometer driver



```
App:08025B28                                                         ; DATA XREF: flash_global_interrupt
App:08025B2C 7C 7A 02 08     c_help          DCD aHelp               ; DATA XREF: help_command_stuff+58↑
App:08025B2C                                                         ; help_command_stuff+66↑o ...
App:08025B2C                                                         ; command: help
App:08025B30 C8 76 02 08                     DCD aPrintsThisTabl     ; desc: shows this table
App:08025B34 C9 23 02 08                     DCD print_commands+1    ; call: print_commands
App:08025B38 42 1D 01 08                     DCD loc_8011D42
App:08025B3C C8 76 02 08                     DCD aPrintsThisTabl     ; "Prints this table"
App:08025B40 C9 23 02 08                     DCD print_commands+1
App:08025B44 84 7A 02 08                     DCD aAccel              ; command: accel
App:08025B48 F4 77 02 08                     DCD aShowLastXYZ        ; desc: show last x y z
App:08025B4C 15 33 02 08                     DCD show_last_xyz+1     ; reference to accel
App:08025B50 8C 7A 02 08                     DCD aBattery            ; command: battery
App:08025B54 48 72 02 08                     DCD aShowBatteryAnd     ; "Show battery and charger state"
App:08025B58 EF 38 02 08                     DCD battery_and_charger_state+1
App:08025B5C 94 7A 02 08                     DCD aDevice             ; command: device
App:08025B60 DC 76 02 08                     DCD aShowDeviceRecord   ; "Show device record"
App:08025B64 55 3A 02 08                     DCD show_device_record+1
```

Configuration Registers

Register Address	Purpose	Bit Mask	Description
20	Data rate and power mode	57	All axes active, **100Hz** →
21	Filter mode	57	Highpass filter enabled
22	Interrupts	00	All interrupts disabled
23	Endian data selection and self test	00	LSB at lower address and normal self test mode
24	Boot mode	57	Normal boot mode and FIFO enabled
25	Interrupts	00	All interrupts disabled
2e	FIFO control register	8f	FIFO stream mode
30	Interrupt event register	00	All interrupts disabled
32	Interrupt threshold register	00	All interrupts disabled

Looks okay ☺
Can we copy
this with
>25Hz?

Accelerometer Live Mode

- Live mode normally only updates if activity data changes

- Modifications for accelerometer live mode:
  - Maximum update **rate** → ~66 Hz ☺
  - Copy **x,y,z** accelerometer data to live mode variables
  - Ensure **backward compatibility** by only enabling accelerometer live mode after a special command

Use Cases

- **Develop** accelerometer-based applications on any **Bluetooth** capable platform.

- Possibility to **port** these applications later on with C/Nexmon and wirelessly flash them onto your Fitbit, e.g. recognition of different types of movements and gestures with the same **low battery usage**.

Hackable Models & Versions

Encrypted wireless **firmware modifications** (requires memory readout):

Tracker	Firmware Version
One	5.60 (before October 2017)
Flex	7.81 (before October 2017)
Charge HR	18.102 (older…)

If you buy new trackers online, they have a firmware < October 2017 ☺

Summary

Project Status

- Cooperation with the **Fitbit Security** team for **Responsible Disclosure**
  - Communication encrypted with PGP ☺
  - Professional categorization of security issues we report.
  - They get early versions of our publications, we get early feedback.

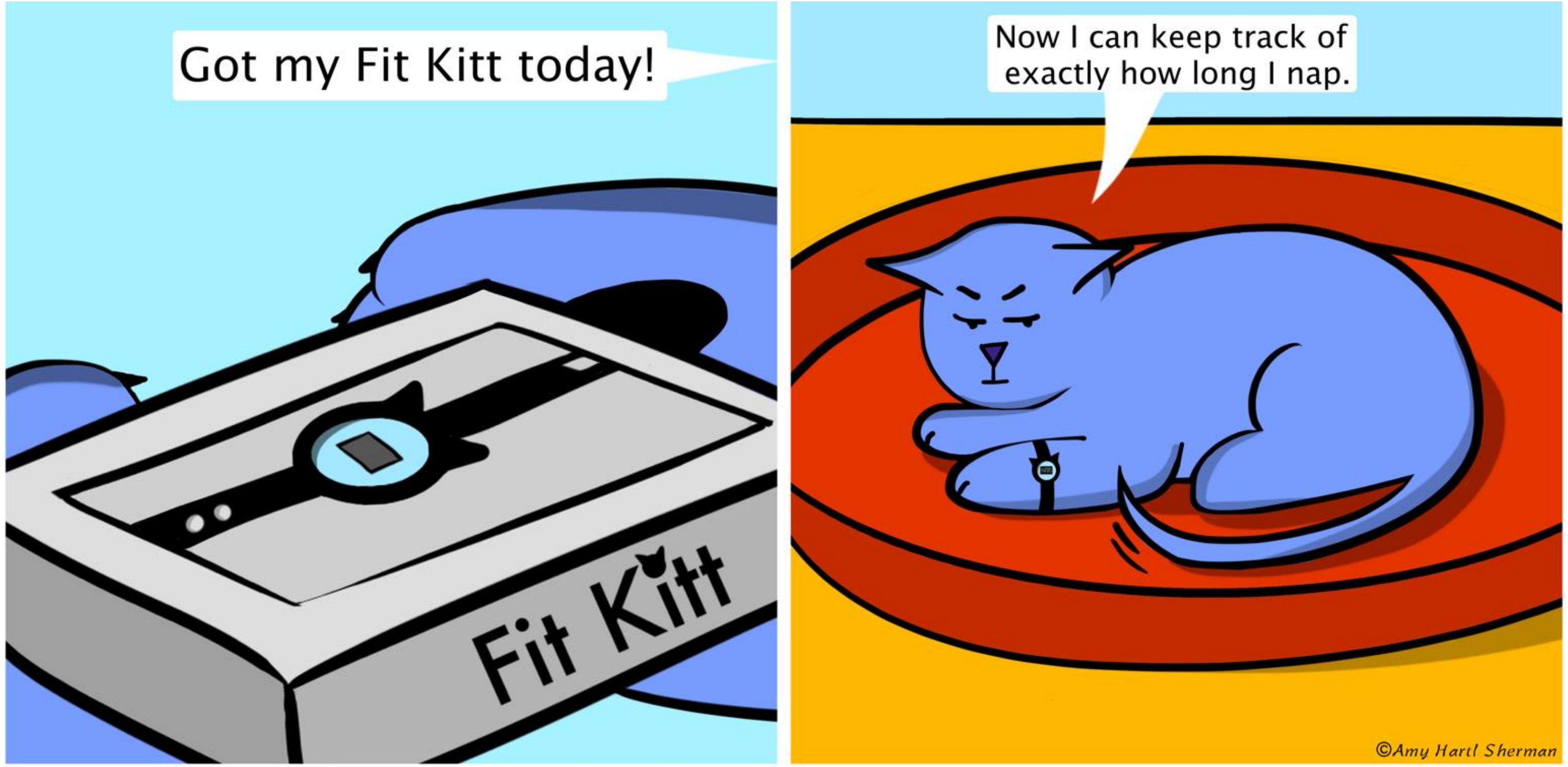
- Open source tool to run **server-independent actions** on fitness trackers, such as **live mode, memory extraction** and firmware **flashing** & a framework to craft your own firmware:

  https://github.com/seemoo-lab/fitness-app

  https://github.com/seemoo-lab/fitness-firmware

Summary

1. Go out and flash your neighbors�hinweis devices.
2. Keep control of your own data.
3. Run any code on your Fitbit.



Q & A