Mobile Information Systems

Lecture 07: Privacy & Security

© 2015-24 Dr. Florian Echtler
Bauhaus-Universität Weimar
Aalborg University

Key issue: security/privacy (recap)

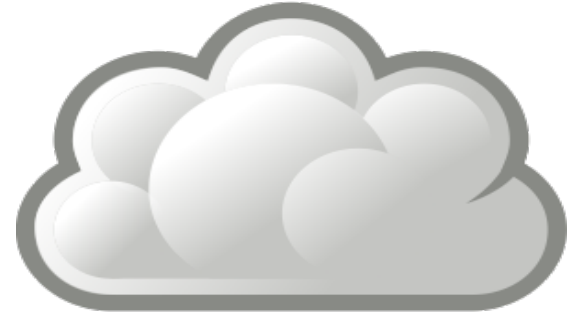
- Huge amounts of private & personal data on mobile devices
  - Contact information, messages & e-mails
  - Visited websites, pictures
  - PIN/TAN codes
- Many people want access to that data
  - Google, Facebook, Microsoft (for selling ads)
  - NSA, GCHQ, BND etc. (for catching criminals)
  - Hackers (for stealing/extorting your money)

Problem 1: (lack of) encryption

- No pervasive encryption on mobile devices
  - Strong opposition from government snoopers (up to demanding “key escrow”)
  - Device encryption
    - Protects flash memory with *user-derived* key
    - When missing: lost/found devices often trivial to access
  - Data encryption
    - Data in transit often unencrypted
    - Can be intercepted/modified

Problem 2: cloud services

- Widespread voluntary use of cloud services
  - Google, Facebook, Dropbox, Whatsapp, Amazon, …
  - Main reasons:
    - Substitution of online resources for on-device resources
    - Social components, communication, sharing
  - Requires trusting at least one, usually several 3rd parties (outsourcing)
  - Mostly without proper *end-to-end* encryption (cf. 1) → data freely accessible to cloud providers



Problem 3: (lack of) data protection

- Private data not protected “by default”
  - Outdated legal framework (but see EU GDPR)
  - Not strictly enforced by mobile OS
  - No fine-granular access control

  → access to contact list, messages, location, …
  possible for nosy/malicious apps
- Even if *data* itself encrypted,
  *meta*data often still readable
  (cf. WhatsApp)

Problem 4: intransparency

- Many components outside user control
  - Baseband module (own CPU, RAM, OS)
  - Device OS (e.g. iOS, many Android devices)
- No independent review possible
  - Bugs, security holes and/or backdoors may exist
  - Android is *supposedly* open-source …
  - … but *many* exceptions (Google apps, vendor modifications, …)
  - Updates sloppy after 1-2 years, almost non-existent after 3

Attacks on mobile devices

- In increasing (arbitrary) order of severity:
  - Re-use/sale of personal data
  - Snooping, ”shoulder surfing”
  - Nosy apps, spyware, malware
  - Network-based attacks (WiFi sniffer, IMSI catcher)
  - Physical device access/theft

Attacks: re-use/sale of personal data

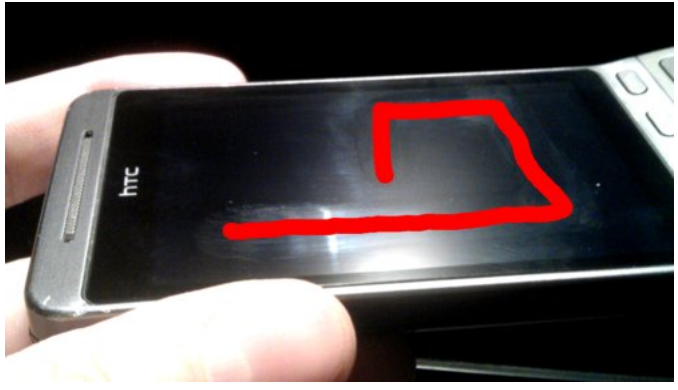
- Most cloud providers have …
  - Full access to uploaded data
  - ToS which allow them secondary use
  - Slowly changing due to GDPR
- Additional problems:
  - Providers acquired by other companies → new ToS
  - Other cloud users (e.g. photo tagging)

- Alternatives: personal cloud, e.g. NextCloud?

Attacks: ”shoulder surfing”

Image source (FU): https://f0rki.at/current-state-of-android-physical-security.html

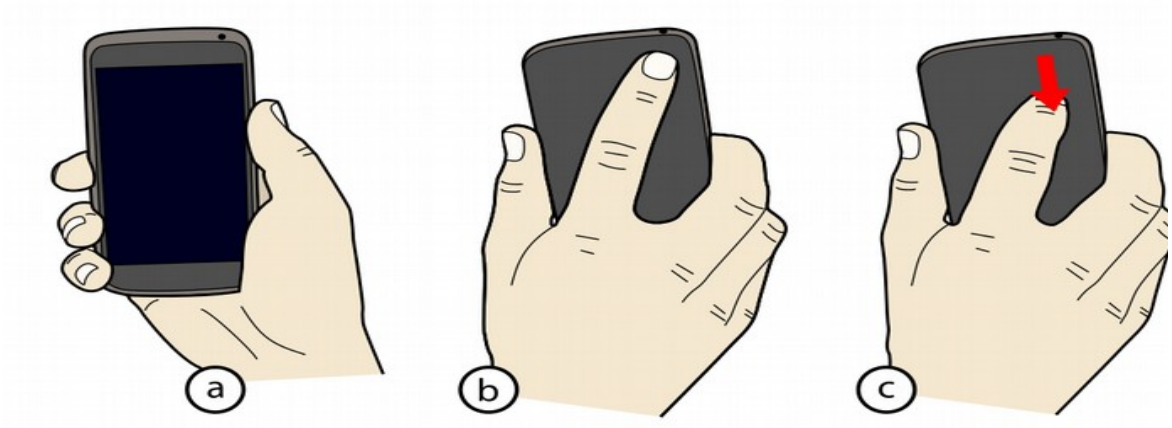
- “Unlock patterns” & short PINs easy to spot
- “Smudge attacks”
  - Fingers always leave small grease traces
  - Traces can be used to infer patterns/PINs



Attacks: ”shoulder surfing” (2)

Image source (FU): https://dl.acm.org/citation.cfm?id=2481330

- Possible methods of defense:
  - “Back-of-device authentication” (De Luca et al., CHI 2013) (requires extra sensor on back side)
  - Combination of PIN input with eye tracking, grasp sensing, …
  - Randomization of number layout (drawbacks?)



Attacks: spyware & malware

- Definition: malicious software which tries to intercept information or mis-use resources
  - In mobile context: mostly infostealers (spyware)
  - Less available resources (CPU/bandwidth) than on desktop/laptop computers
  - Secondary goal: resilience against removal
- Android = main target for malware
  - ~ 80% market share
  - Less strict security model than iOS
  - Slow/sloppy review process in app stores

Android security model

- Based on Linux security model
- Each app has own “account” (UID)
  - No direct access to other apps' data/files
  - Only via OS-provided interfaces (permissions)
- Problems:
  - Permissions can only be approved as always/once/no
  - Exact implications often unclear for user
    - E.g. messaging app requires contact data & internet access (obvious for messaging service)
    - Can also be used to “exfiltrate” whole address book

Attacks: spyware & malware (2)

- Most Android malware are *trojans*
  - Provide some legitimate service/function
  - Hide illicit activities in background
- Dedicated malware relatively rare
  - E.g. ZeuS (Windows) + ZitMo (ZeuS-in-the-Mobile)
  - Zeus on PC prompts user to connect mobile device, installs mobile malware component
  - Zeus intercepts banking website credentials, ZitMo intercepts SMS with TAN

  → full access to bank account!

Attacks: spyware & malware (3)

- Can (sometimes) be analyzed in *sandbox*
  - Android emulator with additional logging (syscalls, network traffic, …)
  - Simulated user input to trigger malicious behaviour
- “Arms race”:
  - Malware tries to identify sandbox environment
  - Sandbox tries to look as realistic as possible
  - E.g. IMEI check, vibration → accelerometer, sound → microphone, waiting for specific user input …

Attacks: network-based attacks

- WLAN (802.11x) – passive sniffing attacks
  - Unencrypted (many hotels)
  - WEP (outdated & thoroughly broken)
  - WPA(2) – pre-shared key & handshake must be known (key often public, handshake can be forced)
  - WPA2 Enterprise (e.g. eduroam) – currently assumed to be secure (but can you trust every eduroam AP?)
  - WPA3 – also not fully secure, known attack scenarios
- → use separate VPN on top of WLAN

Attacks: network-based attacks (2)

- WWAN
  - Identifying information
    - IMSI (Int. Mobile *Subscriber* ID) = SIM card
    - IMEI (Int. Mobile *Equipment* ID) = baseband module
  - Location monitoring
    - Cell-based location always available in backend
    - Uses SS7 (Signalling System 7) backend network

Attacks: network-based attacks (3)

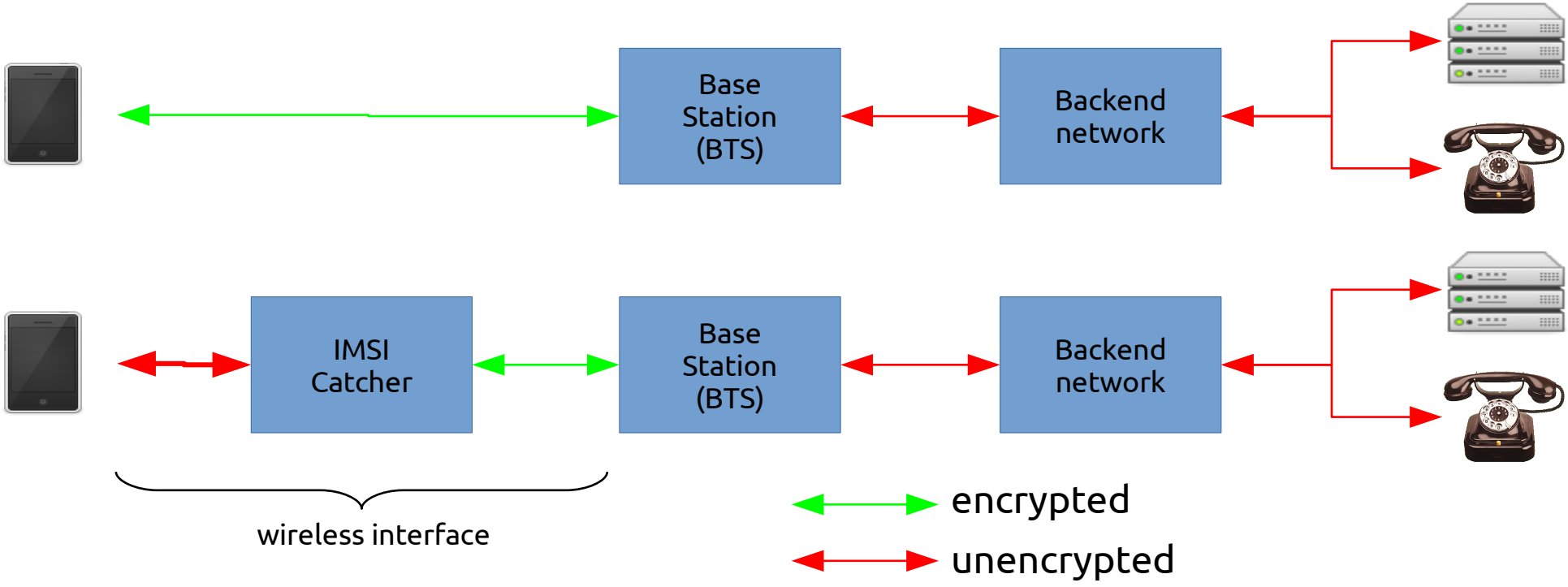
- Signalling System 7 (SS7)
  - Ancient protocol suite (since ~ 1975), used for …
    - call setup, billing etc. between carriers
    - carrier-internal configuration (roaming between MSCs)
  - Many bugs & security issues
  - Accessible to “rogue operators” - allows …
    - re-routing of calls and messages
    - tracking cell-based location of IMSI/IMEI
    - disabling call encryption
  - SMS re-routing recently used for bank fraud (mobile TANs intercepted)

Attacks: network-based attacks (4)

- WWAN - call/data monitoring
  - Uses *IMSI catcher* = “fake base station”
  - Mobile devices must authenticate to base stations, but *not* vice versa → *anybody can operate a BTS!*
  - Strong signal when close to target → phone switches to IMSI catcher
  - IMSI catcher can downgrade/disable encryption
  - Data forwarded to regular network, but decrypted/recorded beforehand
  - Possible using open-source tools (OpenBTS)

Attacks: network-based attacks (5)

- IMSI catcher: classic “MITM” (man-in-the-middle) attack

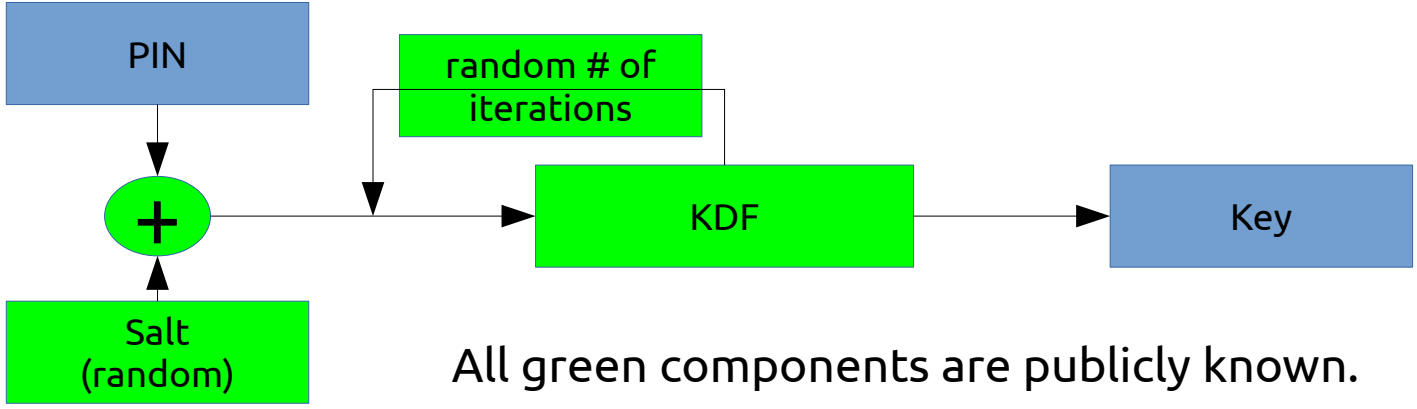


Attacks: physical device access

- Security rule-of-thumb (“evil maid”):
  - Once attacker has physical access, you've lost.
  - Last line of defense: very strong encryption (public-key cryptosystem, 1000s of bits key length)
- Problem: key management - impossible to type/remember 1000s of key bits
  - Alternative 1: key needs to be derived from/ protected by smaller number of bits → use KDF
  - Alternative 2: key needs to be stored in a separate security device, e.g. smartcard (such as SIM card)

Attacks: physical device access (2)

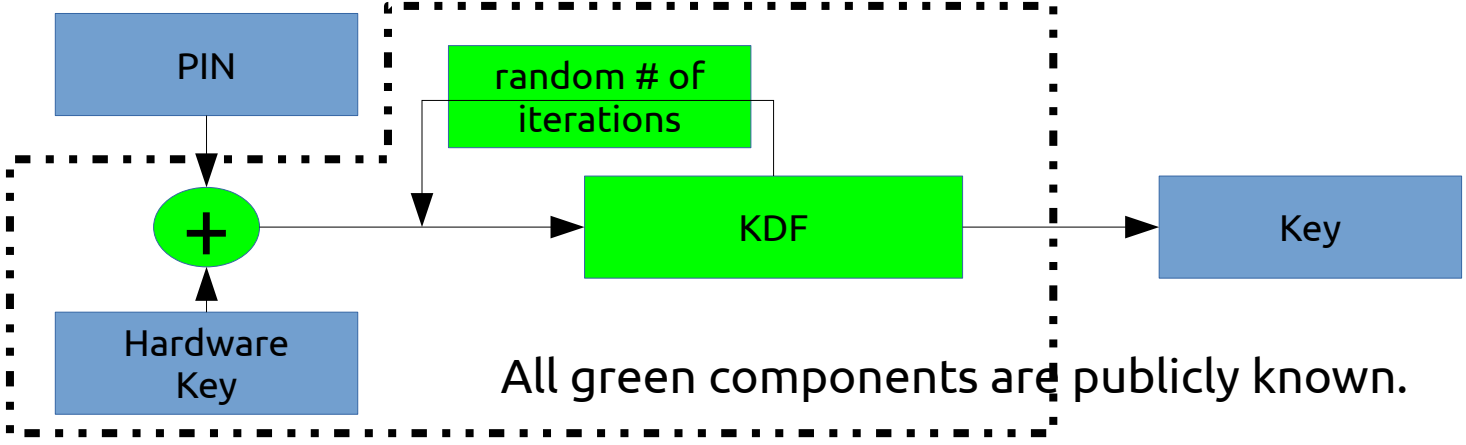
- KDF = Key Derivation Function
  - Protection against brute-force attacks
  - Complex function, 1000s of iterations
  - KDF should not take more than ~ 1 second on mobile device for fast unlock → problems?



All green components are publicly known.

Attacks: physical device access (3)

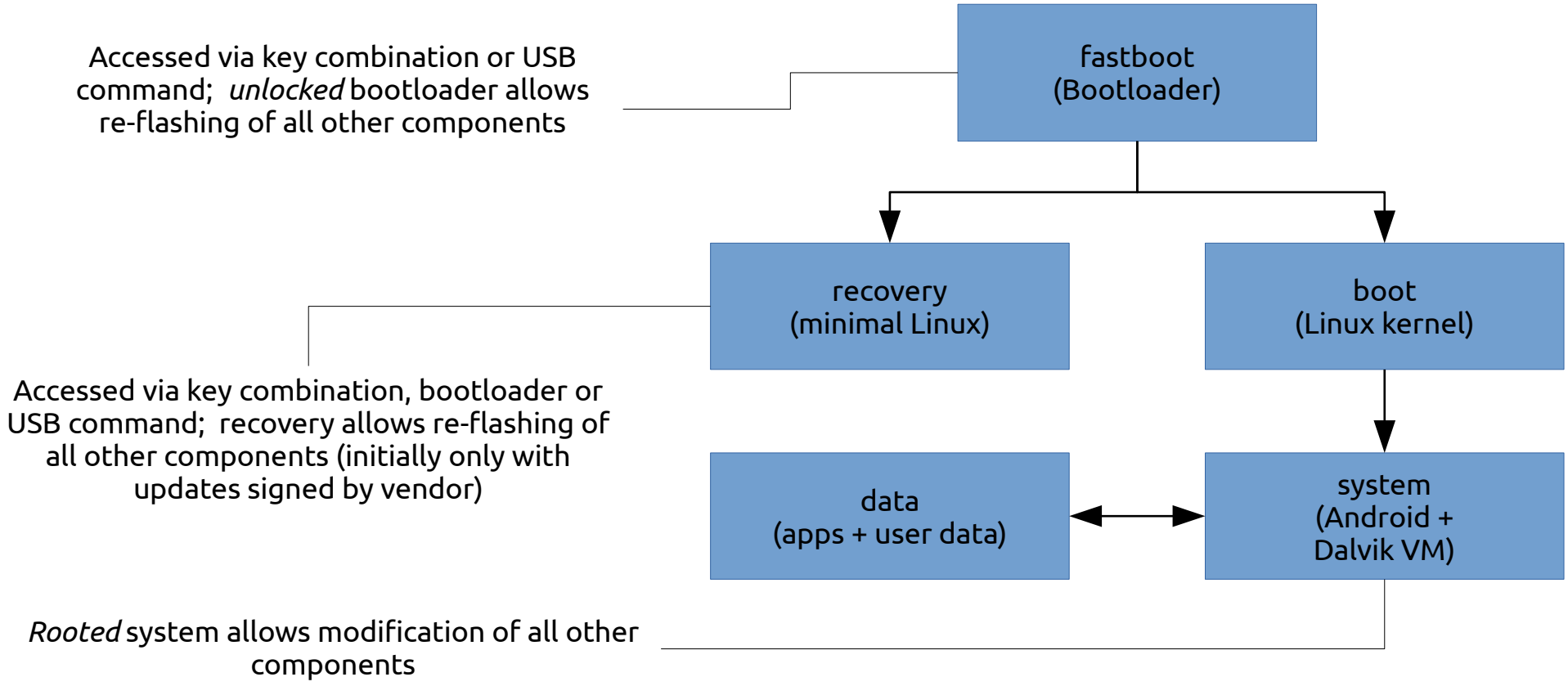
- Software-only KDF: no mitigation against offline attacks with high-end computers
  - Alternative: “Secure Enclave” (iPhone) with internally-stored hardware key
  - Wipes HW key after # of wrong tries



Attacks: physical device access (4)

- Alternative key storage: smart card
  - Small dedicated computer (CPU, RAM, ROM)
  - Common example: SIM card
    - However: (currently) not used to store device keys, only WWAN network/baseband keys
  - Features:
    - Irrevocable wipe after *n* wrong PINs
    - Tamper-proof: e.g. light sensor → triggers chip wipe
    - Attacks usually require advanced knowledge of internals, sometimes via an electron microscope

Android startup process



Attack scenarios on found device

- Device is locked and/or PIN protected
  (note: device PIN != SIM card PIN)
  - Bootloader unlocked
    - Flash & boot custom recovery image
    - Create backup of data partition
    - Analyze offline → PIN protection useless (see KDF)
  - Bootloader locked
    - Simply unlock bootloader? → will erase data partition
    - Attempt to re-flash recovery anyway (bugs in BL?)
    - Ultimate solution: direct hardware access to flash chips
    - “Cold-Boot”: cool down RAM, reboot, read out remains

Encryption done right?

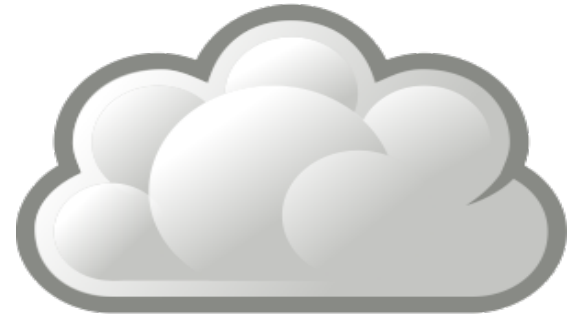
- PIN > 4 digits → user acceptance sinks *rapidly*
  - External tokens possible (USB/NFC), but rarely used
  - Best alternative: fingerprint (but can also be cloned)
- Communication
  - Always use secure channels (HTTPS)
  - Very difficult to account for all failure modes
    - Can often be intercepted, e.g. with MITMProxy
    - Possible countermeasures: Certificate Transparency, Certificate Pinning
- Cloud-stored data still rarely encrypted

Authentication done right?

Image source (FU): http://www.heise.de/.../Android-Smartphone-mit-Iris-Scanner-2650067.html

- Two-factor authentication = something you *know* (e.g. PIN) + something you *own*
  - Biometric: face, fingerprint, iris pattern, …
  - Hardware: wireless token, chipcard, …
- Phone itself often used as second factor
  - NemID/MitID, SMS one-time codes
  - Problematic when phone itself may be insecure/compromised

Cloud services done right?



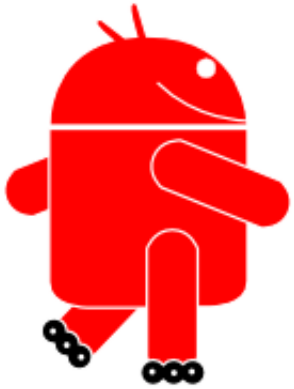
- Peer-to-peer approaches
  - E.g. Tor, Bittorrent, Bitcoin
  - Problem: discovery of mobile peers, multi-hop routing → high network load
- End-to-end encryption
  - Some 3rd party tools: e.g. Boxcryptor, Truecrypt
  - Damages business model of many cloud providers
  - Zero-knowledge computing → computations on encrypted data *without* intermediate decryption

Data protection done right?

- Fine-granular access management
  - Current state: take it or leave it
  - App permissions can only be approved or denied as a whole
  - Slightly better on iOS: individual decisions possible
  - Most apps will simply not work if access denied
- Possible solution: fake personal data
  - E.g. provide plausible-looking randomly generated contacts instead of real ones
  - Apps shouldn't be able to tell the difference

Transparency done right?

Image source (FU): http://www.replicant.us/



- Replicant/GrapheneOS: Android derivatives
  - Contains *only* open-source components
  - No Google apps, services & store
  - Hardware support limited (older devices)
- Osmocom
  - Fully open-source GSM/UMTS baseband implementation
- “Open” phones
  - OpenMoko: open hard-/software smartphone, outdated
  - FairPhone: hybrid, reasonably open, more recent

The End