NEVER BEFORE A SPECTACLE LIKE
THE FALL OF THE SMART CARDS
The thunder of war split them!
The fires of passion burnt them!
The swords of barbarians bled them!
The evil of tyranny toppled them!
ERIK POLL
PRESENTS
ERIC VERHEUL
FABIAN VAN DEN BROEK
THE FALL OF THE SMART CARDS
TECHNICOLOR®
CO-STARRING
JOHN IRELAND · MEL FERRER · OMAR SHARIF
AND ANTHONY QUAYLE · Directed by ANTHONY MANN · Music by DIMITRI TIOMKIN
Original Screenplay by BEN BARZMAN · BASILIO FRANCHINA · PHILIP YORDAN · Produced by SAMUEL BRONSTON · ULTRA-PANAVISION®
A PARAMOUNT RELEASE
"THE FALL OF THE ROMAN EMPIRE" COPYRIGHT© 1964 by BRONSTON-ROMA PRODUCTIONS Printed in U.S.A.

Mobile authentication

Vague plans for possible future work

Erik Poll

with inspiration from

Eric Verheul & Fabian van den Broek

Mobile *online* authentication

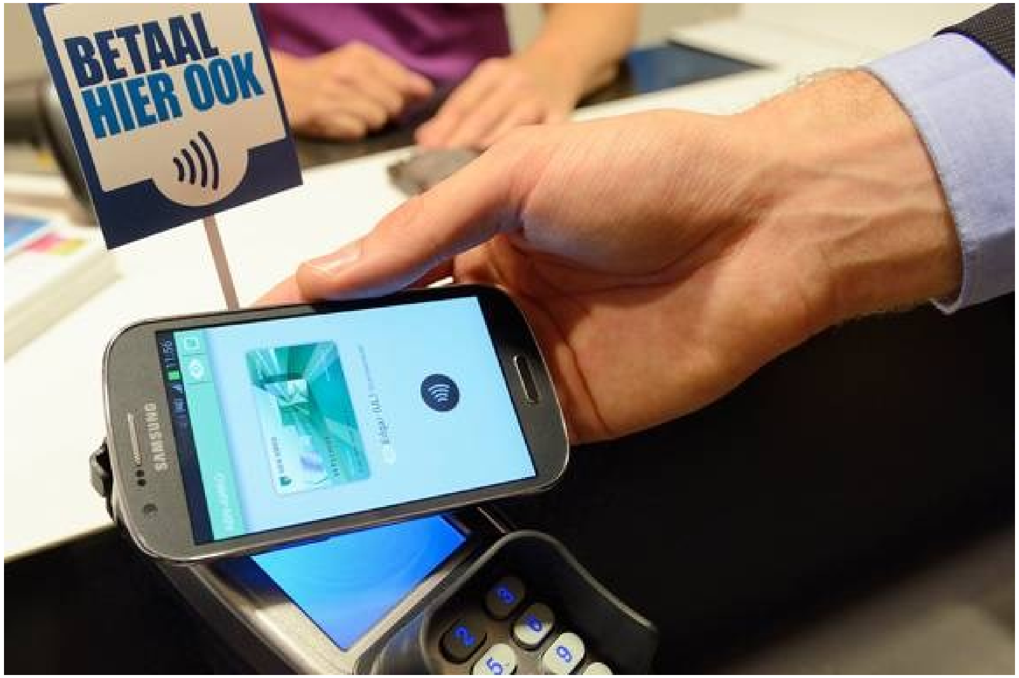
Vague plans for possible future work

Erik Poll

with inspiration from

Eric Verheul & Fabian van den Broek

Exit smartcards, enter apps

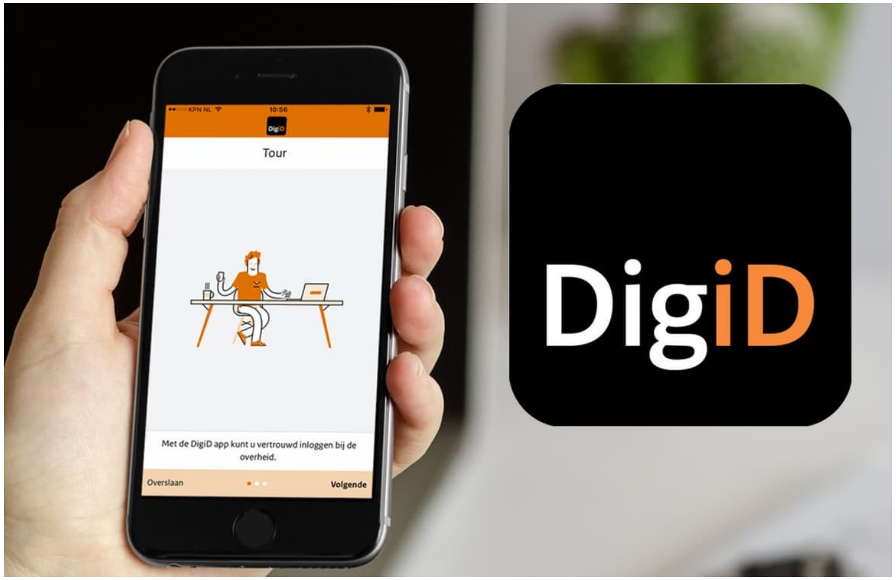


Exit smartcards

ov-chipkaart
Op deze kaart zijn de Algemene Voorwaarden OV-chipkaart van Trans Link Systems B.V. van toepassing. Zie www.ov-chipkaart.nl of bel met 0900-0980 (gebruikelijke belkosten).

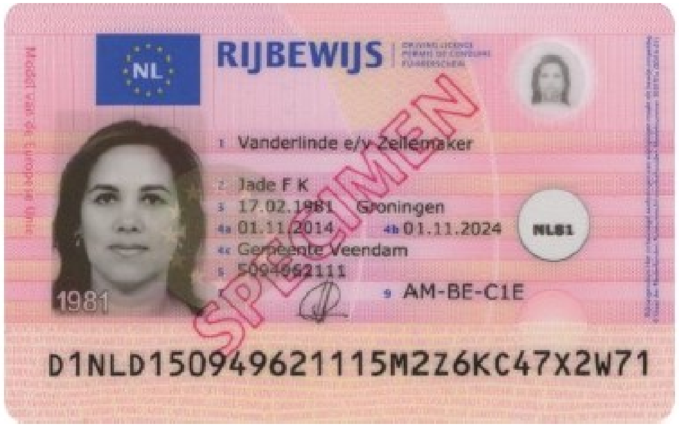
Exit smartcards

NEDERLANDSE IDENTITEITSKAART
KONINKRIJK DER NEDERLANDEN
KINGDOM OF THE NETHERLANDS
IDENTITY CARD
ROYAUME DES PAYS-BAS
CARTE D'IDENTITÉ
documentnummer / document no.
SPECI2014
naam / surname
De Bruijn
voornamen / given names
Willeke Liselotte
geslacht / sex nationaliteit / nationality
V/F Nederlandse
geboortedatum / date of birth
10 MAA/MAR 1965
datum van afgifte / date of issue
09 MAA/MAR 2014
geldig tot / date of expiry
09 MAA/MAR 2024
handtekening / signature
W.L. de Bruijn
SPECIMEN



Met de DigiD app kunt u vertrouwd inloggen bij de overheid.

Exit smartcards



Model van de Europese Unie

NL

RIJBEWIJS | DRIVING LICENCE PERMIS DE CONDUIRE FÜHRERSCHEIN

1 Vanderlinde e/v Zellemaker

2 Jade F K

3 17.02.1981 Groningen

4a 01.11.2014 4b 01.11.2024

4c Gemeente Veendam

5 5094962111

9 AM-BE-C1E

NL81

1981

SPECIMEN

D1NLD150949621115M2Z6KC47X2W71

NSW Driver Licence

Refreshed 5 Nov 2019 02:27pm

NSW

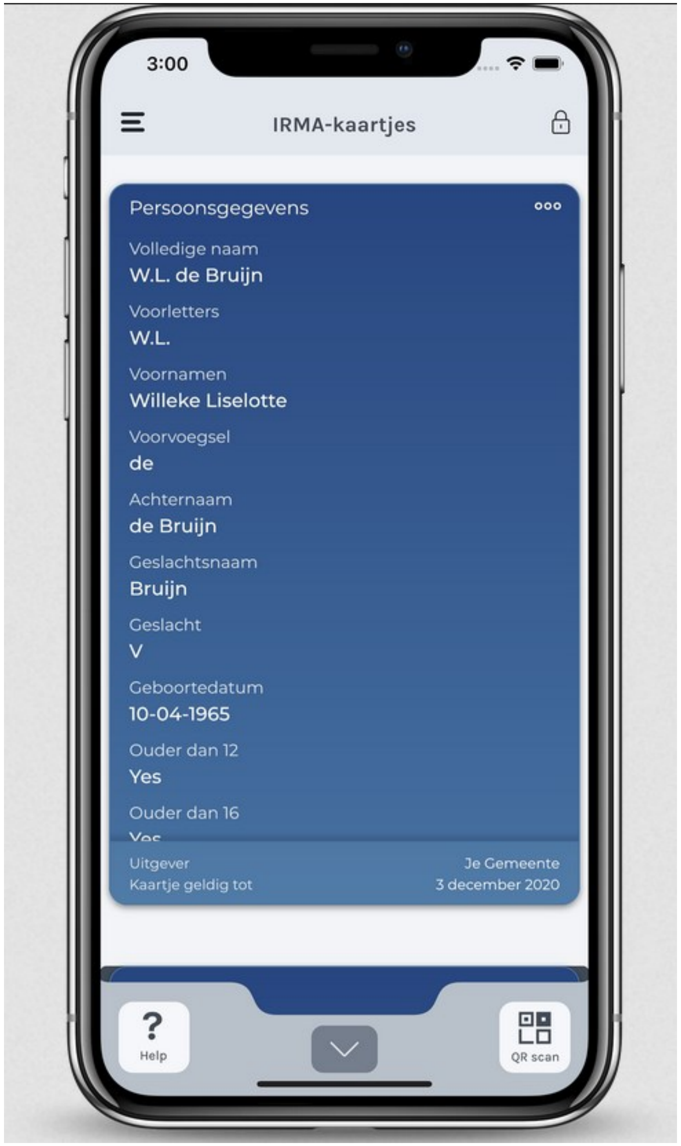
Steve Douglas CITIZEN

LICENCE NUMBER 132842687

EXPIRY

May 2023

Exit smartcards



Persoonsgegevens

Volledige naam
W.L. de Bruijn

Voorletters
W.L.

Voornamen
Willeke Liselotte

Voorvoegsel
de

Achternaam
de Bruijn

Geslachtsnaam
Bruijn

Geslacht
V

Geboortedatum
10-04-1965

Ouder dan 12
Yes

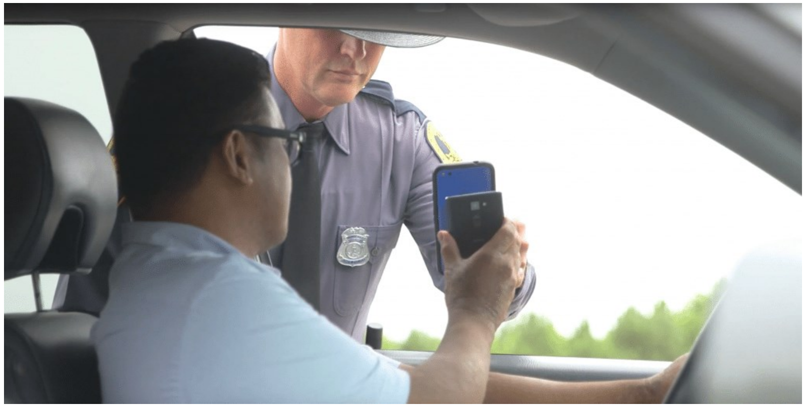
Ouder dan 16
Yes

Uitgever Je Gemeente
Kaartje geldig tot 3 december 2020

Skipping smartcards?

Another trend: Offline ⟶ Online

- Offline use in the physical world
- Online use in the cyberspace
- Combinations
  - incl. digital onboarding



Very different risks! Eg attacks in physical world usually

- do not scale
- come with risk of getting caught

For example

- The crappy crypto in ov-chipcards in the end never caused big issues
- How much of a problem is extracting keys from a Dutch passport chip really?

Why did we use smartcards anyway?

authentication

Pet peeve: there is no A for Authentication in CIA

ACAI would be a better acronym

Why did we use smartcards anyway?

Important ways to do authentication

1. ‘Bio’metrics

    human fingerprints and device fingerprints

2. Passwords (incl. PIN codes & cookies)

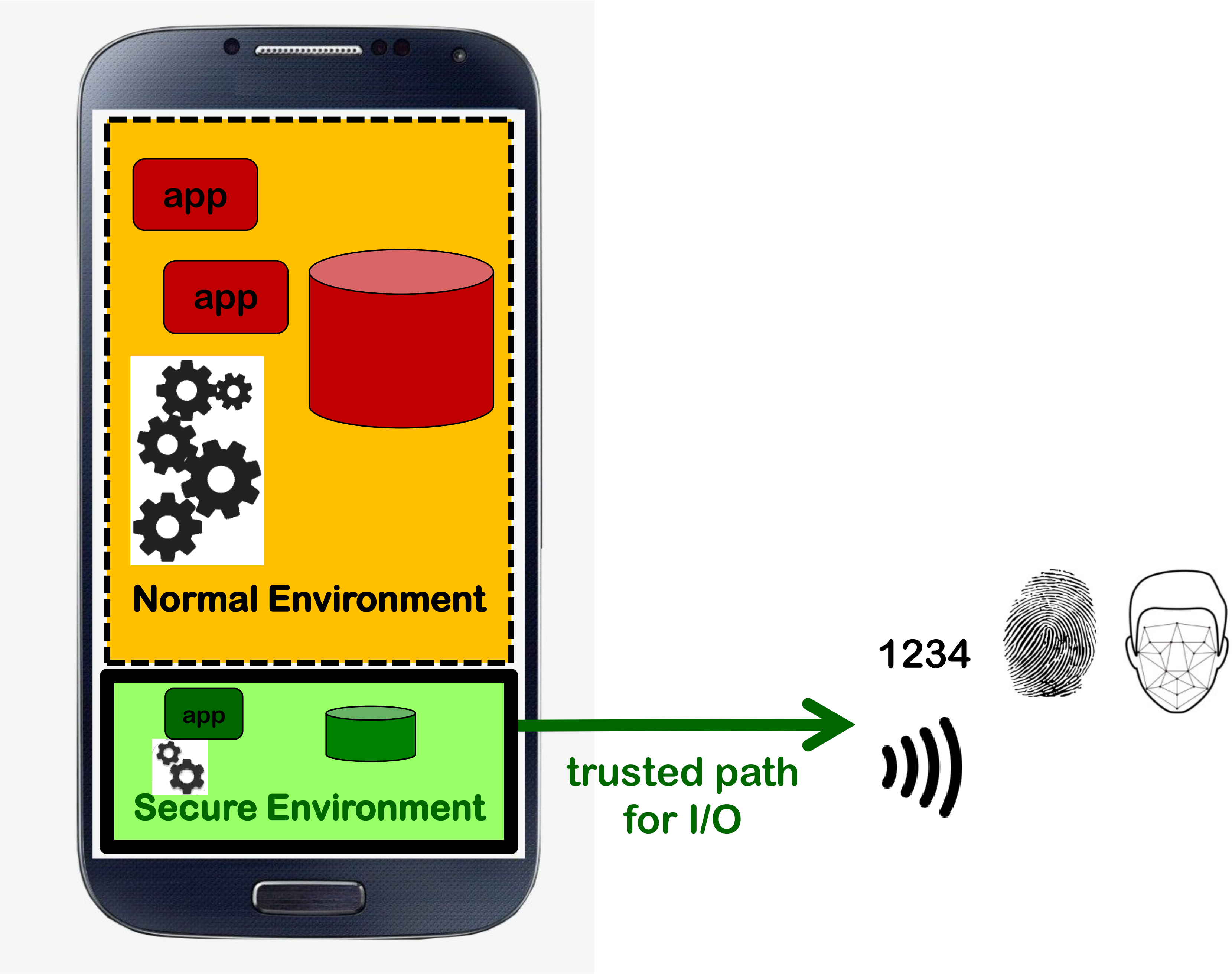
3. Challenge-response

    using security questions or crypto

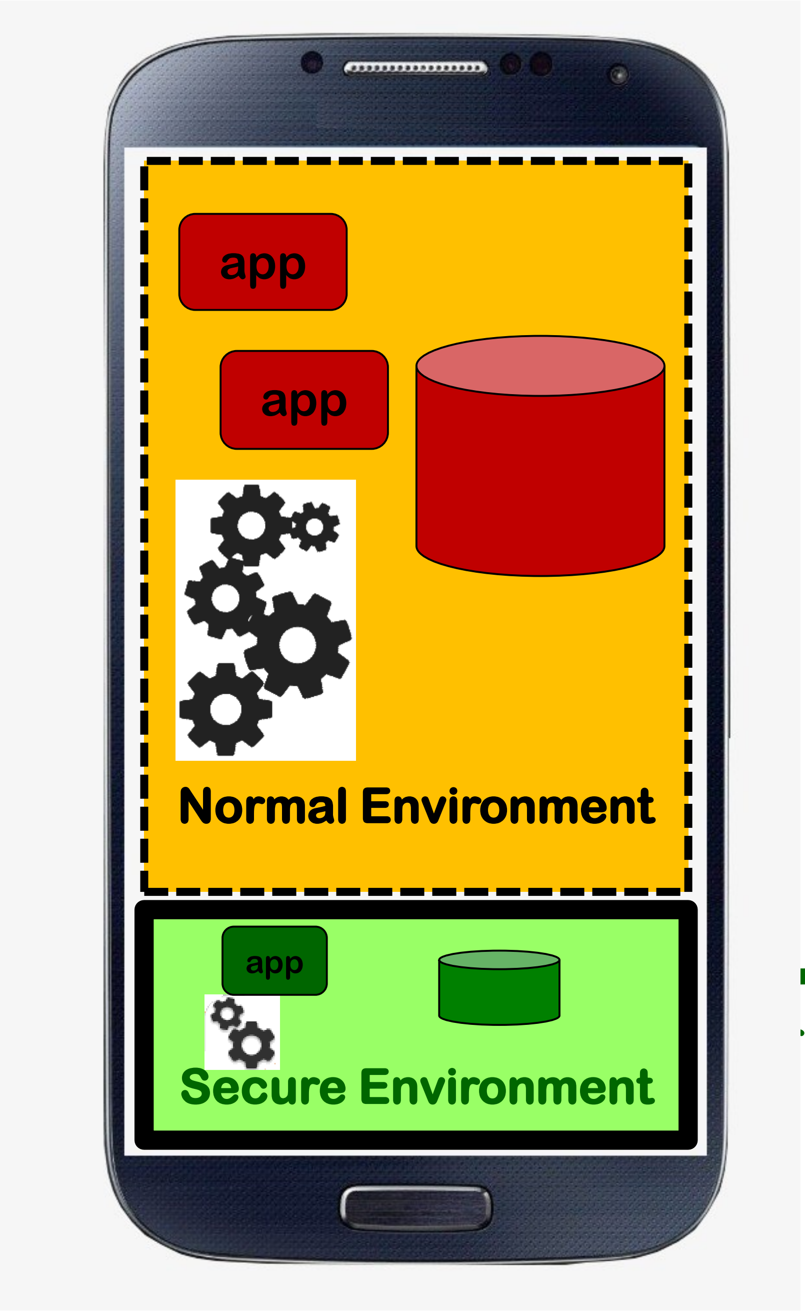
and combinations:

eg access control to challenge-response with PIN code

Secure Environments *in* mobile phones

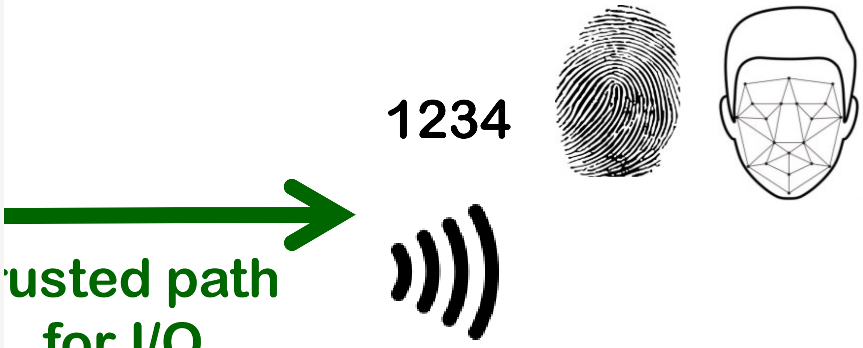


Secure Environments *in* mobile phones



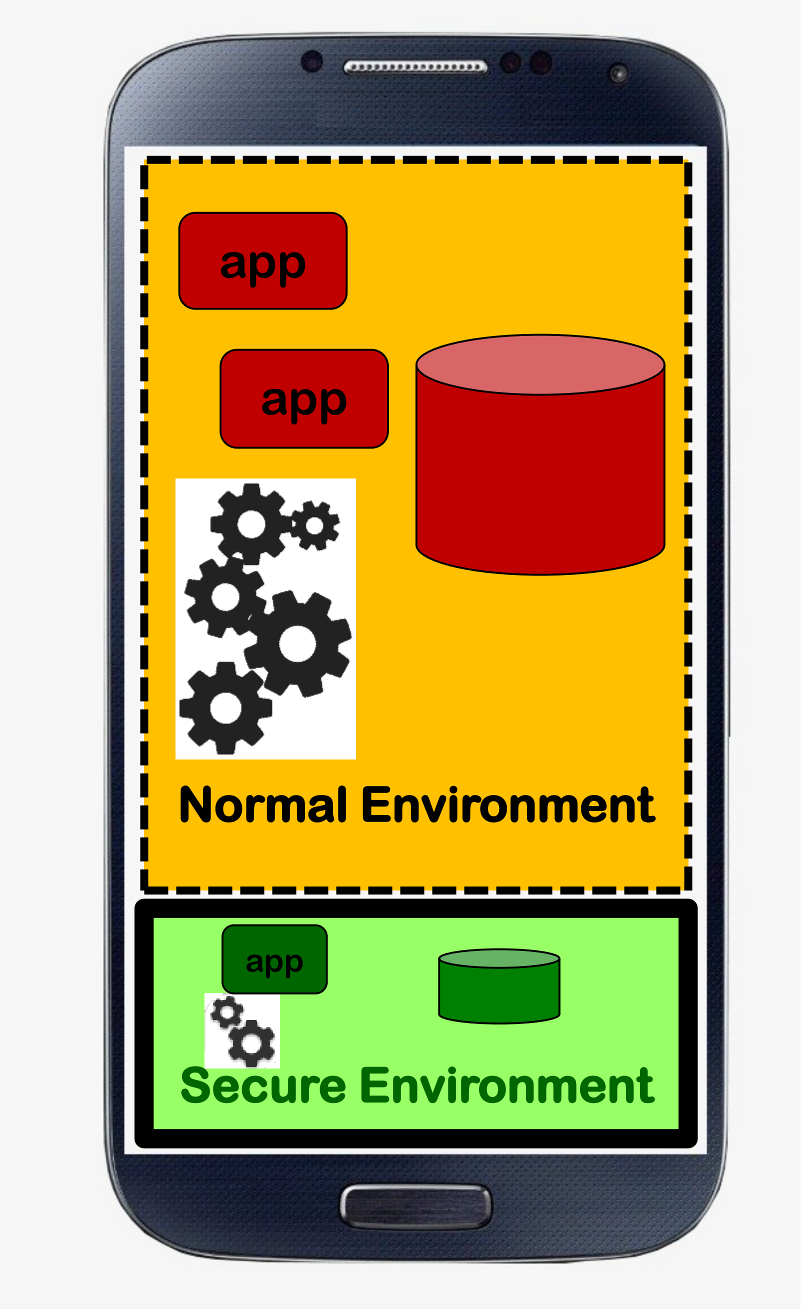
What can SE do?

- processing for crypto and access control checks
- RNG
- data storage for keys, PINs, biometrics
- Fixed functionality provided by OEM, or extensible with trustlets



rusted path
for I/O

Secure Environments *in* mobile phones



How?

1. *physically* separate
   a) SIM card
   b) Secure Element (RIP?)
   c) Apple Secure Enclave &
      Android Strongbox Keymaster
2. *virtually* separate
   a) ARM TrustZone TEE
      (getting less fashionable?)
   b) Whitebox crypto (😂)

Should & will we use SEs?

Given that hardware-backed SEs, esp. Apple Secure Enclave or Android Strongbox Keymaster, are becoming more standard:

- *Should* we use them?
  - Security pros & cons? (for which properties, for which attacker models)
    - Also: pros & cons wrt. usability & privacy?
  - For which types of services are these important/relevant?
  - Will/do eIDas regulations require use?
- *Will* we use them?
  - Technical hassle in implementation & support
  - Legal hassle with additional parties: Apple, Google, OEM, ...
- If we use them, how do we evaluate security?

Attacker models

1. Physical access to mobile phone

   a) with/without PIN or biometrics (fingerprint/faceID)

   b) with/without side channel attacks

2. Compromised main OS

   Does not allow key extraction, but does allow key usage.
   Maybe fingerprint needs to be phished for this.

Risk(*using* keys with 2) >> Risk(*extracting* keys with 1b)

4. Compromised app

   Attacks on apps don't happen much; attack vectors are limited.
   Special case: compromised browser

5. Compromised SE overlaps with 1b; with much smaller risk than 2 or 3

6. Phishing
   with as possible ingredients: malicious app, malicious website, spoofed communications (eg. fake caller ID) or confusing redirects (eg. app ⟶ browser and vv.)

Security of *mobile phone with SE* vs *smartcard*

– More complex and (hence) less secure

+ Mobile phone can do I/O

+ Mobile phone can do biometrics

+ Loss of control: dependency on 3rd party device, OS, app store

- New and more powerful attacker models, in addition to usual attacks on SEs/smartcards
  1) Compromised OS 3) Compromised app
  2) Compromised SE 4) Malicious app

- Nearly always online
  This is both good (eg. for monitoring & response and for updating)
  and bad (as attack vector & for phishing)

– One SE can hold many credentials
  Like a multi-application smartcard. Bad for phishing.

- Enrolment & revocation are totally different:
  – complex, but + cheaper & more flexible

Security of *app + SE* vs *app + smartcard*

**Isn't comparing mobile phones with smartcards unfair,
because smartcard will also need a terminal and maybe an app?**

– **Bigger attack surface in time**

– **If attacker can comprise OS or app, then keys in SE can be used**

+ **SE can control access using biometrics, which smartcards can't.** (Idem for PIN?)

+ **Only one app can use a specific SE key** (assuming OS is not compromised), whereas any app can fool a user in holding their smartcard against the phone

- **Enrolment & revocation are totally different:**
  - complex, but + cheaper & more flexible

Security of *app with SE* vs *app without SE*

Isn't the real (and easier) choice between using the SE or not?

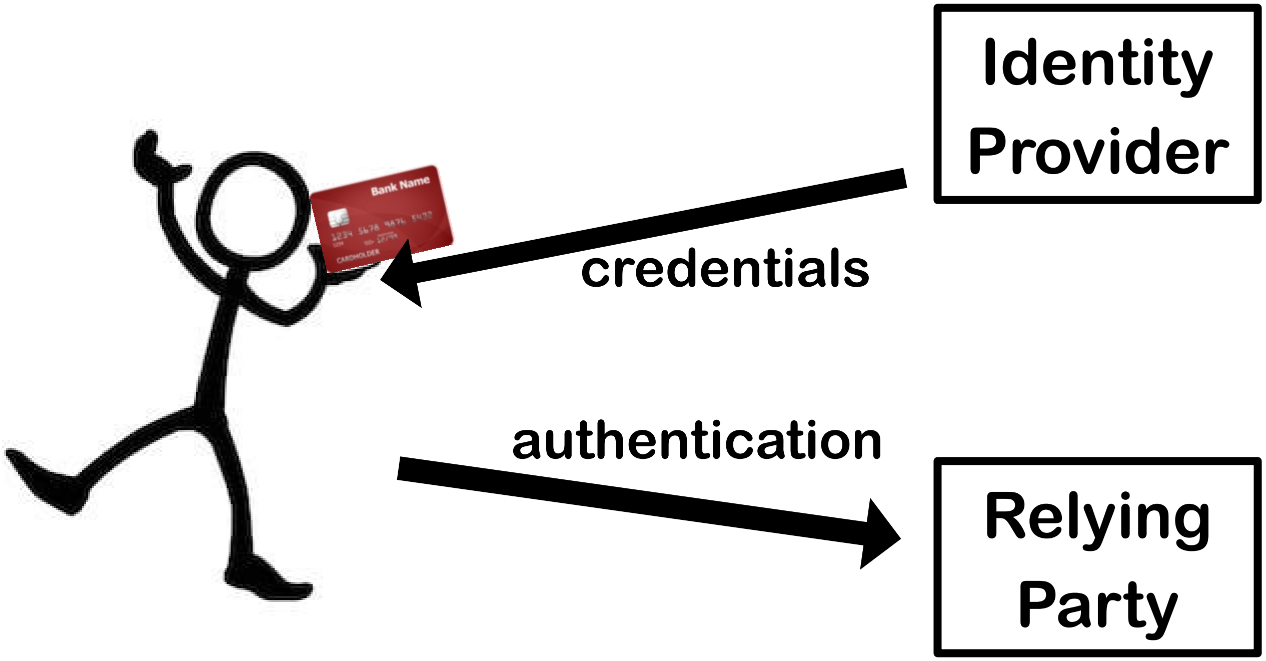
+ **Key protection by SE much more secure than by main OS**

But if the OS is compromised, how much security does SE really buy us?

Maybe the risk of bad publicity & resulting impact on trust is more important

– **Privacy risk of involving OEM in issuance, and using one physical SE for multiple across different applications?**

Using smartcard



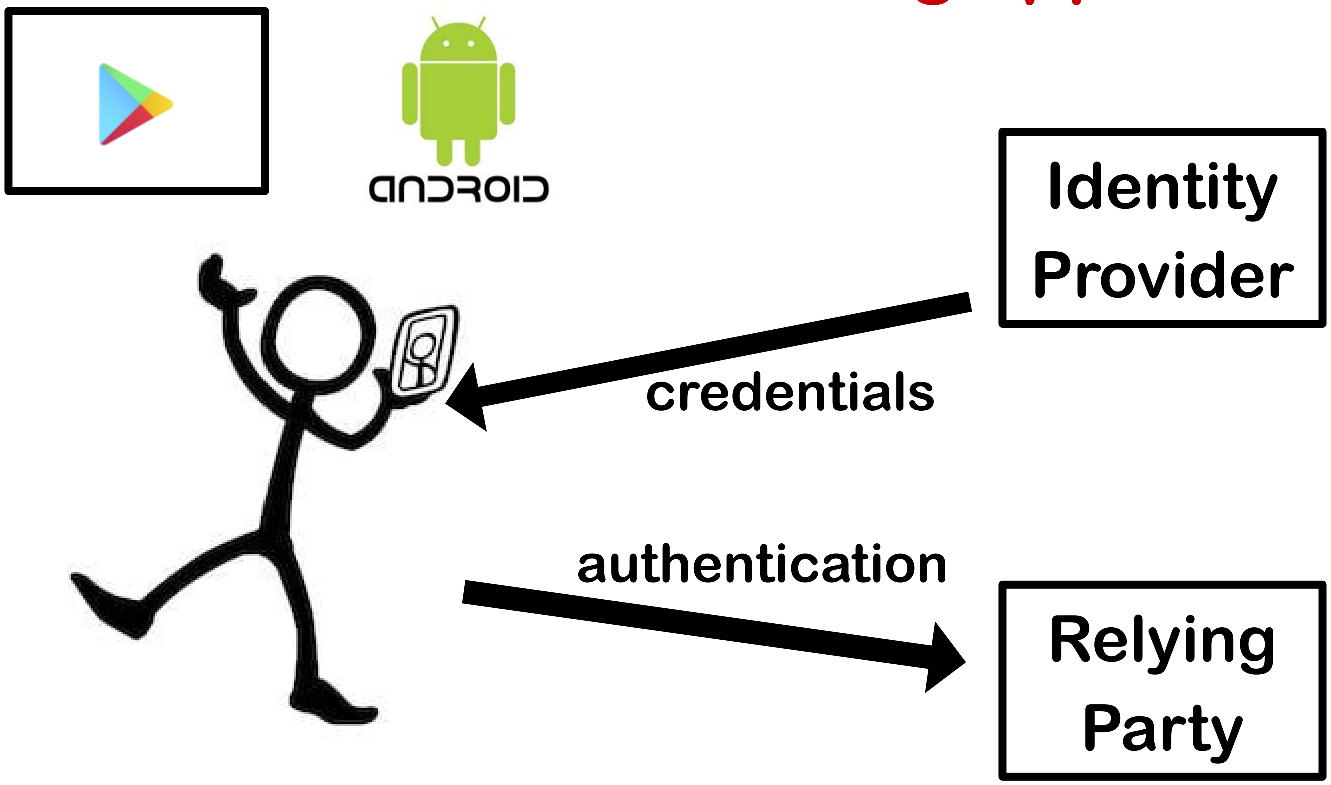
Identity Provider

credentials

authentication

Relying Party

Using app



Identity Provider

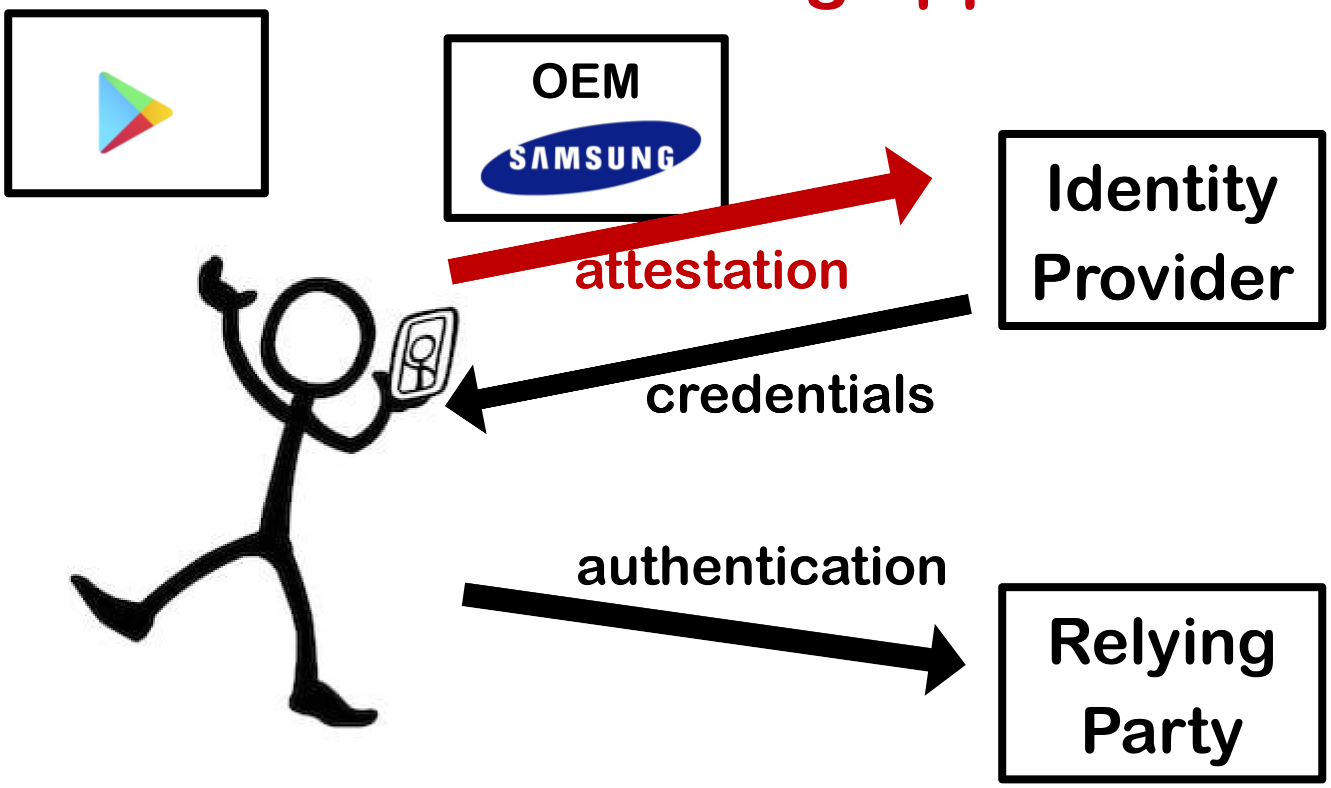
credentials

authentication

Relying Party

- Playstore/OS/OEM involved in distributing & updating apps (and OS)
- OS can play role in monitoring & eg attesting that phone is unrooted
- OS in TCB for everything

Using app + SE



- OS in TCB for many things,
  but not in TCB for confidentiality of crypto keys, PINs & biometrics
- OEM involved in key generation & key attestation
  - commercial/legal hassle & privacy issues?

Discussion: security evaluation

- Security requirements for SEs require smartcard-like side-channel resistance to key extraction
  - Android 11 spec (Sect 9.11.2) requires BSI-CC-PP-0084-2014 or similar & recommends EAL 5 augmented by AVA_VAN.5.

- Isn't that distracting us from the much bigger risk of malware that simply uses the SE's functionality?

DHL scam text: How to avoid 'parcel arriving' message scam after Android FluBot warning

David Hughes 09/05/2021

- Or the even bigger risk of phishing?

How to break classic online authentication?

1. Steal credentials from the server
   Eg hack server of Yahoo and steal password database

2. Steal credentials from the user
   Eg install keylogger to intercept username & password

3. Phishing
   Eg trick using visiting fake website to reveal password

   - The root cause is weak authentication of *service* by the *user*, not vv

   - Risk much bigger for websites than for apps

Claim: Risk of 3 much larger than risk of 2

How to break *mobile* online authentication?

1. Steal credentials from the IdP/Relying party
   No longer works, because we use challenge-response

2. Steal the credentials from the user's mobile phone

   a) Side-channel attack on SE to extract private keys

   b) Compromise OS to access SE signing functionality

   c) Trick user into installing fake app and enrolling again

3. Phishing

   - Authentication scheme can provide some protection against this!
     Recall Eric Verheul's talk last week.

   - Using apps rather than websites can protect against this.
     Eg why not file tax return *inside* the DigID app?

Learning from experiences of banks?

**Banks introducing smartcards in 1990s; governments followed decade later.
The same now happens with mobile-based solutions.**

***Can e-government services learn from banking experience?
Or are security risks for the types of applications too different?***

Lessons learnt:

- **Banks often don't use the SE for payment apps; the risk is reduced by short-lived keys (aka EMV Tokenization)**
- **Banks typically don't use the SE for banking apps.
Still, banking apps are more secure than banking websites because**
  - **Websites are easier to spoof & use in phishing attacks**

    The latest phishing trend, Whatsapp fraud, does work for apps (and websites), because phishing happens outside the banking app
  - **Web browser is easier & more rewarding to hack than apps;
man-in-the-browser attacks easier than man-in-the-app attacks**

Meta-observation about phishing

Don’t we focus too much on authentication *of* the user
and overlook the (bigger?) problem of authentication *by* the user?